TUGAS AKHIR – EE 184801

ANALISA KINERJA TEKNIK HYBRID BEAMFORMING PADA SISTEM LARGE SCALE ARRAY MULTIUSER MISO TERHADAP KANAL ACAK

Dicky Rahmadi Prasetya
NRP 07111640000093

Dosen Pembimbing
Dr. Ir. Puji Handayani, M.T.
Prof. Ir. Gamantyo Hendrantoro, M.Eng.,Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – EE 184801

ANALISA KINERJA TEKNIK HYBRID BEAMFORMING PADA SISTEM LARGE SCALE ARRAY MULTIUSER MISO TERHADAP KANAL ACAK

Dicky Rahmadi Prasetya
NRP 07111640000093

Dosen Pembimbing
Dr. Ir. Puji Handayani, M.T.
Prof. Ir. Gamantyo Hendrantoro, M.Eng.,Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

FINAL PROJECT – EE 184801

PERFORMANCE ANALYSIS OF HYBRID BEAMFORMING TECHNIQUES USING LARGE SCALE ARRAY MULTIUSER MISO SYSTEM ON RANDOM CHANNELS

Dicky Rahmadi Prasetya
NRP 07111640000093

Dosen Pembimbing
Dr. Ir. Puji Handayani, M.T.
Prof. Ir. Gamantyo Hendrantoro, M.Eng.,Ph.D.

Electrical Engineering Department
Faculty of Intelegent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Analisa Kinerja Teknik Hybrid Beamforming Pada Sistem Large Scale Array Multiuser MISO Terhadap Kanal Acak**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2020

Dicky Rahmadi Prasetya
NRP. 0711 16 4000 0093

ANALISA KINERJA TEKNIK HYBRID BEAMFORMING PADA SISTEM LARGE SCALE ARRAY MULTIUSER MISO TERHADAP KANAL ACAK

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Telekomunikasi Multimedia
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dr. Ir. Puji Handayani, M.T.
NIP. 196605101992032002

SURABAYA
JUNI, 2020

[Halaman Ini Sengaja Dikosongkan]

ANALISA KINERJA TEKNIK HYBRID BEAMFORMING PADA SISTEM LARGE SCALE ARRAY MULTIUSER MISO TERHADAP KANAL ACAK

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Telekomunikasi Multimedia
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing II

Prof. Ir. Gamantyo Hendrantoro, M.Eng.,Ph.D.
NIP. 197011111993031002

SURABAYA
JUNI, 2020

[Halaman Ini Sengaja Dikosongkan]

ANALISA KINERJA TEKNIK HYBRID BEAMFORMING PADA SISTEM LARGE SCALE ARRAY MULTIUSER MISO TERHADAP KANAL ACAK

Nama : **Dicky Rahmadi Prasetya**
NRP : **07111640000093**
Pembimbing I : **Dr. Ir. Puji Handayani, M.T.**
Pembimbing II : **Prof. Ir. Gamantyo Hendrantoro, M.Eng.,Ph.D.**

ABSTRAK

Untuk mendukung teknologi 5G dalam memberikan layanan data kecepatan tinggi diperlukan large scale array yang menggunakan antena dalam jumlah banyak untuk mengarahkan pola radiasi ke user tertentu. Teknologi beamforming pada teknologi telekomunikasi sebelumnya menggunakan metode *full digital beamforming* yang tentu tidak cocok jika diterapkan pada teknologi telekomunikasi 5G. Hal ini disebabkan karena metode *full digital beamforming* membutuhkan satu buah *RF chain* tiap antena yang tentu akan menyebabkan peningkatan biaya dan peningkatan konsumsi daya dari komponen RF chain pada frekuensi yang tinggi.

Pada tugas akhir ini digunakan metode *hybrid beamforming* yang mana mampu mengurangi jumlah penggunaan dari RF chain. Pada tugas akhir ini digunakan sistem multiuser MISO dengan tujuan untuk menguji kinerja metode *hybrid beamforming* pada sistem antena tersebut. Kanal yang digunakan dalam pengujian kinerja dari metode *hybrid beamforming* adalah kanal *independent rayleigh fading* dan *uniformly random line-of-sight*. Dalam menganalisa kinerja dari metode *hybrid beamforming* akan digunakan parameter *spectral efficiency versus signal-to-noise ratio (SNR)*. Dari hasil analisa didapatkan bahwa metode HB memiliki rata-rata kinerja SE lebih baik pada kanal IRF dibanding pada kanal URLOS. Kinerja tersebut dapat dibuktikan dengan pengujian pada SNR 0 dB hingga 30 dB, rata-rata kinerja dari metode HB pada kanal IRF sebesar 17,4758 bits/s/Hz sedangkan pada kanal URLOS sebesar 12,3334 bits/s/Hz.

Kata Kunci: Hybrid Beamforming, Multiuser MISO Antena, Independent Rayleigh Fading, UR-LoS, Spectral Efficiency, Signal-to -Noise Ratio (SNR).

[Halaman Ini Sengaja Dikosongkan]

PERFORMANCE ANALYSIS OF HYBRID BEAMFORMING TECHNIQUES USING LARGE SCALE ARRAY MULTIUSER MISO SYSTEM ON RANDOM CHANNELS

Name : **Dicky Rahmadi Prasetya**
NRP : **07111640000093**
Supervisor I : **Dr. Ir. Puji Handayani, M.T.**
Supervisor II : **Prof. Ir. Gamantyo Hendrantoro, M.Eng.,Ph.D.**

ABSTRACT

To support 5G technology in providing high speed data services, a large scale array is needed that uses large amounts of antenna to direct the radiation pattern to a particular user. The beamforming technology in the previous telecommunications technology uses the full digital beamforming method which is certainly not suitable if applied to the 5G telecommunications technology. This is because the full digital beamforming method requires one RF chain per antenna which will certainly cause increased costs and increased power consumption of RF chain components at high frequencies.

This final project uses hybrid beamforming method which is able to reduce the amount of use of the RF chain. In this final project the MISO multiuser system is used with the aim to test the performance of the hybrid beamforming method on the antenna system. The canals used in testing the performance of the hybrid beamforming method are independent rayleigh fading channels and uniformly random line-of-sight. In analyzing the performance of the hybrid beamforming method the parameters of spectral efficiency versus signal-to-noise ratio (SNR) will be used. From the analysis it was found that the HB method has an average SE performance better on IRF channels than on URLOS channels. This performance can be proven by testing at SNR 0 dB to 30 dB, the average performance of the HB method on IRF channels is 17.4758 bits / s / Hz while on URLOS channels it is 12.3334 bits / s / Hz.

Keywords: Hybrid Beamforming, Multiuser MISO Antena, Independent Rayleigh Fading, UR-LoS, Spectral Efficiency, Singal-to- Noise Ratio (SNR)

[Halaman Ini Sengaja Dikosongkan]

KATA PENGANTAR

Puji syukur penulis ucapkan kehadirat Allah SWT, karena atas nikmat dan rahmat karunianya penulis dapat menyelesaikan tugas akhir yang berjudul:

Analisa Kinerja Teknik Hybrid Beamforming pada Sistem Large Scale Array Multiuser MISO terhadap Kanal Acak

Tugas akhir ini disusun untuk memenuhi syarat menyelesaikan studi Strata-1 di departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember Surabaya. Selama penyusunan tugas akhir penulis mendapatkan banyak dorongan, bimbingan, arahan, dan uluran tangan dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT atas limpahan rahmat-Nya, petunjuk-Nya, serta karunia-Nya.
2. Kedua orang tua yang tercinta, Crisna Juda Prasetya, ST, Dwi Indrijani S.Sos, serta adik yang tersayang Raditya Bagas Prasetya yang selalu memberi dukungan dan semangat.
3. Ibu Dr. Ir. Puji Handayani, M.T. dan Bapak Prof. Dr. Ir. Gamantyo Hendrantoro, Ph.D. yang telah memberikan arahan, bimbingan, dukungan, serta motivasi sehingga penulis mampu menyelesaikan tugas akhir.
4. Teman seperjuangan tugas akhir tim "Hybrid Beamforming" Putu Agus Kartika Adhi Pratama.
5. Keluarga pertama di kampus perjuangan, e56, juga keluarga Telkom yang selalu memberikan dukungan dan motivasi dalam proses pengerjaan tugas akhir.

Semoga Tugas Akhir ini dapat bermanfaat bagi pembacanya baik mahasiswa maupun peneliti. Penulis menyadari bahwa dalam tugas akhir ini masih banyak kekurangan sehingga diperlukan kritik dan saran untuk menyempurnakannya.

Surabaya, Juni 2020

Dicky Rahmadi Prasetya

[Halaman Ini Sengaja Dikosongkan]

DAFTAR ISI

PERNYATAAN KEASLIAN i
ABSTRAK i
ABSTRACT iii
KATA PENGANTAR v
DAFTAR ISI vii
DAFTAR GAMBAR x
DAFTAR TABEL xiii
1 BAB I PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Perumusan Masalah 1
1.3 Tujuan Tugas Akhir 2
1.4 Batasan Masalah 2
1.5 Metodologi 2
1.6 Sistematika Penulisan 3
1.7 Relevansi 4
2 BAB II TEORI PENUNJANG 5
2.1 Teknologi 5G 5
2.1.1 Perkembangan Teknologi 5G 5
2.1.2 Milimeter-Wave 7
2.2 Hybrid Beamforming 8
2.2.1 Analog dan Digital Beamforming 9
2.3 Sistem Multiuser MISO 9
2.3.1 Desain precoding sistem Multiuser MISO 11
2.4 Kanal 15
2.4.1 Independent Rayleigh Fading 15
2.4.2 Uniformly Random Line-of-Sight (URLoS) 16

2.5 Pola Radiasi 17
3 BAB III METODOLOGI PENGERJAAN 19
3.1 Parameter Sistem 19
3.2 Pemodelan Sistem 21
3.2.1 Pemodelan Transmitter 21
3.2.2 Pemodelan Kanal 23
3.2.3 Pemodelan Receiver 23
3.2.4 Demapping 24
3.2.5 Simulasi Analisa *Bit Error Rate* 24
3.2.6 Simualsi *Spectral Efficiency versus Signal-to-Noise Ratio (SNR)* 24
4 BAB IV HASIL DAN ANALISA DATA 26
4.1 Perbandingan kinerja antara *Hybrid* dan *Full Digital Beamforming* pada kanal *Independent Rayleigh Fading* dan *Uniformly Random Line-of-Sight* 26
4.1.1 Kanal *Independent Rayleigh Fading* 26
4.1.2 Kanal Uniformly Random Line-of-Sight 29
4.2 Perbandingan kinerja metode HB pada kanal IRF dan URLOS dan perbandingan kinerja metode FDB pada kanal IRF dan URLOS 33
4.2.1 Perbandingan metode HB pada kanal IRF dan URLOS.. 33
4.2.2 Perbandingan metode FDB pada kanal IRF dan URLOS 36
4.3 Perbandingan kinerja metode HB pada kanal IRF dan URLOS yang memiliki perbedaan jumlah RF *Chain*, jumlah antena pemancar, dan jumlah antena penerima 39
4.3.1 Perbandingan metode HB dengan perbedaan jumlah RF *Chain* menggunakan kanal IRF 39
4.3.2 Perbandingan metode HB dengan perbedaan jumlah RF Chain menggunakan kanal URLOS 42

4.3.3 Perbandingan metode HB dengan perbedaan jumlah antenna pemancar pada kanal IRF 46
4.3.4 Perbandingan menggunakan metode HB dengan perbedaan jumlah antenna pemancar pada kanal URLOS 49
4.3.5 Perbandingan menggunakan metode HB dengan perbedaan jumlah user/antena penerima pada kanal IRF 52
4.3.6 Perbandingan menggunakan metode HB dengan perbedaan jumlah user/antena penerima pada kanal URLOS 54
4.4 Tambahan 58
4.4.1 Korelasi Kanal 58
4.4.2 Perbandingan metode HB dengan perbedaan sudut pada kanal URLOS yang dibandingkan dengan kanal IRF 59
4.4.3 Perbandingan metode HB dengan perbedaan jumlah antena pemancar pada kanal URLOS 5 derajat dan URLOS 20 derajat 61
4.4.4 Pola radiasi dari metode HB pada kanal IRF dan pada kanal URLOS 20 derajat 64
4.5 Diskusi 69
5 BAB V KESIMPULAN DAN SARAN 75
5.1 Kesimpulan 75
5.2 Saran 76
Daftar Pustaka 77
LAMPIRAN 79
6 BIOGRAFI PENULIS 290

DAFTAR GAMBAR

Gambar 2.1 . Pengembangan yang dilakukan dari teknologi telekomunikasi sebelumnya ke teknologi telekomunikasi tahun 2020 [2] 5
Gambar 2.2 . Pengembangan yang dilakukan dari teknologi telekomunikasi sebelumnya ke teknologi telekomunikasi tahun 2020 [2] 5
Gambar 2.3 Skenario Pengembangan Teknologi Telekomunikasi diatas tahun 2020 [2] 6
Gambar 2.4 Kriteria yang harus dimiliki 5G [2] 7
Gambar 2.5 Blok diagram teknik hybrid beamforming[4] 8
Gambar 2.6 Blok diagram multiuser MISO 10
Gambar 2.7 Blok diagram sistem MISO[8] 10
Gambar 2.8 Flowchart Algoritma 13
Gambar 2.9 *Uniform linear array* dengan λ/2 dengan hamburan isotropic[11] 16
Gambar 2.10 λ/2 uniform linear array dengan lingkungan propagasi line-of-sight[11] 17
Gambar 3.1 Diagram alur pengerjaan tugas akhir 20
Gambar 3.2 Diagram model sistem hybrid dan full digital beamforming pada Multiuser MISO 21
Gambar 3.3 Plot stem dari sampel bit BPSK sinyal informasi 22
Gambar 3.4 Model receiver multiuser MISO 23
Gambar 4.1 Grafik perbandingan kinerja SE antara metode HB dan FDB pada kanal IRF 27
Gambar 4.2 Grafik perbandingan kinerja BER antara metode HB dan FDB pada kanal IRF 28
Gambar 4.3 Grafik perbandingan kinerja SE antara metode HB dan FDB pada kanal URLOS 30
Gambar 4.4 Grafik perbandingan kinerja BER antara metode HB dan FDB pada kanal URLOS 31
Gambar 4.5 Grafik perbandingan kinerja SE metode HB pada kanal IRF dan URLOS 33
Gambar 4.6 Grafik perbandingan kinerja BER metode pada kanal IRF dan URLOS 34
Gambar 4.7 Grafik perbandingan kinerja SE metode FDB pada kanal IRF dan URLOS 36

Gambar 4.8 Grafik perbandingan kinerja BER metode pada kanal IRF dan URLOS 37
Gambar 4.9 Grafik perbandingan kinerja SE pada kanal IRF dengan menggunakan beda jumlah RF *chain* 39
Gambar 4.10 Grafik perbandingan kinerja BER pada kanal IRF dengan menggunakan beda jumlah RF *chain* 40
Gambar 4.11 Grafik perbandingan kinerja SE pada kanal URLOS dengan menggunakan beda jumlah RF *chain* 43
Gambar 4.12 Grafik perbandingan kinerja BER pada kanal URLOS dengan menggunakan beda jumlah RF *chain* 44
Gambar 4.13 Grafik perbandingan kinerja SE pada kanal IRF dengan menggunakan beda jumlah antena pemancar 46
Gambar 4.14 Grafik perbandingan kinerja BER pada kanal IRF dengan menggunakan beda jumlah antena pemancar 47
Gambar 4.15 Grafik perbandingan kinerja SE pada kanal URLOS dengan menggunakan beda jumlah antena pemancar 49
Gambar 4.16 Grafik perbandingan kinerja BER pada kanal URLOS dengan menggunakan beda jumlah antena pemancar 50
Gambar 4.17 Grafik perbandingan kinerja SE pada kanal IRF dengan menggunakan beda jumlah antena penerima 52
Gambar 4.18 Grafik perbandingan kinerja BER pada kanal IRF dengan menggunakan beda jumlah antena penerima 53
Gambar 4.19 Grafik perbandingan kinerja SE pada kanal URLOS dengan menggunakan beda jumlah antena penerima 55
Gambar 4.20 Grafik perbandingan kinerja BER pada kanal URLOS dengan menggunakan beda jumlah antena penerima 56
Gambar 4.21 Grafik perbandingan kinerja SE metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF 59
Gambar 4.22 Grafik perbandingan kinerja BER metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF 60
Gambar 4.23 Grafik perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat 62
Gambar 4.24 Grafik perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat 63
Gambar 4.25 Plot polar kanal IRF 64
Gambar 4.26 Plot polar kanal IRF dengan rekomendasi sudut 64
Gambar 4.27 Plot rectangular kanal IRF dengan rekomendasi sudut .. 65

Gambar 4.28 Plot rectangular kanal IRF 65
Gambar 4.29 Plot polar kanal URLOS 5 derajat 66
Gambar 4.30 Plot polar kanal URLOS 5 derajat dengan plot sudut 6, 11, 16, dan 21 66
Gambar 4.31 Plot rectangular kanal URLOS 5 derajat..... 67
Gambar 4.32 Plot rectangular kanal URLOS 5 derajat dengan plot sudut 6, 11, 16, dan 21 67
Gambar 4.33 Plot rectangular kanal URLOS 20 derajat..... 68
Gambar 4.34 Plot polar kanal URLOS 20 derajat dengan plot sudut -40, -20, 0, 20 68
Gambar 4.35 Plot rectangular kanal URLOS 20 derajat..... 69
Gambar 4.36 Plot rectangular kanal URLOS 20 derajat dengan plot sudut 69

DAFTAR TABEL

Tabel 4.1 Data SE metode HB dan FDB pada kanal IRF 27
Tabel 4.2 Data BER metode HB dan FDB pada kanal IRF 28
Tabel 4.3 Data SE metode HB dan FDB pada kanal URLOS 30
Tabel 4.4 Data BER metode HB dan FDB pada kanal URLOS 31
Tabel 4.5 Data SE metode HB pada kanal IRF dan URLOS 33
Tabel 4.6 Data BER metode HB pada kanal IRF dan URLOS 34
Tabel 4.7 Data SE metode FDB pada kanal IRF dan URLOS 36
Tabel 4.8 Data BER metode FDB pada kanal IRF dan URLOS 37
Tabel 4.9 Data SE pada kanal IRF dengan perbedaan jumlah RF *chain* 40
Tabel 4.10 Data BER pada kanal IRF dengan perbedaan jumlah RF *chain* 41
Tabel 4.11 Data SE pada kanal URLOS dengan perbedaan jumlah RF *chain* 43
Tabel 4.12 Data BER pada kanal URLOS dengan perbedaan jumlah RF *chain* 44
Tabel 4.13 Data SE pada kanal IRF dengan perbedaan jumlah antena pemancar 47
Tabel 4.14 Data BER pada kanal IRF dengan perbedaan jumlah antena pemancar 47
Tabel 4.15 Data SE pada kanal URLOS dengan perbedaan jumlah antena pemancar 49
Tabel 4.16 Data BER pada kanal URLOS dengan perbedaan jumlah antena pemancar 50
Tabel 4.17 Data SE pada kanal IRF dengan perbedaan jumlah antena penerima 52
Tabel 4.18 Data BER pada kanal IRF dengan perbedaan jumlah antena penerima 53
Tabel 4.19 Data SE pada kanal URLOS dengan perbedaan jumlah antena penerima 55
Tabel 4.20 Data BER pada kanal URLOS dengan perbedaan jumlah antena penerima 56
Tabel 4.21 Korelasi kanal IRF 58
Tabel 4.22 Korelasi kanal URLOS 5 derajat 58
Tabel 4.23 Korelasi kanal URLOS 20 derajat 58
Tabel 4.24 Data perbandingan kinerja SE metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF 59

Tabel 4.25 Data perbandingan kinerja BER metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF 60
Tabel 4.26 Data perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat..... 62
Tabel 4.27 Data perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat..... 63

BAB I PENDAHULUAN

1.1 Latar belakang

Beberapa tahun terakhir, teknologi serta industri telekomunikasi berkembang pesat. Dibuktikan dengan peningkatan penggunaan trafik dari tahun ke tahun. Diikuti dengan pertambahan jumlah pengguna perangkat telekomunikasi yang sejalan dengan penggunaan platform(aplikasi) yang terus meningkat, membuat penggunaan trafik semakin padat tiap tahunnya. Penggunaan video yang juga meningkat tiap tahun baik dari jumlah pengguna, maupun dari tingkat kualitasnya, juga menuntut untuk dapat menggunakan layanan telekomunikasi yang memiliki data rate kencang [1]. Teknologi telekomunikasi 5 Generation atau 5G hadir dengan menawarkan banyak keunggulan dari teknologi sebelumnya diharapkan mampu menjawab kebutuhan serta tantangan di era mendatang [2]. Teknologi 5G memanfaatkan frekuensi yang lebih tinggi dari teknologi sebelumnya, menggunakan gelombang millimeter atau millimeter-wave yang memiliki panjang gelombang dengan rentang 1 mm – 10 mm atau menggunakan frekuensi dengan rentang 30 GHz – 300 GHz [3]. Sejumlah penelitian telah dilakukan terkait dengan penggunaan millimeter-wave untuk aplikasi 5G [4] [5], dan dari hasil penelitian yang telah dilakukan, beberapa diantaranya telah memenuhi kriteria atau standar atau key indicator yang dimiliki 5G [2].

Sistem yang digunakan pada teknologi sebelumnya, yakni full-digital precoding memiliki kompleksitas yang tinggi, selain itu sistem tersebut juga tidak efisien, dan membutuhkan biaya yang cukup besar. Penelitian ini akan membahas mengenai teknik hybrid beamforming, yang tidak hanya menggunakan digital precoding saja, namun juga menggunakan analog precoding. Teknik hybrid beamforming mampu mengurangi kompleksitas sistem, membuat sistem lebih efisien, juga memangkas biaya yang diperlukan untuk merancang sistem tersebut [6].

Penelitian ini menggunakan sistem multiuser MISO (multi-input-single-output) yang nantinya akan diuji pada kanal *independent rayleigh fading* dan *uniformly random line-of-sight*. Kemudian akan dapat diamati kinerja *spectral efficiency* dari metode *hybrid beamforming* pada kedua kanal.

1.2 Perumusan Masalah

Rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana pengaruh perubahan RF *chain* terhadap kinerja dari metode *hybrid beamforming* dengan sistem Multiuser MISO?
2. Bagaimana kinerja dari metode Hybrid Beamforming dengan sistem Multiuser MISO pada kanal *independent rayleigh fading* dan *uniformly random line-of-sight*?

1.3 Tujuan Tugas Akhir

Adapun tujuan dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Menguji kinerja dari metode *hybrid beamforming* terhadap kanal *independent rayleigh fading* dan *uniformly random line-of-sight*.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Modulasi yang digunakan untuk *bit stream* informasi adalah BPSK.
2. Teknik yang digunakan dalam penelitian ini yaitu analog precoding dan digital precoding.
3. Antena dengan sistem Multiuser MISO.
4. Jenis kanal yang diujikan yaitu Rayleigh Fading Channel dan Uniformly Random Line-of-Sight (UR-LoS) Channel.
5. Parameter pengujian yang dianalisa yakni pendekatan *spectrum efficiency versus signal-to-noise ratio* (SNR).

1.5 Metodologi

Metodologi yang digunakan pada penelitian tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

   Pada tahap studi literatur, hal yang perlu dicari oleh penulis adalah referensi dari publikasi ilmiah, jurnal ilmilah, buku pendukung, dan sumber lainnya sebagai penunjang untuk mengerjakan tugas akhir ini. Kata kunci yang digunakan antara lain adalah millimeter-wave, modulasi BPSK, teknik hybrid beamforming, antenna multiuser MISO, kanal Independent Rayleigh Fading, dan kanal Uniformly Random Line-of-Sight (UR-LoS).
2. Pemodelan dan Perancangan

   Tahapan selanjutnya adalah memodelkan semua bagian yang ingin disimulasikan dan diujikan mulai dari metode *hybrid beamforming,* metode *full digital beamforming,* kanal *independent rayleigh fading,* kanal *uniformly random line-of-*

sight, juga kondisi-kondisi yang ingin diujikan. Tahap simulasi ini dilakukan dengan bantuan software MATLAB.

3. Hasil dan Analisa Data

   Hasil dari pengujian pada dua kanal yang berbeda akan di analisis hasilnya menggunakan parameter *spectral efficiency versus Signal-to-Noise Ratio (SNR)*. Hasil pengujian antara kedua kanal juga akan dibandingkan pada kondisi-kondisi yang telah ditentukan.

4. Kesimpulan dan Saran

   Dari hasil pengujian dan analisa yang telah dilakukan akan ditarik kesimpulan-kesimpulan yang menggambarkan hasil dari simulasi. Kemudian akan diberikan saran mengenai kegiatan penelitian yang telah dilakukan.

1.6 Sistematika Penulisan

Sistematika penulisan pada laporan tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan.
Pada bab ini dijelaskan mengenai latar belakang, permasalahan, tujuan, batasan masalah, metodologi, sistematika penulisan dan relevansi pada tugas akhir.

BAB II Teori Penunjang
Pada bab ini dijelaskan mengenai teori perkembangan teknologi 5G, spektrum yang digunakan pada teknologi 5G, teori hybrid beamforming, macam antena serta kanal yang akan digunakan dalam simulasi tugas akhir.

BAB III Metodologi
Pada bab ini dijelaskan menganai perancangan dan pemodelan sistem Multiuser MISO pada kabal *independent rayleigh fading* dan *uniformly random line-of-sight*

BAB IV Hasil dan Analisa Data
Pada bab ini dijelaskan mengenai analisa terhadap hasil simulasi yang telah dilakukan menggunakan metode *hybrid beamforming* dengan sistem Multiuser MISO

BAB V Kesimpulan dan Saran

Pada bab ini penulis akan memberikan kesimpulan dari hasil analisa simulasi yang telah dilakukan serta memberi saran yang dapat di implementasikan pada penelitian mendatang.

1.7 Relevansi

Hasil dari penelitian tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

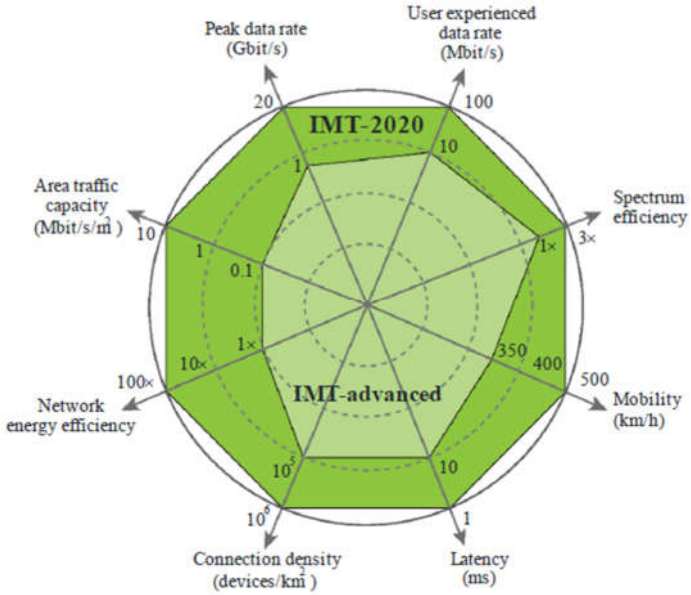
1. Memberikan gambaran mengenai kinerja dari sistem Multiuser MISO pada kanal *independent rayleigh fading* dan *uniformly random line-of-sight*.

BAB II TEORI PENUNJANG

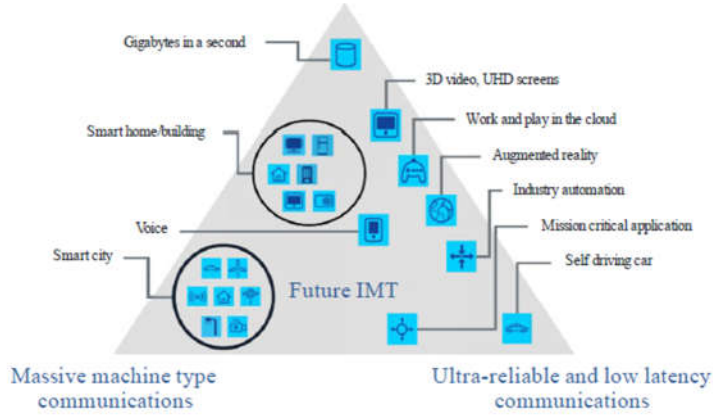
2.1 Teknologi 5G

2.1.1 Perkembangan Teknologi 5G

Perkembangan zaman selalu diikuti dengan perkembangan teknologi, khusunya teknologi telekomunikasi. Teknologi telekomunikasi semakin dituntut untuk melayani pengguna dalam skala yang massif yang disertai dengan kebutuhan-kebutuhan yang mengikutinya. Dahulu teknologi telekomunikasi yang hanya memfasilitasi komunikasi antar manusia, kini memiliki tantangan untuk dapat memfasilitasi komunikasi antar mesin. Tantangan kolaborasi teknologi telekomunikasi dalam melayani manusia dan mesin memiliki harapan agar kedepannya menjadi fasilitator dalam mengembangkan sektor industry, pendidikan, kesehatan, keamanan, transportasi, dan edukasi[2].



Gambar 2.1 . Pengembangan yang dilakukan dari teknologi telekomunikasi sebelumnya ke teknologi telekomunikasi tahun 2020 [2]



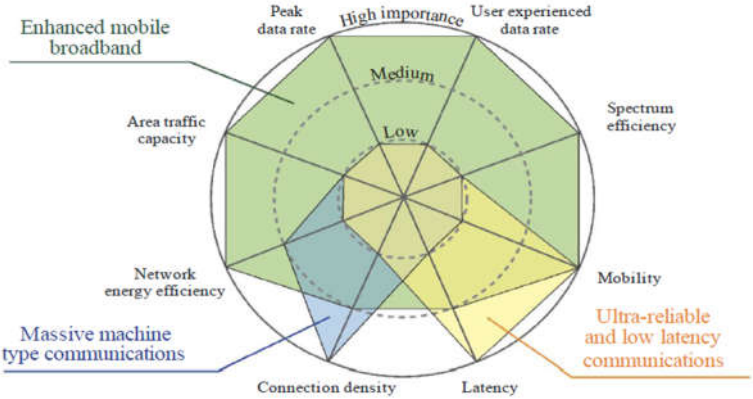
Gambar 2.3 Skenario Pengembangan Teknologi Telekomunikasi diatas tahun 2020 [2]

Melihat lebih detail peran dari teknologi telekomunikasi di masa mendatang, maka teknologi telekomunikasi harus memiliki beberapa kriteria sebagai berikut[2]:

1. Infrastruktur nirkabel untuk menghubungkan dunia: teknologi telekomunikasi akan menjadi pilar yang mewadahi layanan seluler dan pertukaran informasi. Kedepannya, pengguna pribadi maupun pengguna profesional akan difasilitasi dengan berbagai macam aplikasi dan layanan yang mampu menunjang kehidupan yang lebih baik.
2. Membuka peluang terjadinya pasar baru di bidang informasi dan teknologi(digital): perkembangan teknologi telekomunikasi kelak akan membuka peluang terbukanya pasar baru di bidang informasi dan teknologi. Seperti *big data analyzer, data science, programmer*.
3. Menjembatani kesenjangan platform digital: teknologi telekomunikasi kedepan diharapkan mampu mengurangi gap yang ada dengan membuat *platform* maupun *device* yang mudah digunakan dan dikendalikan.
4. Cara baru dalam komunkasi: teknologi telekomunikasi dituntut untuk menyediakan layanan yang mampu menghubungkan manusia tanpa dibatasi oleh jarak dan waktu.

5. Bentuk baru dalam pendidikan: perkembangan teknologi telekomunikasi diharapkan mampu mempermudah manusia dalam belajar dengan menyediakan *digital textbooks, cloud-based storage of knowledge*, juga membuat *e-learning* dan sebagainya.

Secara garis besar, perkembangan teknologi 5G dibagi menjadi tiga garis besar: *Enhanced mobile broadband (eMBB)*, *Ultra-reliable and low latency communications (URLLC)*, dan *Massive machine type communications (mMTC)*. Tiap pengembangan memiliki ranah kembang yang akan dispesifikkan berdasarkan pengguna layanan telekomunikasi. Seperti halnya URRLC berfokus pada *self-driving car*, *augmented reality*, *industry automation*, yang memiliki ciri *low latency* dan *High Reliability (low error rate)*. Sedangkan mMTC lebih berfokus pada perancangan dan pembangunan *smart city* dengan kriteria yang wajib dimiliki yaitu *improved link budget, low device complexity, long device battery life, high density device deployment*. Komunikasi antar manusia termasuk kedalam *Enchanced Mobile Broadband* yang mana akan dirancang memiliki kritetia *low latency dan higher spectral efficiency*[2].



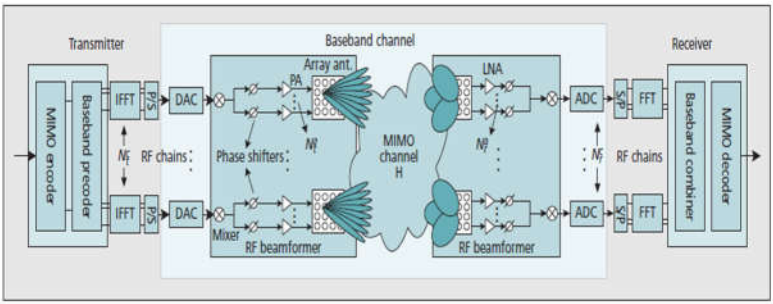
Gambar 2.4 Kriteria yang harus dimiliki 5G [2]

2.1.2 Milimeter-Wave

Teknologi 5G yang hadir, menggunakan *millimeter-wave* dengan rentang panjang gelombang 1 mm – 10 mm, atau dengan rentang frekuensi sekitar 30 GHz – 300 GHz[3][4][5]. Rentang frekuensi 30 GHz – 300 GHz dipilih karena selain memiliki kelebihan bandwidth yang sangat lebar, juga mampu memfasilitasi terjadinya transmisi data rate yang kencang (*high speed data rate transmission*). Berbeda dengan penggunaan frekuensi rendah, memang panjang gelombang yang dihasilkan lebih panjang, yang mampu melayani area yang luas, juga dan menembus bangunan gedung dengan mudah, *millimeter-wave* yang memiliki panjang gelombang yang pendek mampu lebih menghemat, mengoptimalkan, juga mengefisiensi spektrum frekuensi yang digunakan, juga menambah tingkat keamanan transmisi[3]. Beberapa penelitian telah dilakukan dan mendapatkan hasil yang sesuai harapan yakni *data rate* kencang yang menembus angka satu hingga puluhan Gbps, sehingga millimeter-wave bisa digunakan dan diterapkan untuk teknologi 5G, dengan segala pertimbangan dan kelebihan yang ditawarkan[4][5].

2.2 Hybrid Beamforming

Precoding adalah teknik yang mendukung terjadinya beamforming pada multi antenna, pada kasus ini array antenna. Pada teknologi sebelumnya, precoding yang digunakan adalah *full-digital precoding* yang mana membutuhkan satu *RF chain* tiap satu elemen antenanya. Precoding pada teknologi sebelumnya dirasa kurang layak digunakan untuk array antenna dalam skala besar, dikarenakan tingginya biaya dan tingginya konsumsi daya dari komponen *RF chain* apabila dioperasikan menggunakan *millimeter-wave*.



Gambar 2.5 Blok diagram teknik hybrid beamforming[4]

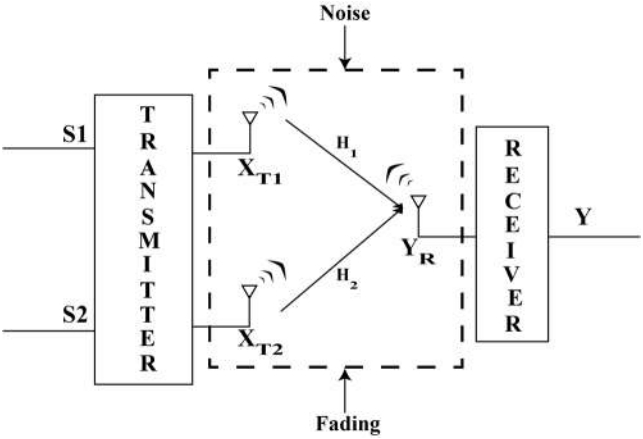
2.2.1 Analog dan Digital Beamforming

Analog beamforming adalah sebuah metode yang telah diusulkan lebih dari 50 tahun yang lalu. Konsep dari analog beamforming ini adalah untuk mengontrol fase dari tiap sinyal transmisi menggunakan *phase shifter* dengan harga yang terjangkau. Antena analog beamforming tersusun oleh matriks hybrid dan *fixed phase shifter*. *Switch*/saklar frekuensi radio selektif digunakan untuk memfasilitasi terjadinya fungsi *beam steering* yang berfungsi dalam mengarahkan beam [12].

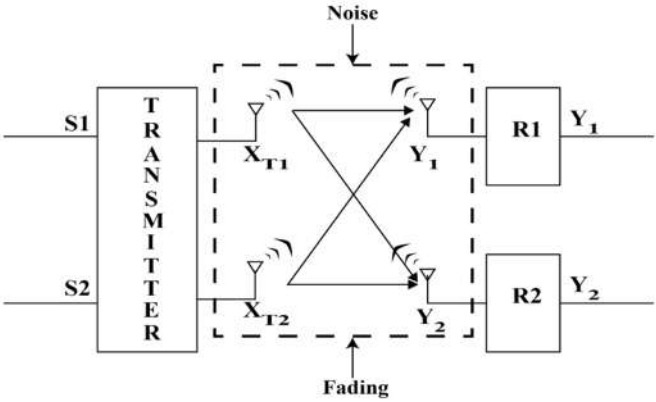
Digital beamforming memiliki banyak kegunaan seperti estimasi DOA (*direction-of-arrival*), pengontrolan terprogram dari pola radiasi antena, dan pengarahan adaptif terhadap beamnya. Beamforming lama (*full digital beamforming*) hanya bekerja/diterapkan pada baseband yang dapat berfungsi untuk mengontrol fase dan amplitude dari sinyal. Frekuensi *carrier* sinyal yang telah terproses pada baseband tadi perlu untuk mendapat *up converter* dari RF chain yang berisikan *digital-to-analog converter* (DAC), *mixer,* and *power amplifier*. Respon dari RF chain tersebut nantinya akan dikombinasikan dengan elemen antena, dengan kata lain tiap elemen array antena harus diperkuat oleh sebuah RF chain. Hal ini yang menyebabkan penggunaan metode *full digital beamforming* dengan jumlah antena yang relatif besar (akibat penggunaan frekuensi yang tinggi) tidak cocok, akibat dari penggunaan daya yang boros dan biaya yang dibutuhkan untuk merealisasikan mahal [12].

2.3 Sistem Multiuser MISO

Penggunaan *millimeter-wave* sebagai media transmisi menuntut pemancar untuk mampu memfasilitasi terbentuknya *beamforming* yang sangat terarah ke pengguna. *Millimeter-wave* juga membuka peluang untuk penggunaan antenna dalam jumlah yang banyak dalam satu wujud dikarenakan memiliki panjang gelombang yang pendek dibanding dengan penggunaan frekuensi pada teknologi sebelumnya[6]. Dua hal ini menjadi penyebab munculnya kemungkinan penggunaan banyak antenna (atau *array antenna*) pada pemancar. Semakin banyak antenna yang digunakan, maka bentuk beam yang dibentuk akan semakin terarah[7].



Gambar 2.7 Blok diagram sistem MISO[8]



Gambar 2.6 Blok diagram multiuser MISO

Sistem MISO atau multi-input-single-output adalah sistem dengan antena pemancar jamak dan hanya memiliki satu penerima dengan satu antena tiap penerima, sedangkan sistem multi user MISO adalah sistem dengan antena pemancar jamak yang memiliki beberapa penerima atau

user equipment (UE) dengan satu antena tiap penerima. Kapasitas kanal MISO bisa ditentukan dengan rumus berikut [8]:

$$C = M_t B log_2\left(1 + \frac{S}{N}\right) \quad (2.1)$$

Keterangan:

C = Kapasitas kanal (bits/s)
Mt = Jumlah antena transmisi
B = Bandwidth (Hz)
$\frac{S}{N}$ = Signal-to-noise ratio

Spectral efficiency berbeda dengan kapasitas kanal. Kapasitas kanal adalah sebuah parameter yang menjelaskan kapasitas data rate yang dapat ditampung oleh suatu kanal per satuan waktu (sekon) dengan satuan bits/s. Sedangkan *spectral efficiency* adalah sebuah parameter yang mana menjelaskan pengefisienan penggunaan dari spektrum frekuensi, yang dapat diketahui dengan SE = C / B, dimana SE adalah *spectral efficiency* dalam satuan bits/s/Hz, C adalah kapasitas kanal dan B adalah bandwidth. Kemudian, *spectral efficiency* yang dimaksud hanya pada sisi beamforming dan tidak pada keseluruhan sistem telekomunikasi.

2.3.1 Desain precoding sistem Multiuser MISO

2.3.1.1 Desain Digital Precoder

Pada tugas akhir ini dipertimbangkan penggunaan ZF (*Zero forcing*) beamforming dengan alokasi daya sebagai bagian dari *digital precoder* dengan dimensi rendah untuk mengatasi *inter-user interference*. *Digital precoder* dengan *zero forcing* dapat dicari dengan rumus berikut [4]:

$$\mathbf{V}_D^{ZF} = \mathbf{V}_{RF}^H \mathbf{H}^H (\mathbf{H}\mathbf{V}_{RF}\mathbf{H}^H)^{-1}\mathbf{P}^{\frac{1}{2}} = \widetilde{\mathbf{V}}_D \mathbf{P}^{\frac{1}{2}} \quad (2.2)$$

dimana $\mathrm{H} = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_K]^H, \widetilde{\mathbf{V}}_D = \mathbf{V}_{RF}^H \mathbf{H}^H (\mathbf{H}\mathbf{V}_{RF}\mathbf{V}_{RF}^H\mathbf{H}^H)^{-1}$ dan $\mathbf{P} = diag(p_1, \cdots, p_K)$ dengan p_K mendenotasikan daya yang diterima pada user ke-k. Dengan mengasumsikan RF *precoder* yang mungkin/layak digunakan, semua desain daya tersebut dapat diperoleh dari persamaan berikut:

$$\max_{p_1, \cdots, p_K} \sum_{k=1}^{K} \beta_k \, log_2\left(1 + \frac{p_k}{\sigma^2}\right) \quad (2.3)$$

$$s.t. Tr\left(\widetilde{\mathbf{Q}}\mathbf{P}\right) \leq P, \quad (2.4)$$

dimana $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{V}}_D^H \mathbf{V}_{RF}^H \mathbf{V}_{RF} \widetilde{\mathbf{V}}_D$. Solusi optimal untuk menyelesaikan permasalahan tersebut dapat diperoleh dengan cara *water-filling* yaitu:

$$p_k = \frac{1}{\tilde{q}_{kk}} max\left\{\frac{\beta_k}{\lambda} - \tilde{q}_{kk}\sigma^2, 0\right\}, \quad (2.5)$$

dimana $\tilde{q}_{kk}$ adalah elemen diagonal ke-k dari $\widetilde{\mathbf{Q}}$ dan λ dipilih sebagaimana sehingga $\sum_{k=1}^{K} max\left\{\frac{\beta_k}{\lambda} - \tilde{q}_{kk}\sigma^2, 0\right\} = P$.

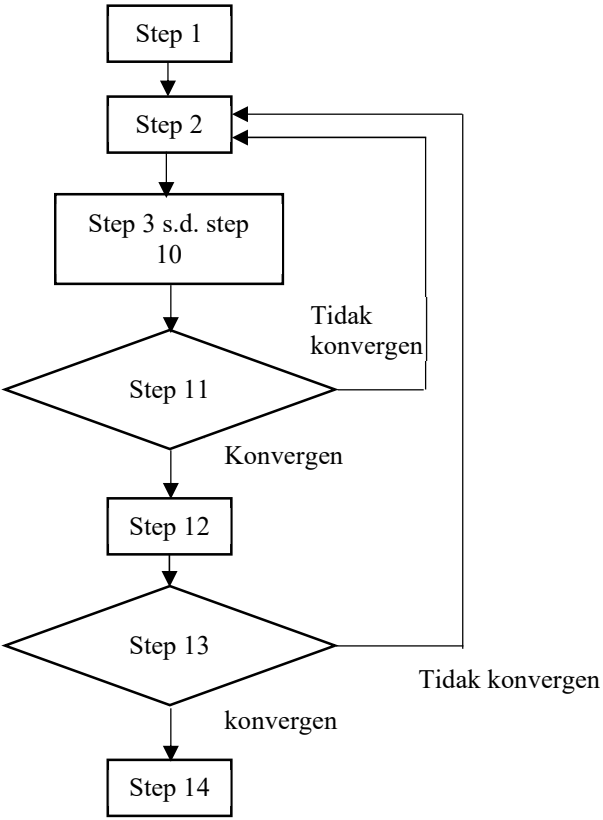
2.3.1.2 Desain RF/Analog Precoder

Desain *hybrid precoder* untuk sistem Multiuser MISO dapat dilihat pada algoritma berikut [4]:

Parameter yang dibutuhkan: β_k, P, σ^2

1. Mulai dengan $\mathbf{V}_{RF}$ yang mungkin/layak dan $\mathbf{P} = \mathbf{I}_K$
2. **for** $j = 1 \rightarrow N^{RF}$ **do**
3. hitung $\mathbf{A}_j = \mathbf{P}^{-\frac{1}{2}}\mathbf{H}\widetilde{\mathbf{V}}_{\mathrm{RF}}^{j}\left(\widetilde{\mathbf{V}}_{\mathrm{RF}}^{j}\right)^{H}\mathbf{H}^{H}\mathbf{P}^{-\frac{1}{2}}$
4. **For** $i = 1 \rightarrow M$ **do**
5. hitung $\zeta_{ij}^{B}, \zeta_{ij}^{D}, \eta_{ij}^{B}, \eta_{ij}^{D}$
6. hitung $\theta_{i,j}^{(1)}$ dan $\theta_{i,j}^{(2)}$ berdasarkan persamaan (3.12)
7. cari $\theta_{ij}^{opt} = argmin\left(\hat{f}\left(\theta_{i,j}^{(1)}\right), \hat{f}\left(\theta_{i,j}^{(2)}\right)\right)$
8. tetapkan $\mathbf{V}_{RF}(i,j) = e^{-j\theta_{ij}^{opt}}$.
9. **end for**
10. **end for**
11. Cek konvergensi dari RF *precoder*. Jika sudah, lanjutkan; jika belum, kembali ke langkah 2.
12. Cari $\mathbf{P} = diag[p_1, \cdots, p_K]$ menggunakan *water-filling* seperti pada persamaan (3.4)
13. Cek konvergensi dari keseluruhan algoritma. Jika sudah, lanjutkan; jika belum, kembali ke langkah 2.
14. Hitung $\mathbf{V}_D = \mathbf{V}_{RF}^{H}\mathbf{H}^{H}(\mathbf{H}\mathbf{V}_{RF}\mathbf{V}_{RF}^{H}\mathbf{H}^{H})^{-1}\mathbf{P}^{\frac{1}{2}}$.

Pada awal algoritma pencarian analog/RF precoder disebutkan bahwa memerlukan beberapa parameter, yang diasumsikan seperti $\beta_k = 1, P = 1\ Watt, dan\ \sigma^2 = 0.1\ sampai\ 1000$.



Gambar 2.8 Flowchart Algoritma

Pada step pertama, maksud dari $\mathbf{V}_{RF}$ yang mungkin/layak adalah $\mathbf{V}_{RF} = \mathbf{1}_{N \times N^{RF}}$ dan $\mathbf{P}$ adalah matriks identitas yang berukuran K x K. Pada step ketiga, yang dimaksud dengan $\widetilde{\mathbf{V}}_{\mathrm{RF}}^{j}$ adalah sub-matriks dari $\mathbf{V}_{\mathrm{RF}}$ yang mana kolom ke-j telah dihilangkan. Pada step kelima, $\zeta_{ij}^{B}, \zeta_{ij}^{D}, \eta_{ij}^{B}, \eta_{ij}^{D}$ dapat dicari melalui algoritma-algoritma berikut:

$$\widetilde{\mathbf{H}} = \mathbf{P}^{-\frac{1}{2}}\mathbf{H} \tag{2.6}$$

$$\mathbf{B}_j = \widetilde{\mathbf{H}}^{H}\mathbf{A}_j^{-2}\widetilde{\mathbf{H}} \tag{2.7}$$

$$\mathbf{D}_j = \tilde{\mathbf{H}}^H \mathbf{A}_j^{-1} \tilde{\mathbf{H}} \quad (2.8)$$

$$\zeta_{ij}^B = \mathbf{B}_j(i,i) + 2Re\left\{\sum_{\boldsymbol{k\neq i, m\neq i}} \mathbf{V}_{\boldsymbol{RF}}^*(k,j)\mathbf{B}_j(k,m)\mathbf{V}_{RF}(m,j)\right\}, \quad (2.9)$$

$$\zeta_{ij}^D = \mathbf{D}_j(i,i) + 2Re\left\{\sum_{\boldsymbol{k\neq i, m\neq i}} \mathbf{V}_{\boldsymbol{RF}}^*(k,j)\mathbf{D}_j(k,m)\mathbf{V}_{RF}(m,j)\right\}, \quad (2.10)$$

$$\eta_{ij}^B = \sum_{l\neq i} \mathbf{B}_j(i,l)\, \mathbf{V}_{RF}(l,j), \quad (2.11)$$

$$\eta_{ij}^D = \sum_{l\neq i} \mathbf{D}_j(i,l)\, \mathbf{V}_{RF}(l,j), \quad (2.12)$$

Pada step keenam, $\theta_{i,j}^{(1)}$ dan $\theta_{i,j}^{(2)}$ dapat dicari melalui algoritma berikut:

$$\theta_{i,j}^{(1)} = -\phi_{i,j} + sin^{-1}\left(\frac{z_{ij}}{|c_{ij}|}\right),$$

$$\theta_{i,j}^{(2)} = \pi - \phi_{i,j} - sin^{-1}\left(\frac{z_{ij}}{|c_{ij}|}\right), \quad (2.13)$$

dimana $c_{ij} = \left(1 + \zeta_{ij}^D\right)\eta_{ij}^B - \zeta_{ij}^B \eta_{ij}^D$, $z_{ij} = Im\left\{2\left(\eta_{ij}^B\right) * \eta_{ij}^D\right\}$, dan

$$\phi_{i,j} = \begin{cases} sin^{-1}\left(\frac{\boldsymbol{Im}\{c_{ij}\}}{|c_{ij}|}\right), jika\ Re\{c_{ij}\} \geq \mathbf{0}, \\ \pi - sin^{-1}\left(\frac{\boldsymbol{Im}\{c_{ij}\}}{|c_{ij}|}\right), jika\ Re\{c_{ij}\} < \mathbf{0}. \end{cases} \quad (2.14)$$

Pada step ketujuh, $\hat{f}\left(\theta_{i,j}^{(1)}\right)$ dan $\hat{f}\left(\theta_{i,j}^{(2)}\right)$ dapat dicari dengan cara mensubstitusikan , $\theta_{i,j}^{(1)}$ dan $\theta_{i,j}^{(2)}$ kedalam $\hat{f}(\mathbf{V}_{RF})$, dimana:

$$\hat{f}(\mathbf{V}_{RF}) = NTr\left(\mathbf{A}_j{}^{-1}\right) - N\frac{\zeta_{ij}^B + 2Re\left\{\mathbf{V}_{RF}^*(i,j)\eta_{ij}^B\right\}}{1+\zeta_{ij}^D + 2Re\left\{\mathbf{V}_{RF}^*(i,j)\eta_{ij}^D\right\}}, \quad (2.15)$$

dimana

$$\mathbf{V}_{RF}(i,j) = e^{-j\theta_{i,j}} \quad (2.16)$$

dimana

$$|\mathbf{V}_{RF}(i,j)|^2 = 1, \forall i,j. \quad (2.17)$$

sehingga didapat θ_{ij}^{opt} sebagai berikut:

$$\theta_{ij}^{opt} = \underset{\theta_{i,j}^{(1)}, \theta_{i,j}^{(2)}}{argmin}\left(\hat{f}\left(\theta_{i,j}^{(1)}\right), \hat{f}\left(\theta_{i,j}^{(2)}\right)\right) \quad (2.18)$$

Setelah menjalankan serangkaian proses diatas, kita telah bisa Menyusun sebuah iterative mulai dari membangkitkan RF *precoder* yang mungkin/layak dan melakukan *update*/pembaharuan elemen-elemen RF *precoder* secara sekuensial berdasarkan θ_{ij}^{opt} yang didapat hingga algoritma konvergen terhadap minimizer lokal $\hat{f}(\mathbf{V}_{RF})$. Kemudian dengan menggunakan RF *precoder* yang didapat, dapat dilakukan pencarian nilai **P** menggunakan teori water-filling seperti pada algoritma (3.4). Iterasi

terhadap kedua langkah ini dilakukan terus menerus hingga mendapatkan hasil yang konvergen untuk keseluruhan algoritma[4].

2.4 Kanal

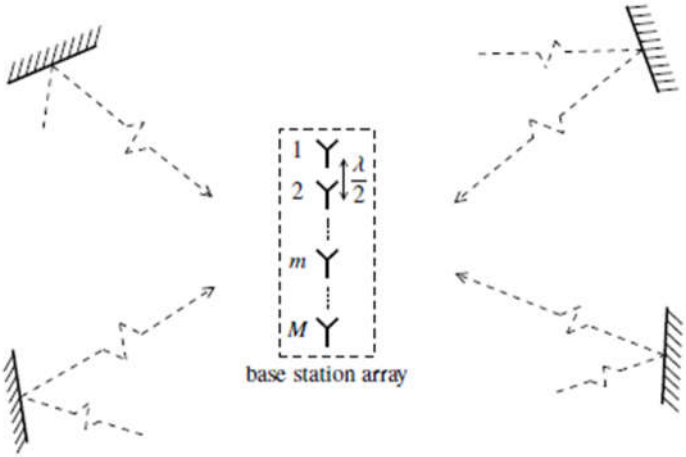
Kanal adalah segala hal yang ada diantara pemancar dan penerima. Dalam hal ini bisa diartikan sebagai media transmisi dalam telekomunikasi. Wujud dari media transmisi ini juga bermacam, seperti: *free space* (kondisi ideal), atmosfer, kabel, dan optik[9]. Sinyal yang melewati media transmisi, tentu akan mengalami perubahan akibat adanya penguatan dan pelemahan. Terjadinya perubahan ini disebabkan oleh adanya peristiwa refleksi, difraksi, dan scattering[10]. Media transmisi yang diasumsikan tidak memiliki penyebab rugi-rugi transmisi(kanal ideal) bisa disebut dengan *free space loss*. Pada penelitian ini, digunakan dua macam kanal untuk menguji hasil dari *beamforming*, yaitu kanal *Independent Rayleigh Fading* dan *Uniformly Random Line-of-Sight*.

2.4.1 Independent Rayleigh Fading

Independent Rayleigh Fading adalah kanal yang memiliki lingkungan atau media yang padat dan memiliki hamburan isotropik. Dengan mengasumsikan bahwa setiap antenna mampu melihat superposisi dari banyak sinyal yang merupakan hamburan independen. Dengan demikian, *fading* bersifat independen di tiap antenna yang berbeda[10]. Pada *independent Rayleigh fading* berlaku $E\{|g_k^m|^2\} = \beta_k$ dan $E\left\{g_k^m * g_{k'}^{m'}\right\} = 0$, ketika $k' \neq k$ atau $m' \neq m$. Dengan hukum *large numbers*:

$$\frac{1}{M}\|g_k\|^2 \rightarrow \beta_k, \qquad M \rightarrow \infty, \qquad k = 1, \ldots, K, \tag{2.19}$$

$$\frac{1}{M} g_k^H g_{k'} \rightarrow 0, \qquad M \rightarrow 0, \qquad k \neq k'. \tag{2.20}$$



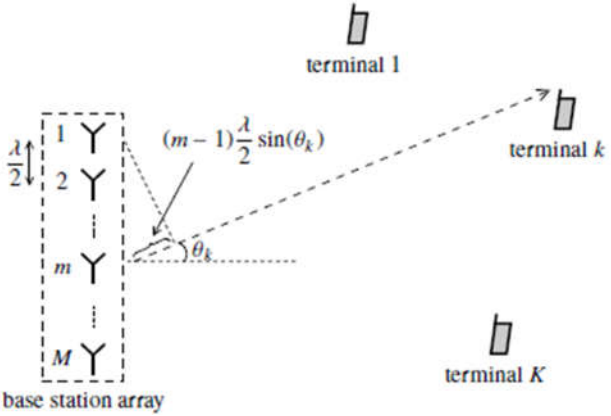
Gambar 2.9 *Uniform linear array* dengan λ/2 dengan hamburan isotropic[11]

2.4.2 Uniformly Random Line-of-Sight (URLoS)

Berbeda dengan *Independent Rayleigh Fading*, yang mana tiap antenna bersifat independen dalam menangkap sinyal hamburan isotropic, pada *Uniformly Random Line-of-Sight*, semua terminal dalam kondisi *line-of-sight*. Pada kanal ini, tiap sinyal dari terminal yang masuk, saling mempengaruhi kualitas antenna penerima yang disebabkan oleh adanya perbedaan jarak antar antenna dan sudut datang sinyal terhadap lokasi antenna penerima[11]. Pada gambar di bawah, diasumsikan terminal ke-k terletak pada medan jauh dari antena array dan memiliki sudut θ_k yang diukur relatif terhadap *array boresight*, maka:

$$g_k = \sqrt{\beta_k} e^{i\emptyset_k} \left[1 \; e^{-i\pi \sin(\theta_k)} \; \dots \; e^{-i(M-1)\pi \sin(\theta_k)}\right]^T, \qquad (2.21)$$

dimana θ_k bernilai acak yang terdistribusi secara *uniform* antara -π dan π, yang memodelkan pergeseran fasa yang terkait dengan rentang acak antara antena array dan terminal ke-k.



Gambar 2.10 λ/2 uniform linear array dengan lingkungan propagasi line-of-sight[11]

Jika ada dua terminal k dan k', dengan sudut θ_k dan $\theta_{k'}$ sedemikian rupa sehingga $\theta_k \neq \theta_{k'}$,

$$\frac{1}{M} g_k^H g_{k'} = \frac{1}{M}\sqrt{\beta_k \beta_{k'}} e^{-i\left(\theta_k - \theta_{k'}\right)} \sum_{m=0}^{M-1} e^{im\pi\left(\sin(\theta_k) - \sin\left(\theta_{k'}\right)\right)}$$

$$= \frac{1}{M}\sqrt{\beta_k \beta_{k'}} e^{-i(\theta_k - \theta_{k'})} \frac{1 - e^{iM\pi(\sin(\theta_k) - \sin(\theta_{k'}))}}{1 - e^{i\pi(\sin(\theta_k) - \sin(\theta_{k'}))}}$$

$$\to 0. \qquad M \to \infty, \tag{2.22}$$

juga,

$$\frac{1}{M}\|g_k\|^2 = \beta_k, \qquad k = 1, \cdots, K. \tag{2.23}$$

2.5 Pola Radiasi

Pola radiasi diperoleh menggunakan rumus *array factor*. Dari plot pola radiasi ini dapat diketahui pola radiasi dari metode hybrid precoding atau hybrid beamforming yang telah digunakan. Persamaan rumus *array factor* dituliskan sebagai berikut:

$$AF(\psi) = \sum_{n=0}^{N-1} An\, e^{jn\beta d\cos\theta + \alpha} \tag{2.24}$$

$$f(\psi) = \frac{AF(\psi)}{\max[AF(\psi)]} \tag{2.25}$$

Dimana $AF(\psi)$ adalah *array factor*. N adalah jumlah antena jamak yang digunakan, An adalah magnitudo dari matriks $\mathbf{V}_{RF}\mathbf{V}_D$, β adalah ketetapan fasa yakni $\frac{2\pi}{\lambda}$, d adalah jarak antar elemen antena yakni $\frac{\lambda}{2}$, θ adalah sudut yang dibentuk antara elemen antena pada pemancar (*base station*) dengan elemen antena pada penerima, dan α adalah fasa dari matriks $\mathbf{V}_{RF}\mathbf{V}_D$.

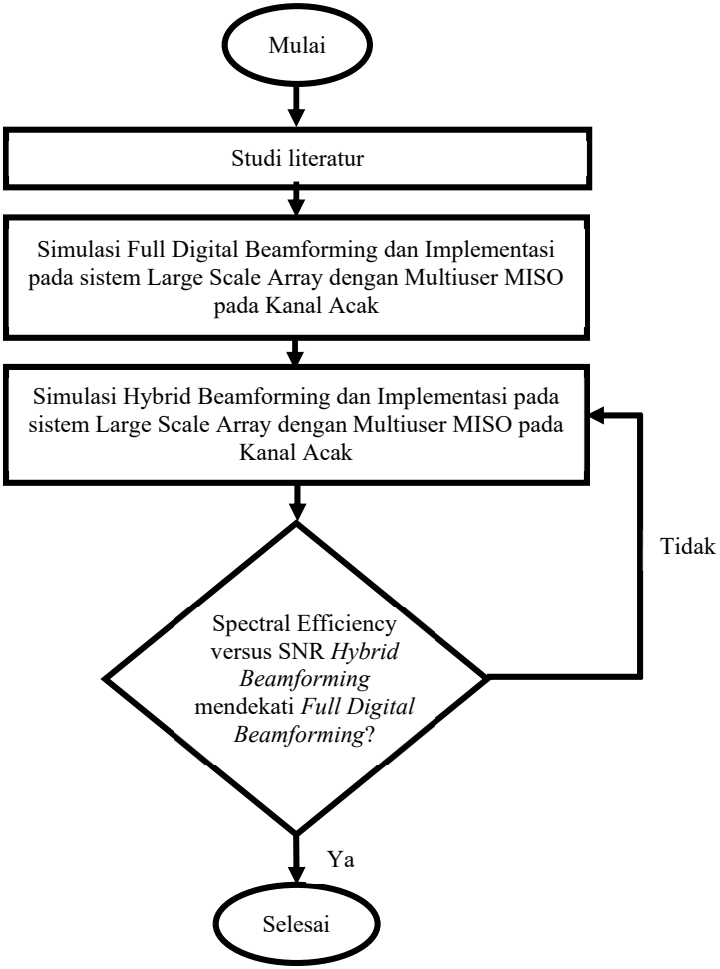
BAB III METODOLOGI PENGERJAAN

Pada bab tiga ini dijelaskan mengenai tahapan-tahapan dalam pengerjaan tugas akhir. Akan dipaparkan juga mengenai alur pengerjaan tugas akhir serta pemodelan sistem yang akan digunakan dalam pengerjaan kedepannya. Tugas akhir ini membahas tentnag analisa kinerja dari metode hybrid beamforming pada sistem antena multiuser MISO. Kinerja dari metode hybrid beamforming akan dibandingkan terhadap metode fulldigital beamforming. Parameter yang akan digunakan dalam menguji kinerja metode tersebut adalah *spectral efficiency versus signal-to-noise ratio (SNR)* yang mana akan di ujikan pada kanal *independent Rayleigh fading* dan kanal *uniformly random line-of-sight (UR-LoS)*. Akan dilakukan beberapa kali pengujian dengan jumlah elemen yang berbeda yang kemudian akan dianalisa sehingga akan diperolah konfigurasi jumlah rf chain pada *digital dan analog precoding* yang memiliki *spectrall efficiency versus signal-to-noise ratio* yang mendekati metode *full digital beamforming*. Dalam pemodelan simulasi ini digunakan *software Matlab* versi R2019b.

3.1 Parameter Sistem

Pada tugas akhir ini akan digunakan beberapa parameter dalam proses simulasi seperti:

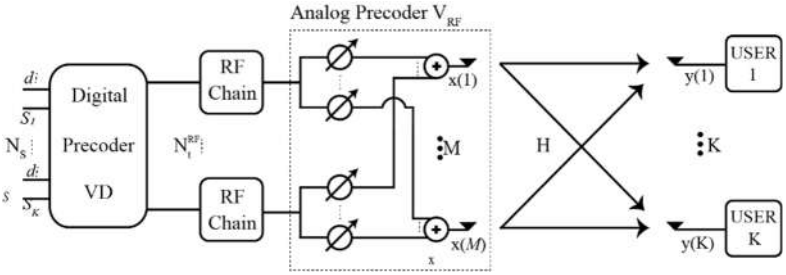
a) Modulasi yang digunakan untuk *bit stream* informasi adalah BPSK.
b) Konfigurasi sistem yang di variasikan adalah jumlah rf chain, jumlah antena pemancar, dan jumlah user/antena penerima.
c) *Hybrid beamforming* dimodelkan dengan jumlah RF chain sebanyak setengah dari jumlah RF chain pada *full digital beamforming*.
d) Antena dengan sistem Multiuser MISO.
e) Kanal yang digunakan adalah kanal *Independent Rayleigh Fading* dan kanal *Uniformly Random Line-of-Sight (UR-LoS)*.
f) Menggunakan pendekatan *spectrum efficiency versus signal-to-noise ratio (SNR)* dalam pengujian kinerja metode beamforming.



Gambar 3.1 Diagram alur pengerjaan tugas akhir

3.2 Pemodelan Sistem

Secara garis besar model sistem yang akan dipakai pada tugas akhir ini digambarkan pada Gambar 3.2, yang selanjutkan akan dijelaskan lebih detail mengenai bagian-bagiannya.



Gambar 3.2 Diagram model sistem hybrid dan full digital beamforming pada Multiuser MISO

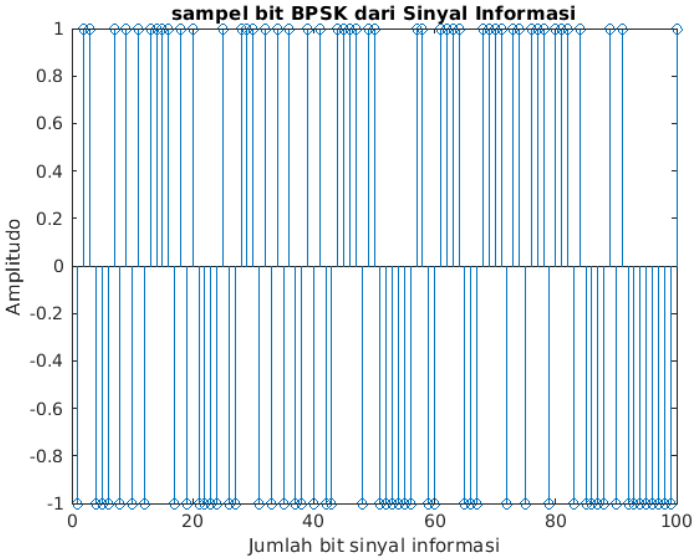
3.2.1 Pemodelan Transmitter

Transmitter adalah bagian yang memiliki fungsi untuk mengirimkan/mentransmisikan sinyal informasi yang diinginkan. Pada gambar 3.2 dapat kita lihat bahwa bagian transmitter memiliki komponen-komponen yang akan dilalui oleh sinyal informasi sebelum nantinya akan dikirimkan atau dipancarkan oleh antena pemancar. Terdapat beberapa komponen seperti sinyal informasi(*s*), user (K), *data stream* (*d*), jumlah antena (M), *digital precoder* (V_D)*, analog precoder* (V_{RF})*, RF chain, dan phase shifter.* Jika kita lihat mulai awal, sinyal informasi akan masuk ke *Digital Precoder* (V_D) terlebih dahulu yang akan diolah oleh precoder lalu dikirim ke *RF chain* yang ada. Setelah itu, sinyal yang sudah melalui *digital precoder* akan masuk ke komponen *analog precoder* (V_{RF}) yang berisi komponen *phase shifter*. Setelah itu sinyal informasi yang telah melewati serangkaian proses akan di pancarkan oleh antena pemancar dan disebut sinyal transmisi(*x*). Secara matematis akan ditulis sebagai berikut:

$$x = \mathbf{V}_{RF}\mathbf{V}_D s = \sum_{l=1}^{K} \mathbf{V}_{RF}\mathbf{V}_{D_l} s_l \qquad (3.1)$$

Penggunaan *full digital beamforming* pada spektrum frekuensi *milimeter wave* dirasa tidak mungkin karena membutuhkan jumlah *RF chain* yang banyak yakni 1 *RF chain* tiap antena, yang mana membutuhkan banyak biaya dalam merealisasikannya. Dengan metode *hybrid beamforming*, penggunaan *RF chain* dapat ditekan menjadi sejumlah $N_{RF} = N_S$, dimana $N_S = K\,d$, dan *K* merepresentasikan jumlah

user dan *d* merepresentasikan jumlah data stream tiap usernya. Pada pengerjaan tugas akhir ini akan dibangkitkan jumlah bit bpsk sinyal informasi sebanyak 1000-bit tiap usernya. Pada model umum sistem multiuser MISO ini, akan digunakan jumlah user sebanyak 4 user, sehingga total dari jumlah bit bpsk sinyal informasi yang akan dibangkitkan yakni sebanyak 4000-bit sinyal. Berikut adalah plot stem dari 100 sampel bit BPSK dalam bentuk baseband ekivalen:



Gambar 3.3 Plot stem dari sampel bit BPSK sinyal informasi

3.2.2.1 Desain Digital Precoder

Matriks digital precoding ($\mathbf{V}_D$) dapat diperoleh setelah memenuhi beberapa persyaratan, yakni: matriks $\mathbf{V}_{RF}$ telah memenuhi syarat konvergen (pada persamaan 11) dan pencarian daya menggunakan teorema *water-filling* telah mencapai konvergen (pada persamaan 13). Setelah itu akan didapat $\mathbf{V}_D$ yang relevan sesuai dengan keadaan atau kondisi sistem.

3.2.2.2 Desain RF/Analog Precoder

Dimulai dengan menentukan matriks $\mathbf{V}_{RF} = 1_{NxN^{RF}}$, kemudian dilanjutkan hingga step ke 11 dan akan berulang hingga mendapatkan matriks $\mathbf{V}_{RF}$ yang konvergen. Kemudian dilanjutkan pada pencarian nilai **P** menggunakan $\mathbf{V}_{RF}$ yang telah konvergen yang nantinya akan digunakan dalam perhitungan pencarian matriks $\mathbf{V}_D$.

3.2.2 Pemodelan Kanal

Untuk menguji kinerja dari dua metode yakni *hybrid beamforming* dan *full digital beamforming* maka pada tugas akhir ini digunakan dua tipe kanal yang berbeda, yaitu kanal *independent rayleigh fading* dan kanal *uniformly random line-of-sight*. Karena dua tipe kanal ini memiliki karakteristik yang berbeda, maka kita dapat melihat perbandingan antara kinerja kedua metode tersebut.

3.2.2.1 Kanal Independent Rayleigh Fading

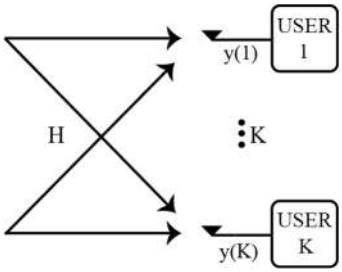
Kanal *independent rayleigh fading* adalah kanal yang memiliki hamburan isotropik yang padat dan tidak memiliki lintasan *line of sight*. Pada simulasi pemodelan kanal, digunakan kanal *independent rayleigh* yang memiliki elemen acak independen dengan *mean* nol dan terdistribusi kompleks normal $(0,\beta_k)$ dimana β_k diasumsikan bernilai 1.

3.2.2.2 Kanal Uniformly Random Line-of-Sight

Kanal *uniformly random line-of-sight* adalah tipe kanal yang tidak memiliki hamburan lokal dan semua penerimanya (*terminal*) memiliki lintasan *line-of-sight* (langsung) terhadap antena pemancar. Pada percobaan ini akan digunakan selisih sudut antar user sebesar 5 derajat.

3.2.3 Pemodelan Receiver

Model dari receiver yang digunakan pada tugas akhir ini digambarkan sebagai berikut:



Gambar 3.4 Model receiver multiuser MISO

Dapat dilihat pada hambar diatas bahwa penerima pada sistem *multiuser* MISO hanya berupa antena *receive* saja. Sinyal informasi yang telah melalui serangkaian proses pada pemancar dan juga telah melewati kanal, akan diterima dan disimolkan dengan *y*. Secara matematis dapat dituliskan seperti berikut:

$$y_k = \mathbf{H}_k \mathbf{V}_{RF} \mathbf{V}_D \boldsymbol{s}_k + \mathbf{z}_k \tag{3.2}$$

3.2.4 Demapping

Sinyal yang diterima (y) masih dalam bentuk sinyal transmit, sehingga perlu diolah untuk mengubah sinyal transmit menjadi sinyal informasi kembali. Sehingga pada simulasi ini digunakan demapping seperti berikut:

$$\tilde{y}_k = \begin{cases} 1, jika\ real(y) > 0 \\ -1, jika\ real(y) < 0 \end{cases} \tag{3.3}$$

dimana $\tilde{y}_k$ merepresentasikan sinyal terima yang telah diolah menjadi seperti sinyal informasi yang dikirimkan.

3.2.5 Simulasi Analisa *Bit Error Rate*

Setelah mendapatkan serta mengolah sinyal terima yang telah nemiliki format yang sama dengan sinyal informasi yang dikirimkan, langkah selanjutnya adalah melakukan pengecekan *bit error rate*. Pengecekan ini bertujuan untuk mengetahui apakah ada kesalahan yang terjadi pada saat proses pengiriman bit sinyal informasi. Pengecekan *bit error rate* ini juga dapat melihat bagaimana kondisi kelayakan sistem dalam proses transmisi. Simulasi *bit error rate* yang dilakukan pada tugas akhir ini dijelaskan melalui persamaan 3.20

$$BER = \frac{jumlah\ bit\ error}{total\ bit} \tag{3.4}$$

3.2.6 Simualsi *Spectral Efficiency versus Signal-to-Noise Ratio (SNR)*

Simulasi *spectral efficiency versus signal-to-noise ratio (SNR)* adalah langkah terakhir yang dilakukan agar dapat mengetahui kinerja dari kedua metode yang diinginkan. Dari grafik *spectral efficiency versus signal-to-noise ratio (SNR)* ini nantinya dapat dianalisa mengenai hubungan antara kenaikan nilai SNR dengan kualitas kinerja kedua metode yang dilihat pada grafik *spectral efficiency*. *Spectral efficiency* disimbolkan dengan R_k yang mana adalah *rate expression* untuk user k [4].

$$R_k = log_2\left(1 + \frac{\left|\mathbf{h}_k^H \mathbf{V}_{RF} \mathbf{V}_{D_k}\right|^2}{\sigma^2 + \sum_{l \neq k} \left|\mathbf{h}_k^H \mathbf{V}_{RF} \mathbf{V}_{D_l}\right|^2}\right)$$

Dimana $\mathbf{V}_{D_l}$ adalah kolom ke-l dari digital precoder. Menggunakan sifat dari *Zero Forcing beamforming*, maka $\left|\mathbf{h}_k^H \mathbf{V}_{RF} \mathbf{V}_{D_k}\right| = \sqrt{p_k}$ dan $\left|\mathbf{h}_k^H \mathbf{V}_{RF} \mathbf{V}_{D_l}\right| = 0$ untuk semua $l \neq k$.

BAB IV HASIL DAN ANALISA DATA

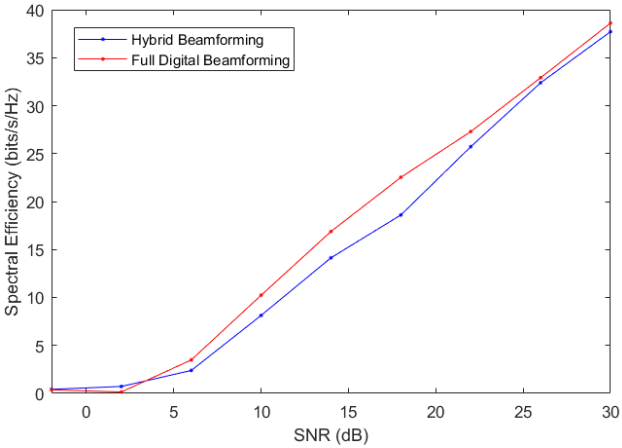
Pada bab ini akan ditampilkan hasil simulasi dari teknik *hybrid beamforming* (HB) dan *full digital beamforming* (FDB) yang diujikan pada dua kanal yaitu kanal *independent rayleigh fading* (IRF) dan *uniformly random line-of-sight* (URLOS). Kemudian akan dianalisa kinerja kedua metode terhadap dua kanal yang berbeda. Pada bab ini akan dibagi menjadi tiga bagian, yang pertama berisi perbandingan kinerja antara HB dan FDB pada dua kanal yang berbeda. Hal ini bertujuan untuk memeriksa apakah metode HB telah sesuai atau belum. Bagian kedua berisi hasil simulasi antara kanal IRF dengan URLOS saat menggunakan metode HB dan pada saat metode FDB. Pada bagian ketiga akan ditampilkan perbandingan kinerja SE pada masing-masing kanal dengan metode HB yang memiliki jumlah RF *chain* (NRF) yang berbeda, jumlah antena pemancar yang berbeda, dan jumlah antena penerima yang berbeda. Dari perbedaan komponen yang telah disebutkan diatas, parameter *spectral efficiency* (SE) versus *signal-to-noise ratio* (SNR) dan *Bit Error Rate* (BER) akan membantu dalam menganalisa kinerja dari kedua metode.

4.1 Perbandingan kinerja antara *Hybrid* dan *Full Digital Beamforming* pada kanal *Independent Rayleigh Fading* dan *Uniformly Random Line-of-Sight*

Pada simulasi ini akan didapatkan grafik perbandingan kinerja dari metode HB dan FDB yang akan dilihat dari SE vs SNR. Pada percobaan ini akan digunakan 64 pemancar (Tx), 4 *user* (4 antena penerima/Rx), serta 1 RF chain untuk metode HB dan 2 RF chain untuk metode FDB.

4.1.1 Kanal *Independent Rayleigh Fading*

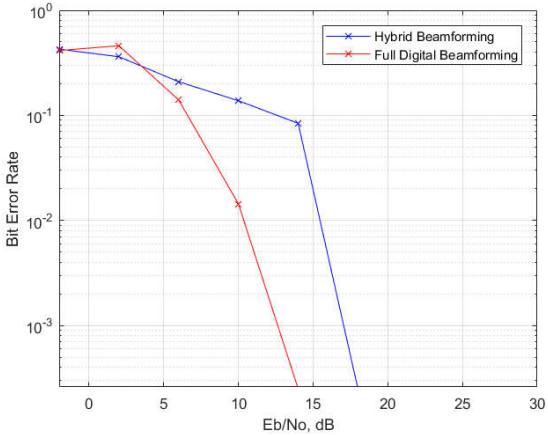
Hasil dari simulasi dapat dilihat pada gambar 4.1 dan gambar 4.2. Adapun tabel 4.1 dan tabel 4.2 yang akan membantu memperjelas dari pembacaan gambar 4.1 dan gambar 4.2.



Gambar 4.1 Grafik perbandingan kinerja SE antara metode HB dan FDB pada kanal IRF

Tabel 4.1 Data SE metode HB dan FDB pada kanal IRF

SNR	Spectral Efficiency	
	Hybrid Beamforming	Full Digital Beamforming
-2	0,4223	0,3686
2	0,7246	0,1458
6	2,39	3,488
10	8,132	10,23
14	14,14	16,88
18	18,6	22,53
22	25,72	27,29
26	32,39	32,91
30	37,71	38,61



Gambar 4.2 Grafik perbandingan kinerja BER antara metode HB dan FDB pada kanal IRF

Tabel 4.2 Data BER metode HB dan FDB pada kanal IRF

SNR	BER	
	Hybrid Beamforming	Full Digital Beamforming
-2	0,4272	0,4143
2	0,3613	0,4601
6	0,2091	0,1401
10	0,1378	0,01436
14	0,084	0,00025
18	0,00025	0,00025
22	0,00025	0,00025
26	0,00025	0,00025
30	0,00025	0,00025

Kinerja dari metode HB dan FDB yang ditampilan pada gambar 4.1 dan tabel 4.1 menunjukkan bahwa pada saat SNR rendah (0-10 dB) metode HB memiliki rata-rata kinerja SE sebesar 3,74887 bits/s/Hz dan metode FDB memiliki rata-rata kinerja SE sebesar 4,62127 bits/s/Hz. Pada saat SNR sedang (10-20 dB) metode HB memiliki rata-rata kinerja SE sebesar 16,37 bits/s/Hz dan metode FDB memiliki rata-rata kinerja SE

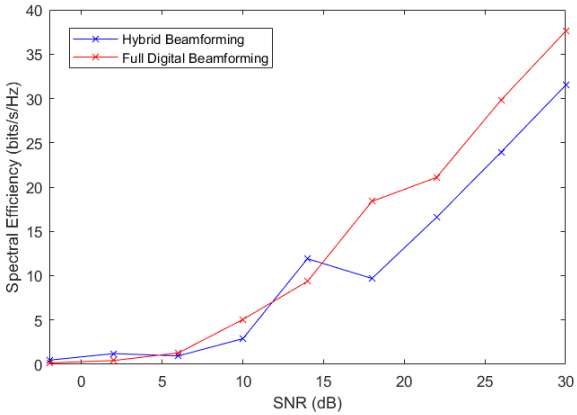
sebesar 19,705 bits/s/Hz. Pada saat SNR tinggi (20-30 dB) metode HB memiliki rata-rata kinerja SE sebesar 31,94 bits/s/Hz dan metode FDB memiliki rata-rata kinerja SE sebesar 32,9367 bits/s/Hz. Untuk keseluruhan SNR dari 0 dB hingga 30 dB (rata-rata keseluruhan), metode HB memiliki rata-rata kinerja SE sebesar 17,4758 bits/s/Hz dan metode FDB memiliki rata-rata kinerja SE sebesar 19,0105 bits/s/Hz.

Pengujian BER pada kedua metode ini digambarkan pada gambar 4.2 dan dipaparkan data tiap plotnya pada tabel 4.2. Dapat dilihat pada SNR rendah (0-10 dB) tepatnya pada saat SNR = 2 dB, metode HB memiliki BER sebesar 0,3613 sedangkan FDB memiliki BER 0,4601 yang mana masih lebih tinggi. Akan tetapi pada saat SNR = 6 dB, metode FDB dengan BER 0,1401 telah berada pada posisi di bawah HB dengan BER 0,2091. Pada SNR sedang (10-20), saat SNR = 14 dB dapat dilihat bahwa metode FDB telah mencapai BER 0,00025 yang artinya kemungkinan terjadi error kecil, sedangkan untuk metode HB memiliki BER 0,084 yang mana masih cukup tinggi. Pada SNR tinggi (20-30 dB) kedua metode telah memiliki kemungkinan untuk terjadi error kecil dengan BER 0,00025.

Jika dilihat dari gambar 4.1 dan tabel 4.1, simulasi ini membuktikan bahwa kinerja SE metode HB telah mendekati kinerja SE metode FDB. Sedangkan jika melihat perbandingan BER pada gambar 4.2 dan tabel 4.2, metode FDB lebih dulu memiliki tingkat kemungkinan error yang rendah dengan BER 0,00025 sedangkan walaupun metode HB masih memiliki tingkat error yang relatif tinggi pada saat SNR sedang, metode HB akan menyusul memiliki tingkat error yang rendah dengan BER 0,00025 pada saat SNR tinggi.

4.1.2 Kanal Uniformly Random Line-of-Sight

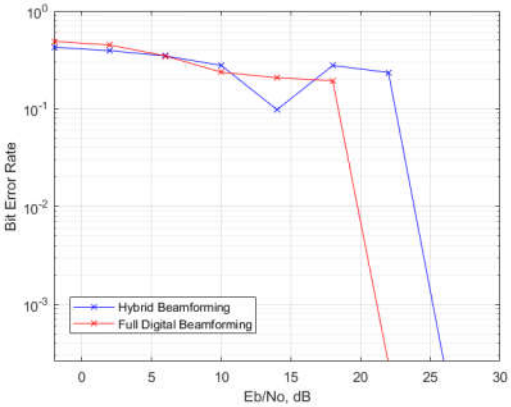
Hasil dari simulasi dapat dilihat pada gambar 4.3 dan gambar 4.4. Adapun tabel 4.3 dan tabel 4.3 yang akan membantu memperjelas dari pembacaan gambar 4.3 dan gambar 4.4.



Gambar 4.3 Grafik perbandingan kinerja SE antara metode HB dan FDB pada kanal URLOS

Tabel 4.3 Data SE metode HB dan FDB pada kanal URLOS

SNR	Spectral Efficiency	
	Hybrid Beamforming	Full Digital Beamforming
-2	0,4716	0,1572
2	1,206	0,4304
6	0,9474	1,283
10	2,897	5,062
14	11,91	9,355
18	9,687	18,39
22	16,58	21.08
26	23,95	29,82
30	31,49	37,58



Gambar 4.4 Grafik perbandingan kinerja BER antara metode HB dan FDB pada kanal URLOS

Tabel 4.4 Data BER metode HB dan FDB pada kanal URLOS

SNR	BER	
	Hybrid Beamforming	Full Digital Beamforming
-2	0,4254	0,4882
2	0,3922	0,4484
6	0,3462	0,3461
10	0,2772	0,2364
14	0,09769	0,2074
18	0,2766	0,1934
22	0,2346	0,00025
26	0,00025	0,00025
30	0,00025	0,00025

Dapat dilihat kinerja dari kedua metode pada gambar 4.3 dan tabel 4.3 menunjukkan pada saat SNR rendah (0-10 dB) metode HB memiliki

rata-rata kinerja SE sebesar 1,68347 bits/s/Hz dan metode FDB memiliki rata-rata kinerja SE sebesar 2,25847 bits/s/Hz. Pada saat SNR sedang (10-20 dB) metode HB memiliki rata-rata kinerja SE sebesar 10,7985 bits/s/Hz dan metode FDB memiliki rata-rata kinerja SE sebesar 13,8725 bits/s/Hz. Pada saat SNR tinggi (20-30 dB) metode HB memiliki rata-rata kinerja SE sebesar 24,0067 bits/s/Hz dan metode FDB memiliki rata-rata kinerja SE sebesar 22,4667 bits/s/Hz. Sementara untuk total rata-rata kinerja kedua metode dari 0 dB sampai 30 dB, HB memiliki rata-rata kinerja total sebesar 12,3334 bits/s/Hz dan FDB memiliki rata-rata kinerja total sebesar 12,7401 bits/s/Hz.

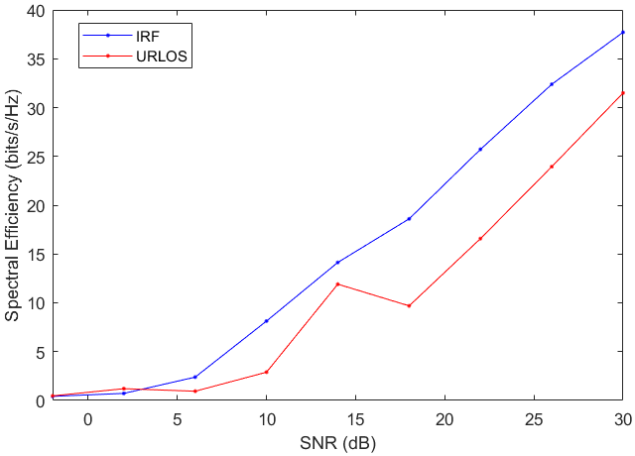
Pengujian BER pada kedua metode ini digambarkan pada gambar 4.4 dan dipaparkan data tiap plotnya pada tabel 4.4. Dapat dilihat pada SNR rendah (0-10 dB) saat SNR = 2, metode HB dengan BER 0,3922 berada sedikit di bawah metode FDB dengan BER 0,4484. Pada saat SNR = 10 dB, metode FDB dengan BER 0.2364 berada sedikit di bawah metode HB yang memiliki BER 0,3922. Juga dapat dilihat pada SNR rendah, kedua metode masih memiliki tingkat kemungkinan terjadi error yang cukup tinggi. Pada SNR sedang (10-20 dB) saat SNR = 4 dB, metode HB dengan BER 0,09769 berada di bawah metode FDB yang memiliki BER 0,2074. Akan tetapi pada saat SNR = 18 dB, metode HB yang memiliki BER = 0,2766 kembali naik menjadi semakin tinggi dan berada diatas metode FDB yang memiliki BER = 0,1934. Pada SNR sedang ini dapat dilihat bahwa metode HB sempat berada di bawah metode FDB pada saat SNR = 14 dB yang kemudian naik kembali pada saat SNR = 18 dB, sedangkan metode FDB belum menunjukkan penurunan BER yang signifikan. Pada SNR tinggi (20-30 dB) saat SNR = 22 dB, metode FDB telah mencapai BER 0,00025, sedangkan metode HB masih memiliki angka BER yang tinggi yaitu 0,2346. Pada saat SNR = 26 dB hingga 30 dB, kedua metode baik HB dan FDB telah menunjukkan angka BER yang rendah yaitu 0.00025. Pada SNR tinggi dapat dilihat bahwa pada saat SNR = 22 dB, metode FDB lebih dulu memiliki angka BER yang kecil, yang artinya memiliki kemungkinan error yang kecil, yang mana berarti precoding ini akan bekerja dengan baik sejak SNR = 22 dB, sedangkan metode HB baru memiliki angka BER yang rendah yaitu 0,00025 sejak SNR = 26 dB.

Dapat dilihat baik dari gambar 4.3 dan tabel 4.3 juga gambar 4.4 dan tabel 4.4, metode HB telah berhasil mendekati kinerja SE dan BER dari metode FDB. Ini juga membuktikan bahwa metode HB dapat memiliki kinerja SE dan BER yang mendekati metode FDB pada kanal URLOS.

4.2 Perbandingan kinerja metode HB pada kanal IRF dan URLOS dan perbandingan kinerja metode FDB pada kanal IRF dan URLOS

4.2.1 Perbandingan metode HB pada kanal IRF dan URLOS

Pada bagian ini akan ditampilkan hasil simulasi metode HB pada kanal IRF dan kanal URLOS. Dari simulasi ini, akan dibandingkan kinerja metode HB pada kedua kanal yang berbeda. Metode HB disini menggunakan jumlah antenna pemancar (tx) sejumlah 64 antena, user sejumlah 4, dan jumlah RF chain sebanyak 1.

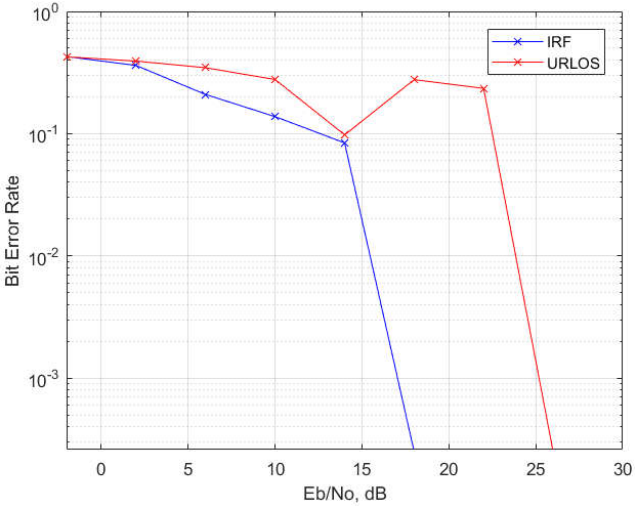


Gambar 4.5 Grafik perbandingan kinerja SE metode HB pada kanal IRF dan URLOS

Tabel 4.5 Data SE metode HB pada kanal IRF dan URLOS

SNR	Spectral Efficiency (HB)	
	IRF	URLOS
-2	0,4223	0,4716
2	0,7246	1,206
6	2,39	0,9474
10	8,132	2,897
14	14,14	11,91

18	18,6	9,687
22	25,72	16,58
26	32,39	23,95
30	37,71	31,49



Gambar 4.6 Grafik perbandingan kinerja BER metode pada kanal IRF dan URLOS

Tabel 4.6 Data BER metode HB pada kanal IRF dan URLOS

SNR	BER (HB)	
	IRF	URLOS
-2	0,4272	0,4254
2	0,3613	0,3922
6	0,2091	0,3462
10	0,1378	0,2772
14	0,084	0,09769
18	0,00025	0,2766
22	0,00025	0,2346
26	0,00025	0,00025
30	0,00025	0,00025

Dapat dilihat melalui gambar 4.5 dan tabel 4.5 pada saat SNR rendah (0-10 dB) metode HB pada kanal IRF memiliki rata-rata kinerja SE sebesar 3,74887 bits/s/Hz yang mana lebih tinggi jika dibandingkan dengan metode HB pada kanal URLOS yang memiliki rata-rata kinerja SE sebesar 1,68347 bits/s/Hz. Pada SNR sedang (10-20 dB) metode HB pada kanal IRF memiliki rata-rata kinerja SE sebesar 16,37 bits/s/Hz yang lebih tinggi dibandingkan dengan metode HB pada kanal URLOS yang memiliki rata-rata kinerja SE sebesar 10,7985 bits/s/Hz. Pada SNR tinggi (20-30 dB) metode HB pada kanal IRF memiliki rata-rata kinerja SE sebesar 31,94 bits/s/Hz yang lebih tinggi dibandingkan metode HB pada kanal URLOS yang memiliki rata-rata kinerja SE sebesar 24,0067 bits/s/Hz. Untuk total rata-rata kinerja pada SNR 0 sampai 30 dB, metode HB pada kanal IRF memiliki total rata-rata kinerja SE sebesar 17,4758 bits/s/Hz, sedangkan metode HB pada kanal URLOS memiliki total rata-rata kinerja SE sebesar 12,3334 bits/s/Hz. Pada grafik 4.5 dan tabel 4.5 dapat dilihat bahwa secara keseluruhan, metode HB pada kanal IRF memiliki kinerja SE yang lebih baik dibandingkan dengan metode HB pada kanal URLOS.

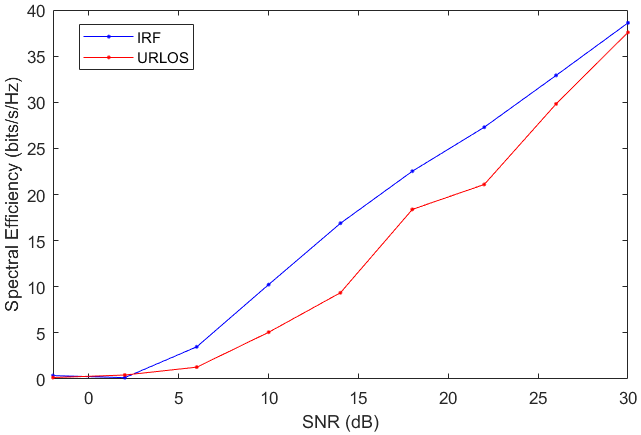
Pengujian BER pada metode HB dengan kanal IRF dan kanal URLOS dijelaskan dan dipaparkan oleh gambar 4.6 dan tabel 4.6. Dari gambar 4.6 dan tabel 4.6 dapat kita lihat dan ketahui bahwa pada SNR rendah (0-10 dB) saat SNR = 2 dB, metode HB pada kanal IRF yang memiliki BER sebesar 0.3613 berada sedikit dibawah metode HB pada kanal URLOS yang memiliki BER sebesar 0,3922. Begitu juga untuk SNR = 6 dB dan SNR = 10 dB, metode HB pada kanal IRF memiliki BER sebesar 0.2091 dan 0,1378 yang mana berada di bawah metode HB pada kanal URLOS dengan BER sebesar 0,3462 dan 0,2772. Pada SNR sedang (10-20 dB), saat SNR = 14 dB, metode HB dengan kanal IRF memiliki BER sebesar 0,084 yang mana berada di bawah metode HB dengan kanal URLOS yang memiliki BER sebesar 0,09769. Saat SNR = 18 dB, metode HB dengan kanal IRF terlihat telah memiliki BER sebesar 0,00025, sedangkan metode HB dengan kanal URLOS memiliki BER sebesar 0,2766. Pada saat SNR tinggi (20-30 dB), untuk metode HB dengan kanal IRF, telah memiliki BER yang konstan sebesar 0,00025. Sedangkan untuk metode HB dengan kanal URLOS, pada saat SNR = 22 dB masih memiliki nilai BER yang cukup tinggi yaitu sebesar 0,2346, dan akan menuju 0,00025 pada SNR = 26 dB yang konstan hingga SNR = 30 dB. Dari sini dapat dilihat pada secara rata-rata, baik saat SNR rendah, sedang,

maupun tinggi, metode HB dengan kanal IRF memiliki BER yang lebih rendah dibandingkan dengan metode HB dengan kanal URLOS.

Secara keseluruhan dari gambar 4.6 dan tabel 4.6 menunjukkan bahwa jenis kanal memiliki pengaruh terhadap kinerja metode HB. Secara keseluruhan baik dari kinerja SE maupun tingkat BER, metode HB dengan kanal IRF menunjukkan performa yang lebih baik dibandingkan dengan metode HB dengan kanal URLOS.

4.2.2 Perbandingan metode FDB pada kanal IRF dan URLOS

Pada bagian ini akan ditampilkan hasil simulasi metode FDB pada kanal IRF dan kanal URLOS. Dari simulasi ini, akan dibandingkan kinerja metode FDB pada kedua kanal yang berbeda. Metode FDB disini menggunakan jumlah antenna pemancar (tx) sejumlah 64 antena, user sejumlah 4, dan jumlah RF chain sebanyak 2.

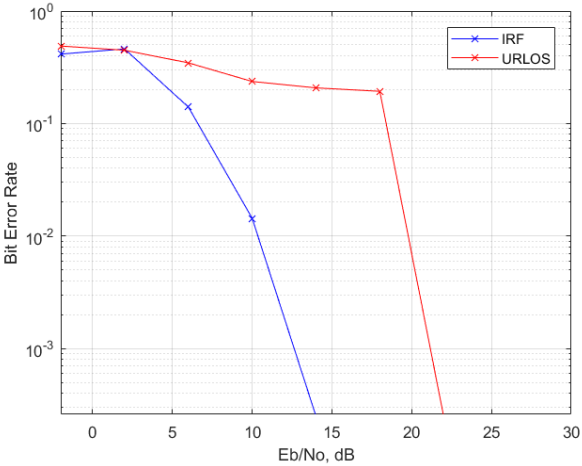


Gambar 4.7 Grafik perbandingan kinerja SE metode FDB pada kanal IRF dan URLOS

Tabel 4.7 Data SE metode FDB pada kanal IRF dan URLOS

SNR	Spectral Efficiency (FDB)	
	IRF	URLOS
-2	0,3686	0,1572
2	0,1458	0,4304
6	3,488	1,283
10	10,23	5,062

14	16,88	9,355
18	22,53	18,39
22	27,29	21.08
26	32,91	29,82
30	38,61	37,58



Gambar 4.8 Grafik perbandingan kinerja BER metode pada kanal IRF dan URLOS

Tabel 4.8 Data BER metode FDB pada kanal IRF dan URLOS

SNR	BER (FDB)	
	IRF	URLOS
-2	0,4143	0,4882
2	0,4601	0,4484
6	0,1401	0,3461
10	0,01436	0,2364
14	0,00025	0,2074
18	0,00025	0,1934
22	0,00025	0,00025
26	0,00025	0,00025
30	0,00025	0,00025

Dapat dilihat pada Gambar 4.7 dan Tabel 4.7 pada saat SNR rendah (0-10 dB) metode FDB pada kanal IRF memiliki rata-rata kinerja SE sebesar 4,62127 bits/s/Hz yang lebih tinggi dibandingkan dengan metode FDB pada kanal URLOS memiliki rata-rata kinerja SE sebesar 2,25847 bits/s/Hz. Pada SNR sedang (10-20 dB) metode FDB pada kanal IRF memiliki rata-rata kinerja SE sebesar 19,705 bits/s/Hz yang mana lebih tinggi jika dibandingkan dengan metode FDB pada kanal URLOS yang memiliki rata-rata kinerja SE sebesar 4,6775 bits/s/Hz. Pada SNR tinggi (20-30 dB) metode FDB pada kanal IRF memiliki rata-rata kinerja SE sebesar 32,9367 bits/s/Hz yang mana lebih tinggi dibandigkan dengan metode FDB pada kanal URLOS yang memiliki rata-rata kinerja SE sebesar 22,4667 bits/s/Hz. Secara keseluruhan, dapat dilihat baik pada gambar grafik maupun data tabel bahwa rata-rata kinerja SE pada metode FDB dengan kanal IRF lebih baik dibanding dengan rata-rata kinerja SE pada metode FDB dengan kanal URLOS. Juga dapat diketahui bahwa untuk SNR 0 dB sampai 30 dB, metode FDB dengan kanal IRF memiliki rata-rata kinerja SE sebesar 19,0105 bits/s/Hz yang mana lebih tinggi dibandingkan dengan rata-rata kinerja SE dengan metode FDB pada kanal URLOS sebesar 12,7401 bits/s/Hz.

Pada Gambar 4.8 dan Tabel 4.8 dapat dilihat pada saat SNR rendah (0-10 dB) tepatnya saat SNR = 2 dB, metode FDB dengan kanal IRF dengan BER sebesar 0,4601 berada sedikit diatas metode FDB dengan kanal URLOS dengan BER sebesar 0,4484. Saat SNR = 6 dB, metode FDB dengan kanal IRF mulai menunjukkan penurunan BER yang signifikan dengan BER sebesar 0,1401, sedangkan metode FDB dengan kanal URLOS hanya mengalami sedikit penurunan dengan BER sebesar 0,3461. Pada saat SNR sedang (10-20 dB), saat SNR = 14 dB, metode FDB dengan kanal IRF telah mencapai titik BER yang paling rendah yang dicapainya yaitu pada BER sebesar 0,00025, sedangkan metode FDB dengan kanal URLOS masih memiliki BER yang tinggi yaitu sebesar 0,2074. Pada saat SNR tinggi (20-30m dB) saat SNR = 22 dB, metode FDB dengan kanal IRF masih konstan berada di titik terendahnya yakni BER sebesar 0,00025 yang terus konstan hingga mencapai SNR = 30 dB. Sedangkan metode FDB dengan kanal URLOS baru mencapai titik yang paling rendah yang bisa dicapai pada saat SNR = 22 dB dengan BER sebesar 0,00025 yang terus konstan hingga SNR = 30 dB.

Dari Gambar 4.7 dan Tabel 4.7 dapat diketahui bahwa secara keseluruhan rata-rata kinerja SE dari metode FDB dengan kanal IRF lebih baik dibandingkan dengan metode FDB dengan kanal URLOS. Juga pada

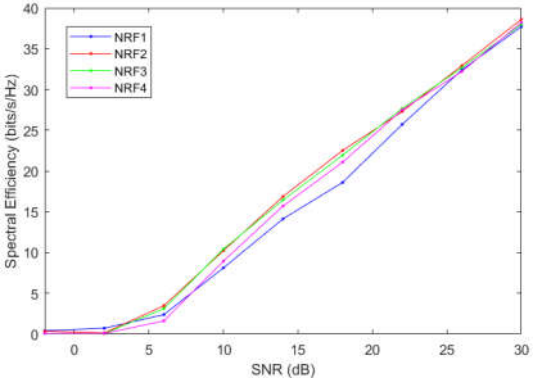
gambar 4.8 dan tabel 4.8 dapat diketahui bahwa metode FDB dengan kanal IRF memiliki kemungkinan error yang kecil sejak SNR = 14 dB hingga SNR = 30 dB, dengan BER sebesar 0,00025. Sedangkan untuk metode FDB pada kanal URLOS memiliki kemungkinan error yang kecil sejak SNR = 24 dB hingga SNR = 30 dB.

4.3 Perbandingan kinerja metode HB pada kanal IRF dan URLOS yang memiliki perbedaan jumlah RF *Chain*, jumlah antena pemancar, dan jumlah antena penerima

Pada bagian ini akan ditampilkan perbandingan simulasi kinerja SE pada kanal IRF dan URLOS yang dibandingkan dengan menggunakan perbedaan jumlah NRF, jumlah antenna pemancar, dan jumlah antenna penerima. Hasil dari simulasi akan dibandingkan kinerjanya dengan menggunakan parameter *spectrum efficiency* (SE) *versus signal-to-noise ratio* (SNR).

4.3.1 Perbandingan metode HB dengan perbedaan jumlah RF *Chain* menggunakan kanal IRF

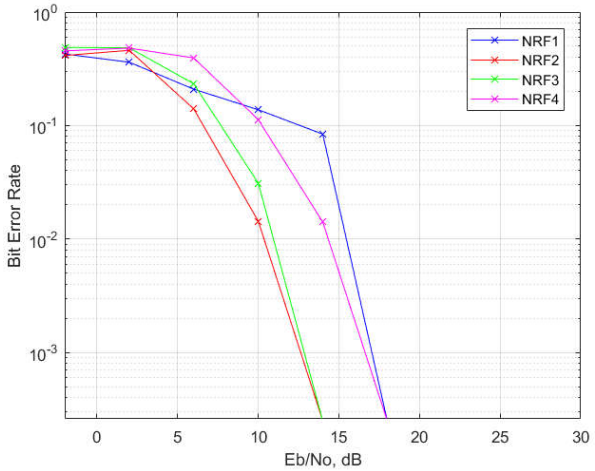
Pada simulasi ini akan dibandingkan kinerja SE yang memiliki perbedaan jumlah RF chain menggunakan kanal IRF. Pengujian ini memiliki tujuan untuk menguji kinerja dari SE apabila memiliki jumlah RF *chain* yang melebihi FDB (NRF2 = 2*Ns). Jumlah pemancar yang digunakan adalah sebanyak 64, jumlah antena penerima yang digunakan adalah sebanyak 4, dan jumlah RF *chain* yang digunakan adalah NRF1 = 1*N, NRF2 = 2*Ns, NRF3 = 3*Ns, dan NRF4 = 4*Ns.



Gambar 4.9 Grafik perbandingan kinerja SE pada kanal IRF dengan menggunakan beda jumlah RF *chain*

Tabel 4.9 Data SE pada kanal IRF dengan perbedaan jumlah RF *chain*

SNR	Spectral Efficiency			
	NRF1	NRF2	NRF3	NRF4
-2	0,4223	0,3686	0,08338	0,08717
2	0,7246	0,1458	0,06314	0,09251
6	2,39	3,488	3,133	1,615
10	8,132	10,23	10,42	8,963
14	14,14	16,88	16,44	15,71
18	18,6	22,53	21,93	21,09
22	25,72	27,29	27,67	27,51
26	32,39	32,91	32,66	32,2
30	37,71	38,61	37,98	38,21



Gambar 4.10 Grafik perbandingan kinerja BER pada kanal IRF dengan menggunakan beda jumlah RF *chain*

Tabel 4.10 Data BER pada kanal IRF dengan perbedaan jumlah RF *chain*

SNR	BER			
	NRF1	NRF2	NRF3	NRF4
-2	0,4272	0,4143	0,4878	0,4552
2	0,3613	0,4601	0,4842	0,481
6	0,2091	0,1401	0,2335	0,3942
10	0,1378	0,01436	0,03106	0,1129
14	0,084	0,00025	0,00025	0,01411
18	0,00025	0,00025	0,00025	0,00025
22	0,00025	0,00025	0,00025	0,00025
26	0,00025	0,00025	0,00025	0,00025
30	0,00025	0,00025	0,00025	0,00025

Dari Gambar 4.9 dan Tabel 4.9, dapat diamati bahwa pada saat SNR rendah (0-10 dB) rata-rata kinerja SE dari NRF3 dengan rata-rata kinerja SE sebesar 4,53871 bits/s/Hz lebih tinggi dibanding dengan yang lain, disusul oleh rata-rata kinerja SE NRF1 dengan rata-rata kinerja SE sebesar 3,74887 bits/s/Hz, kemudian disusul oleh rata-rata kinerja SE NRF4 sebesar 3,55684 bits/s/Hz, dan yang paling memiliki kinerja paling rendah pada SNR rendah adalah NRF2 dengan rata-rata kinerja SE sebesar 0.0486. Pada saat SNR sedang (10-20 dB) dapat diamati bahwa rata-rata kinerja SE yang paling tinggi adalah NRF2 sebesar 19,705 bits/s/Hz, kemudian disusul oleh rata-rata kinerja SE NRF3 sebesar 19,185 bits/s/Hz, lalu rata-rata kinerja SE NRF4 sebesar 18,4 bits/s/Hz, dan rata-rata kinerja SE yang paling rendah adalah rata-rata kinerja SE NRF1 sebesar 16,37 bits/s/Hz. Pada saat SNR tinggi (20-30 dB) dapat diamati bahwa rata-rata kinerja SE yang paling tinggi adalah rata-rata kinerja SE NRF2 sebesar 32,9367 bits/s/Hz, kemudian disusul oleh rata-rata kinerja SE NRF3 sebesar 32,77 bits/s/Hz, lalu rata-rata kinerja SE NRF4 sebesar 32,64 bits/s/Hz, dan yang memiliki rata-rata kinerja SE paling rendah adalah NRF1 sebesar 31,94 bits/s/Hz. Dapat diamati juga, secara menyeluruh dari SNR 0 hingga 30 dB, rata-rata kinerja SE NRF2 lebih unggul dibanding yang lain dengan rata-rata kinerja SE sebesar 19,0105 bits/s/Hz, disusul oleh rata-rata kinerja SE NRF3 sebesar 18,787 bits/s/Hz, kemudian rata-rata kinerja SE NRF4 sebesar 18,1738 bits/s/Hz, dan yang paling rendah adalah rata-rata kinerja SE dari NRF1 sebesar

17,4758 bits/s/Hz. Jika melihat Gambar 4.9, kita dapat mengetahui bahwa kinerja SE dari NRF3 dan NRF4 hampir mendekati NRF2, akan tetapi telah diamati melalui rata-rata dari 0 hingga 30 dB bahwa rata-rata kinerja SE yang paling baik dimiliki oleh NRF2 (FDB), disusul oleh NRF3, NRF4, dan yang terakhir NRF1(HB).

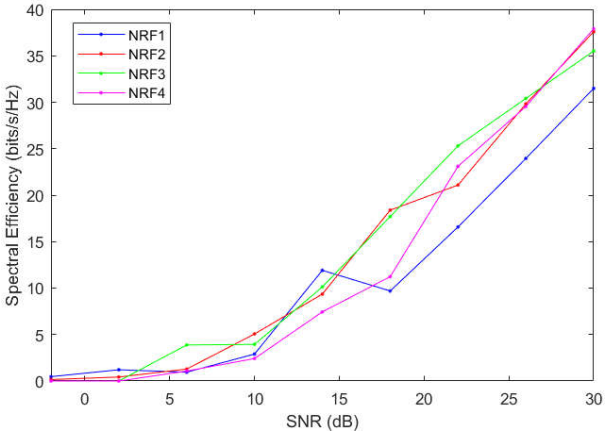
Kemdian dapat diamati grafik BER pada Gambar 4.10 dan data BER dari Tabel 4.10. Pada SNR rendah (0-10 dB) dapat dilihat pada saat SNR = 2 dB, keempat model *precoding* belum menunjukkan perbedaan yang signifikan, dapat dilihat NRF1 memiliki BER sebesar 0,3613, NRF2 memiliki BER sebesar 0,4601, NRF3 memiliki BER sebesar 0,4842, dan NRF4 memiliki BER sebesar 0,481. Akan tetapi saat SNR = 10, sudah bisa dilihat penurunan yang cukup signifikan. Jika dibandingkan, maka saat SNR = 10 dengan NRF2 memiliki BER yang paling rendah dibanding lainnya dengan BER sebesar 0,01436, kemudian disusul oleh NRF3 yang memiliki BER sebesar 0,03106 yang berada sedikit diatas NRF2, lalu disusul oleh NRF4 dengan BER sebesar 0,1129 dan NRF1 dengan BER sebesar 0,1378 yang mana masih memiliki kemungkinan tingkat kesalahan yang cukup tinggi. Pada saat SNR sedang (10-20 dB) saat SNR = 14 dB, dapat dilihat bahwa NRF2 dan NRF3 lebih dulu mencapai titik BER yang paling rendah yang dapat dicapainya yaitu BER sebesar 0,00025. Sedangkan NRF4 dan NRF1 masih memiliki BER yang cukup tinggi dimana BER NRF4 = 0,01411 yang kemudian disusul oleh NRF1 dengan BER 0,084. Pada saat SNR = 18 dB, semua model *precoding* dengan beda jumlah RF *chain* untuk NRF1 dan NRF4 telah mencapai titik BER paling rendah di 0,00025, juga NRF2 dan NRF3 masih berada pada titik itu, dan keseluruhan model *precoding* dengan beda RF *chain* konstan hingga SNR tinggi (SNR = 30 dB).

Dari pengamatan terhadap Gambar 4.9, Tabel 4.9, Gambar 4.10, dan Tabel 4.10, dapat diketahui bahwa penambahan jumlah RF *chain* pada pemancar tidak terlalu memiliki dampak yang signifikan terhadap kinerja SE, juga dapat dilihat tidak memiliki dampak yang signifikan terhadap pengujin BER.

4.3.2 Perbandingan metode HB dengan perbedaan jumlah RF Chain menggunakan kanal URLOS

Pada simulasi ini akan dibandingkan kinerja SE yang memiliki perbedaan jumlah RF chain menggunakan kanal URLOS. Pengujian ini memiliki tujuan untuk menguji kinerja dari SE apabila memiliki jumlah RF *chain* yang melebihi FDB (NRF2 = 2*Ns) yang kemudian akan dibandingkan dengan kinerja dari metode HB dan FDB. Jumlah pemancar

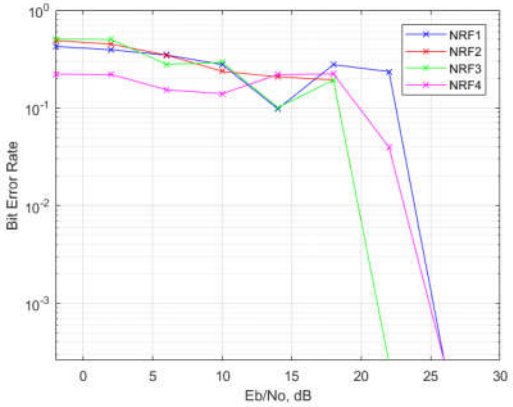
yang digunakan adalah sebanyak 64, jumlah antena penerima yang digunakan adalah sebanyak 4, dan jumlah RF *chain* yang digunakan adalah NRF1 = 1*N, NRF2 = 2*Ns, NRF3 = 3*Ns, dan NRF4 = 4*Ns.



Gambar 4.11 Grafik perbandingan kinerja SE pada kanal URLOS dengan menggunakan beda jumlah RF *chain*

Tabel 4.11 Data SE pada kanal URLOS dengan perbedaan jumlah RF *chain*

SNR	Spectral Efficiency			
	NRF1	NRF2	NRF3	NRF4
-2	0,4716	0,1572	0	0
2	1,206	0,4304	0	0
6	0,9474	1,283	3,879	1,07
10	2,897	5,062	3,938	2,416
14	11,91	9,355	10,12	7,437
18	9,687	18,39	17,7	11,23
22	16,58	21.08	25,3	23,1
26	23,95	29,82	30,41	29,54
30	31,49	37,58	35,51	37,9



Gambar 4.12 Grafik perbandingan kinerja BER pada kanal URLOS dengan menggunakan beda jumlah RF *chain*

Tabel 4.12 Data BER pada kanal URLOS dengan perbedaan jumlah RF *chain*

SNR	BER			
	NRF1	NRF2	NRF3	NRF4
-2	0,4254	0,4882	0,5065	0,2215
2	0,3922	0,4484	0,4995	0,2196
6	0,3462	0,3461	0,2779	0,1529
10	0,2772	0,2364	0,2915	0,1394
14	0,09769	0,2074	0,1007	0,2194
18	0,2766	0,1934	0,1921	0,2226
22	0,2346	0,00025	0,00025	0,03958
26	0,00025	0,00025	0,00025	0,00025
30	0,00025	0,00025	0,00025	0,00025

Pada Gambar 4.11 dan Tabel 4.11 dapat diamati bahwa untuk SNR rendah (0-10 dB), rata-rata kinerja yang paling tinggi ditempati oleh NRF3 dengan rata-rata kinerja SE sebesar 2,60567 bits/s/Hz, kemudian dibawahnya ditempati oleh NRF2 dengan rata-rata kinerja SE sebesar 2,25847 bits/s/Hz, lalu NRF1 dengan rata-rata kinerja SE sebesar 1,68347 bits/s/Hz, dan NRF4 dengan rata-rata kinerja SE sebesar 1,162 bits/s/Hz. Pada SNR sedang (10-20 dB) rata-rata kinerja SE NRF3 sebesar 13,91

bits/s/Hz menempati urutan kinerja yang paling tinggi, kemudian NRF2 yang memiliki selisih sedikit dengan rata-rata kinerja SE sebesar 13,8725 bits/s/Hz, lalu NRF1 yang memiliki rata-rata kinerja SE sebesar 10,7985 bits/s/Hz, dan yang peling rendah adalah NRF4 dengan rata-rata kinerja SE sebesar 9,3335 bits/s/Hz. Pada SNR tinggi (20-30 dB) rata-rata kinerja SE NRF3 berada yang paling tinggi dengan rata-rata kinerja SE sebesar 30,4067 bits/s/Hz, kemudian NRF4 dengan rata-rata kinerja SE sebesar 30,18 bits/s/Hz, NRF1 dengan rata-rata kinerja SE 24,0067 bits/s/Hz, dan NRF2 dengan besar rata-rata kinerja SE 22,4667 bits/s/Hz. Jika diamati dari SNR 0 dB sampai 30 dB, maka rata-rata kinerja SE tertinggi adalah saat menggunakan 3 RF *chain* (NRF3) dengan rata-rata kinerja SE sebesar 15,8571 bits/s/Hz, diikuti oleh NRF4 yang memiliki rata-rata kinerja SE sebesar 14,0866 bits/s/Hz, lalu NRF2 dengan rata-rata kinerja SE sebesar 12,7401 bits/s/Hz, dan yang paling rendah adalah NRF1 dengan rata-rata kinerja SE sebesar 12,3334 bits/s/Hz.

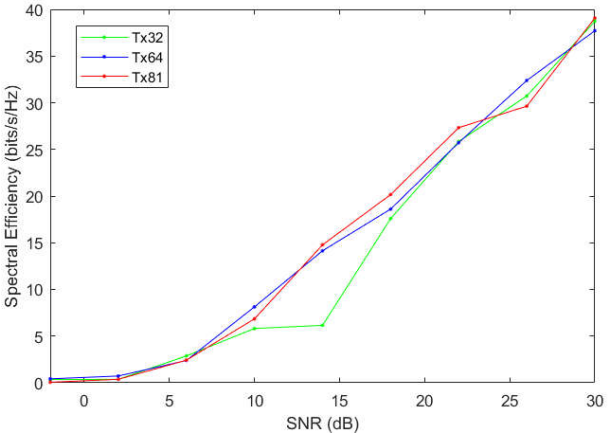
Kemudian mengamati Gambar 4.12 dan Tabel 4.12, saat SNR rendah (0-10 dB) saat SNR = 2 dB, NRF4 memiliki BER yang paling rendah dibanding lainnya dengan BER sebesar 0,2196, sementara diatasnya terdapat NRF1 dengan BER sebesar 0,3922, NRF2 dengan BER sebesar 0,4484, dan yang paling tinggi NRF3 dengan BER sebesar 0,4995. Saat SNR = 10 dB, NRF4 masih memiliki BER yang paling rendah dibanding lainnya dengan BER sebesar 0.1394, kemudian diatasnya terdapat NRF2 dengan BER sebesar 0,2364, lalu NRF1 dengan BER sebesar 0,2772, dan yang paling tinggi adalah NRF3 dengan BER sebesar 2.915. Dari sini dapat dilihat bahwa untuk SNR rendah, NRF4 paling stabil meskipun angka BERnya masih cukup tinggi, sedangkan untuk NRF1, NRF2, dan NRF3, masih memiliki angka BER yang belum konstan/stabil dan memiliki BER yang tinggi. Saat SNR sedang (10-20 dB) saat SNR = 14 dB, kinerja BER yang paling rendah dimiliki oleh NRF1 dengan besar BER 0,09769 yang memiliki sedikit selisih dengan NRF3 dengan BER sebesar 0,1007, kemudian NRF2 yang berada diatas NRF1 dan NRF3 dengan BER sebesar 0,2074, dan yang paling besar adalah BER dari NRF4 dengan besar BER sebesar 0,2194. Kemudian pada saat SNR = 18 dB, BER paling rendah adalah NRF3 dengan BER sebesar 0,1921, disusul NRF2 dengan BER sebesar 0,1934, lalu NRF4 dengan BER sebesar 0,2226, dan yang paling tinggi adalah NRF1 dengan BER sebesar 0,2766. Pada saat SNR tinggi (20-30 dB), saat SNR = 22 dB, NRF2 dan NRF3 telah mencapai titik paling rendah yang dapat dicapainya dengan BER sebesar 0,00025 dan konstan hingga SNR = 30

dB. Sedangkan pada SNR = 22 dB, untuk NRF4 dengan BER sebesar 0,03958 berada di bawah NRF1 dengan BER 0,2346 yang mana angka BER nya masih cukup tinggi. NRF1 dan NRF4 baru mencapai BER terendah pada SNR = 26 dB dengan BER sebesar 0,00025 yang konstan hingga SNR = 30 dB.

Dari hasil pengamatan Gambar 4.11, Tabel 4.11, Gambar 4.12, dan Tabel 4.12, dapat dilihat bahwa penambahan RF *chain* pada kanal URLOS untuk NRF3 dan NRF4 mendapat rata-rata kinerja SE yang lebih baik dibanding dengan NRF2 (FDB) dan NRF1 (HB), akan tetapi untuk BER, NRF3 memiliki angka BER yang lebih rendah dibanding NRF2 (FDB) pada SNR sedang dan SNR tinggi, sedangkan NRF4 memiliki BER diatas NRF1 (HB) namun masih dibawah NRF2 (FDB) dan NRF3.

4.3.3 Perbandingan metode HB dengan perbedaan jumlah antenna pemancar pada kanal IRF

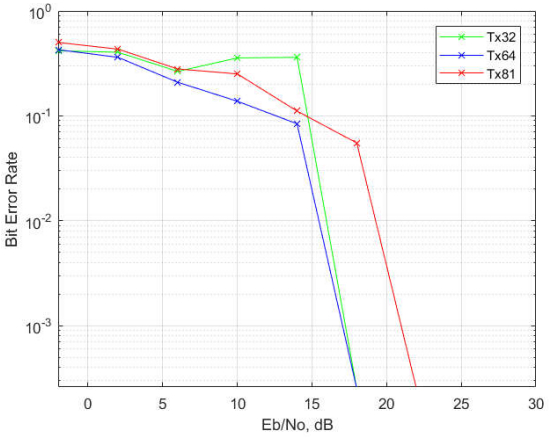
Pada bagian ini akan dibandingkan kinerja dari jumlah antenna pemancar menggunakan metode *hybrid* pada kanal IRF. Jumlah antenna pemancar yang digunakan adalah 81 (Tx81), 64 (Tx64), dan 32 (Tx32) antena. Menggunakan RF *chain* sebanyak 1*Ns, dan menggunakan 4 antena penerima.



Gambar 4.13 Grafik perbandingan kinerja SE pada kanal IRF dengan menggunakan beda jumlah antena pemancar

Tabel 4.13 Data SE pada kanal IRF dengan perbedaan jumlah antena pemancar

SNR	Spectral Efficiency		
	Tx32	Tx64	Tx81
-2	0,351	0,4223	0,05848
2	0,3677	0,7246	0,3707
6	2,875	2,39	2,421
10	5,817	8,132	6,849
14	6,151	14,14	14,77
18	17,59	18,6	20,16
22	25,84	25,72	27,33
26	30,72	32,39	29,63
30	38,73	37,71	39,07



Gambar 4.14 Grafik perbandingan kinerja BER pada kanal IRF dengan menggunakan beda jumlah antena

Tabel 4.14 Data BER pada kanal IRF dengan perbedaan jumlah antena pemancar

SNR	BER		
	Tx32	Tx64	Tx81
-2	0,417	0,4272	0,5003
2	0,4041	0,3613	0,4328

6	0,2652	0,2091	0,2799
10	0,3565	0,1378	0,2511
14	0,3612	0,084	0,111
18	0,00025	0,00025	0,05514
22	0,00025	0,00025	0,00025
26	0,00025	0,00025	0,00025
30	0,00025	0,00025	0,00025

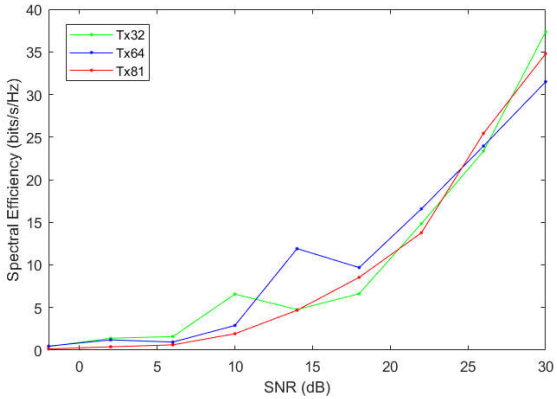
Pada Gambar 4.13 dan Tabel 4.13, dapat diamati bahwa pada saat SNR rendah (0-10 dB) rata-rata kinerja SE dari Tx32 adalah 3,0199 bits/s/Hz, yang berada dibawah Tx81 dengan rata-rata kinerja SE sebesar 3.21357 bits/s/Hz dan dibawah Tx64 dengan rata-rata kinerja SE sebesar 3,74887 bits/s/Hz. Saat SNR sedang (10-20 dB) rata-rata kinerja SE Tx32 sebesar 11,8705 bits/s/Hz dibawah rata-rata kinerja SE Tx64 sebesar 16,37 bits/s/Hz, juga dibawah rata-rata kinerja SE Tx81 sebesar 17,465 bits/s/Hz. Kemudian untuk SNR tinggi (20-30 dB) rata-rata kinerja SE Tx32 berada paling bawah dengan rata-rata kinerja sebesar 31,7633 bits/s/Hz, kemudian diatasnya terdapat rata-rata kinerja SE dari Tx64 sebesar 31,94 bits/s/Hz, dan yang paling tinggi adalah rata-rata kinerja SE dari Tx81 sebesar 32,01 bits/s/Hz. Jika dilihat pada saat SNR 0 dB hingga 30 dB, maka rata-rata kinerja SE yang paling kecil adalah rata-rata kinerja SE dari Tx32 sebesar 16,0113 bits/s/Hz, kemudian diatasnya terdapat rata-rata kinerja SE dari Tx64 sebesar 17,4758 bits/s/Hz, dan yang paling tinggi adalah rata-rata kinerja SE Tx81 sebesar 17,5751 bits/s/Hz.

Pada Gambar 4.14 dan Tabel 14.14 dapat dilihat saat SNR rendah (0-10 dB) saat SNR = 10 dB, Tx64 memiliki angka BER yang paling kecil yaitu sebesar 0.1378, kemudian diatasnya terdapat Tx81 dengan BER sebesar 0,2511, dan yang tertinggi adalah Tx32 dengan BER sebesar 0,3565. Saat SNR sedang (10-20 dB) dapat dilihat bahwa saat SNR = 14 dB, Tx64 dengan BER sebesar 0,084 merupakan angka BER yang paling kecil yang kemudian diatasnya terdapat BER dari Tx81 sebesar 0,111, dan yang paling tinggi adalah BER dari Tx32 sebesar 0,3612. Pada saat SNR = 18 dB, Tx32 dan Tx64 lebih dulu mencapai titik BER yang paling rendah yang bisa dicapai, yakni 0,00025, sedangkan Tx81 masih memiliki BER yang cukup tinggi di angka 0,05514. Untuk SNR tinggi (20-30 dB), pada saat SNR 22, Tx81 baru mencapai titik terendah yang bisa dicapai dengan BER sebesar 0,00025, dan ketiganya memiliki BER yang konstan di angka 0,00025 hingga SNR = 30 dB.

Jika mengaamati Gambar 4.13 dan Tabel 4.13 menunjukkan bahwa semakin banyak jumlah antena pemancar, maka rata-rata kinerja SE yang didapat juga semakin tinggi. Sedangkan jika melihat Gambar 4.14 dan Tabel 4.14 menunjukkan bahwa Tx64 memiliki kinerja yang terbaik dibanding Tx32 dan Tx81.

4.3.4 Perbandingan menggunakan metode HB dengan perbedaan jumlah antenna pemancar pada kanal URLOS

Pada bagian ini akan dibandingkan kinerja dari jumlah antenna pemancar menggunakan metode *hybrid* pada kanal URLOS. Jumlah antenna pemancar yang digunakan adalah 81 (Tx81), 64 (Tx64), dan 32 (Tx32) antena. Menggunakan RF *chain* sebanyak 1*Ns, dan menggunakan 4 antena penerima.

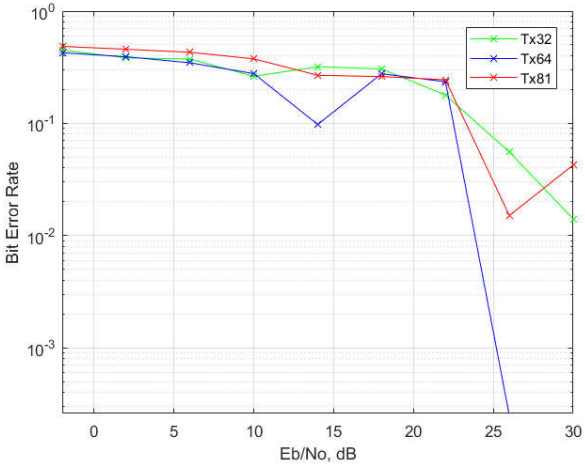


Gambar 4.15 Grafik perbandingan kinerja SE pada kanal URLOS dengan menggunakan beda jumlah antena pemancar

Tabel 4.15 Data SE pada kanal URLOS dengan perbedaan jumlah antena pemancar

SNR	Spectral Efficiency		
	Tx32	Tx64	Tx81
-2	0,4207	0,4716	0,1373
2	1,395	1,206	0,394
6	1,592	0,9474	0,6221
10	6,57	2,897	1,927
14	4,762	11,91	4,681

18	6,626	9,687	8,544
22	14,84	16,58	13,79
26	23,35	23,95	25,43
30	37,35	31,49	34,78



Gambar 4.16 Grafik perbandingan kinerja BER pada kanal URLOS dengan menggunakan beda jumlah antena pemancar

Tabel 4.16 Data BER pada kanal URLOS dengan perbedaan jumlah antena pemancar

SNR	BER		
	Tx32	Tx64	Tx81
-2	0,4497	0,4254	0,48583
2	0,3834	0,3922	0,4569
6	0,3737	0,3462	0,4296
10	0,2626	0,2772	0,3755
14	0,3193	0,09769	0,268
18	0,3053	0,2766	0,2609
22	0,01787	0,2346	0,2444
26	0,05564	0,00025	0,01508
30	0,01406	0,00025	0,04247

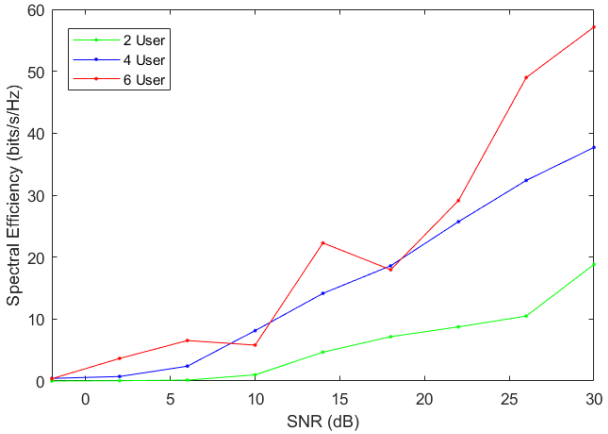
Dapat dilihat pada Gambar 4.15 dan Tabel 4.15, dapat diamati pada saat SNR rendah (0-10 dB), rata-rata kinerja SE untuk Tx32 adalah sebesar 3,18567 bits/s/Hz, sedangkan rata-rata kinerja SE untuk Tx64 adalah sebesar 1,68347 bits/s/Hz, dan untuk Tx81 memiliki rata-rata kinerja SE sebesar 0,98103 bits/s/Hz. Saat SNR sedang (10-20 dB), rata-rata kinerja SE untuk Tx32 adalah sebesar 5,694 bits/s/Hz, sedangkan rata-rata kinerja SE untuk Tx64 adalah sebesar 10,7985 bits/s/Hz, dan rata-rata kinerja SE untuk Tx81 adalah sebesar 6,6125 bits/s/Hz. Untuk SNR tinggi (20-30 dB) rata-rata kinerja SE untuk Tx32 adalah 25,18 bits/s/Hz, sedangkan rata-rata kinerja SE untuk Tx64 adalah 24,0067 bits/s/Hz, dan rata-rata kinerja SE untuk Tx81 adalah 24,6667 bits/s/Hz. Dari sini dapat diketahui bahwa untuk SNR rendah, rata-rata kinerja SE Tx32 lebih baik dibanding yang lain, untuk SNR sedang rata-rata kinerja SE Tx64 lebih baik dibanding yang lain, dan untuk SNR tinggi rata-rata kinerja SE Tx32 adalah yangh paling baik dibanding yang lain. Kemudian untuk SNR 0 dB hingga 30 dB, Tx64 memiliki rata-rata kinerja SE paling baik yaitu sebesar 12,3334 bits/s/Hz, disusul oleh Tx32 dengan rata-rata kinerja SE sebesar 12,0606 bits/s/Hz, dan yang paling rendah adalah rata-rata kinerja SE dari Tx81 yaitu sebesar 11,271 bits/s/Hz.

Dapat dilihat pada Gambar 4.16 dan Tabel 4.16, dapat diamati pada saat SNR rendah (0-10 dB), pada SNR = 10 dB, Tx32 memiliki BER yang paling rendah yaitu sebesar 0.2626, kemudian Tx64 memiliki BER sebesar 0.2772, sedangkan Tx81 memiliki BER yang paling tinggi yaitu 0.3755. Pada saat SNR sedang (10-20 dB), pada SNR = 14 dB, Tx64 memiliki BER sebesar 0,09769 yang merupakan yang paling kecil walaupun masih memiliki tingkat kemungkinan error yang tinggi, kemudian Tx81 yang memiliki BER sebesar 0,268 yang berada diatas Tx64, dan Tx32 memiliki BER sebesar 0,3193. Kemudian saat SNR tinggi (20-30 dB) pada saat SNR = 26 dB, Tx64 telah mencapai titik terendah yang bisa dicapainya yaitu 0,00025 dan terus konstan hingga SNR = 30 dB. Sementara pada saat SNR = 26 dB, Tx81 yang memiliki BER sebesar 0,01508 berada di bawah Tx32 dengan BER 0,05564. Saat SNR = 30 dB, Tx32 memiliki BER sebesar 0,01406, sementara Tx81 memiliki BER sebesar 0,04247.

Dari Gambar 4.15, Tabel 4.15, Gambar 4.16, dan Tabel 4.16 dapat diamati bahwa pemancar dengan jumlah 64 antena lebih baik dari segi rata-rata kinerja SE dan angka BER dibandingkan dengan jumlah antena 32 dan antena 81 pada kanal URLOS.

4.3.5 Perbandingan menggunakan metode HB dengan perbedaan jumlah user/antena penerima pada kanal IRF

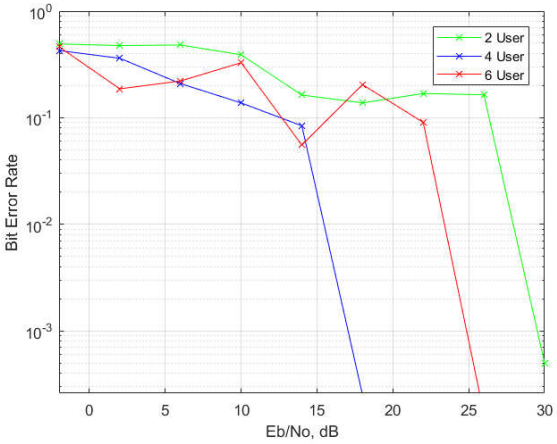
Pada bagian ini akan ditampilkan perbandingan metode HB dengan perbedaan jumlah user. Jumlah antena pemancar yang digunakan adalah 64, jumlah RF *chain* yang digunakan adalah 1*Ns, sementara jumlah antena penerima yang digunakan adalah 2 user, 4 user, dan 6 user.



Gambar 4.17 Grafik perbandingan kinerja SE pada kanal IRF dengan menggunakan beda jumlah antena penerima

Tabel 4.17 Data SE pada kanal IRF dengan perbedaan jumlah antena penerima

SNR	Spectral Efficiency		
	2 User	4 User	8 User
-2	0	0,4223	0,3678
2	0,04838	0,7246	3,643
6	0,1429	2,39	6,53
10	0,9937	8,132	5,795
14	4,656	14,14	22,29
18	7,158	18,6	17,96
22	8,741	25,72	29,13
26	10,48	32,39	48,99
30	18,81	37,71	57,51



Gambar 4.18 Grafik perbandingan kinerja BER pada kanal IRF dengan menggunakan beda jumlah antena penerima

Tabel 4.18 Data BER pada kanal IRF dengan perbedaan jumlah antena penerima

SNR	BER		
	2 User	4 User	8 User
-2	0,4928	0,4272	0,4627
2	0,4754	0,3613	0,1867
6	0,4815	0,2091	0,2213
10	0,3897	0,1378	0,3285
14	0,1633	0,084	0,05544
18	0,1378	0,00025	0,2044
22	0,1689	0,00025	0,09057
26	0,1641	0,00025	0,0001667
30	0,0005	0,00025	0,0001667

Pada Gambar 4.17 dan Tabel 4.17 dapat diamati bahwa pada saat SNR rendah (0-10 dB) saat menggunakan 2 user memiliki rata-rata kinerja SE sebesar 0,39499 bits/s/Hz, sedangkan saat menggunakan 4 user memiliki rata-rata kinerja SE sebesar 3,74887 bits/s/Hz, dan untuk 6 user memiliki rata-rata kinerja SE sebesar 5,532267 bits/s/Hz. Kemudian saat SNR sedang (10-20 dB) rata-rata kinerja SE yang diperoleh pada 2 user adalah 5,907 bits/s/Hz, sedangkan rata-rata kinerja SE yang diperoleh

pada 4 user adalah 16,37 bits/s/Hz, dan untuk 6 user diperoleh rata-rata kinerja SE sebesar 20,125 bits/s/Hz. Pada saat SNR tinggi (20-30 dB), rata-rata kinerja SE yang diperoleh pada 2 user adalah 12,677 bits/s/Hz, sedangkan rata-rata kinerja SE yang diperoleh pada 4 user adalah 31,94 bits/s/Hz, dan untuk 6 user rata-rata kinerja SE yang diperoleh adalah 45,21 bits/s/Hz. Sementara untuk SNR dari 0 dB hingga 30 dB, rata-rata kinerja SE yang diperoleh pada 2 user adalah 6,37875 bits/s/Hz, sedangkan rata-rata kinerja SE yang diperoleh pada 4 user adalah 17,4758 bits/s/Hz, dan untuk 6 user rata-rata kinerja yang diperoleh adalah 23,981 bits/s/Hz.

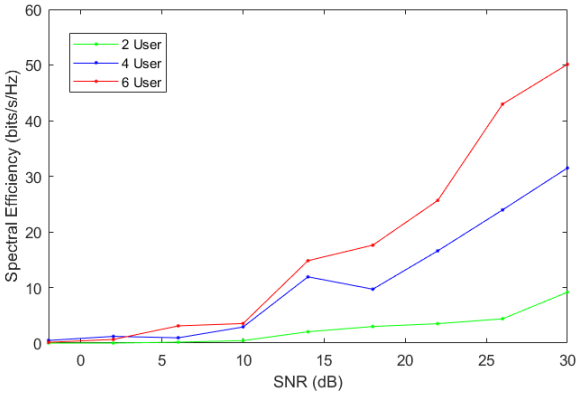
Pada Gambar 4.18 dan tabel 4.18, dapat diamati bahwa pada saat SNR rendah (0-10 dB) tepatnya saat SNR = 10 dB, BER dari user 2 memiliki angka yang paling tinggi yaitu 0,4754, kemudian terdapat BER dari user 6 di bawah user 2 yang memiliki BER sebesar 0,3285, dan user 4 dengan BER yang paling kecil yaitu 0,1378. Kemudian saat BER sedang (10-20 dB) pada saat SNR = 18 dB, metode HB dengan 4 user telah mencapai titik BER terendah yang dapat dicapainya yaitu 0,00025 yang konstan hingga SNR = 30 dB. Kemudian pada saat SNR = 18, metode HB dengan 6 user memiliki BER dengan angka yang paling tinggi yaitu 0,2044 dan metode HB dengan 2 user memiliki BER sebesar 0,1378. Pada saat SNR tinggi (20-30 dB), saat SNR = 26 dB, metode HB dengan user 6 telah mencapai titik yang paling rendah yang dapat dicapai yaitu 0,00017 dan konstan hingga SNR = 30 dB, sementara metode HB dengan 2 user memiliki BER sebesar 0,1641. Saat SNR = 30 dB, user 2 telah mencapai BER yang paling rendah yang dapat dicapainya sebesar 0,0005.

Dari Gambar 4.17 dan Tabel 4.17 diamati bahwa metode HB dengan 2 user memiliki rata-rata kinerja SE yang paling bawah, sedangkan untuk metode HB dengan 4 user memiliki rata-rata kinerja SE yang berada di atas user 2, dan yang paling tinggi adalah metode HB dengan 6 user. Meskipun begitu, pada pengamatan Gambar 4.18 dan Tabel 4.18, diketahui bahwa BER yang terbaik dimiliki oleh metode HB dengan 4 user, yang kemudian disusul oleh metode HB dengan 6 user yang baru bisa mencapai kondisi sedikit error pada SNR = 26 dB, dan yang terburuk adalah metode HB dengan menggunakan 2 user.

4.3.6 Perbandingan menggunakan metode HB dengan perbedaan jumlah user/antena penerima pada kanal URLOS

Pada bagian ini akan ditampilkan perbandingan metode HB dengan perbedaan jumlah user. Jumlah antena pemancar yang digunakan adalah

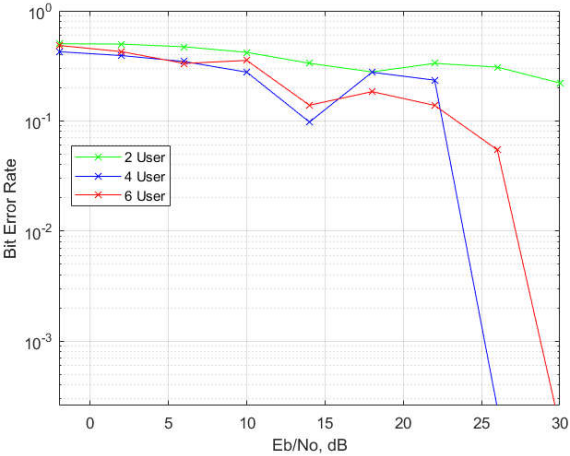
64, jumlah RF *chain* yang digunakan adalah 1*Ns, sementara jumlah antena penerima yang digunakan adalah 2 user, 4 user, dan 6 user.



Gambar 4.19 Grafik perbandingan kinerja SE pada kanal URLOS dengan menggunakan beda jumlah antena penerima

Tabel 4.19 Data SE pada kanal URLOS dengan perbedaan jumlah antena penerima

SNR	Spectral Efficiency		
	2 User	4 User	8 User
-2	0	0,4716	0,1362
2	0	1,206	0,6535
6	0,1626	0,9474	3,092
10	0,4456	2,897	3,514
14	2,033	11,91	14,82
18	2,979	9,687	17,61
22	3,484	16,58	25,64
26	4,36	23,95	42,97
30	9,15	31,49	50,11



Gambar 4.20 Grafik perbandingan kinerja BER pada kanal URLOS dengan menggunakan beda jumlah antena penerima

Tabel 4.20 Data BER pada kanal URLOS dengan perbedaan jumlah antena penerima

SNR	BER		
	2 User	4 User	8 User
-2	0,5026	0,4254	0,4851
2	0,4984	0,3922	0,4259
6	0,4712	0,3462	0,3329
10	0,4198	0,2772	0,3544
14	0,3339	0,09769	0,1389
18	0,2786	0,2766	0,1845
22	0,3334	0,2346	0,1384
26	0,3081	0,00025	0,05467
30	0,221	0,00025	0,0001667

Pada Gambar 4.19 dan Tabel 4.19 dapat diamati bahwa pada saat SNR rendah (0-10 dB) rata-rata kinerja SE dari metode HB dengan 2 user adalah sebesar 0,20273 bits/s/Hz, sedangkan rata-rata kinerja SE dari metode HB dengan 4 user adalah 1,68347 bits/s/Hz, dan untuk 6 user memperoleh rata-rata kinerja SE sebesar 2,41983. Pada saat SNR sedang

(10-20 dB) rata-rata kinerja SE dari metode HB dengan 2 user adalah sebesar 2,506 bits/s/Hz, sedangkan metode HB dengan 4 user memperoleh rata-rata kinerja SE sebesar 10,7985 bits/s/Hz, dan untuk 6 user mendapatkan rata-rata kinerja SE sebesar 16,215 bits/s/Hz. Pada SNR tinggi (20-30 dB) metode HB dengan 2 user memperoleh rata-rata kinerja SE sebesar 5,66467 bits/s/Hz, sedangkan metode HB dengan 4 user memperoleh rata-rata kinerja SE sebesar 24,0067 bits/s/Hz, dan untuk 6 user mendapatkan rata-rata kinerja SE sebesar 39,5733 bits/s/Hz. Sementara untuk SNR dari 0 dB hingga 30 dB, metode HB dengan 2 user memperoleh rata-rata kinerja SE sebesar 2,82678 bits/s/Hz, sedangkan metode HB dengan 4 user memiliki rata-rata kinerja SE sebesar 12,3334 bits/s/Hz, dan untuk 6 user memperoleh rata-rata kinerja SE sebesar 19,8012 bits/s/Hz.

Pada Gambar 4.20 dan Tabel 4.20 dapat diamati pada saat SNR rendah (0-10 dB) saat SNR = 10 dB, metode HB dengan 2 user memiliki BER sebesar 0,4198, sedangkan metode HB dengan 4 user memiliki BER sebesar 0,2772, dan untuk metode HB dengan 6 user memiliki BER sebesar 0,3544. Pada saat SNR sedang (10-20 dB), saat SNR = 18 dB, metode HB dengan 2 user memiliki BER sebesar 0,2786, sedangkan metode HB dengan 4 user memiliki BER sebesar 0,2766, dan untuk metode HB dengan 6 user memiliki BER sebesar 0,1845. Pada saat SNR tinggi (20-30 dB) saat SNR = 26 dB, metode HB dengan user 4 lebih dulu mencapai titik paling rendah yang dapat dicapai yaitu BER sebesar 0.00025 dan terus konstan hingga SNR = 30 dB. Saat SNR = 26 dB, metode HB dengan 2 user dan 6 user masih memiliki angka BER yang tinggi yaitu sebesar 0,3081 dan 0,05467. Pada saat SNR = 30 dB, metode HB dengan 2 user masih memiliki angka BER yang tinggi yaitu sebesar 0,221, sedangkan metode HB dengan 6 user mencapai titik terendah yang bisa dicapai yaiu sebesar 0,00017.

Dari pengamatan Gambar 4.19, dan Tabel 4.19, dapat diketahui bahwa untuk rata-rata kinerja SE yang paling tinggi dicapai oleh metode HB dengan 6 user, kemudian diikuti oleh metode HB dengan 6 user, dan yang paling rendah adalah metode HB dengan 2 user. Meskipun begitu, dapat diamati juga pada Gambar 4.20 dan Tabel 4.20, bahwa metode HB dengan 4 user memiliki angka BER yang paling baik dibanding dengan jumlah antena penerima lainnya.

4.4 Tambahan

Pada sub bab ini akan diberikan penjelasan mengenai korelasi kanal, juga akan ditampilkan perbandingan metode HB dengan perbedaan sudut antar pada kanal URLOS yang dibandingkan dengan kanal IRF, perbandingan metode HB dengan perbedaan jumlah antena pemancar pada kanal URLOS, dan pola radiasi.

4.4.1 Korelasi Kanal

Tabel 4.21 Korelasi kanal IRF

Tabel Korelasi Kanal IRF				
	1	2	3	4
1	1	-0,074	0,0504	0,0669
2	-0,074	1	0,0362	0,0476
3	0,0504	0,0362	1	-0,1016
4	0,0669	0,0476	-0,1016	1

Tabel 4.22 Korelasi kanal URLOS 5 derajat

Tabel Korelasi Kanal URLOS 5 drjt				
	1	2	3	4
1	1	-0,0227	-0,0283	0,0318
2	-0,0227	1	-0,0329	0,0358
3	-0,0283	-0,0329	1	0,066
4	0,0318	0,0358	0,066	1

Tabel 4.23 Korelasi kanal URLOS 20 derajat

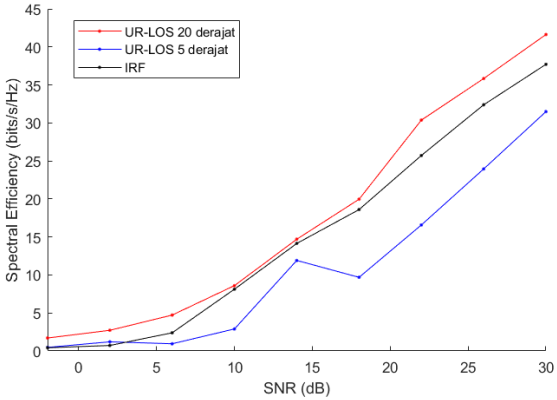
Tabel Korelasi Kanal URLOS 20 drjt				
	1	2	3	4
1	1	0,0304	0	-0,0149
2	0,0304	1	0	0,0049
3	0	0	1	0
4	-0,0149	0,0049	0	1

Pada sub bab berisi data dari korelasi yang terjadi pada masing-masing kanal. Pada tabel 4.21 terdapat data korelasi kanal IRF yang mana

bersifat independen antar elemennya, dimana maksud independen adalah antar kanalnya tidak saling mempengaruhi. Berbeda halnya dengan pada kanal URLOS yang akan saling mempengaruhi. Dapat dilihat pada tabel 4.22 dan 4.23, dapat dilihat serta dibandingkan tiap-tiap elemennya bahwa korelasi pada kanal URLOS 20 derajat lebih kecil jika dibandingkan dengan korelasi pada kanal URLOS 5 derajat. Semakin tinggi nilai dari korelasi kanal, maka kinerja baik SE dan BER akan semakin menurun.

4.4.2 Perbandingan metode HB dengan perbedaan sudut pada kanal URLOS yang dibandingkan dengan kanal IRF

Akan ditampilkan perbandingan antara metode HB pada kanal IRF dengan metode HB pada kanal URLOS 5 derajat dan 20 derajat untuk mengetahui pengaruh dari korelasi kanal.

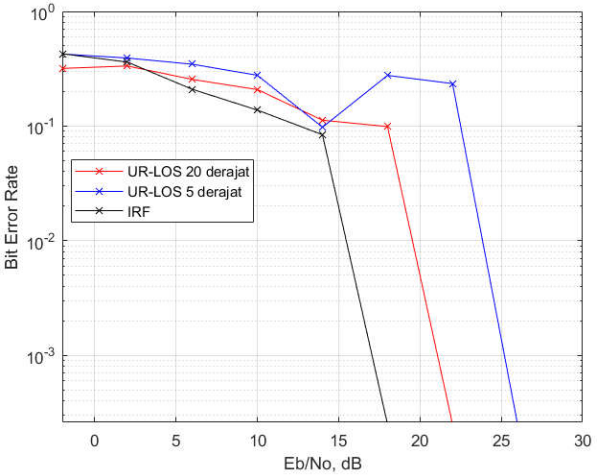


Gambar 4.21 Grafik perbandingan kinerja SE metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF

Tabel 4.24 Data perbandingan kinerja SE metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF

SNR	Spectral efficiency		
	IRF	URLOS5	URLOS20
-2	0,4223	0,4716	1,705
2	0,7246	1,206	2,721
6	2,39	0,9474	4,723
10	8,132	2,897	8,6

14	14,14	11,91	14,7
18	18,6	9,687	19,96
22	25,72	16,58	30,37
26	32,39	23,95	35,84
30	37,71	31,49	41,63



Gambar 4.22 Grafik perbandingan kinerja BER metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF

Tabel 4.25 Data perbandingan kinerja BER metode HB pada kanal URLOS 5 derajat, URLOS 20 derajat, dan IRF

HB IRF vs URLOS 5,20			
SNR	BER		
	IRF	URLOS5	URLOS20
-2	0,4272	0,4254	0,3184
2	0,3613	0,3922	0,3342
6	0,2091	0,3462	0,2552
10	0,1378	0,2772	0,2079
14	0,084	0,09769	0,1124
18	0,00025	0,2766	0,09889

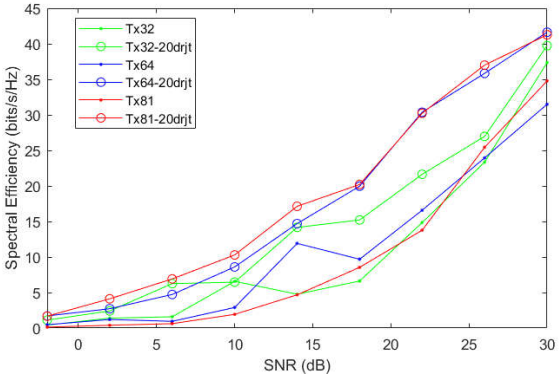
22	0,00025	0,2346	0,00025
26	0,00025	0,00025	0,00025
30	0,00025	0,00025	0,00025

Pada Gambar 4.21 dan Tabel 4.24 dapat dilihat bahwa terdapat peningkatan kinerja pada kanal URLOS 20 derajat dibandingkan dengan kanal URLOS 5 derajat. Hal ini membuktikan bahwa pada simulasi yang dilakukan sebelumnya dengan kanal URLOS 5 derajat terjadi korelasi kanal yang cukup mempengaruhi kinerja dari SE pada kanal URLOS. Dapat dilihat dari Gambar 33 bahwa kanal URLOS 20 derajat memiliki grafik kinerja SE yang paling baik dibandingkan dengan kanal IRF dan kanal URLOS 5 derajat.

Pada Gambar 4.22 dan Tabel 4.25, dapat dilihat bahwa terdapat perbaikan kinerja BER pada kanal URLOS 20 derajat jika dibandingkan dengan kanal URLOS 5 derajat. Dapat dilihat bahwa kinerja BER 0,00025 pada kanal URLOS 20 derajat dapat dicapai sejak SNR 22 dB, sedangkan pada kanal URLOS 5 derajat baru dapat mencapai pada saat SNR 26 dB. Secara keseluruhan kinerja BER IRF adalah yang paling baik jika dengan yang lain, diikuti dengan kinerja BER kanal URLOS 20 derajat, dan yang paling buruk adalah kinerja BER kanal URLOS 5 derajat.

4.4.3 Perbandingan metode HB dengan perbedaan jumlah antena pemancar pada kanal URLOS 5 derajat dan URLOS 20 derajat

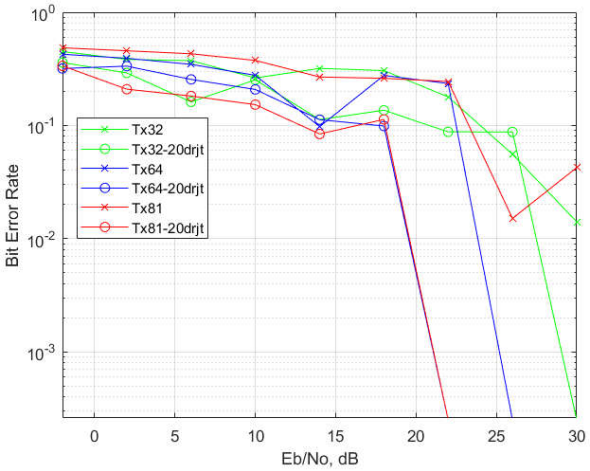
Akan ditampilkan perbandingan antara perbedaan jumlah antena pada kanal URLOS 5 derajat dan URLOS 20 derajat untuk mengetahui pengaruh dari korelasi kanal yang terjadi.



Gambar 4.23 Grafik perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat

Tabel 4.26 Data perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat

SNR	Spectral efficiency					
	32	64	81	32(20)	64(20)	81(20)
-2	0,4207	0,4716	0,1373	1,124	1,705	1,678
2	1,395	1,206	0,394	2,406	2,721	4,116
6	1,592	0,9474	0,6221	6,253	4,723	6,914
10	6,57	2,897	1,927	6,457	8,6	10,29
14	4,762	11,91	4,681	14,17	14,7	17,13
18	6,626	9,687	8,544	15,21	19,96	20,21
22	14,84	16,58	13,79	21,63	30,37	30,22
26	23,35	23,95	25,43	26,97	35,84	37
30	37,35	31,49	34,78	39,74	41,63	41,27



Gambar 4.24 Grafik perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat

Tabel 4.27 Data perbandingan kinerja SE metode HB dengan beda jumlah antena pemancar pada kanal URLOS 5 derajat dan 20 derajat

Beda Jumlah Pemancar Kanal URLOS						
SNR	BER					
	32	64	81	32(20)	64(20)	81(20)
-2	0,4497	0,4254	0,48583	0,3594	0,3184	0,3359
2	0,3834	0,3922	0,4569	0,2911	0,3342	0,2092
6	0,3737	0,3462	0,4296	0,1615	0,2552	0,1819
10	0,2626	0,2772	0,3755	0,2529	0,2079	0,1527
14	0,3193	0,09769	0,268	0,1127	0,1124	0,08408
18	0,3053	0,2766	0,2609	0,1364	0,09889	0,1131
22	0,01787	0,2346	0,2444	0,08772	0,00025	0,00025
26	0,05564	0,00025	0,01508	0,08767	0,00025	0,00025
30	0,01406	0,00025	0,04247	0,00025	0,00025	0,00025

Dapat dilihat pada Gambar 4.23 dan Tabel 4.26 bahwa kinerja SE pada kanal URLOS 20 derajat meningkat pada semua jumlah pemancar

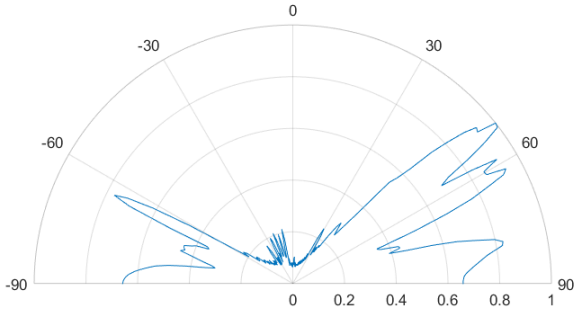
jika dibandingkan dengan kanal URLOS 5 derajat. Hal ini juga membuktikan bahwa simulasi yang dilakukan sebelumnya mengalami korelasi kanal yang membuat kinerja SE semakin menurun.

Dapat dilihat pada Gambar 4.24 dan Tabel 4.27 bahwa terjadi perbaikan kinerja BER pada semua jumlah pemancar pada kanal URLOS 20 derajat jika dibandingkan dengan kinerja BER URLOS 5 derajat. Ini membuktikan bahwa pada saat simulasi BER URLOS 5 derajat terdapat korelasi kanal yang menyebabkan kinerja BER menjadi lebih buruk.

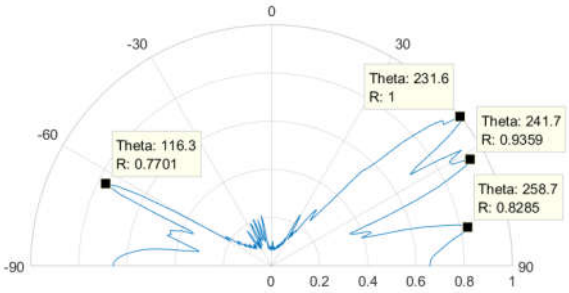
4.4.4 Pola radiasi dari metode HB pada kanal IRF dan pada kanal URLOS 20 derajat

4.4.4.1 Pola radiasi pada kanal IRF

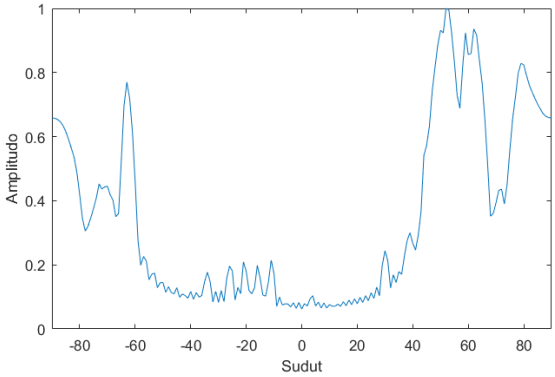
Berikut adalah plot pola radiasi pada kanal IRF:



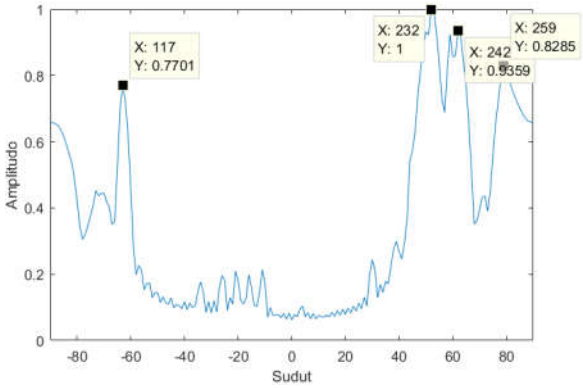
Gambar 4.25 Plot polar kanal IRF



Gambar 4.26 Plot polar kanal IRF dengan rekomendasi sudut



Gambar 4.28 Plot rectangular kanal IRF



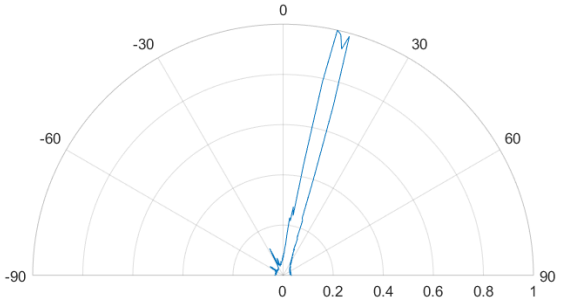
Gambar 4.27 Plot rectangular kanal IRF dengan rekomendasi sudut

Dapat dilihat pada Gambar 37 dan Gambar 38 adalah plot polar dari pola radiasi kanal IRF, dimana karena pada simulasi ini digunakan jumlah antena penerima sebanyak empat, maka akan diberikan empat rekomendasi lokasi dari antena penerima yang akan mendapatkan pancaran radiasi dengan amplitudo yang tinggi. Karena menggunakan antena ULA (*Uniform Linear Array*) maka arah pancaran pola radiasinya adalah azimuth/horizontal. Pancaran dari ULA adalah dengan rentang derajat -90 hingga 90, dimana pada plot rectangular, -90

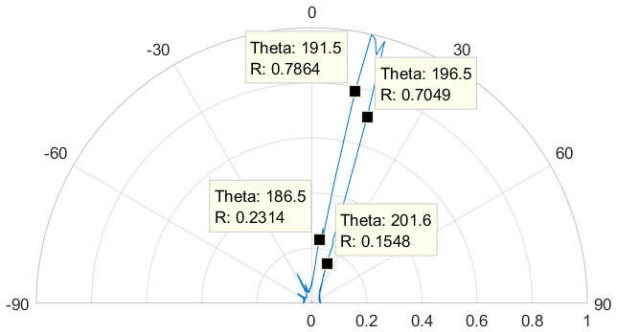
dituliskan dengan 90 derajat, 0 dituliskan dengan 180 derajat, dan 90 dituliskan dengan 270 derajat. Maka rekomendasi sudut yang disarankan adalah -64 derajat yang terletak pada 116,3 derajat pada plot polar dan rectangular, kemudian 51 derajat yang terletak pada 231,6 derajat pada plot polar dan rectangular, selanjutnya 61 derajat yang terletak pada 241,7 derajat pada plot polar dan rectangular, dan yang terakhir adalah 79 derajat yang terletak pada 258,7 pada plot polar dan rectangular.

4.4.4.2 Pola radiasi pada kanal URLOS 5 derajat

Berikut adalah plot pola radiasi pada kanal URLOS 5 derajat:

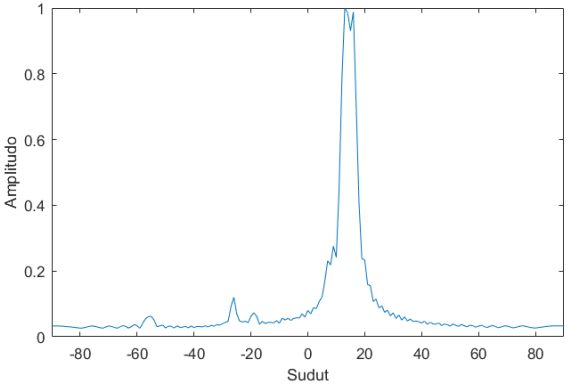


Gambar 4.29 Plot polar kanal URLOS 5 derajat

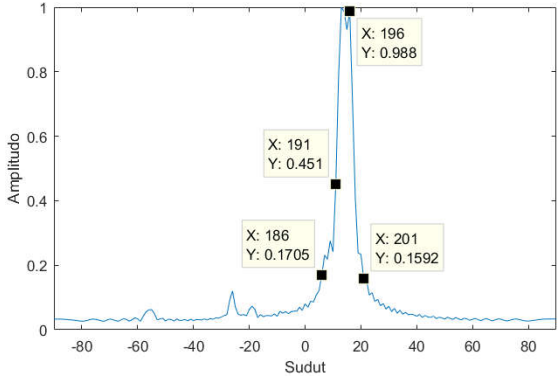


Gambar 4.30 Plot polar kanal URLOS 5 derajat dengan plot sudut 6, 11, 16, dan 21

Pada kanal URLOS 5 derajat, sudut yang digunakan dalam simulasi adalah 6 derajat yang terletak pada 186 derajat, 11 derajat yang terletak pada 191 derajat, 16 derajat yang terletak pada 196 derajat, dan 21 derajat yang terletak pada 201 derajat.



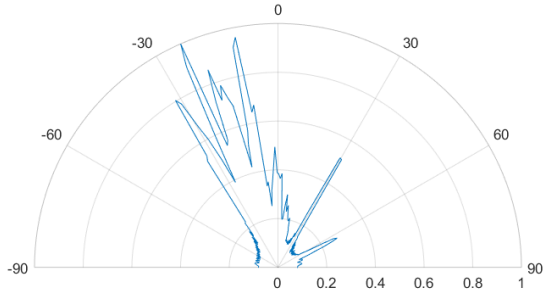
Gambar 4.31 Plot rectangular kanal URLOS 5 derajat



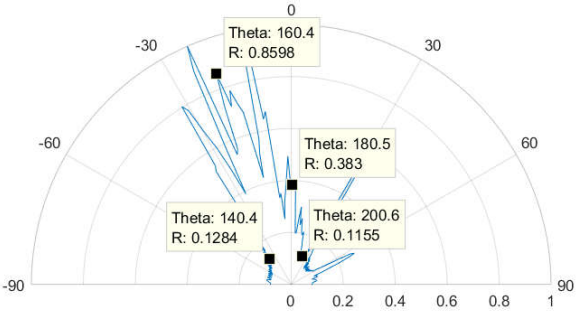
Gambar 4.32 Plot rectangular kanal URLOS 5 derajat dengan plot sudut 6, 11, 16, dan 21

4.4.4.3 Pola radiasi pada kanal URLOS 20 derajat

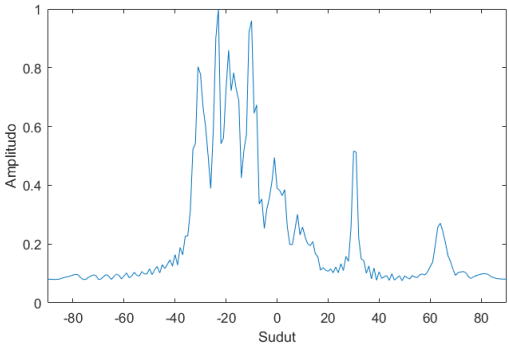
Pada kanal URLOS 20 derajat, sudut yang digunakan dalam simulasi adalah -40 derajat yang terletak pada 140 derajat, -20 derajat yang terletak pada 160 derajat, 0 derajat yang terletak pada 180 derajat, dan 20 derajat yang terletak pada 220 derajat.



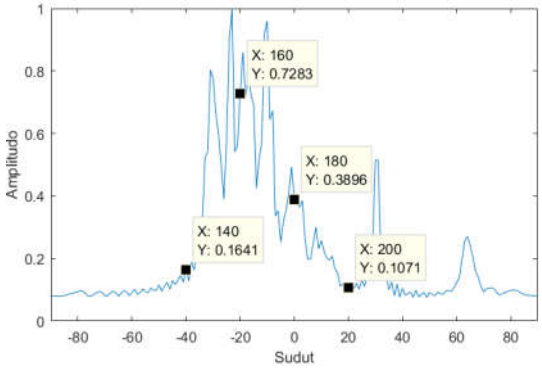
Gambar 4.33 Plot rectangular kanal URLOS 20 derajat



Gambar 4.34 Plot polar kanal URLOS 20 derajat dengan plot sudut -40, -20, 0, 20



Gambar 4.35 Plot rectangular kanal URLOS 20 derajat



Gambar 4.36 Plot rectangular kanal URLOS 20 derajat dengan plot sudut

4.5 Diskusi

Pada perbandingan kinerja antara Hybrid dan Full Digital Beamforming pada kanal IRF, kinerja dari metode HB dapat menyerupai kinerja dari metode FDB dengan syarat memiliki jumlah RF chain setengah dari jumlah RF chain yang dimiliki oleh FDB. Dibuktikan dengan rata-rata kinerja SE untuk SNR tinggi (20-30 dB), metode HB memiliki rata-rata kinerja SE sebesar 31,94 bits/s/Hz sedangkan metode

FDB memiliki rata-rata kinerja SE sebesar 32,9367 bits/s/Hz. Dari hasil analisis BER juga diketahui bahwa metode FDB dapat mencapai kinerja BER yang paling baik (paling rendah) tepatnya pada 0,00025 pada saat SNR = 14 dB, sedangkan metode HB pada saat SNR = 14 dB memiliki BER sebesar 0,084 dan baru dapat mencapai BER yang paling baik tepatnya 0,00025 saat SNR = 18 dB.

Pada perbandingan kinerja antara Hybrid dan Full Digital Beamforming pada kanal URLOS, kinerja dari metode HB dapat menyerupai kinerja dari metode FDB dengan syarat memiliki jumlah RF chain setengah dari jumlah RF chain yang dimiliki oleh FDB. Dibuktikan dengan rata-rata kinerja SE untuk SNR tinggi (20-30 dB), metode HB memiliki rata-rata kinerja SE sebesar 24,0067 bits/s/Hz sedangkan metode FDB memiliki rata-rata kinerja SE sebesar 22,4667 bits/s/Hz. Dari hasil analisis BER juga diketahui bahwa metode FDB dapat mencapai kinerja BER yang paling baik (paling rendah) tepatnya pada 0,00025 pada saat SNR = 22 dB, sedangkan metode HB pada saat SNR = 22 dB memiliki BER sebesar 0,2346 dan baru dapat mencapai BER yang paling baik tepatnya 0,00025 saat SNR = 26 dB.

Pada perbandingan kinerja metode HB pada kanal IRF dan URLOS, metode HB pada kanal IRF memiliki rata-rata kinerja SE lebih baik dibandingkan dengan metode HB pada kanal URLOS. Dibuktikan dengan rata-rata kinerja SE untuk SNR dari 0 dB hingga 30 dB, metode HB pada kanal IRF memiliki rata-rata kinerja SE sebesar 17,4758 bits/s/Hz sedangkan metode HB pada kanal URLOS memiliki rata-rata kinerja SE sebesar 12,3334 bits/s/Hz. Dari hasil analisis BER juga diketahui bahwa metode HB pada kanal IRF dapat mencapai kinerja BER yang paling baik (paling rendah) tepatnya pada 0,00025 pada saat SNR = 18 dB, sedangkan metode HB pada kanal URLOS pada saat SNR = 18 dB memiliki BER yang masih tinggi yakni sebesar 0,2766 dan baru dapat mencapai BER yang paling baik tepatnya 0,00025 saat SNR = 26 dB.

Pada perbandingan kinerja metode FDB pada kanal IRF dan URLOS, metode FDB pada kanal IRF memiliki rata-rata kinerja SE lebih baik dibandingkan dengan metode FDB pada kanal URLOS. Dibuktikan dengan rata-rata kinerja SE untuk SNR dari 0 dB hingga 30 dB, metode FDB pada kanal IRF memiliki rata-rata kinerja SE sebesar 19,0105 bits/s/Hz sedangkan metode FDB pada kanal URLOS memiliki rata-rata kinerja SE sebesar 12,7401 bits/s/Hz. Dari hasil analisis BER juga diketahui bahwa metode FDB pada kanal IRF dapat mencapai kinerja BER yang paling baik (paling rendah) tepatnya pada 0,00025 pada saat

SNR = 14 dB, sedangkan metode FDB pada kanal URLOS pada saat SNR = 14 dB memiliki BER yang masih tinggi yakni sebesar 0,2074 dan baru dapat mencapai BER yang paling baik tepatnya 0,00025 saat SNR = 22 dB.

Pada perbandingan jumlah RF chain pada kanal IRF dimana metode HB adalah sistem dengan jumlah RF chain = 1 (NRF1), metode FDB adalah sistem dengan jumlah RF chain = 2 (NRF2), dibandingkan dengan jumlah RF chain = 3 (NRF3), dan jumlah RF chain = 4 (NRF4), menunjukkan bahwa penambahan jumlah RF chain tidak menambah performa dibandingkan dengan pada saat jumlah NRF2 (metode FDB), akan tetapi meningkatkan kinerja SE dan BER jika dibandingkan dengan NRF1 (metode HB). Hal tersebut dibuktikan dengan rata-rata kinerja SE untuk SNR tinggi (20-30 dB), NRF2 memiliki rata-rata kinerja SE yang paling tinggi yaitu sebesar 19,0105 bits/s/Hz, kemudian NRF3 memiliki rata-rata kinerja SE dibawahnya yaitu sebesar 18,787 bits/s/Hz, kemudian NRF4 memiliki rata-rata kinerja SE sebesar 18,1738 bits/s/Hz, dan yang paling rendah adalah NRF1 dengan rata-rata kinerja SE sebesar 17,4758 bits/s/Hz. Dari pengamatan gambar dan tabel BER, dapat dilihat bahwa metode FDB dengan NRF2 (2 RF chain) memiliki angka BER yang baik, diikuti oleh NRF3, kemudian NRF4 dan NRF1. Hal tersebut dapat dilihat pada saat SNR = 14 dB, NRF2 dan NRF3 lebih dulu mencapai titik terendah yaitu BER sebesar 0,00025, sedangkan pada SNR = 14 dB, NRF4 dan NRF1 memiliki BER sebesar 0,01411 dan 0,084. NRF4 dan NRF1 baru dapat mencapai BER yang paling rendah saat SNR = 18 dB.

Pada perbandingan jumlah RF chain pada kanal URLOS dimana metode HB adalah sistem dengan jumlah RF chain = 1 (NRF1), metode FDB adalah sistem dengan jumlah RF chain = 2 (NRF2), dibandingkan dengan jumlah RF chain = 3 (NRF3), dan jumlah RF chain = 4 (NRF4), menunjukkan bahwa penambahan jumlah RF chain membuat rata-rata kinerja SE meningkat jika dibandingkan dengan NRF2(metode FDB) dan NRF1 (metode HB). Hal tersebut dibuktikan dengan rata-rata kinerja SE untuk SNR tinggi (20-30 dB), rata-rata kinerja SE untuk NRF3 adalah sebesar 15,8571 bits/s/Hz, kemudian rata-rata kinerja SE untuk NRF4 adalah sebesar 14,0866 bits/s/Hz, lalu rata-rata kinerja SE untuk NRF2 adalah sebesar 12,7401 bits/s/Hz, dan rata-rata kinerja SE untuk NRF1 adalah sebesar 12,3334 bits/s/Hz. Dari pengamatan pada gambar dan tabel BER dapat dilihat bahwa metode FDB (NRF2) dan NRF3 pada saat SNR = 22 dB mencapai angka BER yang paling kecil yaitu 0,00025, sedangkan

pada saat SNR = 22 dB, NRF1 dan NRF4 masih memiliki BER yang tinggi di angka 0,2346 dan 0,03958.

Pada perbandingan metode HB pada kanal IRF dengan menggunakan jumlah pemancar yang berbeda dimana jumlah antena pemancar yang digunakan adalah 32 (Tx32), 64 (Tx64), dan 81 (Tx81), membuktikan bahwa dari penambahan jumlah antena membuat rata-rata kinerja SE untuk antena dengan jumlah pemancar yang lebih banyak meningkat jika dibandingkan dengan jumlah pemancar yang sedikit. Dibuktikan dengan rata-rata kinerja SE untuk SNR dari 0 dB hingga 30 dB, rata-rata kinerja SE yang diperoleh untuk Tx81 adalah 17,5751 bits/s/Hz, kemudian untuk Tx64 adalah 17,4758 bits/s/Hz, dan untuk Tx32 adalah 38,73 bits/s/Hz. Sedangkan saat mengamati gambar dan tabel BER dapat dilihat bahwa Tx64 adalah jumlah antena pemancar yang memiliki BER yang paling baik. Dapat dilihat pada SNR = 18 dB, BER dari Tx64 telah mencapai angka yang paling kecil yaitu 0,00025, begitu juga dengan Tx32. Sedangkan pada saat SNR = 18 dB, BER dari Tx81 memiliki angka sebesar 0,05514 dan baru mencapai BER yang paling rendah pada saat SNR = 22 dB. Meskipun Tx32 dan Tx64 pada saat SNR = 18 dB mampu mencapai BER paling kecil secara bersamaan, tetapi dari pengamatan grafik pada gambar dan pada tabel dapat dilihat bahwa untuk rentang SNR 0 dB hingga 30 dB, Tx64 selalu memiliki nilai BER yang lebih kecil.

Sedangkan pada perbandingan metode HB pada kanal URLOS dengan menggunakan jumlah pemancar yang berbeda dimana jumlah antena pemancar yang digunakan adalah 32 (Tx32), 64 (Tx64), dan 81 (Tx81), membuktikan bahwa dari penambahan jumlah antena pemancar tidak menambah rata-rata kinerja SE dari metode HB pada kanal URLOS. Dibuktikan dengan rata-rata kinerja SE untuk SNR dari 0 dB hingga 30 dB, rata-rata kinerja SE yang diperoleh untuk Tx64 adalah 12,3334 bits/s/Hz, kemudian untuk Tx32 adalah 12,0606 bits/s/Hz, dan untuk Tx81 adalah 11,271 bits/s/Hz. Pada gambar dan tabel BER dapat dilihat bahwa hanya Tx64 yang mampu mencapai kondisi BER yang sangat kecil. Dibuktikan dengan pada saat SNR = 26 dB, BER dari Tx64 mampu mencapai 0,00025, sedangkan untuk Tx32 masih tinggi dengan nilai BER sebesar 0,05564, juga Tx81 dengan BER sebesar 0,01508. Hingga SNR = 30 dB, Tx32 dan Tx81 belum mampu mencapai angka BER yang rendah.

Pada perbandingan metode HB pada kanal IRF dengan menggunakan perbedaan jumlah user/antena penerima 2, 4, dan 6, membuktikan bahwa dari penambahan jumlah user/antenna penerima memperoleh rata-rata

kinerja SE yang semakin meningkat. Dibuktikan dengan rata-rata kinerja SE untuk SNR dari 0 dB hingga 30 dB, rata-rata kinerja SE yang diperoleh untuk user 2 adalah 6,37875 bits/s/Hz, kemudian rata-rata kinerja SE untuk user 4 adalah 17,4758 bits/s/Hz, dan rata-rata kinerja SE untuk user 6 adalah 23,981 bits/s/Hz. Meskipun user 6 memiliki rata-rata kinerja SE yang paling tinggi, user 6 masih belum bisa mencapai angka BER yang terbaik. Dibuktikan dengan pada saat SNR = 18 dB, user 4 telah berhasil mencapai angka BER paling rendah yaitu 0,00025, sedangkan user 6 memiliki BER sebesar 0,2044, dan user 2 memiliki BER sebesar 0,1378. User 6 baru mencapai BER yang paling rendah saat SNR = 26 dB, sedangkan user 2 baru mencapai BER yang paling rendah saat 30 dB. Dari sini dapat diketahui bahwa meskipun user 6 memiliki rata-rata kinerja SE yang paling baik, tetapi user 6 memiliki BER yang kurang baik.

Pada perbandingan metode HB pada kanal URLOS dengan menggunakan perbedaan jumlah user/antena penerima 2, 4, dan 6, membuktikan bahwa dari penambahan jumlah user/antenna penerima memperoleh rata-rata kinerja SE yang semakin meningkat. Dibuktikan dengan rata-rata kinerja SE untuk SNR dari 0 dB hingga 30 dB, rata-rata kinerja SE yang diperoleh untuk user 2 adalah 2,82678 bits/s/Hz, kemudian rata-rata kinerja SE untuk user 4 adalah 12,3334 bits/s/Hz, dan rata-rata kinerja SE untuk user 6 adalah 19,8012 bits/s/Hz. Dari pengamatan terhadap gambar dan tabel BER dapat diamati bahwa user 2 memiliki pencapaian BER yang paling buruk mulai dari 0 dB hingga 30 dB. User 6 walaupun memiliki rata-rata kinerja SE yang paling tinggi, akan tetapi memiliki pencapaian BER yang buruk, dibuktikan dengan BER yang dicapai baru menunjukkan angka yang sangat rendah saat SNR benar-benar mendekati 30 dB. Sedangkan user 4 memiliki pencapaian BER terbaik, dibuktikan dengan pada saat SNR = 26, user 4 telah berhasil mencapai BER yang paling rendah yang mampu dicapai, yakni sebesar 0,00025.

Pada sub bab tambahan telah dijelaskan mengenai pengaruh korelasi kanal. Dimulai dari penjabaran data korelasi kanal pada kanal IRF, URLOS 5 derajat, dan URLOS 20 derajat. Diketahui bahwa pada kanal URLOS 5 derajat memiliki korelasi kanal yang lebih tinggi jika dibandingkan dengan URLOS 20 derajat. Kemudian untuk pengecekan efek dari terjadinya korelasi kanal maka dilakukan pengujian terhadap kinerja SE dan BER pada metode HB dengan membandingkan pada kanal IRF, URLOS 5 derajat, dan URLOS 20 derajat. Diketahui dari hasil kinerja SE dan BER bahwa pada kanal URLOS 5 derajat dimana terjadi

korelasi kanal yang cukup besar, mengalami penurunan kinerja SE dan BER yang telah dibuktikan dengan data gambar dan tabel. Kemudian dilakukan juga pengujian kinerja SE dan BER selanjutnya dengan melakukan perbandingan terhadap jumlah antena pemancar yang berbeda pada kanal URLOS, didapatkan bahwa, untuk kanal URLOS 20 derajat pada semua jumlah antena pemancarnya (32, 64, dan 81) mengalami pengingkatan kinerja SE dan perbaikan kinerja BER yang telah dibuktikan dengan gambar dan tabel. Dari dua percobaan yang telah dilakukan dapat diketahui bahwa kanal yang memiliki korelasi kanal yang besar akan mengalami penurunan kinerja SE dan BER.

Lalu, dilakukan plot pola radiasi baik polar maupun rectangular agar dapat mengetahui arah dari pola radiasi antena yang disimulasikan. Pada kanal IRF, direkomendasikan agar antena penerima (antena Rx), berada pada daerah yang memiliki pola radiasi dengan amplitudo yang cukup tinggi yaitu pada -64 derajat, 51 derajat, 61 derajat, dan 79 derajat, karena pada saat simulasi mengalami kesulitan dalam melakukan plotting, maka data yang seharusnya digambarkan dengan rentang -90 hingga 90 derajat, akhirnya digambarkan dengan 90 hingga 270 derajat, dimana -64 derajat terletak pada 116,3 derajat, 51 derajat terletak pada 231,6 derajat, 61 derajat diletakkan pada 241,7 derajat, dan 79 derajat diletakkan pada 258,7 derajat. Kemudian pada kanal URLOS 5 derajat digunakan sudut 6, 11, 16, dan 21 derajat, dimana 6 derajat terletak pada 186 derajat, 11 derajat terletak pada 191 derajat, 16 derajat terletak pada 196 derajat, dan 21 derajat terletak pada 201 derajat. Kemudian pada kanal URLOS 20 derajat, digunakan sudut -40 derajat yang terletak pada 140 derajat, -20 derajat yang terletak pada 160 derajat, 0 derajat yang terletak pada 180 derajat, dan 20 derajat terletak pada 200 derajat. Dapat dilihat pada gambar plot polar kanal URLOS 5 derajat dimana semua user berada pada posisi main lobe, sedangkan pada kanal URLOS 20 derajat dimana pada -20 dan 0 derajat mendapat main lobe, sementara -40 dan 20 mendapat minor lobe. Analisa yang dapat diberikan adalah terjadi sedikit error terhadap akurasi precoding, sehingga diprediksi bahwa terjadi pergeseran arah pola radiasi terhadap lokasi antena pemancar. Kemudian, meskipun kanal URLOS 20 derajat mendapat pola radiasi yang kurang bagus jika dibanding dengan kanal URLOS 5 derajat, namun kinerja dari SE dan BER dari kanal URLOS 20 derajat berada diatas URLOS 5 derajat, hal tersebut dikarenakan terjadinya korelasi kanal yang menyebabkan penurunan kinerja dari kanal URLOS 5 derajat.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisa dan pembahasan terhadap simulasi kinerja SE dan BER sistem Multiuser MISO pada kanal IRF dan URLOS pada bab 4, maka berikut adalah beberapa hal yang bisa disimpulkan:

1. Sistem MISO adalah akronim dari *Multi Input Single Output*, dimana pada sisi pemancar memiliki jumlah antena lebih dari satu buah sedangkan pada sisi penerima hanya memiliki satu buah antena penerima.
2. Pada kanal IRF dan URLOS, kinerja dari metode HB dapat menyerupai kinerja dari metode FDB dengan syarat memiliki jumlah RF chain setengah dari jumlah RF chain yang dimiliki oleh FDB.
3. Metode HB dan metode FDB pada kanal IRF memiliki rata-rata kinerja SE lebih baik dibandingkan dengan metode HB dan metode FDB pada kanal URLOS.
4. Pada perbandingan jumlah RF chain pada kanal IRF, menunjukkan bahwa penambahan jumlah RF chain tidak menambah performa dibandingkan dengan pada saat jumlah NRF2 (metode FDB), akan tetapi meningkatkan kinerja SE dan BER jika dibandingkan dengan NRF1 (metode HB). Sedangkan pada kanal URLOS, menunjukkan bahwa penambahan jumlah RF chain membuat rata-rata kinerja SE meningkat jika dibandingkan dengan NRF2 (metode FDB) dan NRF1 (metode HB).
5. Pada perbandingan metode HB pada kanal IRF dengan menggunakan perbedaan jumlah pemancar, membuktikan bahwa dari penambahan jumlah antena membuat rata-rata kinerja SE untuk antena dengan jumlah pemancar yang lebih banyak meningkat dibandingkan dengan jumlah pemancar yang sedikit. Sedangkan pada kanal URLOS, membuktikan bahwa penambahan jumlah antena pemancar tidak menambah rata-rata kinerja SE dari metode HB.
6. Pada perbandingan metode HB pada kanal IRF dan URLOS dengan menggunakan perbedaan jumlah user/antena penerima 2, 4, dan 6, membuktikan bahwa dari penambahan jumlah user/antenna penerima memperoleh rata-rata kinerja SE yang semakin meningkat.

7. Pengaruh perbedaan sudut yang tidak cukup jauh menyebabkan terjadinya korelasi pada kanal URLOS yang akan membuat kinerja SE semakin menurun.

5.2 Saran

Setelah melakukan pemodelan, analisa, dan penarikan kesimpulan, penulis memberikan saran untuk pengembangan penelitian pada topik Multiuser MISO sebagai berikut:

1. Untuk mendapatkan grafik SE yang semakin bagus sebaiknya dapat memperbanyak pembangkitan iterasi kanal tiap rentang SNRnya.
2. Perlu ada penelitian lebih lanjut mengenai jenis kanal lainnya pada sistem Multiuser MISO dengan metode *hybrid beamforming*.
3. Menentukan perbedaan sudut antara antena penerima (Rx) yang cukup jauh agar tidak terjadi korelasi kanal.

Daftar Pustaka

1 C. A. Balanis, Antenna Theory: Analysis and Design, Hoboken, New Jersey: John Wiley & Sons, Inc., 2005.

2 I. T. Union, "IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond," Electronic Publication, Geneva, 2015.

3 I. T. Union, "IMT traffic estimates for the years 2020 to 2030," Electronic Publication, Geneva, 2015.

4 F. Sohrabi and W. Yu, "Hybrid Digital and Analog Beamforming Design for Large-Scale Antenna Arrays," *IEEE Journal of Selected Topics in Signal Processing,* vol. 10(3), pp. 510-513, 2016.

5 S. R. Saunders and A. A. Zavala, Antennas and Propagation for Wireless Communication Systems, Chichester, England: John Wiley & Sons Ltd, 2007.

6 W. Roh, J.-Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho and K. Cheun, "Millimeter-Wave Beamforming as an Enabling Technology for 5G Cellular Communications: Theoretical Feasibility and Prototype Results," *IEEE Communications Magazine,* vol. 52(2), pp. 106-113, 2014.

7 T. L. Marzetta, E. G. Larsson, H. Yang and H. Q. Ngo, Fundamentals of Massive MIMO, Cambridge, United Kingdom: Cambridge University Press, 2016.

8 T. Kim, J. Park, J.-Y. Seol, S. Jeong, J. Cho and W. Roh, "Tens of Gbps Support with mmWave Beamforming Systems for Next Generation Communications," in *IEEE Global Communications Conference (GLOBECOM)*, Korea, 2013.

9 M. C. Jeruchim, P. Balaban and K. S. Shanmugan, Simulation of Communication Systems: Modelling, Methodology, and Techniques, New York: Kluwer Academic Publishers, 2002.

10 N. C. Giri, A. Sahoo, J. R. Swain, P. Kumar, A. Nayak and P. Debogoswami, "Capacity & Performance Comparison of

SISO and MIMO System for Next Generation Network (NGN)," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* vol. 3, no. 9, 2014.

11 F. C. Commission, "Millimeter Wave Propagation: Spectrum Management Implications," *Office of Engineering and Technology,* vol. 70, pp. 1-7, 1997.

12 E. Ali, M. Ismail, R. Nordin and N. F. Abdulah, "Beamforming techniques for massive MIMO systems in 5G: overview, classification, and trends for future research.," Frontiers of Information Technology & Electronic Engineering, vol. 18, no. 6, pp. 753-772, 2017.

LAMPIRAN

LISTING PROGRAM

```
%%%
```

```
    Pp                = eye(k);        %step 2
    beta_kp           = ones([1, 1]);

    v_rf         = randi([1], N_tx, N_rf); %step
1
    v_rf1        = randi([1], N_tx, N_rf);
    v_rf2        = randi([1], N_tx, N_rf);
    vrf_opt      = randi([1], N_tx, N_rf);
    argmin       = randi([1], N_tx, N_rf)*5;
    selisih      = 1;
    temp_topt    = randi([1], N_tx, N_rf)*20;
    teta_opt     = ones(N_tx, N_rf);
    teta_opt2    = ones(N_tx, N_rf);
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp     = 0;
            temp2    = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp             = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
```

```
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
                    end
                    end
                    temporary1(ii,J)    = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)    = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3   = 0;
                    temp4   = 0;
                    for L = 1:N_rf
                    if L~= ii
                       v_rf2(L,J)       =
v_rf(L,J);
                       temp3            =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4            =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)      =  temp3;
                    n_ijD(ii,J)      =  temp4;
                    c_ij(ii,J)       =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)       =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
```

```
                    teta_ij2(ii,J)     = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    vrf_ij1(ii,J)      = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)      = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)      =
argmin(ii,J);
                    argmin(ii,J)       =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)      = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
```

```
                break
                end

            if isnan(A)
            variancekanal = 1;
            h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km  = h_km;
            g_kmt = transpose(g_km);
            Pp              = eye(k);
%step 2
            beta_kp         = ones([1, 1]);
            v_rf        = randi([1], N_tx,
N_rf); %step 1
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
```

```
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
                    end
                end
                Pmiring           = Pmiring +
PK2;
                pk                = 1/qkk * PK2;
                Pp(k2,k2)         = pk;
                PPP               = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)            = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)   = sum(Rk);
    SE_IRF1           = real(sum1);
%%%
```

```
        sigbpsk(:,stream)     =
2*genbpsk(:,stream) -1;

        y(:,stream)                    = g_kmt *
vrf_opt * VD * sigbpsk(:,stream);
        detektor                       =
real(y)>0;
        for sinyal2= 1:Ns
            if detektor(sinyal2,stream) ==1
                sinyal_rx(sinyal2, stream) = 1;
            else sinyal_rx(sinyal2,stream) = -1;
            end
        end
    end
    nErr(n,SNRsize) = size(find([detektor-
genbpsk]),1);
    nErr(nErr==0) = 1;
    end

    for n1 = 1:N
    variancekanal1 = 1;
    h_km1  =
sqrt(variancekanal1/2)*(randn([m],[k])+((1i)*ran
dn([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
    g_km1  = h_km1;
    g_kmt1 = transpose(g_km1);
    Pp1               = eye(k);        %step 2
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx, N_rf2);
%step 1
    v_rf11        = randi([1], N_tx, N_rf2);
    v_rf21        = randi([1], N_tx, N_rf2);
    vrf_opt1      = randi([1], N_tx, N_rf2);
    argmin1       = randi([1], N_tx, N_rf2)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf2)*20;
    teta_opt1     = ones(N_tx, N_rf2);
    teta_opt21    = ones(N_tx, N_rf2);
```

```
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf2%step 1
            Hbar1          = Pp1^(-1/2) * g_kmt1;
            v_rfj1         = v_rf1;
            v_rfj1(:,J)    = [];
            A1             = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21            = inv(A1) * inv(A1);
            A11            = inv(A1);
            B1             = Hbar1' * A21 *
Hbar1;
            D1             = Hbar1' * A11 *
Hbar1;
            B21            = Hbar1' * A21 *
Hbar1;
            D21            = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1    = 0;
            temp21   = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
```

```
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
                    for L1 = 1:N_rf2
                    if L1~= ii1
                       v_rf21(L1,J1)     =
v_rf1(L1,J1);
                       temp31            =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                       temp41            =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)        =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)        =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
```

```
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    vrf_ij11(ii1,J1)     = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)     = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)      = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)      = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)     =
argmin1(ii1,J1);
                    argmin1(ii1,J1)      =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)     = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
```

```
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            variancekanal1 = 1;
            h_km1  =
sqrt(variancekanal1/2)*(randn([m],[k])+((1i)*ran
dn([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km1  = h_km1;
            g_kmt1 = transpose(g_km1);
            Pp1              = eye(k);
%step 2
            beta_kp1         = ones([1, 1]);
            v_rf1        = randi([1], N_tx,
N_rf2); %step 1
            end

            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
```

```
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1            = Pmiring1 +
PK21;
                pk1                 = 1/qkk1 *
PK21;
                Pp1(k21,k21)        = pk1;
                PPP1                = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)            = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
        cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_IRF1irf2         = real(sum11);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    jumlah_signal1 = 1000;
```

```
    for stream1 = 1:1:jumlah_signal1
        jumlah_data1            = Ns;
        genbpsk1(:,stream1)     =
rand(jumlah_data1, 1)>0.5;
        sigbpsk1(:,stream1)     =
2*genbpsk1(:,stream1) -1;
        y1(:,stream1)           = g_kmt1 *
vrf_opt1 * VD1 * sigbpsk1(:,stream1);
        detektor1               = real(y1)>0;
        for sinyal21= 1:Ns
            if detektor1(sinyal21,stream1) ==1
                sinyal_rx1(sinyal21, stream1) =
1;
            else sinyal_rx1(sinyal21,stream1) =
-1;
            end
        end
    end
    nErr1(n,SNRsize) = size(find([detektor1-
genbpsk1]),1);
    nErr1(nErr1==0) = 1;
    end
end

simBER     = nErr/(jumlah_signal*Ns);
simBERirf2 = nErr1/(jumlah_signal1*Ns);

figure(1)
plot(SNR_dB, mean(SE_IRF1,2), 'bx-'); hold on;
plot(SNR_dB, mean(SE_IRF1irf2,2), 'rx-'); hold
on;
%plot(SNR_dB, mean(SE_IRF1irf3,2), 'b.-'); hold
on;
%plot(SNR_dB, mean(SE_IRF1irf4,2), 'r.-'); hold
on;
legend('Hybrid Beamforming','Full Digital
Beamforming'); %,'NRF3','NRF4');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR kanal IRF');
```

```
figure(2)
semilogy(SNR_dB,mean(simBER,1),'bx-'); hold on;
semilogy(SNR_dB,mean(simBERirf2,1),'rx-'); hold
on;
%semilogy(SNR_dB,mean(simBERirf3,1),'bx-'); hold
on;
%semilogy(SNR_dB,mean(simBERirf4,1),'rx-'); hold
on;
axis([-10 30 10^-5 0.5])
grid on
legend('Hybrid Beamforming','Full Digital
Beamforming');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation kanal IRF');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATLAB SCRIPT PERBANDINGAN HB VS FDB URLOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;
clc;

m = 64; k = 4; %JUMLAH ANTENA
P               = 1; %Daya dalam Watt
s               = 1; %menurut analisa, WAJIB
[Jml Ns == Jml m];
k_user          = k;
Ns              = s*k_user;
N_tx            = m;
N_rf            = Ns; %untuk syarat Hybrid
Beamforming
N_rf2           = 2*Ns; %untuk syarat Full
Digital Beamforming

SNR_dB  = -2:4:30; %range SNR in dB
SNR_lin = 10.^(SNR_dB/10); % range SNR in linear
power
N =9;

for SNRsize = 1 : length(SNR_dB)
    SNR = SNR_lin(SNRsize);
    variance = P/SNR;
    for n = 1:N
    g_km = [];
    thet    = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12
    g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g2      = 2*g-1; %pembangkitan bpsk
    thett   = (g2.*thet);
    %theta   = asind(thett);

    beta_k  = 1;
```

```
    psi          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it           = randi([0 1],k,1)>=0.5;
    itb          = (it*2)-1;
    psii         = (itb.*psi);

    for kurlos       = 1:1:k
    for murlos       = 1:1:m
        theta_k(murlos,kurlos)  = thett +
(5*(kurlos-1)); %selisih 0.5 derajat antar
antena
        ekurung(murlos,kurlos)  = ((1*exp((-
1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
    end
    ekurungt                    =
transpose(ekurung);
    g_km2(:,kurlos)            = (sqrt(beta_k)
.* exp(1i*psii(kurlos))) * ekurungt(kurlos,:);
    end
    g_kmt       = transpose(g_km2);


    Pp               = eye(k);       %step 2
    beta_kp          = ones([1, 1]);
    v_rf         = randi([1], N_tx, N_rf); %step
1
    v_rf1        = randi([1], N_tx, N_rf);
    v_rf2        = randi([1], N_tx, N_rf);
    vrf_opt      = randi([1], N_tx, N_rf);
    argmin       = randi([1], N_tx, N_rf)*5;
    selisih      = 1;
    temp_topt    = randi([1], N_tx, N_rf)*20;
    teta_opt     = ones(N_tx, N_rf);
    teta_opt2    = ones(N_tx, N_rf);
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
```

```
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp     = 0;
            temp2    = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp             = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2            = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
                    end
                    end
                    temporary1(ii,J)     = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)     = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)     = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3    = 0;
                    temp4    = 0;
                    for L = 1:N_rf
                    if L~= ii
                       v_rf2(L,J)        =
v_rf(L,J);
```

```
                        temp3              =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                        temp4              =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)       =  temp3;
                    n_ijD(ii,J)       =  temp4;
                    c_ij(ii,J)         =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)         =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    vrf_ij1(ii,J)     = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)     = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
```

```
                    argmin2(ii,J)     =
argmin(ii,J);
                    argmin(ii,J)      =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)     = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end

            if isnan(A)
            g_km = [];
            thet    = 6 + (0.5 - (-
6).*rand(1,1)); %pembangkitan theta dari 6
sampai 12 derajat
            g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g2      = 2*g-1; %pembangkitan bpsk
            thett   = (g2.*thet);


            beta_k  = 1;
```

```
            psi         = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it          = randi([0 1],k,1)>=0.5;
            itb         = (it*2)-1;
            psii        = (itb.*psi);

            for kurlos      = 1:1:k
            for murlos      = 1:1:m
                theta_k(murlos,kurlos)  = thett
+ (5*(kurlos-1)); %selisih 0.5 derajat antar
antena
                ekurung(murlos,kurlos)  =
((1*exp((-1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
            end
            ekurungt                    =
transpose(ekurung);
            g_km2(:,kurlos)             =
(sqrt(beta_k) .* exp(1i*psii(kurlos))) *
ekurungt(kurlos,:);
            end
            g_kmt       = transpose(g_km2);
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
```

```
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_URLOS            = real(sum1);
```

```
%%%
```

```
    it1           = randi([0 1],k,1)>=0.5;
    itb1          = (it1*2)-1;
    psii1         = (itb1.*psi1);

    for kurlos1       = 1:1:k
    for murlos1       = 1:1:m
        theta_k1(murlos1,kurlos1)  = thett1 +
(5*(kurlos1-1)); %selisih 0.5 derajat antar
antena
        ekurung1(murlos1,kurlos1)  = ((1*exp((-
1i)*(murlos1-
1)*pi*sind(theta_k1(murlos1,kurlos1)))));
    end
    ekurungt1                      =
transpose(ekurung1);
    g_km21(:,kurlos1)              =
(sqrt(beta_k1) .* exp(1i*psii1(kurlos1))) *
ekurungt1(kurlos1,:);
    end
    g_kmt1       = transpose(g_km21);


    Pp1               = eye(k);        %step 2
    beta_kp1          = ones([1, 1]);
    v_rf1         = randi([1], N_tx, N_rf2);
%step 1
    v_rf11        = randi([1], N_tx, N_rf2);
    v_rf21        = randi([1], N_tx, N_rf2);
    vrf_opt1      = randi([1], N_tx, N_rf2);
    argmin1       = randi([1], N_tx, N_rf2)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf2)*20;
    teta_opt1     = ones(N_tx, N_rf2);
    teta_opt21    = ones(N_tx, N_rf2);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf2%step 1
            Hbar1          = Pp1^(-1/2) * g_kmt1;
```

```
            v_rfj1          = v_rf1;
            v_rfj1(:,J)     = [];
            A1              = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21             = inv(A1) * inv(A1);
            A11             = inv(A1);
            B1              = Hbar1' * A21 *
Hbar1;
            D1              = Hbar1' * A11 *
Hbar1;
            B21             = Hbar1' * A21 *
Hbar1;
            D21             = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1    = 0;
            temp21   = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
```

```
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
                    for L1 = 1:N_rf2
                    if L1~= ii1
                       v_rf21(L1,J1)     =
v_rf1(L1,J1);
                       temp31            =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                       temp41            =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)        =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)        =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    vrf_ij11(ii1,J1)     = exp(-
1i*angle(teta_ij11(ii1,J1)));
```

```
                    vrf_ij21(ii1,J1)      = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)      = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
                cnt1 = cnt1 + 1;
                break
```

```
                end

            if isnan(A1)
            g_km1 = [];
            thet1    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g1       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g21      = 2*g1-1; %pembangkitan
bpsk
            thett1   = (g21.*thet1);
            beta_k1  = 1;
            psi1          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it1           = randi([0
1],k,1)>=0.5;
            itb1          = (it1*2)-1;
            psii1         = (itb1.*psi1);

            for kurlos1       = 1:1:k
            for murlos1       = 1:1:m
            theta_k1(murlos1,kurlos1)  = thett1
+ (5*(kurlos1-1)); %selisih 0.5 derajat antar
antena
            ekurung1(murlos1,kurlos1)  =
((1*exp((-1i)*(murlos1-
1)*pi*sind(theta_k1(murlos1,kurlos1)))));
            end
            ekurungt1                      =
transpose(ekurung1);
            g_km21(:,kurlos1)              =
(sqrt(beta_k1) .* exp(1i*psii1(kurlos1))) *
ekurungt1(kurlos1,:);
            end
            g_kmt1       = transpose(g_km21);
            end

            cnt1 = cnt1+1;
```

```
        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1             = Pmiring1 +
PK21;
                pk1                  = 1/qkk1 *
PK21;
                Pp1(k21,k21)         = pk1;
                PPP1                 = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)             = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
```

```
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
       cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_URLOS2           = real(sum11);
%%%
```

```
simBERurlos      = nErr/(jumlah_signal*Ns);
simBERurlos2     = nErr1/(jumlah_signal1*Ns);

figure(1)
plot(SNR_dB, mean(SE_URLOS,2), 'bx-'); hold on;
plot(SNR_dB, mean(SE_URLOS2,2), 'rx-'); hold on;
%plot(SNR_dB, mean(SE_IRF1irf3,2), 'b.-'); hold
on;
%plot(SNR_dB, mean(SE_IRF1irf4,2), 'r.-'); hold
on;
legend('Hybrid Beamforming','Full Digital
Beamforming'); %,'NRF3','NRF4');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR kanal URLOS');

figure(2)
semilogy(SNR_dB,mean(simBERurlos,1),'bx-'); hold
on;
semilogy(SNR_dB,mean(simBERiurlos2,1),'rx-');
hold on;
%semilogy(SNR_dB,mean(simBERirf3,1),'bx-'); hold
on;
%semilogy(SNR_dB,mean(simBERirf4,1),'rx-'); hold
on;
axis([-10 30 10^-5 0.5])
grid on
legend('Hybrid Beamforming','Full Digital
Beamforming');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation kanal URLOS');
```

```
%%%
```

```
    v_rf        = randi([1], N_tx, N_rf); %step
1
    v_rf1       = randi([1], N_tx, N_rf);
    v_rf2       = randi([1], N_tx, N_rf);
    vrf_opt     = randi([1], N_tx, N_rf);
    argmin      = randi([1], N_tx, N_rf)*5;
    selisih     = 1;
    temp_topt   = randi([1], N_tx, N_rf)*20;
    teta_opt    = ones(N_tx, N_rf);
    teta_opt2   = ones(N_tx, N_rf);
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp    = 0;
            temp2   = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp            = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
```

```
                    end
                    end
                    temporary1(ii,J)    = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)    = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3    = 0;
                    temp4    = 0;
                    for L = 1:N_rf
                    if L~= ii
                       v_rf2(L,J)       =
v_rf(L,J);
                       temp3            =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4            =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)      =  temp3;
                    n_ijD(ii,J)      =  temp4;
                    c_ij(ii,J)       =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)       =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
```

```
                    vrf_ij1(ii,J)      = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)      = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)      =
argmin(ii,J);
                    argmin(ii,J)       =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)      = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end
```

```
            if isnan(A)
            variancekanal = 1;
            h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km  = h_km;
            g_kmt = transpose(g_km);
            Pp              = eye(k);
%step 2
            beta_kp         = ones([1, 1]);
            v_rf        = randi([1], N_tx,
N_rf); %step 1
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
```

```
                        PK2 = 0;
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
        cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_IRF1             = real(sum1);
%%%
```

```
        y(:,stream)                         = g_kmt *
vrf_opt * VD * sigbpsk(:,stream);
        detektor                            =
real(y)>0;
        for sinyal2= 1:Ns
            if detektor(sinyal2,stream) ==1
                sinyal_rx(sinyal2, stream) = 1;
            else sinyal_rx(sinyal2,stream) = -1;
            end
        end
    end
    nErr(n,SNRsize) = size(find([detektor-
genbpsk]),1);
    nErr(nErr==0) = 1;
    end

    for n1 = 1:N
    g_kmm = [];
    thet    = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12 derajat
    g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g2      = 2*g-1; %pembangkitan bpsk
    thett   = (g2.*thet);
    %theta   = asind(thett);

    beta_k  = 1;

    psi         = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it          = randi([0 1],k,1)>=0.5;
    itb         = (it*2)-1;
    psii        = (itb.*psi);

    for kurlos      = 1:1:k
    for murlos      = 1:1:m
        theta_k(murlos,kurlos)  = thett +
(5*(kurlos-1)); %selisih 0.5 derajat antar
antena
```

```
        ekurung(murlos,kurlos)  = ((1*exp((-
1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
    end
    ekurungt                    =
transpose(ekurung);
    g_km2(:,kurlos)             = (sqrt(beta_k)
.* exp(1i*psii(kurlos))) * ekurungt(kurlos,:);
    end
    g_kmt1      = transpose(g_km2);


    Pp1               = eye(k);        %step 2
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx, N_rf); %step
1
    v_rf11        = randi([1], N_tx, N_rf);
    v_rf21        = randi([1], N_tx, N_rf);
    vrf_opt1      = randi([1], N_tx, N_rf);
    argmin1       = randi([1], N_tx, N_rf)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf)*20;
    teta_opt1     = ones(N_tx, N_rf);
    teta_opt21    = ones(N_tx, N_rf);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf%step 1
            Hbar1          = Pp1^(-1/2) * g_kmt1;
            v_rfj1         = v_rf1;
            v_rfj1(:,J)    = [];
            A1             = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21            = inv(A1) * inv(A1);
            A11            = inv(A1);
            B1             = Hbar1' * A21 *
Hbar1;
```

```
            D1              = Hbar1' * A11 *
Hbar1;
            B21             = Hbar1' * A21 *
Hbar1;
            D21             = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1     = 0;
            temp21    = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)     =
temp1;
                    temporary21(ii1,J1)    =
temp21;
                    si_ijB1(ii1,J1)     =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)     =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31    = 0;
                    temp41    = 0;
                    for L1 = 1:N_rf
                    if L1~= ii1
                       v_rf21(L1,J1)      =
v_rf1(L1,J1);
```

```
                        temp31              =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                        temp41              =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)         =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)         =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    vrf_ij11(ii1,J1)     = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)     = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)      = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
```

```
                    f_vrf21(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)      = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            g_kmm = [];
            thet    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
```

```
            g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g2      = 2*g-1; %pembangkitan bpsk
            thett   = (g2.*thet);
            %theta   = asind(thett);

            beta_k  = 1;

            psi         = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it          = randi([0 1],k,1)>=0.5;
            itb         = (it*2)-1;
            psii        = (itb.*psi);

            for kurlos      = 1:1:k
            for murlos      = 1:1:m
                theta_k(murlos,kurlos)  = thett
+ (5*(kurlos-1)); %selisih 0.5 derajat antar
antena
                ekurung(murlos,kurlos)  =
((1*exp((-1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
            end
            ekurungt                    =
transpose(ekurung);
            g_km2(:,kurlos)             =
(sqrt(beta_k) .* exp(1i*psii(kurlos))) *
ekurungt(kurlos,:);
            end
            g_kmt1      = transpose(g_km2);
            end

            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
```

```
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1             = Pmiring1 +
PK21;
                pk1                  = 1/qkk1 *
PK21;
                Pp1(k21,k21)         = pk1;
                PPP1                 = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)             = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
```

```
            end
        cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_URLOS        = real(sum11);
%%%
```

```
plot(SNR_dB, mean(SE_URLOS,2), 'rx-'); hold on;
%plot(SNR_dB, mean(SE_IRF1irf3,2), 'b.-'); hold
on;
%plot(SNR_dB, mean(SE_IRF1irf4,2), 'r.-'); hold
on;
legend('IRF','URLOS');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR HB');

figure(2)
semilogy(SNR_dB,mean(simBER,1),'bx-'); hold on;
semilogy(SNR_dB,mean(simBERurlos,1),'rx-'); hold
on;
%semilogy(SNR_dB,mean(simBERirf3,1),'bx-'); hold
on;
%semilogy(SNR_dB,mean(simBERirf4,1),'rx-'); hold
on;
axis([-10 30 10^-5 0.5])
grid on
legend('IRF','URLOS');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation HB');
```

```
%%%
```

```
    v_rf        = randi([1], N_tx, N_rf); %step
1
    v_rf1       = randi([1], N_tx, N_rf);
    v_rf2       = randi([1], N_tx, N_rf);
    vrf_opt     = randi([1], N_tx, N_rf);
    argmin      = randi([1], N_tx, N_rf)*5;
    selisih     = 1;
    temp_topt   = randi([1], N_tx, N_rf)*20;
    teta_opt    = ones(N_tx, N_rf);
    teta_opt2   = ones(N_tx, N_rf);
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp    = 0;
            temp2   = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp            = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
```

```
                    end
                    end
                    temporary1(ii,J)    = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)    = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3   = 0;
                    temp4   = 0;
                    for L = 1:N_rf
                    if L~= ii
                       v_rf2(L,J)       =
v_rf(L,J);
                       temp3            =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4            =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)      =  temp3;
                    n_ijD(ii,J)      =  temp4;
                    c_ij(ii,J)       =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)       =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
```

```
                    vrf_ij1(ii,J)      = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)      = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)      =
argmin(ii,J);
                    argmin(ii,J)       =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)      = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end
```

```
            if isnan(A)
            variancekanal = 1;
            h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km  = h_km;
            g_kmt = transpose(g_km);
            Pp              = eye(k);
%step 2
            beta_kp         = ones([1, 1]);
            v_rf        = randi([1], N_tx,
N_rf); %step 1
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
```

```
                        PK2 = 0;
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
        cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_IRF1             = real(sum1);
%%%
```

```
        y(:,stream)                         = g_kmt *
vrf_opt * VD * sigbpsk(:,stream);
        detektor                            =
real(y)>0;
        for sinyal2= 1:Ns
            if detektor(sinyal2,stream) ==1
                sinyal_rx(sinyal2, stream) = 1;
            else sinyal_rx(sinyal2,stream) = -1;
            end
        end
    end
    nErr(n,SNRsize) = size(find([detektor-
genbpsk]),1);
    nErr(nErr==0) = 1;
    end

    for n1 = 1:N
    g_kmm = [];
    thet    = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12 derajat
    g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g2      = 2*g-1; %pembangkitan bpsk
    thett   = (g2.*thet);
    %theta   = asind(thett);

    beta_k  = 1;

    psi         = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it          = randi([0 1],k,1)>=0.5;
    itb         = (it*2)-1;
    psii        = (itb.*psi);

    for kurlos      = 1:1:k
    for murlos      = 1:1:m
        theta_k(murlos,kurlos)  = thett +
(5*(kurlos-1)); %selisih 0.5 derajat antar
antena
```

```
        ekurung(murlos,kurlos)  = ((1*exp((-
1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
    end
    ekurungt                    =
transpose(ekurung);
    g_km2(:,kurlos)             = (sqrt(beta_k)
.* exp(1i*psii(kurlos))) * ekurungt(kurlos,:);
    end
    g_kmt1      = transpose(g_km2);


    Pp1               = eye(k);        %step 2
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx, N_rf); %step
1
    v_rf11        = randi([1], N_tx, N_rf);
    v_rf21        = randi([1], N_tx, N_rf);
    vrf_opt1      = randi([1], N_tx, N_rf);
    argmin1       = randi([1], N_tx, N_rf)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf)*20;
    teta_opt1     = ones(N_tx, N_rf);
    teta_opt21    = ones(N_tx, N_rf);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf%step 1
            Hbar1          = Pp1^(-1/2) * g_kmt1;
            v_rfj1         = v_rf1;
            v_rfj1(:,J)    = [];
            A1             = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21            = inv(A1) * inv(A1);
            A11            = inv(A1);
            B1             = Hbar1' * A21 *
Hbar1;
```

```
            D1              = Hbar1' * A11 *
Hbar1;
            B21             = Hbar1' * A21 *
Hbar1;
            D21             = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1     = 0;
            temp21    = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)     =
temp1;
                    temporary21(ii1,J1)    =
temp21;
                    si_ijB1(ii1,J1)     =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)     =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31    = 0;
                    temp41    = 0;
                    for L1 = 1:N_rf
                    if L1~= ii1
                       v_rf21(L1,J1)      =
v_rf1(L1,J1);
```

```
                        temp31              =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                        temp41              =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)         =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)         =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    vrf_ij11(ii1,J1)     = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)     = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)      = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
```

```
                    f_vrf21(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)      = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            g_kmm = [];
            thet    = 6 + (0.5 - (-
6).*rand(1,1)); %pembangkitan theta dari 6
sampai 12 derajat
```

```
            g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g2      = 2*g-1; %pembangkitan bpsk
            thett   = (g2.*thet);


            beta_k  = 1;

            psi         = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it          = randi([0 1],k,1)>=0.5;
            itb         = (it*2)-1;
            psii        = (itb.*psi);

            for kurlos      = 1:1:k
            for murlos      = 1:1:m
                theta_k(murlos,kurlos)  = thett
+ (5*(kurlos-1)); %selisih 0.5 derajat antar
antena
                ekurung(murlos,kurlos)  =
((1*exp((-1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
            end
            ekurungt                    =
transpose(ekurung);
            g_km2(:,kurlos)             =
(sqrt(beta_k) .* exp(1i*psii(kurlos))) *
ekurungt(kurlos,:);
            end
            g_kmt1      = transpose(g_km2);
            end

            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
```

```
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1             = Pmiring1 +
PK21;
                pk1                  = 1/qkk1 *
PK21;
                Pp1(k21,k21)         = pk1;
                PPP1                 = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)             = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
```

```
            end
        cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_URLOS        = real(sum11);
%%%
```

```
plot(SNR_dB, mean(SE_URLOS,2), 'rx-'); hold on;
%plot(SNR_dB, mean(SE_IRF1irf3,2), 'b.-'); hold
on;
%plot(SNR_dB, mean(SE_IRF1irf4,2), 'r.-'); hold
on;
legend('IRF','URLOS');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR FDB');

figure(2)
semilogy(SNR_dB,mean(simBER,1),'bx-'); hold on;
semilogy(SNR_dB,mean(simBERurlos,1),'rx-'); hold
on;
%semilogy(SNR_dB,mean(simBERirf3,1),'bx-'); hold
on;
%semilogy(SNR_dB,mean(simBERirf4,1),'rx-'); hold
on;
axis([-10 30 10^-5 0.5])
grid on
legend('IRF','URLOS');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation FDB');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATLAB SCRIPT PERBANDINGAN JUMLAH RF CHAIN IRF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;
clc;

m = 64; k = 4; %JUMLAH ANTENA
P               = 1; %Daya dalam Watt
s               = 1; %menurut analisa, WAJIB
[Jml Ns == Jml m];
k_user          = k;
Ns              = s*k_user;
N_tx            = m;
N_rf            = Ns; %untuk syarat Hybrid
Beamforming
N_rf2           = 2*Ns; %untuk syarat Full
Digital Beamforming
N_rf3           = 3*Ns;
N_rf4           = 4*Ns;

SNR_dB  = -2:4:30; %range SNR in dB
SNR_lin = 10.^(SNR_dB/10); % range SNR in linear
power
N =9;

for SNRsize = 1 : length(SNR_dB)
    SNR = SNR_lin(SNRsize);
    variance = P/SNR;
    for n = 1:N
    variancekanal = 1;
    h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
    g_km  = h_km;
    g_kmt = transpose(g_km);
    Pp              = eye(k);       %step 2
    beta_kp         = ones([1, 1]);
```

```
    v_rf        = randi([1], N_tx, N_rf); %step
1
    v_rf1       = randi([1], N_tx, N_rf);
    v_rf2       = randi([1], N_tx, N_rf);
    vrf_opt     = randi([1], N_tx, N_rf);
    argmin      = randi([1], N_tx, N_rf)*5;
    selisih     = 1;
    temp_topt   = randi([1], N_tx, N_rf)*20;
    teta_opt    = ones(N_tx, N_rf);
    teta_opt2   = ones(N_tx, N_rf);
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp    = 0;
            temp2   = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp            = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
```

```
                    end
                    end
                    temporary1(ii,J)    = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)    = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3    = 0;
                    temp4    = 0;
                    for L = 1:N_rf
                    if L~= ii
                       v_rf2(L,J)       =
v_rf(L,J);
                       temp3            =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4            =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)      =  temp3;
                    n_ijD(ii,J)      =  temp4;
                    c_ij(ii,J)       =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)       =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
```

```
                    vrf_ij1(ii,J)      = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)      = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)      =
argmin(ii,J);
                    argmin(ii,J)       =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)      = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end
```

```
            if isnan(A)
            variancekanal = 1;
            h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km  = h_km;
            g_kmt = transpose(g_km);
            Pp              = eye(k);
%step 2
            beta_kp         = ones([1, 1]);
            v_rf        = randi([1], N_tx,
N_rf); %step 1
            end

            cnt = cnt+1;

        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
```

```
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_IRF1             = real(sum1);
%%%
```

```
        detektor                        =
real(y)>0;
        for sinyal2= 1:Ns
            if detektor(sinyal2,stream) ==1
                sinyal_rx(sinyal2, stream) = 1;
            else sinyal_rx(sinyal2,stream) = -1;
            end
        end
    end
    nErr(n,SNRsize) = size(find([detektor-
genbpsk]),1);
    nErr(nErr==0) = 1;
    end

    for n2 = 1:N
    variancekanal1 = 1;
    h_km1  =
sqrt(variancekanal1/2)*(randn([m],[k])+((1i)*ran
dn([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
    g_km1  = h_km1;
    g_kmt1 = transpose(g_km1);
    Pp1               = eye(k);       %step 2
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx, N_rf2);
%step 1
    v_rf11        = randi([1], N_tx, N_rf2);
    v_rf21        = randi([1], N_tx, N_rf2);
    vrf_opt1      = randi([1], N_tx, N_rf2);
    argmin1       = randi([1], N_tx, N_rf2)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf2)*20;
    teta_opt1     = ones(N_tx, N_rf2);
    teta_opt21    = ones(N_tx, N_rf2);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf2%step 1
```

```
            Hbar1           = Pp1^(-1/2) * g_kmt1;
            v_rfj1          = v_rf1;
            v_rfj1(:,J1)    = [];
            A1              = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21             = inv(A1) * inv(A1);
            A11             = inv(A1);
            B1              = Hbar1' * A21 *
Hbar1;
            D1              = Hbar1' * A11 *
Hbar1;
            B21             = Hbar1' * A21 *
Hbar1;
            D21             = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1    = 0;
            temp21   = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
```

```
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
                    for L1 = 1:N_rf2
                    if L1~= ii1
                       v_rf21(L1,J1)     =
v_rf1(L1,J1);
                       temp31            =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                       temp41            =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)        =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)        =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
```

```
                    vrf_ij11(ii1,J1)      = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)      = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)      = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
```

```
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            variancekanal1 = 1;
            h_km1  =
sqrt(variancekanal1/2)*(randn([m],[k])+((1i)*ran
dn([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km1  = h_km1;
            g_kmt1 = transpose(g_km1);
            Pp1              = eye(k);
%step 2
            beta_kp1         = ones([1, 1]);
            v_rf1        = randi([1], N_tx,
N_rf2); %step 1
            end

            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d1            = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1              = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1            = svd(M11);
                qkk1               =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
```

```
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1            = Pmiring1 +
PK21;
                pk1                 = 1/qkk1 *
PK21;
                Pp1(k21,k21)        = pk1;
                PPP1                = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)            = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
       cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_IRF1irf2         = real(sum11);
%%%
```

```
        genbpsk1(:,stream1)      =
rand(jumlah_data1, 1)>0.5;
        sigbpsk1(:,stream1)      =
2*genbpsk1(:,stream1) -1;
        y1(:,stream1)            = g_kmt1 *
vrf_opt1 * VD1 * sigbpsk1(:,stream1);
        detektor1                = real(y1)>0;
        for sinyal21= 1:Ns
            if detektor1(sinyal21,stream1) ==1
                sinyal_rx1(sinyal21, stream1) =
1;
            else sinyal_rx1(sinyal21,stream1) =
-1;
            end
        end
    end
    nErr1(n,SNRsize) = size(find([detektor1-
genbpsk1]),1);
    nErr1(nErr1==0) = 1;
    end

%%%
```

```
    vrf_opt11      = randi([1], N_tx, N_rf3);
    argmin11       = randi([1], N_tx, N_rf3)*5;
    selisih11      = 1;
    temp_topt11    = randi([1], N_tx, N_rf3)*20;
    teta_opt11     = ones(N_tx, N_rf3);
    teta_opt211    = ones(N_tx, N_rf3);
    cnt211 = 0;
    while 1
    cnt11 = 0;
        while 1
            for J11 = 1:1:N_rf3%step 1
            Hbar11          = Pp11^(-1/2) *
g_kmt11;
            v_rfj11         = v_rf11;
            v_rfj11(:,J11)    = [];
            A11             = Pp11^(-1/2) *
g_kmt11 * v_rfj11 * v_rfj11' * g_kmt11' *
Pp11^(-1/2);
            A211            = inv(A11) *
inv(A11);
            A111            = inv(A11);
            B11             = Hbar11' * A211 *
Hbar11;
            D11             = Hbar11' * A111 *
Hbar11;
            B211            = Hbar11' * A211 *
Hbar11;
            D211            = Hbar11' * A111 *
Hbar11;
            for ii11 = 1:1:N_tx
            temp11     = 0;
            temp211    = 0;
                    for K11= 1:1:N_tx
                    for M11 = 1:1:N_tx
                        if K11 ~= ii11 && M11 ~=
ii11
                        v_rf111(K11,J11) =
v_rf11(K11,J11);
                        v_rf111(M11,J11) =
v_rf11(M11,J11);
```

```
                        temp11            =
temp11 + (conj(v_rf111(K11,J11)) * B11(K11,M11)
* v_rf111(M11,J11));
                        temp211           =
temp211+ (conj(v_rf111(K11,J11)) * D11(K11,M11)
* v_rf111(M11,J11));
                        end
                    end
                    end
                    temporary111(ii11,J11)    =
temp11;
                    temporary211(ii11,J11)    =
temp211;
                    si_ijB11(ii11,J11)    =
B11(ii11,ii11) + 2*real(temporary111(ii11,J11));
                    si_ijD11(ii11,J11)    =
D11(ii11,ii11) + 2*real(temporary211(ii11,J11));
                    temp311   = 0;
                    temp411   = 0;
                    for L11 = 1:N_rf3
                    if L11~= ii11
                       v_rf211(L11,J11)     =
v_rf11(L11,J11);
                       temp311           =
temp311 + (B211(ii11,L11) * v_rf211(L11,J11));
                       temp411           =
temp411 + (D211(ii11,L11) * v_rf211(L11,J11));
                    end
                    end
                    n_ijB11(ii11,J11)      =
temp311;
                    n_ijD11(ii11,J11)      =
temp411;
                    c_ij11(ii11,J11)       =
((1+si_ijD11(ii11,J11))*n_ijB11(ii11,J11)) -
(si_ijB11(ii11,J11) * n_ijD11(ii11,J11));
                    z_ij11(ii11,J11)        =
imag((2*n_ijB11(ii11,J11)) * n_ijD11(ii11,J11));
                    if real(c_ij11(ii11,J11)) >=
0
```

```
                        psi_ij11(ii11,J11)  =
asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11)
));
                    elseif
real(c_ij11(ii11,J11)) < 0
                        psi_ij11(ii11,J11)  = pi
-
(asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11
))));
                    end
                    teta_ij111(ii11,J11)    =
(((-1)*psi_ij11(ii11,J11)) +
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    teta_ij211(ii11,J11)    =
(pi + ((-1)*psi_ij11(ii11,J11)) -
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    vrf_ij111(ii11,J11)     =
exp(-1i*angle(teta_ij111(ii11,J11)));
                    vrf_ij211(ii11,J11)     =
exp(-1i*angle(teta_ij211(ii11,J11)));
                    f_vrf111(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij111(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    f_vrf211(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij211(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij211(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    argmin211(ii11,J11)     =
argmin11(ii11,J11);
                    argmin11(ii11,J11)      =
min(angle(f_vrf111(ii11,J11)),angle(f_vrf211(ii1
1,J11)));
                    if argmin11(ii11,J11) ==
f_vrf111(ii11,J11) %% INI MENCARI TETA OPTIMUM
```

```
                        teta_opt1(ii11,J11) =
angle(f_vrf111(ii11,J11));
                    elseif argmin11(ii11,J11) ~=
f_vrf111(ii11,J11)
                        teta_opt11(ii11,J11) =
angle(f_vrf211(ii11,J11));
                    end
                    vrf_opt11(ii11,J11)     =
exp(-1i*(teta_opt11(ii11,J11))); %INI MENCARI
VRF_OPT -- STEP 8
                    v_rf11(ii11,J11) =
vrf_opt11(ii11,J11);
            end
            end
            selisih11 = abs(argmin211(ii11,J11)
- argmin11(ii11,J11));
                if selisih11 <= 0.3
                cnt11 = cnt11 + 1;
                break
                end

            if isnan(A11)
            variancekanal11 = 1;
            h_km11  =
sqrt(variancekanal11/2)*(randn([m],[k])+((1i)*ra
ndn([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km11  = h_km11;
            g_kmt11 = transpose(g_km11);
            Pp11              = eye(k1);
%step 2
            beta_kp11         = ones([1, 1]);
            v_rf11        = randi([1], N_tx,
N_rf3); %step 1
            end

            cnt11 = cnt11+1;

        end
        cnt11 = cnt11 + 1;
```

```
        Pmiring11 = 0;
        for k211 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d11            = vrf_opt11'
* g_kmt11' * (g_kmt11 * vrf_opt11 * vrf_opt11' *
g_kmt11')^-1;
                Qbar11              = vbar_d11'
* vrf_opt11' * vrf_opt11 * vbar_d11;
                M111                = vrf_opt11'
* g_kmt11' * g_kmt11 * vrf_opt11; %UNTUK
MENGHITUNG EIGENVALUE VRF*KANAL
                lambda11            = svd(M111);
                qkk11               =
Qbar11(k211,k211);
                for k311 = 1:1:size(M111,1)
                    PK111(k311) =
beta_kp11/lambda11(k311) - qkk11 * variance;
                    if PK111(k311) >= 0 &&
PK111(k311) <= 10
                        PK211 = PK111(k311);
                    elseif PK111(k311) <= 0
                        PK211 = 0;
                    end
                end
                Pmiring11           = Pmiring11
+ PK211;
                pk11                = 1/qkk11 *
PK211;
                Pp11(k211,k211)       = pk11;
                PPP11               = beta_kp11
* log2(1+ (pk11/variance));
                Rk11(k211)           = log2(1+
(sqrt(pk11)/variance));
                if Pmiring11 == 0
                    Pmiring11 = 1;
                end
            end
            if  trace(Qbar11*Pp11) <= P &&
(log10(Pmiring11)*10) >= ((log10(P)*10)-0.5)
```

```
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt211 = cnt211+1;
            break
            end
       cnt211 = cnt211+1;
    end
    VD11                = vbar_d11 *
(Pp11)^(1/2);
    sum111(SNRsize,n)   = sum(Rk11);
    SE_IRF1irf3         = real(sum111);
%%%
```

```
%%%
```

```
            D111            = Hbar111' * A1111 *
Hbar111;
            B2111           = Hbar111' * A2111 *
Hbar111;
            D2111           = Hbar111' * A1111 *
Hbar111;
            for ii111 = 1:1:N_tx
            temp111    = 0;
            temp2111   = 0;
                    for K111= 1:1:N_tx
                    for M111 = 1:1:N_tx
                        if K111 ~= ii111 && M111
~= ii111
                        v_rf1111(K111,J111) =
v_rf111(K111,J111);
                        v_rf1111(M111,J111) =
v_rf111(M111,J111);
                        temp111             =
temp111 + (conj(v_rf1111(K111,J111)) *
B111(K111,M111) * v_rf1111(M111,J111));
                        temp2111            =
temp2111+ (conj(v_rf1111(K111,J111)) *
D111(K111,M111) * v_rf1111(M111,J111));
                        end
                    end
                    end
                    temporary1111(ii111,J111)
= temp111;
                    temporary2111(ii111,J111)
= temp2111;
                    si_ijB111(ii111,J111)    =
B111(ii111,ii111) +
2*real(temporary1111(ii111,J111));
                    si_ijD111(ii111,J111)    =
D111(ii111,ii111) +
2*real(temporary2111(ii111,J111));
                    temp3111   = 0;
                    temp4111   = 0;
                    for L111 = 1:N_rf4
                    if L111~= ii111
```

```
                        v_rf2111(L111,J111)     =
v_rf111(L111,J111);
                        temp3111            =
temp3111 + (B2111(ii111,L111) *
v_rf2111(L111,J111));
                        temp4111            =
temp4111 + (D2111(ii111,L111) *
v_rf2111(L111,J111));
                    end
                    end
                    n_ijB111(ii111,J111)      =
temp3111;
                    n_ijD111(ii111,J111)      =
temp4111;
                    c_ij111(ii111,J111)       =
((1+si_ijD111(ii111,J111))*n_ijB111(ii111,J111))
- (si_ijB111(ii111,J111) *
n_ijD111(ii111,J111));
                    z_ij111(ii111,J111)       =
imag((2*n_ijB111(ii111,J111)) *
n_ijD111(ii111,J111));
                    if real(c_ij111(ii111,J111))
>= 0
                        psi_ij111(ii111,J111)  =
asin(imag(c_ij111(ii111,J111))/abs(c_ij111(ii111
,J111)));
                    elseif
real(c_ij111(ii111,J111)) < 0
                        psi_ij111(ii111,J111)  =
pi -
(asin(imag(c_ij111(ii111,J111))/abs(c_ij111(ii11
1,J111))));
                    end
                    teta_ij1111(ii111,J111)    =
(((-1)*psi_ij111(ii111,J111)) +
asin(z_ij111(ii111,J111)/abs(c_ij111(ii111,J111)
)));
                    teta_ij2111(ii111,J111)    =
(pi + ((-1)*psi_ij111(ii111,J111)) -
```

```
asin(z_ij111(ii111,J111)/abs(c_ij111(ii111,J111)
)));
                    vrf_ij1111(ii111,J111)     =
exp(-1i*angle(teta_ij1111(ii111,J111)));
                    vrf_ij2111(ii111,J111)     =
exp(-1i*angle(teta_ij2111(ii111,J111)));
                    f_vrf1111(ii111,J111)      =
(m * trace(inv(A111))) - (m * (
(si_ijB111(ii111,J111) +
2*real(conj(vrf_ij1111(ii111,J111)) *
n_ijB111(ii111,J111)))/(1 +
si_ijD111(ii111,J111) +
2*real(conj(vrf_ij1111(ii111,J111))*n_ijD111(ii1
11,J111)))));
                    f_vrf2111(ii111,J111)      =
(m * trace(inv(A111))) - (m * (
(si_ijB111(ii111,J111) +
2*real(conj(vrf_ij2111(ii111,J111)) *
n_ijB111(ii111,J111)))/(1 +
si_ijD111(ii111,J111) +
2*real(conj(vrf_ij2111(ii111,J111))*n_ijD111(ii1
11,J111)))));
                    argmin2111(ii111,J111)     =
argmin111(ii111,J111);
                    argmin11(ii111,J111)      =
min(angle(f_vrf1111(ii111,J111)),angle(f_vrf2111
(ii111,J111)));
                    if argmin11(ii111,J111) ==
f_vrf1111(ii111,J111) %% INI MENCARI TETA
OPTIMUM
                        teta_opt1(ii111,J111) =
angle(f_vrf1111(ii111,J111));
                    elseif argmin111(ii111,J111)
~= f_vrf1111(ii111,J111)
                        teta_opt11(ii111,J111) =
angle(f_vrf2111(ii111,J111));
                    end
                    vrf_opt111(ii111,J111)     =
exp(-1i*(teta_opt111(ii111,J111))); %INI MENCARI
VRF_OPT -- STEP 8
```

```
                    v_rf111(ii111,J111) =
vrf_opt111(ii111,J111);
            end
            end
            selisih111 =
abs(argmin2111(ii111,J111) -
argmin111(ii111,J111));
                if selisih111 <= 0.3
                cnt111 = cnt111 + 1;
                break
                end

            if isnan(A111)
            variancekanal111 = 1;
            h_km111  =
sqrt(variancekanal111/2)*(randn([m],[k])+((1i)*r
andn([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km111  = h_km111;
            g_kmt111 = transpose(g_km111);
            Pp111               = eye(k);
%step 2
            beta_kp111          = ones([1, 1]);
            v_rf111         = randi([1], N_tx,
N_rf4); %step 1
            end

            cnt111 = cnt111+1;

        end
        cnt111 = cnt111 + 1;
        Pmiring111 = 0;
        for k2111 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d111             =
vrf_opt111' * g_kmt111' * (g_kmt111 * vrf_opt111
* vrf_opt111' * g_kmt111')^-1;
                Qbar111               =
vbar_d111' * vrf_opt111' * vrf_opt111 *
vbar_d111;
```

```
                M1111                   =
vrf_opt111' * g_kmt111' * g_kmt111 * vrf_opt111;
%UNTUK MENGHITUNG EIGENVALUE VRF*KANAL
                lambda111               =
svd(M1111);
                qkk111                  =
Qbar111(k2111,k2111);
                for k3111 = 1:1:size(M1111,1)
                    PK1111(k3111) =
beta_kp111/lambda111(k3111) - qkk111 * variance;
                    if PK1111(k3111) >= 0 &&
PK1111(k3111) <= 10
                        PK2111 = PK1111(k3111);
                    elseif PK1111(k3111) <= 0
                        PK2111 = 0;
                    end
                end
                Pmiring111              =
Pmiring111 + PK2111;
                pk111                   = 1/qkk111
* PK2111;
                Pp111(k2111,k2111)         =
pk111;
                PPP111                  =
beta_kp111 * log2(1+ (pk111/variance));
                Rk111(k2111)              =
log2(1+ (sqrt(pk111)/variance));
                if Pmiring111 == 0
                    Pmiring111 = 1;
                end
            end
            if  trace(Qbar111*Pp111) <= P &&
(log10(Pmiring111)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2111 = cnt2111+1;
            break
            end
       cnt2111 = cnt2111+1;
    end
```

```
    VD111                  = vbar_d111 *
(Pp111)^(1/2);
    sum1111(SNRsize,n)    = sum(Rk111);
    SE_IRF1irf4           = real(sum1111);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    jumlah_signal111 = 1000;
    for stream111 = 1:1:jumlah_signal111
        jumlah_data111             = Ns;
        genbpsk111(:,stream111)     =
rand(jumlah_data111, 1)>0.5;
        sigbpsk111(:,stream111)     =
2*genbpsk111(:,stream111) -1;
        y111(:,stream111)           = g_kmt111 *
vrf_opt111 * VD111 * sigbpsk111(:,stream111);
        detektor111                =
real(y111)>0;
        for sinyal2111= 1:Ns
            if detektor111(sinyal2111,stream111)
==1
                sinyal_rx111(sinyal2111,
stream111) = 1;
            else
sinyal_rx111(sinyal2111,stream111) = -1;
            end
        end
    end
    nErr111(n,SNRsize) = size(find([detektor111-
genbpsk111]),1);
    nErr111(nErr111==0) = 1;

end



end
simBER     = nErr/(jumlah_signal*Ns);
simBERirf2 = nErr1/(jumlah_signal1*Ns);
```

```
simBERirf3 = nErr11/(jumlah_signal11*Ns);
simBERirf4 = nErr11/(jumlah_signal11*Ns);


figure(1)
plot(SNR_dB, mean(SE_IRF1,2), 'b.-'); hold on;
plot(SNR_dB, mean(SE_IRF1irf2,2), 'r.-'); hold
on;
plot(SNR_dB, mean(SE_IRF1irf3,2), 'g.-'); hold
on;
plot(SNR_dB, mean(SE_IRF1irf4,2), 'm.-'); hold
on;
legend('NRF1','NRF2','NRF3','NRF4');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR kanal IRF');

figure(2)
semilogy(SNR_dB,mean(simBER,1),'bx-'); hold on;
semilogy(SNR_dB,mean(simBERirf2,1),'rx-'); hold
on;
semilogy(SNR_dB,mean(simBERirf3,1),'gx-'); hold
on;
semilogy(SNR_dB,mean(simBERirf4,1),'mx-'); hold
on;
axis([-10 30 10^-5 0.5])
grid on
legend('NRF1','NRF2','NRF3','NRF4');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation kanal IRF');




%%%
```

```
%MATLAB SCRIPT PERBANDINGAN JUMLAH RF CHAIN
URLOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;
clc;

m = 64; k = 4; %JUMLAH ANTENA
P               = 1; %Daya dalam Watt
s               = 1; %menurut analisa, WAJIB
[Jml Ns == Jml m];
k_user          = k;
Ns              = s*k_user;
N_tx            = m;
N_rf            = Ns; %untuk syarat Hybrid
Beamforming
N_rf2           = 2*Ns; %untuk syarat Full
Digital Beamforming
N_rf3           = 3*Ns;
N_rf4           = 4*Ns;

SNR_dB  = -2:4:30; %range SNR in dB
SNR_lin = 10.^(SNR_dB/10); % range SNR in linear
power
N =9;

for SNRsize = 1 : length(SNR_dB)
    SNR = SNR_lin(SNRsize);
    variance = P/SNR;
    for n = 1:N
    g_km = [];
    thet    = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12 derajat
    g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g2      = 2*g-1; %pembangkitan bpsk
    thett   = (g2.*thet);

    beta_k  = 1;
```

```
    psi          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it           = randi([0 1],k,1)>=0.5;
    itb          = (it*2)-1;
    psii         = (itb.*psi);

    for kurlos       = 1:1:k
    for murlos       = 1:1:m
        theta_k(murlos,kurlos)  = thett +
(5*(kurlos-1)); %selisih 0.5 derajat antar
antena
        ekurung(murlos,kurlos)  = ((1*exp((-
1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
    end
    ekurungt                    =
transpose(ekurung);
    g_km2(:,kurlos)             = (sqrt(beta_k)
.* exp(1i*psii(kurlos))) * ekurungt(kurlos,:);
    end
    g_kmt       = transpose(g_km2);




    Pp               = eye(k);       %step 2
    beta_kp          = ones([1, 1]);

    v_rf         = randi([1], N_tx, N_rf); %step
1
    v_rf1        = randi([1], N_tx, N_rf);
    v_rf2        = randi([1], N_tx, N_rf);
    vrf_opt      = randi([1], N_tx, N_rf);
    argmin       = randi([1], N_tx, N_rf)*5;
    selisih      = 1;
    temp_topt    = randi([1], N_tx, N_rf)*20;
    teta_opt     = ones(N_tx, N_rf);
    teta_opt2    = ones(N_tx, N_rf);
```

```
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp    = 0;
            temp2   = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp            = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
                    end
                    end
                    temporary1(ii,J)    = temp;
                    temporary2(ii,J)   = temp2;
                    si_ijB(ii,J)    = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3   = 0;
                    temp4   = 0;
                    for L = 1:N_rf
```

```
                    if L~= ii
                       v_rf2(L,J)        =
v_rf(L,J);
                       temp3             =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4             =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)       =  temp3;
                    n_ijD(ii,J)       =  temp4;
                    c_ij(ii,J)         =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)         =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    vrf_ij1(ii,J)     = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)     = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
```

```
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)     =
argmin(ii,J);
                    argmin(ii,J)      =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)     = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end

            if isnan(A)
            g_km = [];
            thet    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g2      = 2*g-1; %pembangkitan bpsk
            thett   = (g2.*thet);

            beta_k  = 1;
```

```
            psi         = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it          = randi([0 1],k,1)>=0.5;
            itb         = (it*2)-1;
            psii        = (itb.*psi);

            for kurlos      = 1:1:k
            for murlos      = 1:1:m
            theta_k(murlos,kurlos)  = thett +
(5*(kurlos-1)); %selisih 0.5 derajat antar
antena
            ekurung(murlos,kurlos)  = ((1*exp((-
1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
            end
            ekurungt                    =
transpose(ekurung);
            g_km2(:,kurlos)             =
(sqrt(beta_k) .* exp(1i*psii(kurlos))) *
ekurungt(kurlos,:);
            end
            g_kmt       = transpose(g_km2);
            end

            cnt = cnt+1;

        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
```

```
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_URLOS             = real(sum1);
```

```
%%%
```

```
    it1           = randi([0 1],k,1)>=0.5;
    itb1          = (it1*2)-1;
    psii1         = (itb1.*psi1);

    for kurlos1       = 1:1:k
    for murlos1       = 1:1:m
        theta_k1(murlos1,kurlos1)  = thett1 +
(5*(kurlos1-1)); %selisih 0.5 derajat antar
antena
        ekurung1(murlos1,kurlos1)  = ((1*exp((-
1i)*(murlos1-
1)*pi*sind(theta_k1(murlos1,kurlos1)))));
    end
    ekurungt1                      =
transpose(ekurung1);
    g_km21(:,kurlos1)              =
(sqrt(beta_k1) .* exp(1i*psii1(kurlos1))) *
ekurungt1(kurlos1,:);
    end
    g_kmt1       = transpose(g_km21);


    Pp1               = eye(k);        %step 2
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx, N_rf2);
%step 1
    v_rf11        = randi([1], N_tx, N_rf2);
    v_rf21        = randi([1], N_tx, N_rf2);
    vrf_opt1      = randi([1], N_tx, N_rf2);
    argmin1       = randi([1], N_tx, N_rf2)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf2)*20;
    teta_opt1     = ones(N_tx, N_rf2);
    teta_opt21    = ones(N_tx, N_rf2);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf2%step 1
```

```
            Hbar1           = Pp1^(-1/2) * g_kmt1;
            v_rfj1          = v_rf1;
            v_rfj1(:,J1)    = [];
            A1              = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21             = inv(A1) * inv(A1);
            A11             = inv(A1);
            B1              = Hbar1' * A21 *
Hbar1;
            D1              = Hbar1' * A11 *
Hbar1;
            B21             = Hbar1' * A21 *
Hbar1;
            D21             = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1    = 0;
            temp21   = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
```

```
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
                    for L1 = 1:N_rf2
                    if L1~= ii1
                       v_rf21(L1,J1)     =
v_rf1(L1,J1);
                       temp31            =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                       temp41            =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)        =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)        =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
```

```
                    vrf_ij11(ii1,J1)      = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)      = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)      = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
```

```
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            g_km1 = [];
            thet1    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g1       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g21      = 2*g1-1; %pembangkitan
bpsk
            thett1   = (g21.*thet1);
            beta_k1  = 1;
            psi1          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it1           = randi([0
1],k,1)>=0.5;
            itb1          = (it1*2)-1;
            psii1         = (itb1.*psi1);

            for kurlos1      = 1:1:k
            for murlos1      = 1:1:m
            theta_k1(murlos1,kurlos1)  = thett1
+ (5*(kurlos1-1)); %selisih 0.5 derajat antar
antena
            ekurung1(murlos1,kurlos1)  =
((1*exp((-1i)*(murlos1-
1)*pi*sind(theta_k1(murlos1,kurlos1)))));
            end
            ekurungt1                      =
transpose(ekurung1);
            g_km21(:,kurlos1)              =
(sqrt(beta_k1) .* exp(1i*psii1(kurlos1))) *
ekurungt1(kurlos1,:);
            end
            g_kmt1      = transpose(g_km21);
            end
```

```
            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1             = Pmiring1 +
PK21;
                pk1                  = 1/qkk1 *
PK21;
                Pp1(k21,k21)         = pk1;
                PPP1                 = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)             = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
```

```
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
       cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_URLOS2       = real(sum11);
%%%
```

```
    end

%%%
```

```
    Pp11                = eye(k);       %step 2
    beta_kp11           = ones([1, 1]);

    v_rf11          = randi([1], N_tx, N_rf3);
%step 1
    v_rf111         = randi([1], N_tx, N_rf3);
    v_rf211         = randi([1], N_tx, N_rf3);
    vrf_opt11       = randi([1], N_tx, N_rf3);
    argmin11        = randi([1], N_tx, N_rf3)*5;
    selisih11       = 1;
    temp_topt11     = randi([1], N_tx, N_rf3)*20;
    teta_opt11      = ones(N_tx, N_rf3);
    teta_opt211     = ones(N_tx, N_rf3);
    cnt211 = 0;
    while 1
    cnt11 = 0;
        while 1
            for J11 = 1:1:N_rf3%step 1
            Hbar11          = Pp11^(-1/2) *
g_kmt11;
            v_rfj11         = v_rf11;
            v_rfj11(:,J11)    = [];
            A11             = Pp11^(-1/2) *
g_kmt11 * v_rfj11 * v_rfj11' * g_kmt11' *
Pp11^(-1/2);
            A211            = inv(A11) *
inv(A11);
            A111            = inv(A11);
            B11             = Hbar11' * A211 *
Hbar11;
            D11             = Hbar11' * A111 *
Hbar11;
            B211            = Hbar11' * A211 *
Hbar11;
            D211            = Hbar11' * A111 *
Hbar11;
            for ii11 = 1:1:N_tx
            temp11     = 0;
            temp211    = 0;
```

```
                    for K11= 1:1:N_tx
                    for M11 = 1:1:N_tx
                        if K11 ~= ii11 && M11 ~=
ii11
                        v_rf111(K11,J11) =
v_rf11(K11,J11);
                        v_rf111(M11,J11) =
v_rf11(M11,J11);
                        temp11            =
temp11 + (conj(v_rf111(K11,J11)) * B11(K11,M11)
* v_rf111(M11,J11));
                        temp211           =
temp211+ (conj(v_rf111(K11,J11)) * D11(K11,M11)
* v_rf111(M11,J11));
                        end
                    end
                    end
                    temporary111(ii11,J11)    =
temp11;
                    temporary211(ii11,J11)    =
temp211;
                    si_ijB11(ii11,J11)    =
B11(ii11,ii11) + 2*real(temporary111(ii11,J11));
                    si_ijD11(ii11,J11)    =
D11(ii11,ii11) + 2*real(temporary211(ii11,J11));
                    temp311   = 0;
                    temp411   = 0;
                    for L11 = 1:N_rf3
                    if L11~= ii11
                       v_rf211(L11,J11)     =
v_rf11(L11,J11);
                       temp311           =
temp311 + (B211(ii11,L11) * v_rf211(L11,J11));
                       temp411           =
temp411 + (D211(ii11,L11) * v_rf211(L11,J11));
                    end
                    end
                    n_ijB11(ii11,J11)      =
temp311;
```

```
                    n_ijD11(ii11,J11)      =
temp411;
                    c_ij11(ii11,J11)       =
((1+si_ijD11(ii11,J11))*n_ijB11(ii11,J11)) -
(si_ijB11(ii11,J11) * n_ijD11(ii11,J11));
                    z_ij11(ii11,J11)        =
imag((2*n_ijB11(ii11,J11)) * n_ijD11(ii11,J11));
                    if real(c_ij11(ii11,J11)) >=
0
                        psi_ij11(ii11,J11)  =
asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11)
));
                    elseif
real(c_ij11(ii11,J11)) < 0
                        psi_ij11(ii11,J11)  = pi
-
(asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11
))));
                    end
                    teta_ij111(ii11,J11)    =
(((-1)*psi_ij11(ii11,J11)) +
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    teta_ij211(ii11,J11)    =
(pi + ((-1)*psi_ij11(ii11,J11)) -
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    vrf_ij111(ii11,J11)     =
exp(-1i*angle(teta_ij111(ii11,J11)));
                    vrf_ij211(ii11,J11)     =
exp(-1i*angle(teta_ij211(ii11,J11)));
                    f_vrf111(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij111(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    f_vrf211(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij211(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
```

```
2*real(conj(vrf_ij211(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    argmin211(ii11,J11)     =
argmin11(ii11,J11);
                    argmin11(ii11,J11)      =
min(angle(f_vrf111(ii11,J11)),angle(f_vrf211(ii1
1,J11)));
                    if argmin11(ii11,J11) ==
f_vrf111(ii11,J11) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii11,J11) =
angle(f_vrf111(ii11,J11));
                    elseif argmin11(ii11,J11) ~=
f_vrf111(ii11,J11)
                        teta_opt11(ii11,J11) =
angle(f_vrf211(ii11,J11));
                    end
                    vrf_opt11(ii11,J11)     =
exp(-1i*(teta_opt11(ii11,J11))); %INI MENCARI
VRF_OPT -- STEP 8
                    v_rf11(ii11,J11) =
vrf_opt11(ii11,J11);
            end
            end
            selisih11 = abs(argmin211(ii11,J11)
- argmin11(ii11,J11));
                if selisih11 <= 0.3
                cnt11 = cnt11 + 1;
                break
                end

            if isnan(A11)
            g_km11 = [];
            thet11    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g11       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g211      = 2*g11-1; %pembangkitan
bpsk
            thett11   = (g211.*thet11);
```

```
            beta_k11  = 1;
            psi11          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it11           = randi([0
1],k,1)>=0.5;
            itb11          = (it11*2)-1;
            psii11         = (itb11.*psi11);

            for kurlos11       = 1:1:k
            for murlos11       = 1:1:m
            theta_k11(murlos11,kurlos11)  =
thett11 + (5*(kurlos11-1)); %selisih 0.5 derajat
antar antena
            ekurung11(murlos11,kurlos11)  =
((1*exp((-1i)*(murlos11-
1)*pi*sind(theta_k11(murlos11,kurlos11)))));
            end
            ekurungt11                    =
transpose(ekurung11);
            g_km211(:,kurlos11)            =
(sqrt(beta_k11) .* exp(1i*psii11(kurlos11))) *
ekurungt11(kurlos11,:);
            end
            g_kmt11      = transpose(g_km211);
            end

            cnt11 = cnt11+1;

        end
        cnt11 = cnt11 + 1;
        Pmiring11 = 0;
        for k211 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d11            = vrf_opt11'
* g_kmt11' * (g_kmt11 * vrf_opt11 * vrf_opt11' *
g_kmt11')^-1;
                Qbar11              = vbar_d11'
* vrf_opt11' * vrf_opt11 * vbar_d11;
```

```
                M111                = vrf_opt11'
* g_kmt11' * g_kmt11 * vrf_opt11; %UNTUK
MENGHITUNG EIGENVALUE VRF*KANAL
                lambda11            = svd(M111);
                qkk11               =
Qbar11(k211,k211);
                for k311 = 1:1:size(M111,1)
                    PK111(k311) =
beta_kp11/lambda11(k311) - qkk11 * variance;
                    if PK111(k311) >= 0 &&
PK111(k311) <= 10
                        PK211 = PK111(k311);
                    elseif PK111(k311) <= 0
                        PK211 = 0;
                    end
                end
                Pmiring11           = Pmiring11
+ PK211;
                pk11                = 1/qkk11 *
PK211;
                Pp11(k211,k211)       = pk11;
                PPP11               = beta_kp11
* log2(1+ (pk11/variance));
                Rk11(k211)           = log2(1+
(sqrt(pk11)/variance));
                if Pmiring11 == 0
                    Pmiring11 = 1;
                end
            end
            if  trace(Qbar11*Pp11) <= P &&
(log10(Pmiring11)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt211 = cnt211+1;
            break
            end
       cnt211 = cnt211+1;
    end
    VD11                = vbar_d11 *
(Pp11)^(1/2);
```

```
    sum111(SNRsize,n)    = sum(Rk11);
    SE_URLOS3          = real(sum111);
%%%
```

```
    thett111    = (g2111.*thet111);
    beta_k111   = 1;
    psi111          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it111           = randi([0 1],k,1)>=0.5;
    itb111          = (it111*2)-1;
    psii111         = (itb111.*psi111);

    for kurlos111       = 1:1:k
    for murlos111       = 1:1:m
        theta_k111(murlos111,kurlos111)  =
thett111 + (5*(kurlos111-1)); %selisih 0.5
derajat antar antena
        ekurung111(murlos111,kurlos111)  =
((1*exp((-1i)*(murlos111-
1)*pi*sind(theta_k111(murlos111,kurlos111)))));
    end
    ekurungt111                     =
transpose(ekurung111);
    g_km2111(:,kurlos111)             =
(sqrt(beta_k111) .* exp(1i*psii111(kurlos111)))
* ekurungt111(kurlos111,:);
    end
    g_kmt111      = transpose(g_km2111);


    Pp111               = eye(k);       %step 2
    beta_kp111          = ones([1, 1]);

    v_rf111         = randi([1], N_tx, N_rf4);
%step 1
    v_rf1111        = randi([1], N_tx, N_rf4);
    v_rf2111        = randi([1], N_tx, N_rf4);
    vrf_opt111      = randi([1], N_tx, N_rf4);
    argmin111       = randi([1], N_tx, N_rf4)*5;
    selisih111      = 1;
    temp_topt111    = randi([1], N_tx, N_rf4)*20;
    teta_opt111     = ones(N_tx, N_rf4);
    teta_opt2111    = ones(N_tx, N_rf4);
```

```
    cnt2111 = 0;
    while 1
    cnt111 = 0;
        while 1
            for J111 = 1:1:N_rf4%step 1
            Hbar111          = Pp111^(-1/2) *
g_kmt111;
            v_rfj111         = v_rf111;
            v_rfj111(:,J111)   = [];
            A111             = Pp111^(-1/2) *
g_kmt111 * v_rfj111 * v_rfj111' * g_kmt111' *
Pp111^(-1/2);
            A2111            = inv(A111) *
inv(A111);
            A1111            = inv(A111);
            B111             = Hbar111' * A2111 *
Hbar111;
            D111             = Hbar111' * A1111 *
Hbar111;
            B2111            = Hbar111' * A2111 *
Hbar111;
            D2111            = Hbar111' * A1111 *
Hbar111;
            for ii111 = 1:1:N_tx
            temp111    = 0;
            temp2111   = 0;
                    for K111= 1:1:N_tx
                    for M111 = 1:1:N_tx
                        if K111 ~= ii111 && M111
~= ii111
                        v_rf1111(K111,J111) =
v_rf111(K111,J111);
                        v_rf1111(M111,J111) =
v_rf111(M111,J111);
                        temp111             =
temp111 + (conj(v_rf1111(K111,J111)) *
B111(K111,M111) * v_rf1111(M111,J111));
                        temp2111            =
temp2111+ (conj(v_rf1111(K111,J111)) *
D111(K111,M111) * v_rf1111(M111,J111));
```

```
                        end
                    end
                    end
                    temporary1111(ii111,J111)
= temp111;
                    temporary2111(ii111,J111)
= temp2111;
                    si_ijB111(ii111,J111)    =
B111(ii111,ii111) +
2*real(temporary1111(ii111,J111));
                    si_ijD111(ii111,J111)    =
D111(ii111,ii111) +
2*real(temporary2111(ii111,J111));
                    temp3111   = 0;
                    temp4111   = 0;
                    for L111 = 1:N_rf4
                    if L111~= ii111
                       v_rf2111(L111,J111)     =
v_rf111(L111,J111);
                       temp3111            =
temp3111 + (B2111(ii111,L111) *
v_rf2111(L111,J111));
                       temp4111            =
temp4111 + (D2111(ii111,L111) *
v_rf2111(L111,J111));
                    end
                    end
                    n_ijB111(ii111,J111)      =
temp3111;
                    n_ijD111(ii111,J111)      =
temp4111;
                    c_ij111(ii111,J111)       =
((1+si_ijD111(ii111,J111))*n_ijB111(ii111,J111))
- (si_ijB111(ii111,J111) *
n_ijD111(ii111,J111));
                    z_ij111(ii111,J111)       =
imag((2*n_ijB111(ii111,J111)) *
n_ijD111(ii111,J111));
                    if real(c_ij111(ii111,J111))
>= 0
```

```
                        psi_ij111(ii111,J111)  =
asin(imag(c_ij111(ii111,J111))/abs(c_ij111(ii111
,J111)));
                    elseif
real(c_ij111(ii111,J111)) < 0
                        psi_ij111(ii111,J111)  =
pi -
(asin(imag(c_ij111(ii111,J111))/abs(c_ij111(ii11
1,J111))));
                    end
                    teta_ij1111(ii111,J111)    =
(((-1)*psi_ij111(ii111,J111)) +
asin(z_ij111(ii111,J111)/abs(c_ij111(ii111,J111)
)));
                    teta_ij2111(ii111,J111)    =
(pi + ((-1)*psi_ij111(ii111,J111)) -
asin(z_ij111(ii111,J111)/abs(c_ij111(ii111,J111)
)));
                    vrf_ij1111(ii111,J111)     =
exp(-1i*angle(teta_ij1111(ii111,J111)));
                    vrf_ij2111(ii111,J111)     =
exp(-1i*angle(teta_ij2111(ii111,J111)));
                    f_vrf1111(ii111,J111)      =
(m * trace(inv(A111))) - (m * (
(si_ijB111(ii111,J111) +
2*real(conj(vrf_ij1111(ii111,J111)) *
n_ijB111(ii111,J111)))/(1 +
si_ijD111(ii111,J111) +
2*real(conj(vrf_ij1111(ii111,J111))*n_ijD111(ii1
11,J111)))));
                    f_vrf2111(ii111,J111)      =
(m * trace(inv(A111))) - (m * (
(si_ijB111(ii111,J111) +
2*real(conj(vrf_ij2111(ii111,J111)) *
n_ijB111(ii111,J111)))/(1 +
si_ijD111(ii111,J111) +
2*real(conj(vrf_ij2111(ii111,J111))*n_ijD111(ii1
11,J111)))));
                    argmin2111(ii111,J111)     =
argmin111(ii111,J111);
```

```
                    argmin11(ii111,J111)      =
min(angle(f_vrf1111(ii111,J111)),angle(f_vrf2111
(ii111,J111)));
                    if argmin11(ii111,J111) ==
f_vrf1111(ii111,J111) %% INI MENCARI TETA
OPTIMUM
                        teta_opt1(ii111,J111) =
angle(f_vrf1111(ii111,J111));
                    elseif argmin111(ii111,J111)
~= f_vrf1111(ii111,J111)
                        teta_opt11(ii111,J111) =
angle(f_vrf2111(ii111,J111));
                    end
                    vrf_opt111(ii111,J111)     =
exp(-1i*(teta_opt111(ii111,J111))); %INI MENCARI
VRF_OPT -- STEP 8
                    v_rf111(ii111,J111) =
vrf_opt111(ii111,J111);
            end
            end
            selisih111 =
abs(argmin2111(ii111,J111) -
argmin111(ii111,J111));
                if selisih111 <= 0.3
                cnt111 = cnt111 + 1;
                break
                end

            if isnan(A111)
            g_km111 = [];
            thet111    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g111       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g2111      = 2*g111-1; %pembangkitan
bpsk
            thett111   = (g2111.*thet111);
            beta_k111  = 1;
```

```
            psi111          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it111           = randi([0
1],k,1)>=0.5;
            itb111          = (it111*2)-1;
            psii111         = (itb111.*psi111);

            for kurlos111       = 1:1:k
            for murlos111       = 1:1:m
            theta_k111(murlos111,kurlos111)  =
thett111 + (5*(kurlos111-1)); %selisih 0.5
derajat antar antena
            ekurung111(murlos111,kurlos111)  =
((1*exp((-1i)*(murlos111-
1)*pi*sind(theta_k111(murlos111,kurlos111)))));
            end
            ekurungt111                     =
transpose(ekurung111);
            g_km2111(:,kurlos111)           =
(sqrt(beta_k111) .* exp(1i*psii111(kurlos111)))
* ekurungt111(kurlos111,:);
            end
            g_kmt111      = transpose(g_km2111);
            end

            cnt111 = cnt111+1;

        end
        cnt111 = cnt111 + 1;
        Pmiring111 = 0;
        for k2111 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d111           =
vrf_opt111' * g_kmt111' * (g_kmt111 * vrf_opt111
* vrf_opt111' * g_kmt111')^-1;
                Qbar111             =
vbar_d111' * vrf_opt111' * vrf_opt111 *
vbar_d111;
```

```
                M1111                   =
vrf_opt111' * g_kmt111' * g_kmt111 * vrf_opt111;
%UNTUK MENGHITUNG EIGENVALUE VRF*KANAL
                lambda111               =
svd(M1111);
                qkk111                  =
Qbar111(k2111,k2111);
                for k3111 = 1:1:size(M1111,1)
                    PK1111(k3111) =
beta_kp111/lambda111(k3111) - qkk111 * variance;
                    if PK1111(k3111) >= 0 &&
PK1111(k3111) <= 10
                        PK2111 = PK1111(k3111);
                    elseif PK1111(k3111) <= 0
                        PK2111 = 0;
                    end
                end
                Pmiring111              =
Pmiring111 + PK2111;
                pk111                   = 1/qkk111
* PK2111;
                Pp111(k2111,k2111)         =
pk111;
                PPP111                  =
beta_kp111 * log2(1+ (pk111/variance));
                Rk111(k2111)              =
log2(1+ (sqrt(pk111)/variance));
                if Pmiring111 == 0
                    Pmiring111 = 1;
                end
            end
            if  trace(Qbar111*Pp111) <= P &&
(log10(Pmiring111)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2111 = cnt2111+1;
            break
            end
       cnt2111 = cnt2111+1;
    end
```

```
    VD111                   = vbar_d111 *
(Pp111)^(1/2);
    sum1111(SNRsize,n)    = sum(Rk111);
    SE_URLOS4           = real(sum1111);
%%%
```

```
simBERurlos3    = nErr11/(jumlah_signal11*Ns);
simBERurlos4    = nErr11/(jumlah_signal11*Ns);


figure(1)
plot(SNR_dB, mean(SE_URLOS,2), 'b.-'); hold on;
plot(SNR_dB, mean(SE_URLOS2,2), 'r.-'); hold on;
plot(SNR_dB, mean(SE_URLOS3,2), 'g.-'); hold on;
plot(SNR_dB, mean(SE_URLOS4,2), 'm.-'); hold on;
legend('NRF1','NRF2','NRF3','NRF4');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR kanal URLOS');

figure(2)
semilogy(SNR_dB,mean(simBERurlos,1),'bx-'); hold
on;
semilogy(SNR_dB,mean(simBERurlos2,1),'rx-');
hold on;
semilogy(SNR_dB,mean(simBERurlos3,1),'gx-');
hold on;
semilogy(SNR_dB,mean(simBERurlos4,1),'mx-');
hold on;
axis([-10 30 10^-5 0.5])
grid on
legend('NRF1','NRF2','NRF3','NRF4');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation kanal URLOS');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATLAB SCRIPT PERBANDINGAN BEDA JUMLAH
PEMANCAR(TX) IRF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;
```

```
clc;

m = 64; k = 4; %JUMLAH ANTENA
mm= 32;
mmm=64;
P                = 1; %Daya dalam Watt
s                = 1; %menurut analisa, WAJIB
[Jml Ns == Jml m];
k_user           = k;
Ns               = s*k_user;
N_tx             = m;
N_tx2            = mm;
N_tx3            = mmm;
N_rf             = Ns; %untuk syarat Hybrid
Beamforming

SNR_dB  = -2:4:30; %range SNR in dB
SNR_lin = 10.^(SNR_dB/10); % range SNR in linear
power
N =9;

for SNRsize = 1 : length(SNR_dB)
    SNR = SNR_lin(SNRsize);
    variance = P/SNR;
    for n = 1:N
    variancekanal = 1;
    h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
    g_km  = h_km;
    g_kmt = transpose(g_km);
    Pp               = eye(k);       %step 2
    beta_kp          = ones([1, 1]);

    v_rf         = randi([1], N_tx, N_rf); %step
1
    v_rf1        = randi([1], N_tx, N_rf);
    v_rf2        = randi([1], N_tx, N_rf);
    vrf_opt      = randi([1], N_tx, N_rf);
```

```
    argmin      = randi([1], N_tx, N_rf)*5;
    selisih     = 1;
    temp_topt   = randi([1], N_tx, N_rf)*20;
    teta_opt    = ones(N_tx, N_rf);
    teta_opt2   = ones(N_tx, N_rf);
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp     = 0;
            temp2    = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp             = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2            = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
                    end
                    end
                    temporary1(ii,J)     = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)     = B(ii,ii) +
2*real(temporary1(ii,J));
```

```
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3   = 0;
                    temp4   = 0;
                    for L = 1:N_rf
                    if L~= ii
                       v_rf2(L,J)       =
v_rf(L,J);
                       temp3            =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4            =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)      =  temp3;
                    n_ijD(ii,J)      =  temp4;
                    c_ij(ii,J)       =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)       =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    vrf_ij1(ii,J)     = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)     = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
```

```
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)     =
argmin(ii,J);
                    argmin(ii,J)      =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)     = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end

            if isnan(A)
            variancekanal = 1;
            h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
```

```
            g_km  = h_km;
            g_kmt = transpose(g_km);
            Pp              = eye(k);
%step 2
            beta_kp         = ones([1, 1]);
            v_rf        = randi([1], N_tx,
N_rf); %step 1
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
```

```
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_IRF1             = real(sum1);
%%%
```

```
            else sinyal_rx(sinyal2,stream) = -1;
            end
        end
    end
    nErr(n,SNRsize) = size(find([detektor-
genbpsk]),1);
    nErr(nErr==0) = 1;
    end

    for n2 = 1:N
    variancekanal1 = 1;
    h_km1  =
sqrt(variancekanal1/2)*(randn([mm],[k])+((1i)*ra
ndn([mm],[k]))); %small scale fading berdasar
Jumlah antena dan user
    g_km1  = h_km1;
    g_kmt1 = transpose(g_km1);
    Pp1              = eye(k);       %step 2
    beta_kp1         = ones([1, 1]);

    v_rf1        = randi([1], N_tx2, N_rf);
%step 1
    v_rf11       = randi([1], N_tx2, N_rf);
    v_rf21       = randi([1], N_tx2, N_rf);
    vrf_opt1     = randi([1], N_tx2, N_rf);
    argmin1      = randi([1], N_tx2, N_rf)*5;
    selisih1     = 1;
    temp_topt1   = randi([1], N_tx2, N_rf)*20;
    teta_opt1    = ones(N_tx2, N_rf);
    teta_opt21   = ones(N_tx2, N_rf);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf%step 1
            Hbar1          = Pp1^(-1/2) * g_kmt1;
            v_rfj1         = v_rf1;
            v_rfj1(:,J1)    = [];
            A1             = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
```

```
            A21             = inv(A1) * inv(A1);
            A11             = inv(A1);
            B1              = Hbar1' * A21 *
Hbar1;
            D1              = Hbar1' * A11 *
Hbar1;
            B21             = Hbar1' * A21 *
Hbar1;
            D21             = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx2
            temp1    = 0;
            temp21   = 0;
                    for K1= 1:1:N_tx2
                    for M1 = 1:1:N_tx2
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
```

```
                    for L1 = 1:N_rf
                    if L1~= ii1
                       v_rf21(L1,J1)      =
v_rf1(L1,J1);
                       temp31             =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                       temp41             =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)       =
temp31;
                    n_ijD1(ii1,J1)       =
temp41;
                    c_ij1(ii1,J1)         =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)         =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    vrf_ij11(ii1,J1)     = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)     = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)      = (mm *
trace(inv(A1))) - (mm * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
```

```
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)       = (mm *
trace(inv(A1))) - (mm * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)      = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            variancekanal1 = 1;
```

```
            h_km1  =
sqrt(variancekanal1/2)*(randn([mm],[k])+((1i)*ra
ndn([mm],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km1  = h_km1;
            g_kmt1 = transpose(g_km1);
            Pp1               = eye(k);
%step 2
            beta_kp1          = ones([1, 1]);
            v_rf1         = randi([1], N_tx2,
N_rf); %step 1
            end

            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
```

```
                end
                Pmiring1            = Pmiring1 +
PK21;
                pk1                 = 1/qkk1 *
PK21;
                Pp1(k21,k21)        = pk1;
                PPP1                = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)            = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
       cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_IRF132         = real(sum11);
%%%
```

```
        for sinyal21= 1:Ns
            if detektor1(sinyal21,stream1) ==1
                sinyal_rx1(sinyal21, stream1) =
1;
            else sinyal_rx1(sinyal21,stream1) =
-1;
            end
        end
    end
    nErr1(n,SNRsize) = size(find([detektor1-
genbpsk1]),1);
    nErr1(nErr1==0) = 1;
    end

    for n11 = 1:N
    variancekanal11 = 1;
    h_km11  =
sqrt(variancekanal11/2)*(randn([mmm],[k])+((1i)*
randn([mmm],[k]))); %small scale fading berdasar
Jumlah antena dan user
    g_km11  = h_km11;
    g_kmt11 = transpose(g_km11);
    Pp11              = eye(k);       %step 2
    beta_kp11         = ones([1, 1]);

    v_rf11        = randi([1], N_tx3, N_rf);
%step 1
    v_rf111       = randi([1], N_tx3, N_rf);
    v_rf211       = randi([1], N_tx3, N_rf);
    vrf_opt11     = randi([1], N_tx3, N_rf);
    argmin11      = randi([1], N_tx3, N_rf)*5;
    selisih11     = 1;
    temp_topt11   = randi([1], N_tx3, N_rf)*20;
    teta_opt11    = ones(N_tx3, N_rf);
    teta_opt211   = ones(N_tx3, N_rf);
    cnt211 = 0;
    while 1
    cnt11 = 0;
        while 1
            for J11 = 1:1:N_rf%step 1
```

```
            Hbar11          = Pp11^(-1/2) *
g_kmt11;
            v_rfj11         = v_rf11;
            v_rfj11(:,J11)    = [];
            A11             = Pp11^(-1/2) *
g_kmt11 * v_rfj11 * v_rfj11' * g_kmt11' *
Pp11^(-1/2);
            A211            = inv(A11) *
inv(A11);
            A111            = inv(A11);
            B11             = Hbar11' * A211 *
Hbar11;
            D11             = Hbar11' * A111 *
Hbar11;
            B211            = Hbar11' * A211 *
Hbar11;
            D211            = Hbar11' * A111 *
Hbar11;
            for ii11 = 1:1:N_tx3
            temp11    = 0;
            temp211   = 0;
                    for K11= 1:1:N_tx3
                    for M11 = 1:1:N_tx3
                        if K11 ~= ii11 && M11 ~=
ii11
                        v_rf111(K11,J11) =
v_rf11(K11,J11);
                        v_rf111(M11,J11) =
v_rf11(M11,J11);
                        temp11            =
temp11 + (conj(v_rf111(K11,J11)) * B11(K11,M11)
* v_rf111(M11,J11));
                        temp211           =
temp211+ (conj(v_rf111(K11,J11)) * D11(K11,M11)
* v_rf111(M11,J11));
                        end
                    end
                    end
                    temporary111(ii11,J11)    =
temp11;
```

```
                    temporary211(ii11,J11)    =
temp211;
                    si_ijB11(ii11,J11)    =
B11(ii11,ii11) + 2*real(temporary111(ii11,J11));
                    si_ijD11(ii11,J11)    =
D11(ii11,ii11) + 2*real(temporary211(ii11,J11));
                    temp311   = 0;
                    temp411   = 0;
                    for L11 = 1:N_rf
                    if L11~= ii11
                       v_rf211(L11,J11)     =
v_rf11(L11,J11);
                       temp311            =
temp311 + (B211(ii11,L11) * v_rf211(L11,J11));
                       temp411            =
temp411 + (D211(ii11,L11) * v_rf211(L11,J11));
                    end
                    end
                    n_ijB11(ii11,J11)      =
temp311;
                    n_ijD11(ii11,J11)      =
temp411;
                    c_ij11(ii11,J11)       =
((1+si_ijD11(ii11,J11))*n_ijB11(ii11,J11)) -
(si_ijB11(ii11,J11) * n_ijD11(ii11,J11));
                    z_ij11(ii11,J11)       =
imag((2*n_ijB11(ii11,J11)) * n_ijD11(ii11,J11));
                    if real(c_ij11(ii11,J11)) >=
0
                        psi_ij11(ii11,J11)  =
asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11)
));
                    elseif
real(c_ij11(ii11,J11)) < 0
                        psi_ij11(ii11,J11)  = pi
-
(asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11
))));
                    end
```

```
                    teta_ij111(ii11,J11)    =
(((-1)*psi_ij11(ii11,J11)) +
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    teta_ij211(ii11,J11)    =
(pi + ((-1)*psi_ij11(ii11,J11)) -
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    vrf_ij111(ii11,J11)     =
exp(-1i*angle(teta_ij111(ii11,J11)));
                    vrf_ij211(ii11,J11)     =
exp(-1i*angle(teta_ij211(ii11,J11)));
                    f_vrf111(ii11,J11)      =
(mmm * trace(inv(A11))) - (mmm * (
(si_ijB11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    f_vrf211(ii11,J11)      =
(mmm * trace(inv(A11))) - (mmm * (
(si_ijB11(ii11,J11) +
2*real(conj(vrf_ij211(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij211(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    argmin211(ii11,J11)     =
argmin11(ii11,J11);
                    argmin11(ii11,J11)      =
min(angle(f_vrf111(ii11,J11)),angle(f_vrf211(ii1
1,J11)));
                    if argmin11(ii11,J11) ==
f_vrf111(ii11,J11) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii11,J11) =
angle(f_vrf111(ii11,J11));
                    elseif argmin11(ii11,J11) ~=
f_vrf111(ii11,J11)
                        teta_opt11(ii11,J11) =
angle(f_vrf211(ii11,J11));
                    end
```

```
                    vrf_opt11(ii11,J11)     =
exp(-1i*(teta_opt11(ii11,J11))); %INI MENCARI
VRF_OPT -- STEP 8
                    v_rf11(ii11,J11) =
vrf_opt11(ii11,J11);
            end
            end
            selisih11 = abs(argmin211(ii11,J11)
- argmin11(ii11,J11));
                if selisih11 <= 0.3
                cnt11 = cnt11 + 1;
                break
                end

            if isnan(A11)
            variancekanal11 = 1;
            h_km11  =
sqrt(variancekanal11/2)*(randn([mmm],[k])+((1i)*
randn([mmm],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km11  = h_km11;
            g_kmt11 = transpose(g_km11);
            Pp11                = eye(k);
%step 2
            beta_kp11           = ones([1, 1]);
            v_rf11          = randi([1], N_tx3,
N_rf); %step 1
            end

            cnt11 = cnt11+1;

        end
        cnt11 = cnt11 + 1;
        Pmiring11 = 0;
        for k211 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d11            = vrf_opt11'
* g_kmt11' * (g_kmt11 * vrf_opt11 * vrf_opt11' *
g_kmt11')^-1;
```

```
                Qbar11                  = vbar_d11'
* vrf_opt11' * vrf_opt11 * vbar_d11;
                M111                    = vrf_opt11'
* g_kmt11' * g_kmt11 * vrf_opt11; %UNTUK
MENGHITUNG EIGENVALUE VRF*KANAL
                lambda11                = svd(M111);
                qkk11                   =
Qbar11(k211,k211);
                for k311 = 1:1:size(M111,1)
                    PK111(k311) =
beta_kp11/lambda11(k311) - qkk11 * variance;
                    if PK111(k311) >= 0 &&
PK111(k311) <= 10
                        PK211 = PK111(k311);
                    elseif PK111(k311) <= 0
                        PK211 = 0;
                    end
                end
                Pmiring11               = Pmiring11
+ PK211;
                pk11                    = 1/qkk11 *
PK211;
                Pp11(k211,k211)           = pk11;
                PPP11                   = beta_kp11
* log2(1+ (pk11/variance));
                Rk11(k211)               = log2(1+
(sqrt(pk11)/variance));
                if Pmiring11 == 0
                    Pmiring11 = 1;
                end
            end
            if  trace(Qbar11*Pp11) <= P &&
(log10(Pmiring11)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt211 = cnt211+1;
            break
            end
       cnt211 = cnt211+1;
    end
```

```
    VD11                = vbar_d11 *
(Pp11)^(1/2);
    sum111(SNRsize,n)   = sum(Rk11);
    SE_IRF181          = real(sum111);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    jumlah_signal11 = 1000;
    for stream11 = 1:1:jumlah_signal11
        jumlah_data11           = Ns;
        genbpsk11(:,stream11)    =
rand(jumlah_data11, 1)>0.5;
        sigbpsk11(:,stream11)    =
2*genbpsk11(:,stream11) -1;
        y11(:,stream11)          = g_kmt11 *
vrf_opt11 * VD11 * sigbpsk11(:,stream11);
        detektor11              = real(y11)>0;
        for sinyal211= 1:Ns
            if detektor11(sinyal211,stream11)
==1
                sinyal_rx11(sinyal211, stream11)
= 1;
            else sinyal_rx11(sinyal211,stream11)
= -1;
            end
        end
    end
    nErr11(n,SNRsize) = size(find([detektor11-
genbpsk11]),1);
    nErr11(nErr11==0) = 1;

    end
end

simBER     = nErr/(jumlah_signal*Ns);
simBER32 = nErr1/(jumlah_signal1*Ns);
simBER81 = nErr11/(jumlah_signal11*Ns);

figure(1)
```

```
plot(SNR_dB, mean(SE_IRF132,2), 'g.-'); hold on;
plot(SNR_dB, mean(SE_IRF1,2), 'b.-'); hold on;
plot(SNR_dB, mean(SE_IRF181,2), 'r.-'); hold on;
legend('Tx32','Tx64','Tx81');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR kanal IRF
dengan beda TX');

figure(2)
semilogy(SNR_dB,mean(simBER32,1),'gx-'); hold
on;
semilogy(SNR_dB,mean(simBER,1),'bx-'); hold on;
semilogy(SNR_dB,mean(simBER81,1),'rx-'); hold
on;
axis([-10 30 10^-5 0.5])
grid on
legend('Tx32','Tx64','Tx81');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
kanal IRF dengan beda Tx');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATLAB SCRIPT PERBANDINGAN BEDA JUMLAH
PEMANCAR(TX) URLOS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear all;
close all;
clc;

m = 64; k = 4; %JUMLAH ANTENA
mm= 32;
mmm=64;
P               = 1; %Daya dalam Watt
s               = 1; %menurut analisa, WAJIB
[Jml Ns == Jml m];
k_user          = k;
Ns              = s*k_user;
N_tx            = m;
N_tx2           = mm;
N_tx3           = mmm;
N_rf            = Ns; %untuk syarat Hybrid
Beamforming

SNR_dB  = -2:4:30; %range SNR in dB
SNR_lin = 10.^(SNR_dB/10); % range SNR in linear
power
N =9;

for SNRsize = 1 : length(SNR_dB)
    SNR = SNR_lin(SNRsize);
    variance = P/SNR;
    for n = 1:N
    g_km = [];
    thet    = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12
    g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g2      = 2*g-1; %pembangkitan bpsk
    thett   = (g2.*thet);
```

```
    beta_k  = 1;

    psi          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it           = randi([0 1],k,1)>=0.5;
    itb          = (it*2)-1;
    psii         = (itb.*psi);

    for kurlos       = 1:1:k
    for murlos       = 1:1:m
        theta_k(murlos,kurlos)  = thett +
(5*(kurlos-1)); %selisih 0.5 derajat antar
antena
        ekurung(murlos,kurlos)  = ((1*exp((-
1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
    end
    ekurungt                    =
transpose(ekurung);
    g_km2(:,kurlos)             = (sqrt(beta_k)
.* exp(1i*psii(kurlos))) * ekurungt(kurlos,:);
    end
    g_kmt       = transpose(g_km2);

    Pp               = eye(k);       %step 2
    beta_kp          = ones([1, 1]);

    v_rf         = randi([1], N_tx, N_rf); %step
1
    v_rf1        = randi([1], N_tx, N_rf);
    v_rf2        = randi([1], N_tx, N_rf);
    vrf_opt      = randi([1], N_tx, N_rf);
    argmin       = randi([1], N_tx, N_rf)*5;
    selisih      = 1;
    temp_topt    = randi([1], N_tx, N_rf)*20;
    teta_opt     = ones(N_tx, N_rf);
    teta_opt2    = ones(N_tx, N_rf);
```

```
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp    = 0;
            temp2   = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp            = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
                    end
                    end
                    temporary1(ii,J)    = temp;
                    temporary2(ii,J)   = temp2;
                    si_ijB(ii,J)    = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3   = 0;
                    temp4   = 0;
                    for L = 1:N_rf
```

```
                    if L~= ii
                       v_rf2(L,J)        =
v_rf(L,J);
                       temp3             =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4             =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)       =  temp3;
                    n_ijD(ii,J)       =  temp4;
                    c_ij(ii,J)         =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)         =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    vrf_ij1(ii,J)     = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)     = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
```

```
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)     =
argmin(ii,J);
                    argmin(ii,J)      =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)     = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end

            if isnan(A)
            variancekanal = 1;
            h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km  = h_km;
            g_kmt = transpose(g_km);
            Pp              = eye(k);
%step 2
            beta_kp         = ones([1, 1]);
```

```
            v_rf        = randi([1], N_tx,
N_rf); %step 1
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
```

```
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_IRF1             = real(sum1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    jumlah_signal = 1000;
    for stream = 1:1:jumlah_signal
        jumlah_data          = Ns;
        genbpsk(:,stream)      =
rand(jumlah_data, 1)>0.5;
        sigbpsk(:,stream)      =
2*genbpsk(:,stream) -1;

        y(:,stream)                      = g_kmt *
vrf_opt * VD * sigbpsk(:,stream);
        detektor                         =
real(y)>0;
        for sinyal2= 1:Ns
            if detektor(sinyal2,stream) ==1
                sinyal_rx(sinyal2, stream) = 1;
            else sinyal_rx(sinyal2,stream) = -1;
            end
        end
    end
```

```
    nErr(n,SNRsize) = size(find([detektor-
genbpsk]),1);
    nErr(nErr==0) = 1;
    end

    for n2 = 1:N
    g_km1 = [];
    thet1    = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12 derajat
    g1       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g21      = 2*g1-1; %pembangkitan bpsk
    thett1   = (g21.*thet1);
    beta_k1  = 1;
    psi1          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it1           = randi([0 1],k,1)>=0.5;
    itb1          = (it1*2)-1;
    psii1         = (itb1.*psi1);

    for kurlos1       = 1:1:k
    for murlos1       = 1:1:mm
        theta_k1(murlos1,kurlos1)  = thett1 +
(5*(kurlos1-1)); %selisih 0.5 derajat antar
antena
        ekurung1(murlos1,kurlos1)  = ((1*exp((-
1i)*(murlos1-
1)*pi*sind(theta_k1(murlos1,kurlos1)))));
    end
    ekurungt1                      =
transpose(ekurung1);
    g_km21(:,kurlos1)              =
(sqrt(beta_k1) .* exp(1i*psii1(kurlos1))) *
ekurungt1(kurlos1,:);
    end
    g_kmt1       = transpose(g_km21);


    Pp1               = eye(k);       %step 2
```

```
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx2, N_rf);
%step 1
    v_rf11        = randi([1], N_tx2, N_rf);
    v_rf21        = randi([1], N_tx2, N_rf);
    vrf_opt1      = randi([1], N_tx2, N_rf);
    argmin1       = randi([1], N_tx2, N_rf)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx2, N_rf)*20;
    teta_opt1     = ones(N_tx2, N_rf);
    teta_opt21    = ones(N_tx2, N_rf);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf%step 1
            Hbar1          = Pp1^(-1/2) * g_kmt1;
            v_rfj1         = v_rf1;
            v_rfj1(:,J1)    = [];
            A1             = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21            = inv(A1) * inv(A1);
            A11            = inv(A1);
            B1             = Hbar1' * A21 *
Hbar1;
            D1             = Hbar1' * A11 *
Hbar1;
            B21            = Hbar1' * A21 *
Hbar1;
            D21            = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx2
            temp1     = 0;
            temp21    = 0;
                    for K1= 1:1:N_tx2
                    for M1 = 1:1:N_tx2
                        if K1 ~= ii1 && M1 ~=
ii1
```

```
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1           = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21          =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
                    for L1 = 1:N_rf
                    if L1~= ii1
                       v_rf21(L1,J1)     =
v_rf1(L1,J1);
                        temp31           =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                        temp41           =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)        =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
```

```
                    z_ij1(ii1,J1)         =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)   =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)   = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)     = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)     = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    vrf_ij11(ii1,J1)      = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)      = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)       = (mm *
trace(inv(A1))) - (mm * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)       = (mm *
trace(inv(A1))) - (mm * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
```

```
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)     = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            variancekanal1 = 1;
            h_km1  =
sqrt(variancekanal1/2)*(randn([mm],[k])+((1i)*ra
ndn([mm],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km1  = h_km1;
            g_kmt1 = transpose(g_km1);
            Pp1              = eye(k);
%step 2
            beta_kp1         = ones([1, 1]);
            v_rf1        = randi([1], N_tx2,
N_rf); %step 1
            end

            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
```

```
        Pmiring1 = 0;
        for k21 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1             = Pmiring1 +
PK21;
                pk1                  = 1/qkk1 *
PK21;
                Pp1(k21,k21)         = pk1;
                PPP1                 = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)             = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
```

```
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
       cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_IRF132          = real(sum11);
%%%
```

```
    thet11     = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12 derajat
    g11        = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g211       = 2*g11-1; %pembangkitan bpsk
    thett11    = (g211.*thet11);
    beta_k11   = 1;
    psi11          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it11           = randi([0 1],k,1)>=0.5;
    itb11          = (it11*2)-1;
    psii11         = (itb11.*psi11);

    for kurlos11       = 1:1:k
    for murlos11       = 1:1:mmm
        theta_k11(murlos11,kurlos11)  = thett11
+ (5*(kurlos11-1)); %selisih 0.5 derajat antar
antena
        ekurung11(murlos11,kurlos11)  =
((1*exp((-1i)*(murlos11-
1)*pi*sind(theta_k11(murlos11,kurlos11)))));
    end
    ekurungt11                      =
transpose(ekurung11);
    g_km211(:,kurlos11)             =
(sqrt(beta_k11) .* exp(1i*psii11(kurlos11))) *
ekurungt11(kurlos11,:);
    end
    g_kmt11       = transpose(g_km211);


    Pp11               = eye(k);        %step 2
    beta_kp11          = ones([1, 1]);

    v_rf11         = randi([1], N_tx3, N_rf);
%step 1
    v_rf111        = randi([1], N_tx3, N_rf);
    v_rf211        = randi([1], N_tx3, N_rf);
    vrf_opt11      = randi([1], N_tx3, N_rf);
```

```
    argmin11      = randi([1], N_tx3, N_rf)*5;
    selisih11     = 1;
    temp_topt11   = randi([1], N_tx3, N_rf)*20;
    teta_opt11    = ones(N_tx3, N_rf);
    teta_opt211   = ones(N_tx3, N_rf);
    cnt211 = 0;
    while 1
    cnt11 = 0;
        while 1
            for J11 = 1:1:N_rf%step 1
            Hbar11          = Pp11^(-1/2) *
g_kmt11;
            v_rfj11         = v_rf11;
            v_rfj11(:,J11)    = [];
            A11             = Pp11^(-1/2) *
g_kmt11 * v_rfj11 * v_rfj11' * g_kmt11' *
Pp11^(-1/2);
            A211            = inv(A11) *
inv(A11);
            A111            = inv(A11);
            B11             = Hbar11' * A211 *
Hbar11;
            D11             = Hbar11' * A111 *
Hbar11;
            B211            = Hbar11' * A211 *
Hbar11;
            D211            = Hbar11' * A111 *
Hbar11;
            for ii11 = 1:1:N_tx3
            temp11     = 0;
            temp211    = 0;
                    for K11= 1:1:N_tx3
                    for M11 = 1:1:N_tx3
                        if K11 ~= ii11 && M11 ~=
ii11
                        v_rf111(K11,J11) =
v_rf11(K11,J11);
                        v_rf111(M11,J11) =
v_rf11(M11,J11);
```

```
                        temp11            =
temp11 + (conj(v_rf111(K11,J11)) * B11(K11,M11)
* v_rf111(M11,J11));
                        temp211           =
temp211+ (conj(v_rf111(K11,J11)) * D11(K11,M11)
* v_rf111(M11,J11));
                        end
                    end
                    end
                    temporary111(ii11,J11)    =
temp11;
                    temporary211(ii11,J11)    =
temp211;
                    si_ijB11(ii11,J11)    =
B11(ii11,ii11) + 2*real(temporary111(ii11,J11));
                    si_ijD11(ii11,J11)    =
D11(ii11,ii11) + 2*real(temporary211(ii11,J11));
                    temp311   = 0;
                    temp411   = 0;
                    for L11 = 1:N_rf
                    if L11~= ii11
                       v_rf211(L11,J11)     =
v_rf11(L11,J11);
                       temp311            =
temp311 + (B211(ii11,L11) * v_rf211(L11,J11));
                       temp411            =
temp411 + (D211(ii11,L11) * v_rf211(L11,J11));
                    end
                    end
                    n_ijB11(ii11,J11)      =
temp311;
                    n_ijD11(ii11,J11)      =
temp411;
                    c_ij11(ii11,J11)       =
((1+si_ijD11(ii11,J11))*n_ijB11(ii11,J11)) -
(si_ijB11(ii11,J11) * n_ijD11(ii11,J11));
                    z_ij11(ii11,J11)        =
imag((2*n_ijB11(ii11,J11)) * n_ijD11(ii11,J11));
                    if real(c_ij11(ii11,J11)) >=
0
```

```
                        psi_ij11(ii11,J11)  =
asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11)
));
                    elseif
real(c_ij11(ii11,J11)) < 0
                        psi_ij11(ii11,J11)  = pi
-
(asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11
))));
                    end
                    teta_ij111(ii11,J11)    =
(((-1)*psi_ij11(ii11,J11)) +
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    teta_ij211(ii11,J11)    =
(pi + ((-1)*psi_ij11(ii11,J11)) -
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    vrf_ij111(ii11,J11)     =
exp(-1i*angle(teta_ij111(ii11,J11)));
                    vrf_ij211(ii11,J11)     =
exp(-1i*angle(teta_ij211(ii11,J11)));
                    f_vrf111(ii11,J11)      =
(mmm * trace(inv(A11))) - (mmm * (
(si_ijB11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    f_vrf211(ii11,J11)      =
(mmm * trace(inv(A11))) - (mmm * (
(si_ijB11(ii11,J11) +
2*real(conj(vrf_ij211(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij211(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    argmin211(ii11,J11)     =
argmin11(ii11,J11);
                    argmin11(ii11,J11)      =
min(angle(f_vrf111(ii11,J11)),angle(f_vrf211(ii1
1,J11)));
```

```
                    if argmin11(ii11,J11) ==
f_vrf111(ii11,J11) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii11,J11) =
angle(f_vrf111(ii11,J11));
                    elseif argmin11(ii11,J11) ~=
f_vrf111(ii11,J11)
                        teta_opt11(ii11,J11) =
angle(f_vrf211(ii11,J11));
                    end
                    vrf_opt11(ii11,J11)     =
exp(-1i*(teta_opt11(ii11,J11))); %INI MENCARI
VRF_OPT -- STEP 8
                    v_rf11(ii11,J11) =
vrf_opt11(ii11,J11);
            end
            end
            selisih11 = abs(argmin211(ii11,J11)
- argmin11(ii11,J11));
                if selisih11 <= 0.3
                cnt11 = cnt11 + 1;
                break
                end

            if isnan(A11)
            variancekanal11 = 1;
            h_km11  =
sqrt(variancekanal11/2)*(randn([mmm],[k])+((1i)*
randn([mmm],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km11  = h_km11;
            g_kmt11 = transpose(g_km11);
            Pp11              = eye(k);
%step 2
            beta_kp11         = ones([1, 1]);
            v_rf11        = randi([1], N_tx3,
N_rf); %step 1
            end

            cnt11 = cnt11+1;
```

```
        end
        cnt11 = cnt11 + 1;
        Pmiring11 = 0;
        for k211 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d11            = vrf_opt11'
* g_kmt11' * (g_kmt11 * vrf_opt11 * vrf_opt11' *
g_kmt11')^-1;
                Qbar11              = vbar_d11'
* vrf_opt11' * vrf_opt11 * vbar_d11;
                M111                = vrf_opt11'
* g_kmt11' * g_kmt11 * vrf_opt11; %UNTUK
MENGHITUNG EIGENVALUE VRF*KANAL
                lambda11            = svd(M111);
                qkk11               =
Qbar11(k211,k211);
                for k311 = 1:1:size(M111,1)
                    PK111(k311) =
beta_kp11/lambda11(k311) - qkk11 * variance;
                    if PK111(k311) >= 0 &&
PK111(k311) <= 10
                        PK211 = PK111(k311);
                    elseif PK111(k311) <= 0
                        PK211 = 0;
                    end
                end
                Pmiring11           = Pmiring11
+ PK211;
                pk11                = 1/qkk11 *
PK211;
                Pp11(k211,k211)       = pk11;
                PPP11               = beta_kp11
* log2(1+ (pk11/variance));
                Rk11(k211)           = log2(1+
(sqrt(pk11)/variance));
                if Pmiring11 == 0
                    Pmiring11 = 1;
                end
            end
```

```
            if  trace(Qbar11*Pp11) <= P &&
(log10(Pmiring11)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt211 = cnt211+1;
            break
            end
       cnt211 = cnt211+1;
    end
    VD11                = vbar_d11 *
(Pp11)^(1/2);
    sum111(SNRsize,n)   = sum(Rk11);
    SE_IRF181          = real(sum111);
%%%
```

```
    end
end

simBERurlos     = nErr/(jumlah_signal*Ns);
simBERurlos32 = nErr1/(jumlah_signal1*Ns);
simBERurlos81 = nErr11/(jumlah_signal11*Ns);

figure(1)
plot(SNR_dB, mean(SE_URLOS32,2), 'g.-'); hold
on;
plot(SNR_dB, mean(SE_URLOS,2), 'b.-'); hold on;
plot(SNR_dB, mean(SE_URLOS81,2), 'r.-'); hold
on;
legend('Tx32','Tx64','Tx81');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency versus SNR kanal
URLOS beda Tx');

figure(2)
semilogy(SNR_dB,mean(simBERurlos32,1),'gx-');
hold on;
semilogy(SNR_dB,mean(simBERurlos,1),'bx-'); hold
on;
semilogy(SNR_dB,mean(simBERurlos81,1),'rx-');
hold on;
axis([-10 30 10^-5 0.5])
grid on
legend('Tx32','Tx64','Tx81');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation kanal URLOS beda Tx');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATLAB SCRIPT PERBANDINGAN BEDA JUMLAH USER IRF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
close all;
clc;

m = 64; k = 4; %JUMLAH ANTENA
kk= 2;
kkk=6;
P               = 1; %Daya dalam Watt
s               = 1; %menurut analisa, WAJIB
[Jml Ns == Jml m];
k_user          = k;
Ns              = s*k_user;
Ns2             = s*kk;
Ns3             = s*kkk;
N_tx            = m;
N_rf            = Ns; %untuk syarat Hybrid
Beamforming
N_rf2           = Ns2;
N_rf3           = Ns3;

SNR_dB  = -2:4:30; %range SNR in dB
SNR_lin = 10.^(SNR_dB/10); % range SNR in linear
power
N =9;

for SNRsize = 1 : length(SNR_dB)
    SNR = SNR_lin(SNRsize);
    variance = P/SNR;
    for n = 1:N
    variancekanal = 1;
    h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
    g_km  = h_km;
    g_kmt = transpose(g_km);
```

```
    Pp                = eye(k);        %step 2
    beta_kp           = ones([1, 1]);

    v_rf         = randi([1], N_tx, N_rf); %step
1
    v_rf1        = randi([1], N_tx, N_rf);
    v_rf2        = randi([1], N_tx, N_rf);
    vrf_opt      = randi([1], N_tx, N_rf);
    argmin       = randi([1], N_tx, N_rf)*5;
    selisih      = 1;
    temp_topt    = randi([1], N_tx, N_rf)*20;
    teta_opt     = ones(N_tx, N_rf);
    teta_opt2    = ones(N_tx, N_rf);
    cnt2 = 0;
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp     = 0;
            temp2    = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp             = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
```

```
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
                    end
                    end
                    temporary1(ii,J)    = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)    = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)    = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3   = 0;
                    temp4   = 0;
                    for L = 1:N_rf
                    if L~= ii
                       v_rf2(L,J)       =
v_rf(L,J);
                       temp3            =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                       temp4            =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)      =  temp3;
                    n_ijD(ii,J)      =  temp4;
                    c_ij(ii,J)       =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)       =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
```

```
                    teta_ij2(ii,J)     = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    vrf_ij1(ii,J)      = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)      = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)       = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)      =
argmin(ii,J);
                    argmin(ii,J)       =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)      = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
```

```
                break
                end

            if isnan(A)
            variancekanal = 1;
            h_km  =
sqrt(variancekanal/2)*(randn([m],[k])+((1i)*rand
n([m],[k]))); %small scale fading berdasar
Jumlah antena dan user
            g_km  = h_km;
            g_kmt = transpose(g_km);
            Pp              = eye(k);
%step 2
            beta_kp         = ones([1, 1]);
            v_rf        = randi([1], N_tx,
N_rf); %step 1
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
```

```
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
                    end
                end
                Pmiring           = Pmiring +
PK2;
                pk                = 1/qkk * PK2;
                Pp(k2,k2)         = pk;
                PPP               = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)            = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)   = sum(Rk);
    SE_IRF1           = real(sum1);
%%%
```

```
        sigbpsk(:,stream)     =
2*genbpsk(:,stream) -1;

        y(:,stream)                    = g_kmt *
vrf_opt * VD * sigbpsk(:,stream);
        detektor                       =
real(y)>0;
        for sinyal2= 1:Ns
            if detektor(sinyal2,stream) ==1
                sinyal_rx(sinyal2, stream) = 1;
            else sinyal_rx(sinyal2,stream) = -1;
            end
        end
    end
    nErr(n,SNRsize) = size(find([detektor-
genbpsk]),1);
    nErr(nErr==0) = 1;
    end

    for n2 = 1:N
    variancekanal1 = 1;
    h_km1  =
sqrt(variancekanal1/2)*(randn([m],[kk])+((1i)*ra
ndn([m],[kk]))); %small scale fading berdasar
Jumlah antena dan user
    g_km1  = h_km1;
    g_kmt1 = transpose(g_km1);
    Pp1               = eye(kk);       %step 2
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx, N_rf2);
%step 1
    v_rf11        = randi([1], N_tx, N_rf2);
    v_rf21        = randi([1], N_tx, N_rf2);
    vrf_opt1      = randi([1], N_tx, N_rf2);
    argmin1       = randi([1], N_tx, N_rf2)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf2)*20;
    teta_opt1     = ones(N_tx, N_rf2);
    teta_opt21    = ones(N_tx, N_rf2);
```

```
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf2%step 1
            Hbar1          = Pp1^(-1/2) * g_kmt1;
            v_rfj1         = v_rf1;
            v_rfj1(:,J1)    = [];
            A1             = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21            = inv(A1) * inv(A1);
            A11            = inv(A1);
            B1             = Hbar1' * A21 *
Hbar1;
            D1             = Hbar1' * A11 *
Hbar1;
            B21            = Hbar1' * A21 *
Hbar1;
            D21            = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1     = 0;
            temp21    = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
```

```
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
                    for L1 = 1:N_rf2
                    if L1~= ii1
                       v_rf21(L1,J1)     =
v_rf1(L1,J1);
                       temp31            =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                       temp41            =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)        =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)        =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
```

```
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    vrf_ij11(ii1,J1)     = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)     = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)      = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)      = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)     =
argmin1(ii1,J1);
                    argmin1(ii1,J1)      =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)     = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
```

```
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            variancekanal1 = 1;
            h_km1  =
sqrt(variancekanal1/2)*(randn([m],[kk])+((1i)*ra
ndn([m],[kk]))); %small scale fading berdasar
Jumlah antena dan user
            g_km1  = h_km1;
            g_kmt1 = transpose(g_km1);
            Pp1              = eye(kk);
%step 2
            beta_kp1         = ones([1, 1]);
            v_rf1        = randi([1], N_tx,
N_rf2); %step 1
            end

            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:kk
        %   STEP 12 MEMBUAT P
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
```

```
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1            = Pmiring1 +
PK21;
                pk1                 = 1/qkk1 *
PK21;
                Pp1(k21,k21)        = pk1;
                PPP1                = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)            = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
       cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_IRF1_u2        = real(sum11);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    jumlah_signal1 = 1000;
```

```
    for stream1 = 1:1:jumlah_signal1
        jumlah_data1            = Ns2;
        genbpsk1(:,stream1)     =
rand(jumlah_data1, 1)>0.5;
        sigbpsk1(:,stream1)     =
2*genbpsk1(:,stream1) -1;
        y1(:,stream1)           = g_kmt1 *
vrf_opt1 * VD1 * sigbpsk1(:,stream1);
        detektor1               = real(y1)>0;
        for sinyal21= 1:Ns2
            if detektor1(sinyal21,stream1) ==1
                sinyal_rx1(sinyal21, stream1) =
1;
            else sinyal_rx1(sinyal21,stream1) =
-1;
            end
        end
    end
    nErr1(n,SNRsize) = size(find([detektor1-
genbpsk1]),1);
    nErr1(nErr1==0) = 1;
    end

    for n11 = 1:N
    variancekanal11 = 1;
    h_km11  =
sqrt(variancekanal11/2)*(randn([m],[kkk])+((1i)*
randn([m],[kkk]))); %small scale fading berdasar
Jumlah antena dan user
    g_km11  = h_km11;
    g_kmt11 = transpose(g_km11);
    Pp11              = eye(kkk);       %step 2
    beta_kp11         = ones([1, 1]);

    v_rf11        = randi([1], N_tx, N_rf3);
%step 1
    v_rf111       = randi([1], N_tx, N_rf3);
    v_rf211       = randi([1], N_tx, N_rf3);
    vrf_opt11     = randi([1], N_tx, N_rf3);
    argmin11      = randi([1], N_tx, N_rf3)*5;
```

```
    selisih11      = 1;
    temp_topt11    = randi([1], N_tx, N_rf3)*20;
    teta_opt11     = ones(N_tx, N_rf3);
    teta_opt211    = ones(N_tx, N_rf3);
    cnt211 = 0;
    while 1
    cnt11 = 0;
        while 1
            for J11 = 1:1:N_rf3%step 1
            Hbar11          = Pp11^(-1/2) *
g_kmt11;
            v_rfj11         = v_rf11;
            v_rfj11(:,J11)    = [];
            A11             = Pp11^(-1/2) *
g_kmt11 * v_rfj11 * v_rfj11' * g_kmt11' *
Pp11^(-1/2);
            A211            = inv(A11) *
inv(A11);
            A111            = inv(A11);
            B11             = Hbar11' * A211 *
Hbar11;
            D11             = Hbar11' * A111 *
Hbar11;
            B211            = Hbar11' * A211 *
Hbar11;
            D211            = Hbar11' * A111 *
Hbar11;
            for ii11 = 1:1:N_tx
            temp11     = 0;
            temp211    = 0;
                    for K11= 1:1:N_tx
                    for M11 = 1:1:N_tx
                        if K11 ~= ii11 && M11 ~=
ii11
                        v_rf111(K11,J11) =
v_rf11(K11,J11);
                        v_rf111(M11,J11) =
v_rf11(M11,J11);
```

```
                        temp11            =
temp11 + (conj(v_rf111(K11,J11)) * B11(K11,M11)
* v_rf111(M11,J11));
                        temp211           =
temp211+ (conj(v_rf111(K11,J11)) * D11(K11,M11)
* v_rf111(M11,J11));
                        end
                    end
                    end
                    temporary111(ii11,J11)    =
temp11;
                    temporary211(ii11,J11)    =
temp211;
                    si_ijB11(ii11,J11)    =
B11(ii11,ii11) + 2*real(temporary111(ii11,J11));
                    si_ijD11(ii11,J11)    =
D11(ii11,ii11) + 2*real(temporary211(ii11,J11));
                    temp311   = 0;
                    temp411   = 0;
                    for L11 = 1:N_rf3
                    if L11~= ii11
                       v_rf211(L11,J11)     =
v_rf11(L11,J11);
                       temp311            =
temp311 + (B211(ii11,L11) * v_rf211(L11,J11));
                       temp411            =
temp411 + (D211(ii11,L11) * v_rf211(L11,J11));
                    end
                    end
                    n_ijB11(ii11,J11)      =
temp311;
                    n_ijD11(ii11,J11)      =
temp411;
                    c_ij11(ii11,J11)       =
((1+si_ijD11(ii11,J11))*n_ijB11(ii11,J11)) -
(si_ijB11(ii11,J11) * n_ijD11(ii11,J11));
                    z_ij11(ii11,J11)        =
imag((2*n_ijB11(ii11,J11)) * n_ijD11(ii11,J11));
                    if real(c_ij11(ii11,J11)) >=
0
```

```
                        psi_ij11(ii11,J11)  =
asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11)
));
                    elseif
real(c_ij11(ii11,J11)) < 0
                        psi_ij11(ii11,J11)  = pi
-
(asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11
))));
                    end
                    teta_ij111(ii11,J11)    =
(((-1)*psi_ij11(ii11,J11)) +
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    teta_ij211(ii11,J11)    =
(pi + ((-1)*psi_ij11(ii11,J11)) -
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    vrf_ij111(ii11,J11)     =
exp(-1i*angle(teta_ij111(ii11,J11)));
                    vrf_ij211(ii11,J11)     =
exp(-1i*angle(teta_ij211(ii11,J11)));
                    f_vrf111(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij111(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    f_vrf211(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij211(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij211(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    argmin211(ii11,J11)     =
argmin11(ii11,J11);
                    argmin11(ii11,J11)      =
min(angle(f_vrf111(ii11,J11)),angle(f_vrf211(ii1
1,J11)));
                    if argmin11(ii11,J11) ==
f_vrf111(ii11,J11) %% INI MENCARI TETA OPTIMUM
```

```
                        teta_opt1(ii11,J11) =
angle(f_vrf111(ii11,J11));
                    elseif argmin11(ii11,J11) ~=
f_vrf111(ii11,J11)
                        teta_opt11(ii11,J11) =
angle(f_vrf211(ii11,J11));
                    end
                    vrf_opt11(ii11,J11)     =
exp(-1i*(teta_opt11(ii11,J11))); %INI MENCARI
VRF_OPT -- STEP 8
                    v_rf11(ii11,J11) =
vrf_opt11(ii11,J11);
            end
            end
            selisih11 = abs(argmin211(ii11,J11)
- argmin11(ii11,J11));
                if selisih11 <= 0.3
                cnt11 = cnt11 + 1;
                break
                end

            if isnan(A11)
            variancekanal11 = 1;
            h_km11  =
sqrt(variancekanal11/2)*(randn([m],[kkk])+((1i)*
randn([m],[kkk]))); %small scale fading berdasar
Jumlah antena dan user
            g_km11  = h_km11;
            g_kmt11 = transpose(g_km11);
            Pp11              = eye(kkk);
%step 2
            beta_kp11         = ones([1, 1]);
            v_rf11        = randi([1], N_tx,
N_rf3); %step 1
            end

            cnt11 = cnt11+1;

        end
        cnt11 = cnt11 + 1;
```

```
        Pmiring11 = 0;
        for k211 = 1:kkk
        %   STEP 12 MEMBUAT P
                vbar_d11            = vrf_opt11'
* g_kmt11' * (g_kmt11 * vrf_opt11 * vrf_opt11' *
g_kmt11')^-1;
                Qbar11              = vbar_d11'
* vrf_opt11' * vrf_opt11 * vbar_d11;
                M111                = vrf_opt11'
* g_kmt11' * g_kmt11 * vrf_opt11; %UNTUK
MENGHITUNG EIGENVALUE VRF*KANAL
                lambda11            = svd(M111);
                qkk11               =
Qbar11(k211,k211);
                for k311 = 1:1:size(M111,1)
                    PK111(k311) =
beta_kp11/lambda11(k311) - qkk11 * variance;
                    if PK111(k311) >= 0 &&
PK111(k311) <= 10
                        PK211 = PK111(k311);
                    elseif PK111(k311) <= 0
                        PK211 = 0;
                    end
                end
                Pmiring11           = Pmiring11
+ PK211;
                pk11                = 1/qkk11 *
PK211;
                Pp11(k211,k211)       = pk11;
                PPP11               = beta_kp11
* log2(1+ (pk11/variance));
                Rk11(k211)           = log2(1+
(sqrt(pk11)/variance));
                if Pmiring11 == 0
                    Pmiring11 = 1;
                end
            end
            if  trace(Qbar11*Pp11) <= P &&
(log10(Pmiring11)*10) >= ((log10(P)*10)-0.5)
```

```
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt211 = cnt211+1;
            break
            end
       cnt211 = cnt211+1;
    end
    VD11                = vbar_d11 *
(Pp11)^(1/2);
    sum111(SNRsize,n)   = sum(Rk11);
    SE_IRF1_6         = real(sum111);
%%%
```

```
end

simBER          = nErr/(jumlah_signal*Ns);
simBERirf_u2    = nErr1/(jumlah_signal1*Ns2);
simBER_6        = nErr11/(jumlah_signal11*Ns3);

figure(1)
plot(SNR_dB, mean(SE_IRF1_u2,2), 'g.-'); hold
on;
plot(SNR_dB, mean(SE_IRF1,2), 'b.-'); hold on;
plot(SNR_dB, mean(SE_IRF1_6,2), 'r.-'); hold on;
legend('2 User','4 User','6 User');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR kanal IRF beda
user');

figure(2)
semilogy(SNR_dB,mean(simBERirf_u2,1),'gx-');
hold on;
semilogy(SNR_dB,mean(simBER,1),'bx-'); hold on;
semilogy(SNR_dB,mean(simBER_6,1),'rx-'); hold
on;
axis([-10 30 10^-5 0.5])
grid on
legend('2 User','4 User','6 User');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
kanal IRF beda user');
```

```
%%%
```

```
    thett   = (g2.*thet);
    beta_k  = 1;
    psi          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
    it           = randi([0 1],k,1)>=0.5;
    itb          = (it*2)-1;
    psii         = (itb.*psi);

    for kurlos       = 1:1:k
    for murlos       = 1:1:m
        theta_k(murlos,kurlos)  = thett +
(5*(kurlos-1)); %selisih 0.5 derajat antar
antena
        ekurung(murlos,kurlos)  = ((1*exp((-
1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
    end
    ekurungt                    =
transpose(ekurung);
    g_km2(:,kurlos)             = (sqrt(beta_k)
.* exp(1i*psii(kurlos))) * ekurungt(kurlos,:);
    end
    g_kmt       = transpose(g_km2);


    Pp               = eye(k);       %step 2
    beta_kp          = ones([1, 1]);

    v_rf         = randi([1], N_tx, N_rf); %step
1
    v_rf1        = randi([1], N_tx, N_rf);
    v_rf2        = randi([1], N_tx, N_rf);
    vrf_opt      = randi([1], N_tx, N_rf);
    argmin       = randi([1], N_tx, N_rf)*5;
    selisih      = 1;
    temp_topt    = randi([1], N_tx, N_rf)*20;
    teta_opt     = ones(N_tx, N_rf);
    teta_opt2    = ones(N_tx, N_rf);
    cnt2 = 0;
```

```
    while 1
    cnt = 0;
        while 1
            for J = 1:1:N_rf%step 1
            Hbar          = Pp^(-1/2) * g_kmt;
            v_rfj         = v_rf;
            v_rfj(:,J)    = [];
            A             = Pp^(-1/2) * g_kmt *
v_rfj * v_rfj' * g_kmt' * Pp^(-1/2);
            A2            = inv(A) * inv(A);
            A1            = inv(A);
            B             = Hbar' * A2 * Hbar;
            D             = Hbar' * A1 * Hbar;
            B2            = Hbar' * A2 * Hbar;
            D2            = Hbar' * A1 * Hbar;
            for ii = 1:1:N_tx
            temp     = 0;
            temp2    = 0;
                    for K = 1:1:N_tx
                    for M = 1:1:N_tx
                        if K ~= ii && M ~= ii
                        v_rf1(K,J) = v_rf(K,J);
                        v_rf1(M,J) = v_rf(M,J);
                        temp            = temp +
(conj(v_rf1(K,J)) * B(K,M) * v_rf1(M,J));
                        temp2           = temp2+
(conj(v_rf1(K,J)) * D(K,M) * v_rf1(M,J));
                        end
                    end
                    end
                    temporary1(ii,J)     = temp;
                    temporary2(ii,J)    = temp2;
                    si_ijB(ii,J)     = B(ii,ii) +
2*real(temporary1(ii,J));
                    si_ijD(ii,J)     = D(ii,ii) +
2*real(temporary2(ii,J));
                    temp3    = 0;
                    temp4    = 0;
                    for L = 1:N_rf
                    if L~= ii
```

```
                        v_rf2(L,J)        =
v_rf(L,J);
                        temp3             =  temp3
+ (B2(ii,L) * v_rf2(L,J));
                        temp4             =  temp4
+ (D2(ii,L) * v_rf2(L,J));
                    end
                    end
                    n_ijB(ii,J)       =  temp3;
                    n_ijD(ii,J)       =  temp4;
                    c_ij(ii,J)         =
((1+si_ijD(ii,J))*n_ijB(ii,J)) - (si_ijB(ii,J) *
n_ijD(ii,J));
                    z_ij(ii,J)         =
imag((2*n_ijB(ii,J)) * n_ijD(ii,J));
                    if real(c_ij(ii,J)) >= 0
                        psi_ij(ii,J)  =
asin(imag(c_ij(ii,J))/abs(c_ij(ii,J)));
                    elseif real(c_ij(ii,J)) < 0
                        psi_ij(ii,J)  = pi -
(asin(imag(c_ij(ii,J))/abs(c_ij(ii,J))));
                    end
                    teta_ij1(ii,J)    = (((-
1)*psi_ij(ii,J)) +
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    teta_ij2(ii,J)    = (pi +
((-1)*psi_ij(ii,J)) -
asin(z_ij(ii,J)/abs(c_ij(ii,J))));
                    vrf_ij1(ii,J)     = exp(-
1i*angle(teta_ij1(ii,J)));
                    vrf_ij2(ii,J)     = exp(-
1i*angle(teta_ij2(ii,J)));
                    f_vrf1(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij1(ii,J)) * n_ijB(ii,J)))/(1 +
si_ijD(ii,J) +
2*real(conj(vrf_ij1(ii,J))*n_ijD(ii,J)))));
                    f_vrf2(ii,J)      = (m *
trace(inv(A))) - (m * ( (si_ijB(ii,J) +
2*real(conj(vrf_ij2(ii,J)) * n_ijB(ii,J)))/(1 +
```

```
si_ijD(ii,J) +
2*real(conj(vrf_ij2(ii,J))*n_ijD(ii,J)))));
                    argmin2(ii,J)     =
argmin(ii,J);
                    argmin(ii,J)      =
min(angle(f_vrf1(ii,J)),angle(f_vrf2(ii,J)));
                    if argmin(ii,J) ==
f_vrf1(ii,J) %% INI MENCARI TETA OPTIMUM
                        teta_opt(ii,J) =
angle(f_vrf1(ii,J));
                    elseif argmin(ii,J) ~=
f_vrf1(ii,J)
                        teta_opt(ii,J) =
angle(f_vrf2(ii,J));
                    end
                    vrf_opt(ii,J)     = exp(-
1i*(teta_opt(ii,J))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf(ii,J) = vrf_opt(ii,J);
            end
            end
            selisih = abs(argmin2(ii,J) -
argmin(ii,J));
                if selisih <= 0.3
                cnt = cnt + 1;
                break
                end

            if isnan(A)
            g_km = [];
            thet    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g2      = 2*g-1; %pembangkitan bpsk
            thett   = (g2.*thet11);
            beta_k  = 1;
```

```
            psi          = -pi + (pi - (-
pi).*randn(k,1)); %pembangkitan psi dari -pi
sampai pi
            it           = randi([0 1],k,1)>=0.5;
            itb          = (it*2)-1;
            psii         = (itb.*psi);

            for kurlos       = 1:1:k
            for murlos       = 1:1:m
                theta_k(murlos,kurlos)  =
thett11 + (5*(kurlos-1)); %selisih 0.5 derajat
antar antena
                ekurung(murlos,kurlos11)  =
((1*exp((-1i)*(murlos-
1)*pi*sind(theta_k(murlos,kurlos)))));
            end
            ekurungt                      =
transpose(ekurung);
            g_km2(:,kurlos)             =
(sqrt(beta_k) .* exp(1i*psii(kurlos))) *
ekurungt(kurlos,:);
            end
            g_kmt       = transpose(g_km2);
            end

            cnt = cnt+1;


        end
        cnt = cnt + 1;
        Pmiring = 0;
        for k2 = 1:k
        %   STEP 12 MEMBUAT P
                vbar_d              = vrf_opt' *
g_kmt' * (g_kmt * vrf_opt * vrf_opt' * g_kmt')^-
1;
                Qbar                = vbar_d' *
vrf_opt' * vrf_opt * vbar_d;
```

```
                M1                  = vrf_opt' *
g_kmt' * g_kmt * vrf_opt; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda              = svd(M1);
                qkk                 = Qbar(k2,k2);
                for k3 = 1:1:size(M1,1)
                    PK1(k3) = beta_kp/lambda(k3)
- qkk * variance;
                    if PK1(k3) >= 0 && PK1(k3)
<= 10
                        PK2 = PK1(k3);
                    elseif PK1(k3) <= 0
                        PK2 = 0;
                    end
                end
                Pmiring             = Pmiring +
PK2;
                pk                  = 1/qkk * PK2;
                Pp(k2,k2)           = pk;
                PPP                 = beta_kp *
log2(1+ (pk/variance));
                Rk(k2)              = log2(1+
(sqrt(pk)/variance));
                if Pmiring == 0
                    Pmiring = 1;
                end
            end
            if  trace(Qbar*Pp) <= P &&
(log10(Pmiring)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt2 = cnt2+1;
            break
            end
       cnt2 = cnt2+1;
    end
    VD                  = vbar_d * (Pp)^(1/2);
    sum1(SNRsize,n)     = sum(Rk);
    SE_URLOS             = real(sum1);
```

```
%%%
```

```
    it1           = randi([0 1],kk,1)>=0.5;
    itb1          = (it1*2)-1;
    psii1         = (itb1.*psi1);

    for kurlos1       = 1:1:kk
    for murlos1       = 1:1:m
        theta_k1(murlos1,kurlos1)  = thett1 +
(5*(kurlos1-1)); %selisih 0.5 derajat antar
antena
        ekurung1(murlos1,kurlos1)  = ((1*exp((-
1i)*(murlos1-
1)*pi*sind(theta_k1(murlos1,kurlos1)))));
    end
    ekurungt1                      =
transpose(ekurung1);
    g_km21(:,kurlos1)              =
(sqrt(beta_k1) .* exp(1i*psii1(kurlos1))) *
ekurungt1(kurlos1,:);
    end
    g_kmt1       = transpose(g_km21);


    Pp1               = eye(kk);        %step 2
    beta_kp1          = ones([1, 1]);

    v_rf1         = randi([1], N_tx, N_rf2);
%step 1
    v_rf11        = randi([1], N_tx, N_rf2);
    v_rf21        = randi([1], N_tx, N_rf2);
    vrf_opt1      = randi([1], N_tx, N_rf2);
    argmin1       = randi([1], N_tx, N_rf2)*5;
    selisih1      = 1;
    temp_topt1    = randi([1], N_tx, N_rf2)*20;
    teta_opt1     = ones(N_tx, N_rf2);
    teta_opt21    = ones(N_tx, N_rf2);
    cnt21 = 0;
    while 1
    cnt1 = 0;
        while 1
            for J1 = 1:1:N_rf2%step 1
```

```
            Hbar1           = Pp1^(-1/2) * g_kmt1;
            v_rfj1          = v_rf1;
            v_rfj1(:,J1)    = [];
            A1              = Pp1^(-1/2) * g_kmt1
* v_rfj1 * v_rfj1' * g_kmt1' * Pp1^(-1/2);
            A21             = inv(A1) * inv(A1);
            A11             = inv(A1);
            B1              = Hbar1' * A21 *
Hbar1;
            D1              = Hbar1' * A11 *
Hbar1;
            B21             = Hbar1' * A21 *
Hbar1;
            D21             = Hbar1' * A11 *
Hbar1;
            for ii1 = 1:1:N_tx
            temp1    = 0;
            temp21   = 0;
                    for K1= 1:1:N_tx
                    for M1 = 1:1:N_tx
                        if K1 ~= ii1 && M1 ~=
ii1
                        v_rf11(K1,J1) =
v_rf1(K1,J1);
                        v_rf11(M1,J1) =
v_rf1(M1,J1);
                        temp1             = temp1
+ (conj(v_rf11(K1,J1)) * B1(K1,M1) *
v_rf11(M1,J1));
                        temp21            =
temp21+ (conj(v_rf11(K1,J1)) * D1(K1,M1) *
v_rf11(M1,J1));
                        end
                    end
                    end
                    temporary11(ii1,J1)    =
temp1;
                    temporary21(ii1,J1)   =
temp21;
```

```
                    si_ijB1(ii1,J1)    =
B1(ii1,ii1) + 2*real(temporary11(ii1,J1));
                    si_ijD1(ii1,J1)    =
D1(ii1,ii1) + 2*real(temporary21(ii1,J1));
                    temp31   = 0;
                    temp41   = 0;
                    for L1 = 1:N_rf2
                    if L1~= ii1
                       v_rf21(L1,J1)     =
v_rf1(L1,J1);
                       temp31            =
temp31 + (B21(ii1,L1) * v_rf21(L1,J1));
                       temp41            =
temp41 + (D21(ii1,L1) * v_rf21(L1,J1));
                    end
                    end
                    n_ijB1(ii1,J1)      =
temp31;
                    n_ijD1(ii1,J1)      =
temp41;
                    c_ij1(ii1,J1)        =
((1+si_ijD1(ii1,J1))*n_ijB1(ii1,J1)) -
(si_ijB1(ii1,J1) * n_ijD1(ii1,J1));
                    z_ij1(ii1,J1)        =
imag((2*n_ijB1(ii1,J1)) * n_ijD1(ii1,J1));
                    if real(c_ij1(ii1,J1)) >= 0
                        psi_ij1(ii1,J1)  =
asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1)));
                    elseif real(c_ij1(ii1,J1)) <
0
                        psi_ij1(ii1,J1)  = pi -
(asin(imag(c_ij1(ii1,J1))/abs(c_ij1(ii1,J1))));
                    end
                    teta_ij11(ii1,J1)    = (((-
1)*psi_ij1(ii1,J1)) +
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
                    teta_ij21(ii1,J1)    = (pi +
((-1)*psi_ij1(ii1,J1)) -
asin(z_ij1(ii1,J1)/abs(c_ij1(ii1,J1))));
```

```
                    vrf_ij11(ii1,J1)      = exp(-
1i*angle(teta_ij11(ii1,J1)));
                    vrf_ij21(ii1,J1)      = exp(-
1i*angle(teta_ij21(ii1,J1)));
                    f_vrf11(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij11(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    f_vrf21(ii1,J1)       = (m *
trace(inv(A1))) - (m * ( (si_ijB1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1)) *
n_ijB1(ii1,J1)))/(1 + si_ijD1(ii1,J1) +
2*real(conj(vrf_ij21(ii1,J1))*n_ijD1(ii1,J1)))))
;
                    argmin21(ii1,J1)      =
argmin1(ii1,J1);
                    argmin1(ii1,J1)       =
min(angle(f_vrf11(ii1,J1)),angle(f_vrf21(ii1,J1)
));
                    if argmin1(ii1,J1) ==
f_vrf11(ii1,J1) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii1,J1) =
angle(f_vrf11(ii1,J1));
                    elseif argmin1(ii1,J1) ~=
f_vrf11(ii1,J1)
                        teta_opt1(ii1,J1) =
angle(f_vrf21(ii1,J1));
                    end
                    vrf_opt1(ii1,J1)      = exp(-
1i*(teta_opt1(ii1,J1))); %INI MENCARI VRF_OPT --
STEP 8
                    v_rf1(ii1,J1) =
vrf_opt1(ii1,J1);
            end
            end
            selisih1 = abs(argmin21(ii1,J1) -
argmin1(ii1,J1));
                if selisih1 <= 0.3
```

```
                cnt1 = cnt1 + 1;
                break
                end

            if isnan(A1)
            g_km1 = [];
            thet1    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g1       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g21      = 2*g1-1; %pembangkitan
bpsk
            thett1   = (g21.*thet1);
            beta_k1  = 1;
            psi1          = -pi + (pi - (-
pi).*randn(kk,1)); %pembangkitan psi dari -pi
sampai pi
            it1           = randi([0
1],kk,1)>=0.5;
            itb1          = (it1*2)-1;
            psii1         = (itb1.*psi1);

            for kurlos1      = 1:1:kk
            for murlos1      = 1:1:m
                theta_k1(murlos1,kurlos1)  =
thett1 + (5*(kurlos1-1)); %selisih 0.5 derajat
antar antena
                ekurung1(murlos1,kurlos1)  =
((1*exp((-1i)*(murlos1-
1)*pi*sind(theta_k1(murlos1,kurlos1)))));
            end
            ekurungt1                      =
transpose(ekurung1);
            g_km21(:,kurlos1)              =
(sqrt(beta_k1) .* exp(1i*psii1(kurlos1))) *
ekurungt1(kurlos1,:);
            end
            g_kmt1      = transpose(g_km21);
            end
```

```
            cnt1 = cnt1+1;

        end
        cnt1 = cnt1 + 1;
        Pmiring1 = 0;
        for k21 = 1:kk
        %   STEP 12 MEMBUAT P
                vbar_d1             = vrf_opt1' *
g_kmt1' * (g_kmt1 * vrf_opt1 * vrf_opt1' *
g_kmt1')^-1;
                Qbar1               = vbar_d1' *
vrf_opt1' * vrf_opt1 * vbar_d1;
                M11                 = vrf_opt1' *
g_kmt1' * g_kmt1 * vrf_opt1; %UNTUK MENGHITUNG
EIGENVALUE VRF*KANAL
                lambda1             = svd(M11);
                qkk1                =
Qbar1(k21,k21);
                for k31 = 1:1:size(M11,1)
                    PK11(k31) =
beta_kp1/lambda1(k31) - qkk1 * variance;
                    if PK11(k31) >= 0 &&
PK11(k31) <= 10
                        PK21 = PK11(k31);
                    elseif PK11(k31) <= 0
                        PK21 = 0;
                    end
                end
                Pmiring1             = Pmiring1 +
PK21;
                pk1                  = 1/qkk1 *
PK21;
                Pp1(k21,k21)         = pk1;
                PPP1                 = beta_kp1 *
log2(1+ (pk1/variance));
                Rk1(k21)             = log2(1+
(sqrt(pk1)/variance));
                if Pmiring1 == 0
                    Pmiring1 = 1;
```

```
                end
            end
            if  trace(Qbar1*Pp1) <= P &&
(log10(Pmiring1)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt21 = cnt21+1;
            break
            end
       cnt21 = cnt21+1;
    end
    VD1                 = vbar_d1 * (Pp1)^(1/2);
    sum11(SNRsize,n)    = sum(Rk1);
    SE_URLOS_u2        = real(sum11);
%%%
```

```
    end

    for n11 = 1:N
    g_km11 = [];
    thet11    = 6 + (0.5 - (-6).*rand(1,1));
%pembangkitan theta dari 6 sampai 12 derajat
    g11       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
    g211      = 2*g11-1; %pembangkitan bpsk
    thett11   = (g211.*thet11);
    beta_k11  = 1;
    psi11          = -pi + (pi - (-
pi).*randn(kkk,1)); %pembangkitan psi dari -pi
sampai pi
    it11           = randi([0 1],kkk,1)>=0.5;
    itb11          = (it11*2)-1;
    psii11         = (itb11.*psi11);

    for kurlos11       = 1:1:kkk
    for murlos11       = 1:1:m
        theta_k11(murlos11,kurlos11)  = thett11
+ (5*(kurlos11-1)); %selisih 0.5 derajat antar
antena
        ekurung11(murlos11,kurlos11)  =
((1*exp((-1i)*(murlos11-
1)*pi*sind(theta_k11(murlos11,kurlos11)))));
    end
    ekurungt11                    =
transpose(ekurung11);
    g_km211(:,kurlos11)            =
(sqrt(beta_k11) .* exp(1i*psii11(kurlos11))) *
ekurungt11(kurlos11,:);
    end
    g_kmt11      = transpose(g_km211);


    Pp11               = eye(kkk);        %step 2
    beta_kp11          = ones([1, 1]);
```

```
    v_rf11          = randi([1], N_tx, N_rf3);
%step 1
    v_rf111         = randi([1], N_tx, N_rf3);
    v_rf211         = randi([1], N_tx, N_rf3);
    vrf_opt11       = randi([1], N_tx, N_rf3);
    argmin11        = randi([1], N_tx, N_rf3)*5;
    selisih11       = 1;
    temp_topt11     = randi([1], N_tx, N_rf3)*20;
    teta_opt11      = ones(N_tx, N_rf3);
    teta_opt211     = ones(N_tx, N_rf3);
    cnt211 = 0;
    while 1
    cnt11 = 0;
        while 1
            for J11 = 1:1:N_rf3%step 1
            Hbar11          = Pp11^(-1/2) *
g_kmt11;
            v_rfj11         = v_rf11;
            v_rfj11(:,J11)    = [];
            A11             = Pp11^(-1/2) *
g_kmt11 * v_rfj11 * v_rfj11' * g_kmt11' *
Pp11^(-1/2);
            A211            = inv(A11) *
inv(A11);
            A111            = inv(A11);
            B11             = Hbar11' * A211 *
Hbar11;
            D11             = Hbar11' * A111 *
Hbar11;
            B211            = Hbar11' * A211 *
Hbar11;
            D211            = Hbar11' * A111 *
Hbar11;
            for ii11 = 1:1:N_tx
            temp11     = 0;
            temp211    = 0;
                    for K11= 1:1:N_tx
                    for M11 = 1:1:N_tx
                        if K11 ~= ii11 && M11 ~=
ii11
```

```
                        v_rf111(K11,J11) =
v_rf11(K11,J11);
                        v_rf111(M11,J11) =
v_rf11(M11,J11);
                        temp11           =
temp11 + (conj(v_rf111(K11,J11)) * B11(K11,M11)
* v_rf111(M11,J11));
                        temp211          =
temp211+ (conj(v_rf111(K11,J11)) * D11(K11,M11)
* v_rf111(M11,J11));
                        end
                    end
                    end
                    temporary111(ii11,J11)    =
temp11;
                    temporary211(ii11,J11)    =
temp211;
                    si_ijB11(ii11,J11)    =
B11(ii11,ii11) + 2*real(temporary111(ii11,J11));
                    si_ijD11(ii11,J11)    =
D11(ii11,ii11) + 2*real(temporary211(ii11,J11));
                    temp311   = 0;
                    temp411   = 0;
                    for L11 = 1:N_rf3
                    if L11~= ii11
                       v_rf211(L11,J11)     =
v_rf11(L11,J11);
                        temp311          =
temp311 + (B211(ii11,L11) * v_rf211(L11,J11));
                        temp411          =
temp411 + (D211(ii11,L11) * v_rf211(L11,J11));
                    end
                    end
                    n_ijB11(ii11,J11)      =
temp311;
                    n_ijD11(ii11,J11)      =
temp411;
                    c_ij11(ii11,J11)       =
((1+si_ijD11(ii11,J11))*n_ijB11(ii11,J11)) -
(si_ijB11(ii11,J11) * n_ijD11(ii11,J11));
```

```
                    z_ij11(ii11,J11)        =
imag((2*n_ijB11(ii11,J11)) * n_ijD11(ii11,J11));
                    if real(c_ij11(ii11,J11)) >=
0
                        psi_ij11(ii11,J11)  =
asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11)
));
                    elseif
real(c_ij11(ii11,J11)) < 0
                        psi_ij11(ii11,J11)  = pi
-
(asin(imag(c_ij11(ii11,J11))/abs(c_ij11(ii11,J11
))));
                    end
                    teta_ij111(ii11,J11)    =
(((-1)*psi_ij11(ii11,J11)) +
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    teta_ij211(ii11,J11)    =
(pi + ((-1)*psi_ij11(ii11,J11)) -
asin(z_ij11(ii11,J11)/abs(c_ij11(ii11,J11))));
                    vrf_ij111(ii11,J11)     =
exp(-1i*angle(teta_ij111(ii11,J11)));
                    vrf_ij211(ii11,J11)     =
exp(-1i*angle(teta_ij211(ii11,J11)));
                    f_vrf111(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij111(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij111(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    f_vrf211(ii11,J11)      = (m
* trace(inv(A11))) - (m * ( (si_ijB11(ii11,J11)
+ 2*real(conj(vrf_ij211(ii11,J11)) *
n_ijB11(ii11,J11)))/(1 + si_ijD11(ii11,J11) +
2*real(conj(vrf_ij211(ii11,J11))*n_ijD11(ii11,J1
1)))));
                    argmin211(ii11,J11)     =
argmin11(ii11,J11);
```

```
                    argmin11(ii11,J11)      =
min(angle(f_vrf111(ii11,J11)),angle(f_vrf211(ii1
1,J11)));
                    if argmin11(ii11,J11) ==
f_vrf111(ii11,J11) %% INI MENCARI TETA OPTIMUM
                        teta_opt1(ii11,J11) =
angle(f_vrf111(ii11,J11));
                    elseif argmin11(ii11,J11) ~=
f_vrf111(ii11,J11)
                        teta_opt11(ii11,J11) =
angle(f_vrf211(ii11,J11));
                    end
                    vrf_opt11(ii11,J11)     =
exp(-1i*(teta_opt11(ii11,J11))); %INI MENCARI
VRF_OPT -- STEP 8
                    v_rf11(ii11,J11) =
vrf_opt11(ii11,J11);
            end
            end
            selisih11 = abs(argmin211(ii11,J11)
- argmin11(ii11,J11));
                if selisih11 <= 0.3
                cnt11 = cnt11 + 1;
                break
                end

            if isnan(A11)
            g_km11 = [];
            thet11    = 6 + (0.5 - (-
6).*rand(1,1));  %pembangkitan theta dari 6
sampai 12 derajat
            g11       = randi([0 1],1,1)>=0.5;
%pembangkitan biner 0 1
            g211      = 2*g11-1; %pembangkitan
bpsk
            thett11   = (g211.*thet11);
            beta_k11  = 1;
            psi11          = -pi + (pi - (-
pi).*randn(kk,1)); %pembangkitan psi dari -pi
sampai pi
```

```
            it11          = randi([0
1],kk,1)>=0.5;
            itb11         = (it11*2)-1;
            psii11        = (itb11.*psi11);

            for kurlos11      = 1:1:kk
            for murlos11      = 1:1:m
                theta_k11(murlos11,kurlos11)  =
thett11 + (5*(kurlos11-1)); %selisih 0.5 derajat
antar antena
                ekurung11(murlos11,kurlos11)  =
((1*exp((-1i)*(murlos11-
1)*pi*sind(theta_k11(murlos11,kurlos11)))));
            end
            ekurungt11                    =
transpose(ekurung11);
            g_km211(:,kurlos11)            =
(sqrt(beta_k11) .* exp(1i*psii11(kurlos11))) *
ekurungt11(kurlos11,:);
            end
            g_kmt11      = transpose(g_km211);
            end

            cnt11 = cnt11+1;

        end
        cnt11 = cnt11 + 1;
        Pmiring11 = 0;
        for k211 = 1:kkk
        %   STEP 12 MEMBUAT P
                vbar_d11            = vrf_opt11'
* g_kmt11' * (g_kmt11 * vrf_opt11 * vrf_opt11' *
g_kmt11')^-1;
                Qbar11              = vbar_d11'
* vrf_opt11' * vrf_opt11 * vbar_d11;
                M111                = vrf_opt11'
* g_kmt11' * g_kmt11 * vrf_opt11; %UNTUK
MENGHITUNG EIGENVALUE VRF*KANAL
                lambda11            = svd(M111);
```

```
                qkk11                =
Qbar11(k211,k211);
                for k311 = 1:1:size(M111,1)
                    PK111(k311) =
beta_kp11/lambda11(k311) - qkk11 * variance;
                    if PK111(k311) >= 0 &&
PK111(k311) <= 10
                        PK211 = PK111(k311);
                    elseif PK111(k311) <= 0
                        PK211 = 0;
                    end
                end
                Pmiring11            = Pmiring11
+ PK211;
                pk11                 = 1/qkk11 *
PK211;
                Pp11(k211,k211)        = pk11;
                PPP11                = beta_kp11
* log2(1+ (pk11/variance));
                Rk11(k211)            = log2(1+
(sqrt(pk11)/variance));
                if Pmiring11 == 0
                    Pmiring11 = 1;
                end
            end
            if  trace(Qbar11*Pp11) <= P &&
(log10(Pmiring11)*10) >= ((log10(P)*10)-0.5)
%((log10(1/variance)*10)-7) %&& trace(Pp) <= k %
&&
            cnt211 = cnt211+1;
            break
            end
       cnt211 = cnt211+1;
    end
    VD11                 = vbar_d11 *
(Pp11)^(1/2);
    sum111(SNRsize,n)    = sum(Rk11);
    SE_URLOS_u6         = real(sum111);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%
```

```
legend('2 User','4 User','6 User');
xlabel('SNR');
ylabel('Spectral Efficiency');
title('Spectral Efficiency vs SNR kanal URLOS
beda user');

figure(2)
semilogy(SNR_dB,mean(simBERurlos_u2,1),'gx-');
hold on;
semilogy(SNR_dB,mean(simBERurlos,1),'bx-'); hold
on;
semilogy(SNR_dB,mean(simBERurlos_u6,1),'rx-');
hold on;
axis([-10 30 10^-5 0.5])
grid on
legend('2 User','4 User','6 User');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK
modulation kanal URLOS beda user');
```

```
%%%
```

```
title('Pola Radiasi BS pada kanal IRF');

figure(2)
plot(polaradiasi_norm)
title('Pola Radiasi BS pada kanal IRF');
xlabel('Sudut');
ylabel('Amplitudo');

%%% PLOT POLA RADIASI PADA INPUT TX
figure(3);
ax = polaraxes;
polarplot(ax,polaradiasi_norm);
ax.ThetaDir = 'clockwise';
ax.ThetaZeroLocation = 'bottom';
title('Pola Radiasi TX pada kanal IRF');
thetalim([90 270]);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATLAB SCRIPT PLOT POLA RADIASI POLAR DAN
RECTANGULAR METODE HB PADA KANAL URLOS 5 derajat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;
clc;
load('u1workspace_5.mat');
m = 64; k=4;
VT = vrf_opt * VD;
antenaURLOS = sum(VT,2);
amplitudo_AFURLOS = abs(antenaURLOS);
fasa_AFURLOS = angle(antenaURLOS);
AF = zeros(m, 360);


for tx_urlos = 1:m
    beta = 2*pi;
    d = 1/2;
    N1 = 64 ;
    j = sqrt(-1);
    An = amplitudo_AFURLOS(tx_urlos,:);
    alpha = fasa_AFURLOS(tx_urlos,:);
        for theta= 1 : 360
            rad(theta) = (theta*pi)/180;
            for n = 1 : N1
            AF(tx_urlos,theta) =
AF(tx_urlos,theta) + An*exp(j*(n-1)*((beta * d *
sin(rad(theta)) ) + alpha));
            end
            AF(tx_urlos,theta) =
abs(AF(tx_urlos,theta))/N1;
        end
end
polaradiasi = sum(AF,1);
polaradiasi_norm = polaradiasi/max(polaradiasi);
polaradiasi_dB = 20*log10(polaradiasi_norm);

figure(1)
polar(rad,polaradiasi_norm)
```

```
title('Pola Radiasi BS pada kanal urlos');

figure(2)
plot(polaradiasi_norm)
title('Pola Radiasi BS pada kanal urlos');
xlabel('Sudut');
ylabel('Amplitudo');

%%% PLOT POLA RADIASI PADA INPUT TX
figure(3);
ax = polaraxes;
polarplot(ax,polaradiasi_norm);
ax.ThetaDir = 'clockwise';
ax.ThetaZeroLocation = 'bottom';
title('Pola Radiasi TX pada kanal URLoS');
thetalim([90 270]);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%MATLAB SCRIPT PLOT POLA RADIASI POLAR DAN
RECTANGULAR METODE HB PADA KANAL URLOS 20
derajat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
close all;
clc;
load('u1_workspace_20drjt.mat');
m = 64; k=4;
VT = vrf_opt * VD;
antenaURLOS = sum(VT,2);
amplitudo_AFURLOS = abs(antenaURLOS);
fasa_AFURLOS = angle(antenaURLOS);
AF = zeros(m, 360);

for tx_urlos = 1:m
    beta = 2*pi;
    d = 1/2;
    N1 = 64 ;
    j = sqrt(-1);
    An = amplitudo_AFURLOS(tx_urlos,:);
    alpha = fasa_AFURLOS(tx_urlos,:);
        for theta= 1 : 360
            rad(theta) = (theta*pi)/180;
            for n = 1 : N1
            AF(tx_urlos,theta) =
AF(tx_urlos,theta) + An*exp(j*(n-1)*((beta * d *
sin(rad(theta)) ) + alpha));
            end
            AF(tx_urlos,theta) =
abs(AF(tx_urlos,theta))/N1;
        end
end
polaradiasi = sum(AF,1);
polaradiasi_norm = polaradiasi/max(polaradiasi);
polaradiasi_dB = 20*log10(polaradiasi_norm);

figure(1)
polar(rad,polaradiasi_norm)
```

```
title('Pola Radiasi BS pada kanal urlos');

figure(2)
plot(polaradiasi_norm)
title('Pola Radiasi BS pada kanal urlos');
xlabel('Sudut');
ylabel('Amplitudo');

%%% PLOT POLA RADIASI PADA INPUT TX
figure(3);
ax = polaraxes;
polarplot(ax,polaradiasi_norm);
ax.ThetaDir = 'clockwise';
ax.ThetaZeroLocation = 'bottom';
title('Pola Radiasi TX pada kanal URLoS');
thetalim([90 270]);
```

BIOGRAFI PENULIS

Penulis buku Tugas Akhir dengan judul **"Analisa Kinerja Teknik Hybrid Beamforming pada Sistem Large Scale Array dengan Multiuser MISO terhadap Kanal Acak"** bernama lengkap Dicky Rahmadi Prasetya. Penulis lahir di Surabaya, 16 Maret 1998. Penulis menempuh Pendidikan dasar di SD Muhammadiyah Situbondo dari kelas 1 hingga kelas 3 SD, kemudian penulis pindah ke Probolinggo dan melanjutkan sekolah dasar di SD Sukabumi 3 hingga lulus dan melanjutkan di SMP 1 Probolinggo, akan tetapi pada kelas 2 SMP penulis pindah ke SMP 1 Situbondo hingga lulus. Kemudian melanjutkan sekolah di SMAN 2 Lumajang hingga lulus. Dan melanjutkan kuliah di Institut Teknologi Sepuluh Nopember. Selama masa perkuliahan, penulis aktif di organisasi mahasiswa HIMATEKTRO ITS. Penulis memiliki moto hidup "*aku adalah apa yang aku pikirkan*".