



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR -TE 141599

**PERMODELAN DAN ANALISIS PERFORMA JARINGAN
ANTRIAN DARI SISTEM RUJUKAN BERJENJANG BPJS
DI SURABAYA**

KHAIRURIZAL ALFATHDYANTO
NRP. 2212100053

DOSEN PEMBIMBING
Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D.
Nurlita Gamayanti, S.T., M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

**MODELLING AND PERFORMANCE ANALYSIS OF SISTEM
RUJUKAN BERJENJANG BPJS IN SURABAYA AS A
QUEUEING NETWORK**

KHAIRURIZAL ALFATHDYANTO
NRP. 2212100053

SUPERVISING LECTURERS
Prof. Ir. Abdullah Alkaff, M.Sc., Ph. D.
Nurlita Gamayanti, S.T., M.T.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

**PERMODELAN DAN ANALISIS PERFORMA JARINGAN
ANTRIAN DARI SISTEM RUJUKAN BERJENJANG BPJS DI
SURABAYA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Teknik**

**Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui,

Dosen Pembimbing

Dosen Pembimbing


Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D.

NIP 195501231980031002


Nurlita Gamayanti, S.T., M.T.

NIP 197812012002122002



PERMODELAN DAN ANALISIS PERFORMA JARINGAN ANTRIAN DARI SISTEM RUJUKAN BERJENJANG BPJS DI SURABAYA

Nama : Khairurizal Alfathdyanto
NRP : 2212100053
Dosen Pembimbing 1 : Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D.
Dosen Pembimbing 2 : Nurlita Gamayanti, S.T., M.T.

ABSTRAK

Sistem rujukan berjenjang diterapkan oleh BPJS sebagai bagian dari prosedur penjaminan kesehatannya. Pasien yang ingin mendapatkan penjaminan kesehatan pada fasilitas kesehatan (faskes) tertentu harus mendapat rujukan dari faskes tingkat I pasien tersebut terdaftar. Penumpukan terjadi pasien pada faskes lanjutan tertentu disebabkan tidak ada aturan yang membatasi pemberian rujukan [1]. Dalam tugas akhir ini akan dibuat model jaringan antrian dari sistem rujukan BPJS dan dianalisis performa antrian dalam jaringan tersebut. Performa sistem rujukan berjenjang diestimasi menggunakan model sistem antrian hypercube yang dipergunakan dalam masalah antrian perkotaan. Analisis performa jaringan antrian tersebut menunjukkan pengaruh preferensi pasien terhadap penumpukan pasien pada faskes lanjutan. Pemberian rujukan dapat dilakukan dengan cara routing dinamis dan pemberian prioritas pasien. Kedua aturan tersebut disimulasikan untuk menentukan aturan pemberian rujukan yang cocok untuk permasalahan ini. Hasil yang didapatkan adalah preferensi pasien mempengaruhi laju kedatangan pada tiap rumah sakit, penerapan routing dinamis dapat menurunkan utilitas maksimum dan mengurangi rata-rata waktu tunggu, pemberian prioritas pasien memperbaiki performa routing dinamis pada sistem dengan beban kerja tinggi.

Kata kunci: Jaringan Antrian, Prioritisasi User, Routing Dinamis, Sistem Rujukan Berjenjang

Halaman ini sengaja dikosongkan

MODELLING AND PERFORMANCE ANALYSIS OF SISTEM RUJUKAN BERJENJANG BPJS IN SURABAYA AS A QUEUEING NETWORK

Name : Khairurizal Alfathdyanto
NRP : 2212100053
1st Supervisor : Prof. Ir. Abdullah Alkaff, M.Sc., Ph.D.
2nd Supervisor : Nurlita Gamayanti, S.T., M.T.

ABSTRACT

Sistem rujukan berjenjang is implemented by BPJS as part of the health insurance policies. Patients who want to get health insurance at a health facility particular should get a referral from the first level healthcare facility in which they are registered. Congestion of patients happens in certain upper level healthcare facility as there are no rules maintaining the referral system [1]. This final project models sistem rujukan berjenjang BPJS as a network of queueing system and analize the performance of queues in the network. Sistem rujukan berjenjang performances is estimated by hypercube queueing system model which used in urban queueing problem. Analysis of queueing network performances shows the influence patient preferences to buildup congestion of patients in upper level healthcare facilities. Referrals can be done by means of dynamic routing and prioritization of patients. Both of these rules are simulated to determine the appropriate rules of referral. The following results are obtained from the project, patient preferences affect the arrival rate at each hospital, the application of dynamic routing can reduce the maximum utility and reduce the average waiting time, prioritization of patients improve dynamic routing performance on systems with a high workload.

Keywords: Dynamic Routing, Patient Prioritization, Queueing networks, Sistem Rujukan Berjenjang

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah atas limpahan rahmat dari Allah Subhanahu wa Ta'ala sehingga penulis dapat menyelesaikan buku tugas akhir berjudul **“Permodelan dan Analisis Performa Jaringan Antrian dari Sistem Rujukan Berjenjang BPJS di Surabaya”**. Penulis menyadari dalam penyusunan buku tugas akhir ini banyak menemui kendala dan mendapatkan pertolongan dari berbagai pihak. Oleh sebab itu, penulis menyampaikan ucapan terima kasih dan penghargaan setinggi-tingginya kepada:

1. Badan Penyelenggara Jaminan Sosial (BPJS) atas bantuan data utama dipergunakan dalam penyusunan buku tugas akhir ini.
2. Badan Kesatuan Bangsa dan Perlindungan Masyarakat atas pemberian izin untuk melakukan penelitian di lingkungan Dinas Kesehatan Kota Surabaya.
3. Dinas Kesehatan Kota Surabaya atas data pendukung yang melingkupi susunan buku tugas akhir ini.
4. Prof. Ir. Abdullah Alkaff, M.Sc. dan Nurlita Gamayanti ST., MT. selaku dosen pembimbing atas arahan dan bimbingan beliau selama pengerjaan tugas akhir ini.
5. Orang tua penulis yang selalu memberikan dukungan dan motivasi selama proses pengerjaan tugas akhir ini.

Penulis menyadari bahwa dalam penulisan buku tugas akhir ini masih membutuhkan penyempurnaan. Karenanya, masukan dari semua pihak sangat penulis harapkan demi menyempurnakan karya ini. Akhir kata penulis berharap semoga buku tugas akhir ini dapat bermanfaat dan memajukan ilmu pengetahuan dan teknologi khususnya di bidang teknik sistem pengaturan.

Surabaya, 27 Juni 2016

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR PENGESAHAN	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	1
1.3 Batasan Masalah.....	1
1.4 Tujuan Penelitian.....	2
1.5 Sistematika Penulisan	2
1.6 Relevansi.....	3
BAB 2. DASAR TEORI.....	5
2.1 Sistem Antrian.....	5
2.1.1 Notasi Kendall.....	5
2.1.2 Pendefinisian Performa Antrian.....	6
2.1.3 Performa Sistem Antrian M/M/n.....	7
2.2 Hypercube Queueing System	9
2.2.1 Performa Model Hypercube.....	12
2.2.1.1 Beban Kerja Masing-masing Server.....	12
2.2.1.2 Ketakseimbangan Beban Kerja.....	12
2.2.1.3 Proporsi Penempatan Pengguna dari Atom ke Server.....	13
2.2.1.4 Performa Antrian dalam Model Hypercube.....	14
2.2.2 Estimasi Nilai Workload pada Model Hypercube.....	14
2.3 Routing Dinamis.....	17
2.4 Jaringan Antrian.....	19
2.5 Sistem Antrian dengan Prioritas.....	21
2.6 Simulasi Sistem Antrian pada MATLAB.....	23
BAB 3. PERANCANGAN KONSEPTUAL.....	29
3.1 Sistem Rujukan Berjenjang BPJS.....	29
3.2 Identifikasi Permasalahan.....	30
3.3 Permodelan Sistem Rujukan Berjenjang.....	31
3.3.1 Pembagian Kelompok Penyakit Pasien.....	32

3.3.2 Permodelan Rujukan sebagai Sistem Antrian Hypercube.....	34
3.3.2.1 Model Hypercube Sistem Rujukan.....	34
3.3.2.2 Modifikasi Model Antrian Hypercube.....	39
3.4 Perancangan Algoritma Perkiraan Performa Antrian Hypercube	40
3.5 Dynamic Routing untuk Penempatan Pasien.....	43
3.6 Konsep Pemberian Prioritas Pasien.....	44
BAB 4. IMPLEMENTASI SISTEM.....	47
4.1 Pembuatan Program pada MATLAB.....	47
4.1.1 Program Penerjemah Data.....	48
4.1.2 Program Estimasi Performa Hypercube.....	48
4.1.3 Fungsi Penentu Tujuan Routing Dinamis.....	48
4.2 Pembuatan Simulasi Routing Dinamis.....	48
4.3 Pembuatan Simulasi Pemberian Prioritas.....	52
4.4 Parameter Kerja Model.....	54
4.4.1 Pengolahan Data.....	54
4.4.2 Estimasi Parameter Menggunakan Hitungan Matematis.....	58
BAB 5. HASIL PENGUJIAN DAN ANALISIS.....	61
5.1 Estimasi Laju Kedatangan pada Rumah Sakit.....	61
5.2 Pengaruh Routing Dinamis pada Model Sistem Rujukan.....	65
5.2.1 Hasil Simulasi Routing Dinamis.....	65
5.2.2 Analisis Sensitivitas Routing Dinamis.....	69
5.3 Pengaruh Prioritas Pasien pada Model Sistem Rujukan.....	70
BAB 6. PENUTUP.....	75
6.1 Kesimpulan.....	75
6.2 Saran.....	75
DAFTAR PUSTAKA.....	77
LAMPIRAN A – DATA SURVEI.....	A-1
LAMPIRAN B – SCRIPT MATLAB.....	B-1
LAMPIRAN C – MODEL SIMULASI.....	C-1

DAFTAR GAMBAR

Gambar 2.1	Diagram Transisi State untuk Model Antrian M/M/n...	7
Gambar 2.2	Representasi kubus transisi state <i>hypercube</i> 3 server...	11
Gambar 2.3	Diagram Transisi State Pelayanan dari Server-i	15
Gambar 2.4	Blok time-based entity generator dan kotak dialognya	24
Gambar 2.5	Blok N-server dan kotak dialognya.....	25
Gambar 2.6	Blok FIFO queue dan kotak dialognya.....	25
Gambar 2.7	Blok priority queue dan kotak dialognya.....	26
Gambar 2.8	Blok <i>set attribute</i> dan kotak dialognya.....	27
Gambar 3.1	Prosedur Sistem Rujukan Berjenjang.....	31
Gambar 3.2	Skema Konsep Sistem Rujukan.....	33
Gambar 3.3	Skema Estimasi Menggunakan Model <i>Hypercube</i>	35
Gambar 3.4	Skema Pembagian Atom Geografis dan Posisi Server...	36
Gambar 3.5	Diagram Transisi State Model <i>Hypercube</i> H_i	38
Gambar 4.1	Skema Implementasi Model Simulasi Sistem Rujukan	47
Gambar 4.2	Skema Pemasangan Blok Simulink.....	49
Gambar 4.3	Blok Simulink yang Merepresentasikan <i>Routing</i>	49
Gambar 4.4	Blok Simulink yang Merepresentasikan Poli RS.....	50
Gambar 4.5	Konfigurasi blok <i>entity generator</i>	51
Gambar 4.6	Konfigurasi blok <i>set attribute</i>	52
Gambar 4.7	Pemasangan atribut kelas prioritas pasien.....	53
Gambar 4.8	Konfigurasi blok <i>priotity queue</i>	53
Gambar 5.1	Proses Iteratif Estimasi Beban Kerja.....	61
Gambar 5.2	Perbandingan Presentase Perubahan Laju Kedatangan	64
Gambar 5.3	Banyak Antrian Simulasi <i>Routing</i> Dinamis.....	65
Gambar 5.4	Rata-rata waktu tunggu <i>routing</i> dinamis.....	66
Gambar 5.5	Utilisasi pada <i>routing</i> dinamis.....	66
Gambar 5.6	Perbandingan waktu tunggu.....	67
Gambar 5.7	Perbandingan utilisasi maksimum.....	68
Gambar 5.8	Sensitivitas <i>Routing</i> terhadap Laju Kedatangan.....	69
Gambar 5.9	Banyak Antrian Simulasi Pemberian Prioritas.....	70
Gambar 5.10	Rata-rata waktu tunggu pada simulasi prioritas.....	71
Gambar 5.11	Utilisasi rumah sakit pada simulasi prioritas.....	71
Gambar 5.12	Perbandingan waktu tunggu dengan adanya prioritas	72
Gambar 5.13	Perbandingan utilisasi antar metode <i>routing</i>	73
Gambar 5.14	Sensitivitas terhadap Proporsi Pasien Kronis.....	74

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 4.1 Data Laju Kedatangan Tiap Puskesmas.....	55
Tabel 4.2 Proporsi Penyakit Tiap Area (dalam %).	56
Tabel 4.3 Rata-rata laju pelayanan dan banyak dokter tiap RS.....	57
Tabel 4.4 Waktu Tempuh menuju Rumah Sakit	58
Tabel 4.5 Laju Kedatangan Pasien Berdasarkan Tipe Penyakit	59
Tabel 4.6 Rata-rata laju pelayanan pasien kronis dan non-kronis	60
Tabel 5.1 Rata-rata beban kerja server pada masing-masing poli RS	62
Tabel 5.2 Estimasi laju kedatangan poli rumah sakit.....	62
Tabel 5.3 Estimasi waktu tunggu di poli rumah sakit (dalam menit)	63
Tabel 5.4 Estimasi laju kedatangan poli RS pada skenario kedua....	63

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sistem rujukan berjenjang diterapkan oleh BPJS sebagai bagian dari prosedur penjaminan kesehatannya. Pasien yang ingin mendapatkan penjaminan kesehatan harus melalui proses pemeriksaan di faskes tingkat I. Pasien yang membutuhkan penanganan lebih lanjut dapat memperoleh rujukan menuju faskes lanjutan. Sistem yang diterapkan BPJS ini masih menimbulkan masalah seperti menumpuknya pasien di rumah sakit terkenal yang memiliki fasilitas lengkap. Tidak adanya aturan yang membatasi pemberian rujukan menyebabkan tingginya rasio rujukan. Sebagai contoh, Puskesmas 5 Ilir, Sumatera Barat, rasio rujukannya dapat mencapai 60%.^[1] Hal ini juga dipengaruhi beberapa permintaan rujukan yang didasari keinginan pasien untuk dirujuk sehingga menambah rasio rujukan.

Kualitas dari pelayanan BPJS masih diragukan. Salah satu penyebabnya adalah alur rujukan berjenjang yang diterapkan BPJS masih belum dibatasi dengan ketat. Rujukan saat ini masih memberikan kebebasan pada pasien untuk memilih fasilitas kesehatan lanjutan. Hal ini mengakibatkan terjadinya penumpukan pasien di fasilitas kesehatan yang menjadi favorit karena fasilitas yang lebih lengkap maupun pelayanan yang lebih memuaskan. Analisis performa jaringan antrian yang dimodelkan dari sistem rujukan BPJS akan menunjukkan pengaruh rujukan yang tidak diatur terhadap penumpukan pasien di faskes lanjutan tertentu. Hasil dari analisis akan menunjukkan perlunya pemberian aturan yang dapat menyeimbangkan rasio rujukan sehingga tidak terjadi penumpukan pasien.

1.2 Perumusan Masalah

Permasalahan yang diambil pada tugas akhir ini adalah penentuan *routing* pada sistem rujukan berjenjang dengan menggunakan model antrian yang memperhatikan preferensi dan prioritas pengguna.

1.3 Batasan Masalah

Batasan permasalahan beserta asumsi yang diambil untuk tugas akhir ini adalah

1. Pasien merupakan peserta BPJS yang berdomisili di Surabaya.

2. Area permodelan adalah Surabaya Timur.
3. Rumah sakit yang dipergunakan dalam model sebanyak 3 unit.
4. Fasilitas kesehatan tingkat I yang dimodelkan adalah 13 puskesmas di area Surabaya Timur.
5. Proses kedatangan pasien dari area lain di Surabaya dikelompokkan menjadi 5 kelompok area.
6. Proses kedatangan dan pelayanan diasumsikan mengikuti proses Poisson.

1.4 Tujuan Penelitian

Tujuan dari pelaksanaan tugas akhir ini adalah menghasilkan:

1. Model jaringan antrian dari sistem rujukan berjenjang di Surabaya.
2. Performa antrian model jaringan antrian sistem rujukan berjenjang di Surabaya.
3. Pengaruh routing dinamis dan pemberian prioritas pada model sistem rujukan berjenjang.

1.5 Sistematika Penulisan

Sistematika pembahasan dalam buku tugas akhir ini terdiri dari 5 bab yaitu, pendahuluan, tinjauan pustaka, permodelan sistem, analisis performa dan simulasi kemudian kesimpulan dan saran. Uraian untuk tiap bab adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab 1 menguraikan latar belakang, rumusan masalah, tujuan, metodologi, sistematika penulisan dan relevansi dalam tugas akhir ini.

BAB 2 DASAR TEORI

Bab 2 memberikan uraian tentang sistem antrian, model antrian hypercube, jaringan antrian dan teori BCMP.

BAB 3 PERANCANGAN KONSEPTUAL

Bab 3 mendeskripsikan sistem rujukan berjenjang BPJS kemudian membahas pemodelan sistem tersebut dalam model jaringan antrian BCMP dengan sistem antrian hypercube.

BAB 4 IMPLEMENTASI SISTEM

Bab 4 menjelaskan pembuatan script MATLAB dan perancangan simulasi untuk routing dinamis dan pemberian prioritas pasien.

BAB 5 HASIL PENGUJIAN DAN ANALISIS

Bab 5 menunjukkan hasil simulasi yang berupa nilai-nilai performa antrian dan analisis terhadap hasil tersebut.

BAB 6 KESIMPULAN DAN SARAN

Bab 6 menjabarkan kesimpulan yang diambil pada tugas akhir ini dan memberikan saran untuk pengembangan selanjutnya.

1.6 Relevansi

Tugas Akhir ini berkaitan dengan mata kuliah Jaringan Stokastik dan Simulasi Sistem. Pengerjaan dari tugas akhir ini diharapkan dapat memberikan kontribusi pada pengembangan ilmu pengetahuan dan teknologi khususnya dalam bidang teknik sistem. Tugas akhir juga diharapkan dapat menjadi acuan untuk pengembangan kualitas pelayanan kesehatan terutama peningkatan kualitas sistem rujukan berjenjang BPJS. Ilmu teknik sistem yang dipergunakan dalam tugas akhir ini meliputi jaringan antrian dan simulasi sistem.

Halaman ini sengaja dikosongkan

BAB 2

DASAR TEORI

Bab II membahas mengenai dasar-dasar teori yang dipergunakan untuk mengerjakan tugas akhir ini. Pada bab ini akan dibahas dasar-dasar mengenai jaringan poisson, sistem antrian dan model antrian hypercube.

2.1 Sistem Antrian

Sistem antrian secara umum terdiri atas tiga unsur yaitu, sumber pengguna layanan (populasi), antrian, dan server (dapat berisi satu atau lebih server identik) [4]. Pendeskripsian sistem antrian memerlukan informasi tentang tiga elemen dasar sistem antara lain, proses pembangkitan pengguna (proses kedatangan pengguna di sistem), disiplin antrian (urutan pengguna dalam mendapatkan akses ke server) dan proses pelayanan server.

2.1.1 Notasi Kendall

Untuk mengidentifikasi jenis sistem antrian yang dimodelkan biasanya dinyatakan dalam notasi Kendall. Kendall secara khusus membedakan sistem antrian berdasarkan notasi $A/B/n/p/k$. Kode huruf A mewakili jenis proses kedatangan. Kode huruf B mewakili jenis proses pelayanan. n menyatakan jumlah server. k menyatakan kapasitas sistem dan p menyatakan jenis populasi pelanggan (finite/infinite). Kode huruf yang biasa dipergunakan untuk mengisi A dan B antara lain:

1. M = poisson (M merupakan singkatan dari *memoryless*)
2. D = deterministik (Kedatangan dan waktu pelayanan sudah ditentukan dengan pasti)
3. Ek = Distribusi Erlang orde-k
4. Hk = Distribusi hipereksponensial orde-k
5. G = Distribusi umum

Disiplin antrian merupakan aturan bagaimana pengguna dilayani oleh server. Beberapa istilah digunakan mewakili beberapa disiplin antrian yang umum. FIFO mewakili urutan antrian *first-in, first-out*. Pada antrian FIFO, pengguna yang dilayani sesuai urutan kedatangan. LIFO mewakili urutan antrian *last in, first out*. Pada antrian LIFO, pengguna yang datang terakhir akan dilayani pertama. SIRO mewakili urutan antrian *service in random order*.

2.1.2 Pendefinisian Performa Antrian

Saat pengguna ke- i yang tiba di sistem terdapat tiga even penting yang teridentifikasi dari kedatangan pengguna tersebut yaitu, kedatangan pengguna pada sistem antrian, awal dari pelayanan pada pengguna, selesainya pelayanan untuk pengguna. Definisikan waktu terjadinya even sejak awal penghitungan sebagai $t_a(i)$, $t_b(i)$, dan $t_c(i)$ secara berurutan. Maka dapat didefinisikan juga:

1. $x(i) = t_a(i) - t_a(i - 1) =$ Waktu antar kedatangan
2. $s(i) = t_c(i) - t_b(i) =$ Waktu pelayanan pengguna ke- i
3. $w(i) = t_b(i) - t_a(i) =$ Waktu tunggu dalam antrian untuk pengguna ke- i
4. $t(i) = t_c(i) - t_a(i) =$ Total waktu yang dihabiskan dalam sistem antrian

Secara umum, waktu antar kedatangan dan waktu tunggu adalah variabel acak, $X(i)$ dan $S(i)$ yang memiliki fungsi densitas probabilitas $f_{X(i)}(x)$ dan $f_{S(i)}(s)$ secara berurutan. Penggunaan istilah fungsi probabilitas densitas ini dalam keadaan umum sehingga variabel acak $X(i)$ dan $S(i)$ untuk setiap i dapat berupa variabel acak diskrit, kontinu maupun campuran. $X(i)$ dan saling independen dan memiliki distribusi yang sama. Begitu juga dengan variabel acak $S(i)$ diasumsikan memiliki sifat yang sama. Nilai ekspektasi dari kedua variabel acak ini sering muncul dalam analisis sistem antrian sehingga memiliki notasi tersendiri yaitu,

$$\lambda = \frac{1}{E[X]} \quad (2.1)$$

$$\mu = \frac{1}{E[S]} \quad (2.2)$$

λ merepresentasikan laju kedatangan pengguna dalam sistem antrian. μ merepresentasikan laju pelayanan sebuah server apabila server bekerja terus-menerus. Perhatikan bahwa, ketika n server paralel dalam suatu sistem antrian bekerja bersamaan maka ekspektasi laju

pelayanannya menjadi $n\mu$. Kemudian notasikan beban kerja dari suatu sistem antrian sebagai ρ maka,

$$\rho = \frac{\lambda}{n\mu} \quad (2.3)$$

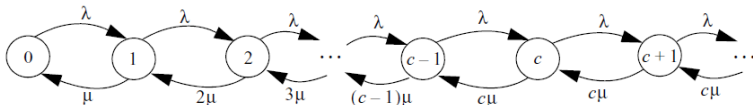
2.1.3 Performa Sistem Antrian M/M/n

Performa sistem antrian secara umum dapat dilihat dari nilai rata-rata waktu tunggu dalam sistem, rata-rata waktu tunggu dalam antrian, rata-rata jumlah pelanggan dalam sistem, rata-rata jumlah pelanggan dalam antrian, proporsi jumlah pelanggan yang ditolak sistem, dan utilisasi server.

Performa antrian dari suatu sistem antrian dapat dihitung dengan menggunakan nilai probabilitas dari masing-masing *state* sistem. Nilai tersebut didapatkan dengan menyelesaikan persamaan beda differensial dari probabilitas transisi *state*. *State* didefinisikan sebagai banyak pengguna dalam sistem. Transisi *state* adalah perpindahan *state* sistem dari satu *state* ke *state* lain.

Transisi *state* dibagi menjadi dua jenis yaitu, transisi “naik” dan transisi “turun”. Transisi naik terjadi ketika pengguna datang ke sistem dan transisi turun terjadi ketika pengguna selesai dilayani. Transisi naik identik dengan kedatangan pengguna memiliki laju kedatangan yang sama (λ) dan berdistribusi poisson untuk setiap *state* dalam model antrian M/M/n.

Sedangkan transisi turunnya memiliki nilai yang sama untuk *state* yang lebih besar dari n ($n\mu$) dan terdistribusi poisson. Akan tetapi untuk *state* $k < n$, nilai laju transisi turunnya adalah $k\mu$. Transisi *state* untuk model M/M/n diilustrasikan menggunakan diagram transisi *state* pada Gambar 2.1.



Gambar 2.1 Diagram Transisi State untuk Model Antrian M/M/n [4]

Notasikan probabilitas *state* i sebagai P_i . Berdasarkan diagram transisi *state* pada Gambar 2.1 persamaan beda differensial untuk model antrian M/M/n dapat dituliskan sebagai berikut

$$\frac{dP_0}{dt} = \mu P_1 - \lambda P_0 \quad (2.4)$$

$$\frac{dP_i}{dt} = (i+1)\mu P_{i+1} + \lambda P_{i-1} - (\lambda + i\mu)P_i, i < n \quad (2.5)$$

$$\frac{dP_i}{dt} = n\mu P_{i+1} + \lambda P_{i-1} - (\lambda + n\mu)P_i, i \geq n \quad (2.6)$$

Dalam kondisi *steady-state* $\frac{dP_i}{dt} = 0$ sehingga ruas kiri dari Persamaan (2.4), (2.5) dan (2.6) menjadi nol. ρ untuk antrian tak hingga merupakan nilai utilitas antrian tersebut. Dengan menyubstitusikan ρ dari Persamaan (2.3) maka dari Persamaan (2.4), (2.5) dan (2.6) akan diperoleh,

$$P_i = \begin{cases} \frac{(n\rho)^i}{i!} P_0, i < c \\ \frac{n^n \rho^i}{n!} P_0, i \geq c \end{cases} \quad (2.7)$$

Karena jumlahan probabilitas semua state dalam sistem bernilai satu, maka didapatkan Persamaan (2.8):

$$\sum_{i=0}^{\infty} P_i = 1 \leftrightarrow P_0 = \left[\sum_{i=1}^{n-1} \left(\frac{(n\rho)^i}{i!} \right) + \frac{(n\rho)^n}{n!} \left(\frac{1}{1-\rho} \right) \right]^{-1} \quad (2.8)$$

Peluang terjadinya antrian dapat dicari melalui Persamaan (2.9)

$$P_Q = \sum_{i=n+1}^{\infty} P_i = \frac{P_0 (n\rho)^n}{n!} \sum_{i=n}^{\infty} \rho^{i-n} = \frac{P_0 (n\rho)^n \rho}{n! (1-\rho)} \quad (2.9)$$

Rata-rata banyak pengguna yang mengantri dapat dihitung menggunakan Persamaan (2.10),

$$E[N_q] = \sum_{i=n}^{\infty} (i-n)P_i = \frac{P_0(n\rho)^n}{n!} \sum_{i=n}^{\infty} (i-n)\rho^{i-n} = \frac{P_0(n\rho)^n \rho}{n! (1-\rho)^2} \quad (2.10)$$

Rata-rata waktu antri pengguna dalam sistem dihitung menggunakan Persamaan (2.11),

$$E[W] = \frac{E[N_q]}{\lambda} = \frac{P_0(n\rho)^n \rho}{n! (1-\rho)^2} = \frac{P_Q}{\lambda(1-\rho)} \quad (2.11)$$

Rata-rata waktu dalam sistem dihitung dengan menambahkan rata-rata waktu layanan dengan rata-rata waktu mengantri. Kemudian, rata-rata banyak pengguna dalam sistem dihitung dengan mengalikan rata-rata waktu dengan λ , seperti pada Persamaan (2.12) dan (2.13).

$$E[T] = E[W] + \frac{1}{\mu} \quad (2.12)$$

$$E[N] = \lambda E[T] \quad (2.13)$$

2.2 *Hypercube Queueing System*

Hypercube queueing system (HQS) merupakan pengembangan dari spatial distributed queue (SDQ) untuk banyak server lebih dari 2 [6]. SDQ adalah model antrian dengan pengguna maupun server terdistribusi secara spasial pada sistem. Permasalahan antrian perkotaan yang dapat dimodelkan sebagai SDQ antara lain sistem pengantaran barang, pelayanan medis darurat, layanan transportasi. Tiga karakteristik utama dalam SDQ adalah (1) ketidakpastian waktu, lokasi dan lama pelayanan, (2) pengaruh besar dari sebaran spasial pengguna maupun server dan (3) penumpukan antrian dapat terjadi dengan berbagai bentuk.

Area dari sistem dimodelkan menjadi sekumpulan atom geografis yang mewakili sumber permintaan pelayanan. Area respon utama adalah area dimana suatu unit ditugaskan ketika siap meskipun unit lain juga dalam keadaan siap. Unit pelayanan tetap diletakkan pada salah satu dari atom tersebut. Ketersediaan server pada HQS dinyatakan sebagai state space sehingga untuk 3 server kondisi state membentuk suatu kubus dan untuk N-server, $N > 3$, state akan membentuk hypercube.

Inti dari model hypercube ini adalah deskripsi geografis dari daerah yang ingin dimodelkan. Daerah dibagi menjadi K bagian yang dapat disebut atom geografis. Atom geografis k ($1 \leq k \leq K$) memunculkan pengguna dengan laju kedatangan masing-masing. Notasikan proporsi laju kedatangan pengguna pada atom k terhadap keseluruhan laju kedatangan pengguna dalam daerah tersebut sebagai f_k . Laju kedatangan keseluruhan pada daerah yang dimodelkan sebagai λ . Maka, laju kedatangan tiap pengguna pada atom k adalah λf_k .

Server dalam model ini dapat diam maupun bergerak. Waktu tempuh antar atom perlu diperhatikan untuk tipe server yang bergerak. Selain itu, proporsi waktu dimana server menetap saat waktu luang diperlukan untuk mengetahui estimasi waktu tempuh rata-rata. Tipe server yang digunakan pada tugas akhir ini adalah tipe server diam.

Sistem antrian yang ingin dimodelkan dapat dilihat sebagai sistem antrian M/M/N jika tidak memperhatikan identitas server. Menurut kapasitas servernya sistem yang dapat dimodelkan ada dua jenis yaitu, kapasitas nol dan kapasitas tak berhingga. Sistem dapat dilihat sebagai sistem antrian M/M/N apabila memenuhi asumsi-asumsi berikut antara lain,

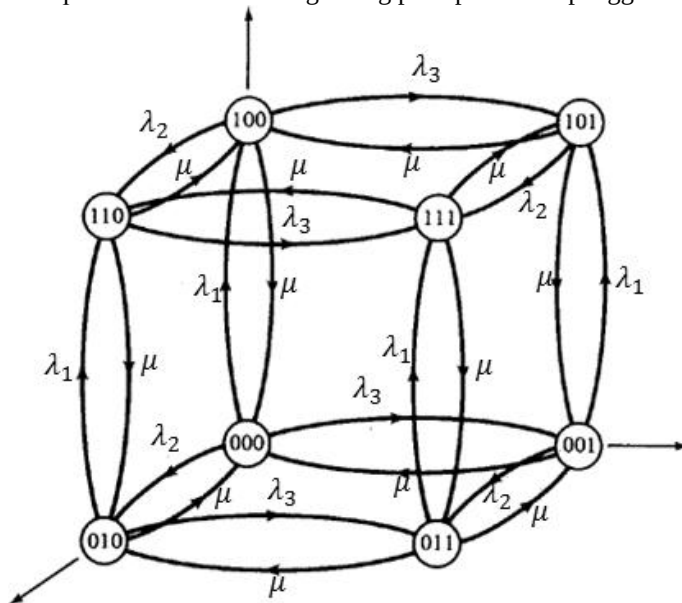
1. Waktu antar kedatangan pengguna memiliki distribusi eksponensial dan hanya satu dari N server yang melayani pengguna tersebut.
2. Waktu pelayanan semua server untuk tiap layanan memiliki distribusi eksponensial dengan rata-rata yang sama ($1/\mu$).
3. Waktu pelayanan tidak dipengaruhi identitas server maupun pengguna dan kondisi sebelumnya pada sistem.
4. Untuk kapasitas antrian tak berhingga, pengguna masuk dalam antrian yang sama dan kemudian akan dilayani dengan disiplin FCFS ketika semua server sedang sibuk.

Representasi *state* dalam model didefinisikan dalam bilangan biner. Bilangan biner tersebut menyatakan kondisi “sibuk” (dinyatakan dengan digit 1) dan “luang” (dinyatakan dengan digit 0). Himpunan bilangan biner terurut $B = \{B_1, B_2, B_3, \dots, B_N\}$ menyatakan state dari keseluruhan sistem, B_i menyatakan kondisi dari server- i . Setiap B berkorespondensi dengan suatu titik koordinat di ruang R^N . Kemudian, C_N dinyatakan sebagai himpunan semua state yang mungkin dari B . Maka, himpunan C_N membentuk sebuah hypercube di ruang R^N .

State space dari model antrian kapasitas nol (tanpa antrian) membentuk sebuah *hypercube* dimensi- N . Tiap titik pada bangun

tersebut berkorespondensi dengan kondisi server dalam sistem. Sementara, state space dari model antrian kapasitas tak berhingga memiliki tambahan yaitu *infinite tail*. *Infinite tail* terjadi ketika pengguna datang saat semua server sedang sibuk. Ilustrasi kubus yang terbentuk dari transisi state model *hypercube* untuk 3 server dengan kapasitas tak berhingga dapat dilihat pada Gambar 2.2

Persamaan *state* ditentukan dari laju transisi antar *state* dalam sistem. Persamaan state dapat dinyatakan dalam sebuah matriks dengan elemen λ_{ij} . λ_{ij} menyatakan laju transisi dari *state* i ke *state* j . Transisi dalam model *hypercube* dibagi menjadi dua jenis transisi “naik” dan “turun”. Transisi “naik” terjadi ketika ada pengguna yang datang dan dilayani oleh server. Transisi “turun” terjadi ketika suatu server selesai melayani pengguna. Karena laju pelayanan tiap server dianggap sama, maka laju transisi “turun” memiliki nilai μ^{-1} . Namun, nilai laju transisi “naik” dapat berbeda karena bergantung pada preferensi pengguna.



Gambar 2.2 Representasi kubus transisi state *hypercube* 3 server[6]

Nilai laju transisi “naik” dapat dicari setelah preferensi dari masing-masing atom geografis sudah ditentukan. Nyatakan S_{ij} sebagai himpunan atom yang mana kedatangan dari atom $s \in S_{ij}$ pada state B_i menyebabkan state sistem menjadi B_j . Anggota himpunan S_{ij} ditentukan dari preferensi atom yang mana memilih server $B_i - B_j$ pada state B_i .

2.2.1 Performa Model Hypercube

Setelah mendapat nilai numerik dari probabilitas tiap *state* selanjutnya dapat dihitung nilai performa antrian dari sistem tersebut. Model dapat digunakan untuk menghitung performa berikut:

1. Beban kerja masing-masing server.
2. Ketakseimbangan beban kerja server dalam wilayah.
3. Proporsi penempatan pengguna dari atom ke server.
4. Rata-rata lama waktu antri pada suatu server.

2.2.1.1 Beban Kerja Masing-Masing Server

Beban kerja server i dinotasikan sebagai ρ_i dan merupakan perbandingan waktu dimana server i sedang sibuk melayani pengguna. Jadi, ρ_i adalah jumlahan dari probabilitas state saat server i sedang sibuk atau $b_i = 1$ sehingga dapat dituliskan dalam Persamaan (2.14)

$$\rho_i = \sum_{\{j|b_i=1\}} P(B_j) \quad (2.14)$$

2.2.1.2 Ketakseimbangan Beban Kerja

Preferensi pengguna untuk memilih server yang melayaninya dapat mengakibatkan ketidakseimbangan beban kerja. Hal ini dikarenakan suatu server menjadi pilihan kebanyakan pengguna dalam area tersebut. Ada beberapa cara untuk menyatakan ketidakseimbangan beban kerja, antara lain:

1. Deviasi absolut

$$\Delta W = \max_{i \neq j, l, j \in N} (|\rho_i - \rho_j|) \quad (2.15)$$

2. Varians

$$\sigma_W^2 = \sum_{i=1}^N \rho_i^2 - \left(\sum_{i=1}^N \rho_i \right)^2 \quad (2.16)$$

3. Standar deviasi

$$\sigma_W = \sqrt{\sum_{i=1}^N \rho_i^2 - \left(\sum_{i=1}^N \rho_i \right)^2} \quad (2.17)$$

4. Persentase deviasi maksimum terhadap mean

$$P = \left(\frac{\Delta W}{\bar{\rho}} \right) \times 100\% \quad (2.18)$$

2.2.1.3 Proporsi Penempatan Pengguna dari Atom ke Server

Proporsi penempatan pengguna dari atom k pada server j merupakan perbandingan laju penempatan pengguna dari atom k pada server j terhadap seluruh laju kedatangan pengguna dalam sistem. Besaran tersebut dinotasikan dengan r_{jk} . Jelas bahwa jumlahan r_{jk} untuk semua asal atom geografis dan server dalam wilayah adalah 1. Penghitungan r_{jk} akan digunakan untuk mencari nilai laju kedatangan pada masing-masing server, rata-rata lama waktu antrian, dan rata-rata lama pengguna di dalam sistem.

Definisikan E_{jk} sebagai himpunan state dimana server j merupakan server yang dipilih oleh pengguna dari atom geografis k pada state tersebut. Maka r_{jk} dapat dituliskan sebagai,

$$r_{jk} = \frac{\sum_{B_i \in E_{jk}} \frac{f_k P(B_i)}{n_{ik}}}{1 - P(B_{2N-1})} \quad (2.19)$$

Penempatan pengguna yang berbeda distrik adalah performa dalam antrian hypercube yang menyatakan banyaknya mobilitas pengguna antar distrik yang berbeda. Distrik dapat didefinisikan sebagai daerah

kerja utama dari server. Hal ini digunakan untuk melihat proporsi pengguna ditempatkan di luar daerah preferensi utamanya maupun proporsi pelayanan pengguna dari luar daerah kerja utama server. Proporsi yang dapat diamati antara lain.

1. Proporsi dari total penempatan di distrik berbeda

$$FI = \sum_{j=1}^N \sum_{k \in (\text{distrik } j)'} r_{jk} \quad (2.20)$$

2. Proporsi dari penempatan ke server j di distrik berbeda

$$FI_j = \frac{\sum_{k \in (\text{distrik } j)'} r_{jk}}{\sum_{j=1}^N r_{jk}} \quad (2.21)$$

3. Proporsi layanan dari distrik i yang dilayani diluar distrik

$$FO_j = \frac{\sum_{j \neq i} \sum_{k \in (\text{distrik } j)'} r_{jk}}{\sum_{j=1}^N \sum_{k \in (\text{distrik } j)'} r_{jk}} \quad (2.22)$$

2.2.1.4 Performa Antrian dalam Model Hypercube

Setelah mendapatkan seluruh proporsi penempatan pengguna dari atom k menuju server j dalam sistem, performa antrian pada tiap server dapat dihitung. Performa antrian dapat dihitung menggunakan fungsi performa antrian pada model M/M/n. Nilai yang perlu didapatkan untuk menghitung performa adalah laju kedatangan menuju server tersebut (λ_j). Laju kedatangan pada server j merupakan jumlahan dari laju kedatangan dari tiap atom yang menuju ke server-j. Nilai λ_j dinyatakan dapat dicari menggunakan Persamaan (2.23)

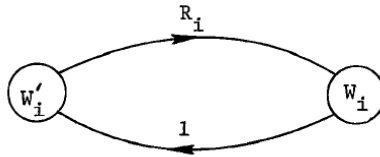
$$\lambda_j = \sum_{k=1}^K \lambda_{jk} = \lambda \sum_{k=1}^K r_{jk} \quad (2.23)$$

2.2.2 Estimasi Nilai Beban Kerja pada Model Hypercube

Dalam suatu model antrian *hypercube* terdapat 2^n state yang berbeda. Hal ini memunculkan 2^n persamaan *state* berbeda untuk model

tersebut. Penyelesaian dari model *hypercube* merupakan solusi dari sistem persamaan yang terdiri dari 2^n persamaan *state*. Hal tersebut menimbulkan kompleksitas perhitungan sehingga solusi dari model *hypercube* ini dapat diperkirakan dengan menggunakan n persamaan yang menyatakan beban kerja server [7].

Definisikan ρ_i sebagai beban kerja server- i (perbandingan waktu server sibuk melayani pengguna) dan w_i sebagai event dimana server- i menangani pengguna. Jelas bahwa $P(w_i) = \rho_i$ sehingga untuk server- i diagram transisi *state* untuk event diilustrasikan pada Gambar 2.3



Gambar 2.3 Diagram Transisi State Pelayanan dari Server- i [9]

R_i merepresentasikan laju penugasan server- i pada suatu pengguna ketika sedang tidak sibuk. Perlu diperhatikan bahwa kompleksitas pada permasalahan ini R_i tidak selalu berupa proses Poisson. Namun, dengan kondisi *steady state* persamaan beda diferensial dari *state* tersebut dapat dinyatakan sebagai berikut,

$$P(w_i) = \rho_i = \frac{R_i}{1 + R_i} \quad (2.24)$$

Permasalahan menentukan nilai ρ_i disederhanakan menjadi penentuan nilai R_i . Definisikan variabel baru R_i' yang merepresentasikan laju penempatan pengguna pada server- i . Dengan fakta bahwa hanya $(1 - \rho_i)$ fraksi dari keseluruhan waktu server- i siap melayani pengguna maka,

$$R_i' = R_i(1 - \rho_i) \quad (2.25)$$

Misalkan G_i^k adalah himpunan atom geografis yang memilih server- i pada pilihan ke- k dan n_{ji} adalah nomer identifikasi server yang

merupakan pilihan ke- i dari atom j . Nilai R_i' dapat dicari dengan menyelesaikan Persamaan (2.26)

$$R_i' = \left(\sum_{j \in G_i^1} \lambda_j \cdot P(w_i') + \sum_{j \in G_i^2} \lambda_j \cdot P(w_{n_{j1}} w_i') + \sum_{j \in G_i^3} \lambda_j \cdot P(w_{n_{j1}} w_{n_{j2}} w_i') + \dots \right. \\ \left. + \sum_{j \in G_i^N} \lambda_j \cdot P(w_{n_{j1}} w_{n_{j2}} \dots w_{n_{jN-1}} w_i') + \frac{\lambda}{N} \right) \quad (2.26)$$

Untuk menyederhanakan Persamaan (2.26), nilai probabilitas *joint* dari kondisi *dispatch* server dapat diestimasi sebagai hasil perkalian utilitas atau faktor ketersediaan dengan faktor pembetulan tertentu sehingga nilai $P(w_i w_j \dots w_n')$ dapat didekati dengan $\rho_i \rho_j \dots Q(N, \rho, k)(1 - \rho_n)$.

Fungsi $Q(N, \rho, k)$ merupakan *queue correction factor* yang nilainya dapat dicari menggunakan tabel pada [9]. Dimana, N menyatakan banyak server dalam sistem antrian, ρ menyatakan rata-rata beban kerja server dalam sistem, dan K menyatakan banyak server yang sibuk sebelum server- n terpilih. Persamaan (2.26) dapat ditulis ulang menjadi,

$$R_i' = \left(\sum_{j \in G_i^1} \lambda_j + \sum_{j \in G_i^2} \lambda_j \cdot Q(N, \rho, 1) \rho_{n_{j1}} + \sum_{j \in G_i^3} \lambda_j \cdot Q(N, \rho, 2) \rho_{n_{j1}} \cdot \rho_{n_{j2}} + \dots \right. \\ \left. + \sum_{j \in G_i^N} \lambda_j \cdot Q(N, \rho, N-1) \rho_{n_{j1}} \cdot \rho_{n_{j2}} \dots \rho_{n_{j(N-1)}} + \frac{\lambda}{N} \right) (1 - \rho_i) \quad (2.27)$$

Dengan menggunakan hubungan antara Persamaan (2.24), (2.25) dan (2.27) akan diperoleh Persamaan (2.28)

$$1 - \rho_i = \left[\sum_{j \in G_i^1} \lambda_j + \sum_{j \in G_i^2} \lambda_j \cdot Q(N, \rho, 1) \rho_{n_{j1}} + \sum_{j \in G_i^3} \lambda_j \cdot Q(N, \rho, 2) \rho_{n_{j1}} \cdot \rho_{n_{j2}} + \dots \right. \\ \left. + \sum_{j \in G_i^N} \lambda_j \cdot Q(N, \rho, N-1) \rho_{n_{j1}} \cdot \rho_{n_{j2}} \dots \rho_{n_{j(N-1)}} + \frac{\lambda}{N(1 - \rho_i)} \right]^{-1} \quad (2.28)$$

Persamaan (2.28) mewakili N persamaan nonlinear yang dipergunakan untuk mengestimasi nilai ρ_i secara iteratif. Sebelum melakukan iterasi tentukan terlebih dahulu nilai toleransi ϵ . Algoritma untuk penentuan nilai ρ_i dijabarkan sebagai berikut,

1. Inisialisasi – Menghitung rata-rata beban kerja server pada antrian M/M/n dengan server sejumlah yang ingin dimodelkan. Set nilai tersebut sebagai nilai awal beban kerja tiap server.
2. Iterasi – Hitung estimasi beban kerja menggunakan persamaan (28) dengan menggunakan nilai pada iterasi sebelumnya.
3. Normalisasi – Normalisasi terhadap rata-rata beban kerja agar memberikan nilai yang sama.
4. Cek Konvergensi – Hentikan iterasi apabila selisih beban kerja antar iterasi sudah di bawah nilai toleransi ϵ .

2.3 Routing Dinamis

Routing dinamis merupakan aturan yang menempatkan pengguna pada server tertentu sesuai dengan kondisi server yang memenuhi kriteria tertentu [8]. *Routing* dinamis dapat dibagi menjadi tiga komponen utama:

1. *Information policy*, yang memutuskan informasi *state* sistem harus dikumpulkan dan bagaimana informasi diperoleh.
2. *Transfer policy*, yang memutuskan apakah pengguna yang datang layak untuk diproses sendiri atau dipindahkan.
3. *Location policy*, yang menentukan node yang layak untuk melayani pengguna yang ditransfer.

Kebijakan informasi dari *routing* dinamis yang dipertimbangkan adalah kebijakan periodik. Artinya, setiap server memberikan informasi kondisi beban (pengguna yang sedang dilayani oleh server tersebut) setiap T waktu kepada semua node lainnya. Dengan asumsi bahwa setiap server mengetahui nilai yang tepat dari bebannya sendiri pada suatu waktu, namun nilai beban pada server lain akan diperkirakan dan

mungkin tidak akurat. Dengan demikian tiap server diasumsikan memiliki nilai taksiran rata-rata waktu layanan dan beban pada semua node.

Aturan *routing* dinamis bekerja seperti berikut. Asumsikan pengguna ditempatkan pada suatu server awal. Kemudian, komponen *transfer policy* diaktifkan untuk menentukan apakah pengguna tersebut memenuhi kriteria untuk dilayani atau dipindah ke server lain. Jika pengguna tersebut dipindah menuju server lain, komponen *location policy* diaktifkan untuk menentukan server yang akan dituju. Penentuan server tersebut menggunakan informasi state sistem yang telah dikumpulkan oleh komponen *information policy*.

Adaptive Virtual Delay (AVD) merupakan salah satu jenis routing dinamis. Kebijakan AVD bertujuan untuk menyeimbangkan delay virtual pada tiap server. Delay virtual tersebut terdiri atas delay virtual layanan dan delay virtual apabila terjadi transfer. Perkiraan delay virtual layanan pada server i dinotasikan sebagai F_i dan delay transfer virtual dinotasikan G . Nilai dari F_i dan G dapat dinyatakan sebagai berikut:

$$F_i = s_i(n_i + 1) \quad (2.29)$$

$$G_{ij} = \frac{t_{ij}}{1 - \rho'_{ij}} \quad (2.30)$$

Keterangan:

- s_i = rata-rata waktu pelayanan server- i ($1/\mu_i$)
- n_i = banyak antrian pada server- i
- t_{ij} = rata-rata waktu tempuh pengguna dari server- i ke server- j
- ρ'_{ij} = tingkat kesibukan jalan antara server- i dan server- j

Delay virtual layanan merupakan total waktu pengguna harus menetap pada server- i jika pengguna itu memasuki server- i pada saat penghitungan. Sementara delay transfer virtual merupakan waktu tempuh antar server apabila pengguna dipindahkan menuju server lain. Adapun deskripsi dari *transfer policy* dan *location policy* untuk AVD sebagai berikut,

Transfer policy adalah perbandingan banyak antrian pada server terhadap batas banyak antrian. Setiap server menggunakan hanya informasi internal untuk menentukan pengguna dilayani atau tidak.

Pengguna akan dipindahkan menuju server lain apabila banyak antrian pada server tersebut melebihi suatu batasan (T_i) dan akan tetap dilayani server tersebut apabila dibawah T_i .

Penentuan nilai T_i pada metode AVD pertama-tama ditentukan T_j sebagai batasan untuk server tercepat-j. Kemudian, untuk server-i ($i \in N, i \neq j$) dengan memasukkan nilai $F_i = F_j$ dan disubstitusikan pada Persamaan (2.29) akan didapat Persamaan (2.31).

$$n_i = \left(\frac{s_i}{s_j} \right) (n_i + 1) - 1 \quad (2.31)$$

Dengan mengganti variabel n_j dan n_i dengan T_j dan T_i secara berturut-turut, maka akan diperoleh nilai batasan untuk semua server.

Location policy merupakan prosedur yang digunakan untuk menentukan server yang melayani pengguna apabila dipindah tempatkan. Prosedur *location policy* dalam AVD antara lain

1. Cari server dengan beban paling ringan.

Pengguna yang datang saat antrian di server-i sedang melebihi batasan. Pada saat tersebut nilai beban di server-i dan server kandidat tujuan j ($j \in N, j \neq i$) dengan aturan berikut

- Jika $F_i > F_j + G_{ij}$, maka server i memiliki lebih sibuk daripada server-j.
- Sebaliknya jika $F_i < F_j + G$, maka server i memiliki tidak lebih sibuk daripada server-j.

Kemudian misalkan $\delta_{ij} = F_i - (F_j + G)$ dan $\delta_i = \max(\delta_{ij})$ Jika δ_i lebih besar dari nol, server-j dengan $\delta_{ij} = \delta_i$ adalah node dengan beban terendah.

2. Mengalokasikan pengguna

Definisikan w sebagai faktor pembobot alokasi dan Δ sebagai nilai bias. Jika $w^c \delta_i > \Delta$ maka pengguna akan dipindah menuju server-j sebanyak c kali. Faktor pembobot w digunakan untuk membatasi pemindahan pengguna. Nilai bias Δ digunakan untuk melindungi sistem dari potensi ketidakstabilan.

2.4 Jaringan Antrian

Jaringan antrian adalah kumpulan sistem antrian yang terhubung dalam suatu jaringan [4]. Pengguna dalam jaringan antrian dapat

dilayani oleh lebih dari satu server sehingga server-server memiliki himpunan pengguna yang sama. Kedatangan pada suatu sistem antrian sering kali berasal dari keberangkat di sistem antrian lainnya.

Jaringan antrian didefinisikan secara lengkap dengan parameter laju kedatangan eksternal, jalur pengguna antar antrian, banyak server, distribusi waktu pelayanan dan disiplin antrian pada tiap antrian. Jaringan dimodelkan sebagai graph dengan node yang mewakili antrian dan edge yang mewakili jalur pengguna. Tiap edge memiliki bobot yang menunjukkan probabilitas *routing*.

Jaringan antrian dapat dikelompokkan menjadi terbuka dan tertutup sesuai sirkulasi pengguna. Pengguna dalam jaringan antrian terbuka dapat berasal dari luar jaringan dan dapat keluar dari jaringan. Sebaliknya, pengguna dalam jaringan tertutup hanya bersirkulasi di dalam jaringan dan banyaknya tetap. Model jaringan sistem rujukan termasuk jaringan terbuka karena pengguna dapat masuk dan keluar dari jaringan.

Jaringan antrian disebut jaringan Markovian apabila waktu pelayanan masing-masing antrian berdistribusi eksponensial, proses kedatangan eksternalnya Poisson, dan transisi antar antrian saling independen. Batasan masalah dalam tugas akhir ini memenuhi kriteria tersebut sehingga model jaringan sistem rujukan dapat disebut Markovian. Analisis jaringan Markovian didasari oleh teorema Burke.

Burke membuktikan bahwa proses keberangkatan dari antrian M/M/n adalah proses Poisson. Secara khusus, teorema Burke menyatakan bahwa untuk antrian M/M/n dalam kondisi *steady state* dengan laju kedatangan λ , hasil berikut akan terpenuhi:

1. Proses keberangkatan adalah proses Poisson dengan laju λ . Jadi, proses keberangkatan secara statistik identik dengan proses kedatangan.
2. Pada setiap waktu t , jumlah pelanggan di sistem independen dari urutan waktu keberangkatan sebelumnya.

Notasikan γ_i mewakili laju kedatangan Poisson eksternal menuju antrian- i . Notasikan p_{ij} sebagai proporsi dari pengguna yang telah dilayani pada antrian- i dan melanjutkan ke antrian- j . p_{i0} menunjukkan proporsi pengguna meninggalkan jaringan setelah dilayani pada antrian- i . Jadi, λ_j yang merupakan laju kedatangan pengguna di antrian- j dapat dinyatakan dalam Persamaan (2.32)

$$\lambda_j = \gamma_j + \sum_{i \neq j} \lambda_i p_{ij} \quad (2.32)$$

Jaringan BCMP adalah salah satu model jaringan antrian yang memiliki penyelesaian product-form. BCMP merupakan pengembangan model Jackson yang memungkinkan *routing* pengguna, distribusi waktu pelayanan dan disiplin antrian yang berbeda untuk tiap jenis pengguna. Disiplin antrian yang dapat dimodelkan dalam jaringan BCMP adalah (1) FCFS dengan distribusi waktu pelayanan yang sama untuk tiap pelanggan, (2) *Processor Sharing* (PS), (3) *Infinite Server* (IS), dan (4) LCFS dengan pengulangan *pre-emptive*.

Dalam suatu jaringan terdapat N antrian dan $R \geq 1$ tipe pengguna. Tiap pengguna dalam network tersebut harus diberi proporsi *routing*. Setiap pengguna bertipe r dapat berubah menjadi tipe s yang menuju node j setelah mendapatkan pelayanan pada suatu node i . Hal ini didefinisikan sebagai $p_{i,r;j,s}$ dan secara akumulatif ditulis dalam matriks routing P dengan entri $\{p_{i,r;j,s}\}$, dengan $i, j = 1, 2, 3, \dots, N$ dan $r, s = 1, 2, 3, \dots, R$.

Kedatangan pengguna baru dari satu stream yang menuju node i dikelompokkan menjadi tipe s dengan proporsi $p_{0, is}$. Pengguna tipe r yang keluar dari jaringan setelah mendapat pelayanan dari node i dengan proporsi $p_{ir, 0}$. Definisikan juga $V_{i,r}$ sebagai proporsi kedatangan pengguna tipe r pada node i maka berlaku,

$$V_{i,r} = p_{0,i,r} + \sum_{s=1}^R \sum_{j=1}^N V_{j,s} \cdot p_{j,s;i,r} \quad (2.33)$$

Probabilitas ada n_i pengguna pada node i dapat dinyatakan dalam,

$$p_i(n_i) = \left(\prod_{j=1}^{n_i} \frac{1}{\mu_i(j)} \right) \left(\sum_{r=1}^R V_{i,r} \right)^{n_i} \quad (2.34)$$

2.5 Sistem Antrian dengan Prioritas

Pada beberapa jenis antrian tidak semua pengguna memiliki urgensi yang sama. Beberapa pengguna membutuhkan perhatian segera sementara yang lain dapat menunggu lebih lama. Dalam situasi tersebut,

pengguna yang datang dapat dikelompokkan dalam kelas prioritas yang berbeda dengan nomor 1 sampai P sehingga prioritas 1 merupakan prioritas tertinggi, kemudian prioritas 2, dan seterusnya, dengan prioritas P menjadi prioritas terendah.

Ada dua jenis dari antrian dengan prioritas, yaitu prioritas *preemptive* dan prioritas *nonpreemptive* [4]. Dalam antrian prioritas *nonpreemptive*, jika pengguna dengan prioritas lebih tinggi datang ketika pengguna dengan prioritas yang lebih rendah sedang dilayani, pengguna prioritas lebih tinggi yang baru datang akan menunggu sampai pelayanan yang sedang dilakukan selesai. Dengan demikian, setiap pengguna yang sedang mendapat layanan akan menyelesaikan layanan tanpa gangguan.

Dalam antrian prioritas *preemptive*, jika pengguna dengan prioritas lebih tinggi datang ketika pengguna dengan prioritas yang lebih rendah sedang dilayani, pengguna prioritas lebih tinggi yang baru datang akan langsung diberikan layanan mendahului pengguna yang sedang dilayani. Kemudian pelanggan sebelumnya dilayani kembali apabila tidak ada pengguna dengan prioritas lebih tinggi. Layanan dapat dilanjutkan dari saat didahului maupun diulang dari awal.

Asumsikan ada p kelas prioritas $P=\{1,2,\dots,p\}$ yang datang dengan laju kedatangan λ_p , dan waktu pelayanan pengguna kelas p memiliki distribusi umum dengan mean $\bar{T}_p = E[T_p]$. Kemudian definisikan,

$$\rho_p = \lambda_p \cdot \bar{T}_p \quad (2.35)$$

$$\lambda = \sum_{i=1}^p \lambda_i \quad (2.36)$$

$$\bar{X} = \sum_{i=1}^p \frac{\rho_i}{\lambda} \quad (2.37)$$

$$\sigma_k = \sum_{i=1}^k \rho_i \quad (2.38)$$

Keterangan:

- λ_p = Laju kedatangan kelas p menuju server
- $\bar{T}_p$ = Rata2 waktu pelayanan kelas p

- ρ_p = Utilisasi server oleh kelas p
- λ = Laju kedatangan keseluruhan menuju server
- $\bar{T}$ = Rata2 waktu pelayanan keseluruhan
- σ_k = Utilisasi server oleh kelas 1 sampai k

Asumsikan bahwa pengguna yang sedang dilayani adalah pengguna kelas p dengan sisa waktu layanan adalah R_p . Ekspektasi sisa waktu layanan kelas p dituliskan pada Persamaan (2.39)

$$E(R_p) = \frac{E(T_p^2)}{2E(T_p)} \quad (2.39)$$

Perhatikan saat pengguna kelas 1 datang ke dalam sistem. Asumsikan banyak pengguna kelas 1 yang sedang mengantri adalah L_1 . Maka ekspektasi waktu tunggu pengguna kelas 1:

$$E[W_1] = E[L_1]E[T_1] + \sum_{i=1}^p \rho_i E[R_i] \quad (2.40)$$

Definisikan $E(R)$ sebagai ekspektasi sisa waktu layanan apapun tipe pengguna yang sedang dilayani saat itu. $E(R)$ dapat dinyatakan sebagai,

$$E(R) = \frac{1}{2} \sum_{i=1}^p \lambda_i E(T_i^2) \quad (2.41)$$

Persamaan (2.40) dapat ditulis ulang menjadi,

$$E(W_1) = E(L_1)E(T_1) + E(R) = \frac{E(R)}{1 - \sigma_1} \quad (2.42)$$

Untuk tingkat prioritas $p > 1$ selanjutnya,

$$E(W_p) = \sum_{i=1}^p E(L_i)E(T_i) + E(R) \quad (2.43)$$

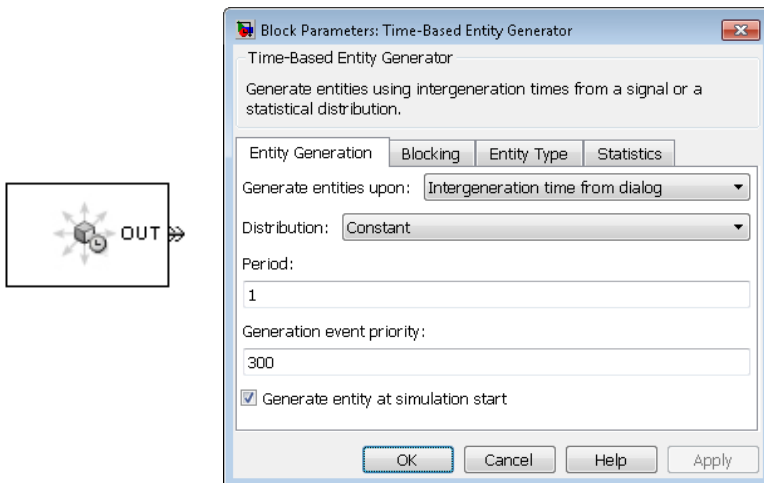
Dengan cara yang sama akan didapat persamaan (2.44)

$$E(W_p) = \frac{E(R)}{(1 - \sigma_{p-1})(1 - \sigma_p)} \quad (2.44)$$

2.6 Simulasi Sistem Antrian pada MATLAB

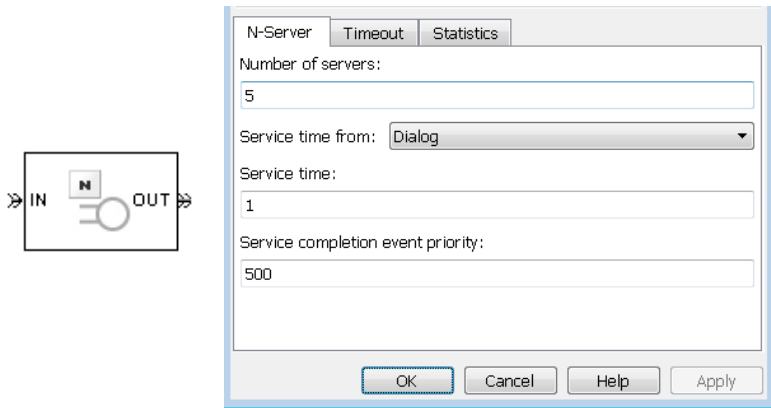
Rancangan sistem antrian akan direalisasikan menggunakan SimuLink pada MATLAB. Simulasi sistem antrian dapat menggunakan toolbox SimEvent yang terdapat pada Simulink. Simevent merupakan toolbox Simulink yang berfungsi menyimulasikan sistem even diskrit seperti proses manufaktur maupun sistem antrian.

Simulasi sistem antrian membutuhkan 3 komponen utama yaitu pengguna, antrian dan server. Pengguna dibentuk melalui blok *entity generator*. Antrian dapat direpresentasikan melalui beberapa blok *queue* sesuai disiplin antriannya. Pada tugas akhir ini dipergunakan blok *FIFO queue* dan *Priority queue*. Kemudian, server direpresentasikan dengan blok *N-server*.



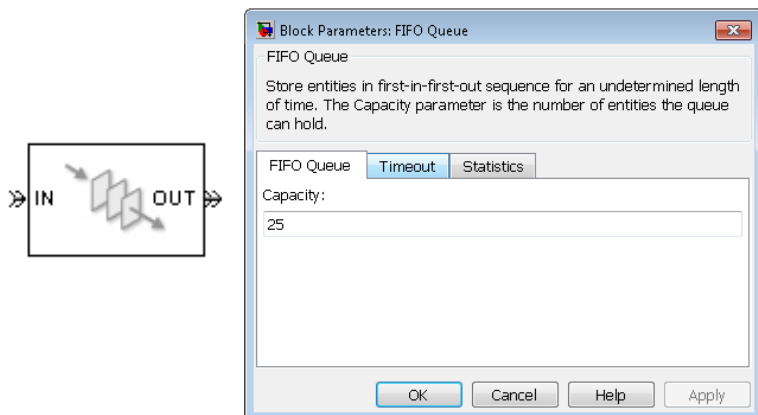
Gambar 2.4 Blok *time-based entity generator* dan kotak dialognya

Blok *time-based entity generator* membentuk entitas atau pengguna menggunakan waktu antar generasi yang dapat berasal dari sinyal atau distribusi statistik. Waktu antar generasi adalah interval waktu antara dua kedatangan pengguna. Nilai waktu antar generasi dapat diatur dalam dialog blok tersebut maupun dari sinyal eksternal. Bentuk blok *time-based entity generator* dan dialognya dapat dilihat pada Gambar 2.4.



Gambar 2.5. Blok *N-server* dan kotak dialognya

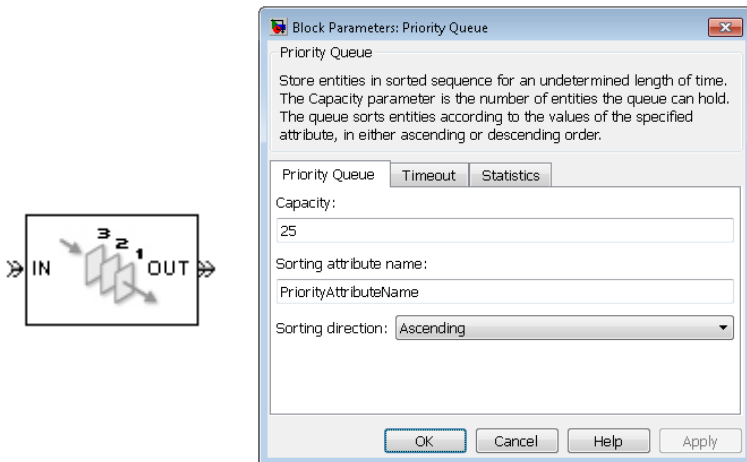
Waktu antar kedatangan tiap entitas dapat diatur sesuai proses kedatangan entitas. Pengaturan waktu antar kedatangan dapat dilakukan melalui dialog tersebut dengan mengganti parameter distribusi sesuai dengan proses kedatangan entitas. Pada Gambar 2.4, proses kedatangan berdistribusi konstan dengan waktu antar kedatangan 1 detik.



Gambar 2.6 Blok *FIFO queue* dan kotak dialognya

Blok *N-server* melayani entitas atau entitas sampai N entitas dalam satu waktu. Blok ini dapat menyimpan sampai N entitas, melayani masing-masing entitas secara terpisah dan independen untuk waktu pelayanan tertentu. Entitas yang selesai dilayani kemudian keluar melalui port OUT. N-server berperilaku seperti N server tunggal yang terhubung secara paralel. Bentuk blok *N-server* dan dialognya dapat dilihat pada Gambar 2.5.

Waktu pelayanan atau durasi pelayanan dapat ditentukan melalui parameter, atribut entitas, atau sinyal eksternal. Hal tersebut dapat di menggunakan kotak dialog blok *N-server*. Blok *N-server* menentukan waktu layanan untuk suatu entitas saat datang pada blok dan diasumsikan dalam satuan detik.



Gambar 2.7 Blok *priority queue* dan kotak dialognya

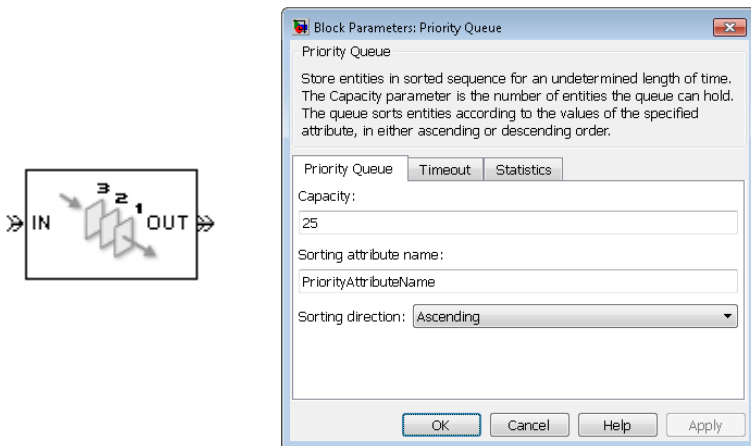
Port IN akan terhalang jika blok ini sudah berisi N entitas. Port IN menjadi terbuka kembali jika setidaknya salah satu dari N entitas sudah keluar. Kondisi port IN terhalang mengindikasikan terjadinya antrian. Untuk menampung entitas yang ingin masuk pada server tersebut dibutuhkan blok *queue*. Berikut ini akan dibahas mengenai dua jenis blok *queue* yaitu *FIFO queue* dan *Priority queue*.

Blok *FIFO queue* menyimpan entitas dengan urutan tertentu sesuai disiplin FIFO. Blok ini menyimpan entitas sesuai kapasitasnya.

Kapasitas blok menyatakan kapasitas antrian pada suatu sistem antrian. Kapasitas blok dapat ditentukan dengan mengisi parameter pada kotak dialognya. Blok ini selalu mengeluarkan entitas melalui port OUT selama tidak terhalang (karena server penuh), tapi menyimpan entitas jika port OUT terhalang. Entitas keluar dari blok ini sesuai urutan masuknya sesuai disiplin antrian FIFO. Bentuk blok *FIFO queue* dan dialognya dapat dilihat pada Gambar 2.6.

Blok *priority queue* menyimpan entitas dengan urutan sesuai prioritas entitas. Blok ini menyimpan entitas sesuai kapasitasnya. Kapasitas blok dapat diatur pada kotak dialog. Perbedaan blok ini dengan *FIFO queue* adalah entitas yang keluar dari blok ini sesuai urutan prioritasnya. Ketika prioritas antara dua entitas sama maka urutan kedua entitas tersebut mengikuti disiplin antrian FIFO. Bentuk blok *priority queue* dan dialognya dapat dilihat pada Gambar 2.7

Penentuan urutan prioritas dari suatu entitas berdasarkan nilai dari suatu atribut. Pemberian atribut pada suatu entitas dapat dilakukan menggunakan blok *set attribute*. Blok tersebut memberikan atribut dengan nama tertentu pada suatu entitas beserta nilai dari atribut tersebut. Nilai atribut dapat ditentukan pada dialog maupun berasal dari sinyal eksternal. Selain untuk menentukan prioritas, atribut juga dapat digunakan untuk menentukan waktu pelayanan. Bentuk blok *set attribute* dan dialognya dapat dilihat pada Gambar 2.8



Gambar 2.8 Blok *set attribute* dan kotak dialognya

Halaman ini sengaja dikosongkan

BAB 3

PERANCANGAN KONSEPTUAL

Pada bagian ini akan dibahas perancangan model jaringan antrian yang merepresentasikan sistem rujukan berjenjang BPJS Surabaya di lingkup Dinas Kesehatan Surabaya area Surabaya Timur. Bagian pertama akan membahas sistem rujukan berjenjang yang akan dimodelkan. Bagian kedua membahas permodelan jaringan antrian sistem rujukan berjenjang. Bagian ketiga akan membahas model simulasi yang dipergunakan.

3.1 Sistem Rujukan Berjenjang BPJS

Sistem rujukan dalam pelayanan kesehatan adalah penyelenggaraan pelayanan kesehatan yang mengatur pelimpahan tugas pelayanan kesehatan secara timbal balik yang dilaksanakan oleh peserta jaminan kesehatan nasional (JKN) dan fasilitas kesehatan yang tergabung dalam penjaminan kesehatan BPJS. Peserta JKN yang ingin mendapatkan penjaminan kesehatan harus mendapatkan pelayanan kesehatan sesuai prosedur sistem rujukan berjenjang ini. Sistem rujukan berjenjang diterapkan BPJS untuk mengatur peserta JKN dan menerapkan konsep fasilitas kesehatan tingkat I sebagai *gatekeeper*.

Fasilitas kesehatan adalah satuan yang menyelenggarakan pelayanan kesehatan perorangan. Fasilitas kesehatan (faskes) dibagi 2 jenis sesuai jenis pelayanan yang diberikan yaitu, faskes tingkat pertama dan faskes tingkat lanjutan.

Pelayanan kesehatan dibagi menjadi 3 jenis yaitu,

1. Pelayanan kesehatan tingkat pertama,
2. Pelayanan kesehatan tingkat kedua (spesialistik), dan
3. Pelayanan kesehatan tingkat ketiga (sub-spesialistik).

Pelayanan kesehatan tingkat pertama merupakan pelayanan kesehatan dasar yang diberikan fasilitas kesehatan tingkat pertama. Pelayanan kesehatan spesialistik merupakan pelayanan kesehatan spesialistik yang dilakukan oleh dokter spesialis yang menggunakan pengetahuan atau teknologi spesialistik. Pelayanan kesehatan sub-spesialistik merupakan pelayanan kesehatan sub-spesialistik yang dilakukan oleh dokter spesialis yang menggunakan pengetahuan atau teknologi sub-spesialistik.

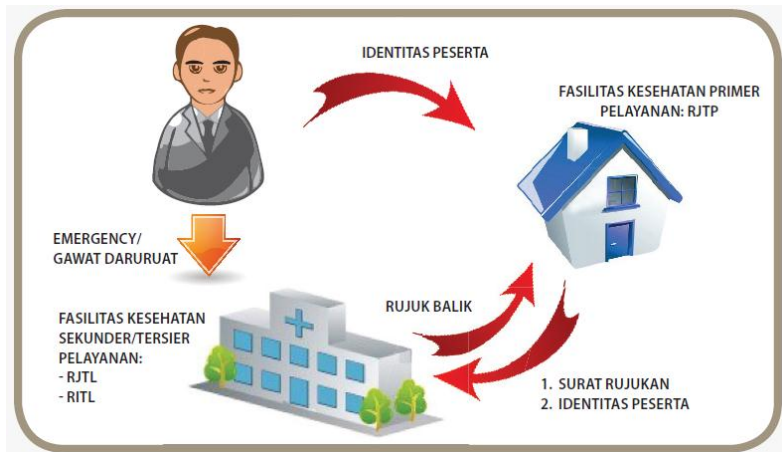
Rujukan merupakan pelimpahan tugas dan tanggung jawab pelayanan kesehatan antar fasilitas kesehatan. Rujukan dibagi menjadi dua jenis yaitu, rujukan vertikal dan horizontal. Model yang dibentuk pada tugas akhir ini hanya memperhatikan rujukan vertikal. Rujukan vertikal adalah rujukan yang dilakukan antar fasilitas kesehatan dengan tingkat pelayanan yang berbeda. Rujukan vertikal dilakukan apabila

1. Pasien membutuhkan pelayanan kesehatan spesialistik atau sub-spesialistik
2. Perujuk tidak dapat melakukan pelayanan kesehatan sesuai kebutuhan pasien
3. Sistem rujukan berjenjang dilakukan secara berjenjang sesuai kebutuhan medis peserta JKN. Prosedur pelaksanaan sistem rujukan berjenjang adalah sebagai berikut:
4. Peserta JKN mendapatkan pelayanan kesehatan dimulai dari faskes tingkat pertama.
5. Jika diperlukan pelayanan kesehatan spesialis, pasien dapat dirujuk ke faskes lanjutan yang menyediakan pelayanan kesehatan spesialis.
6. Jika diperlukan pelayanan kesehatan sub-spesialis, pasien dapat dirujuk ke faskes lanjutan yang menyediakan pelayanan kesehatan sub-spesialis.
7. Peserta JKN dapat dirujuk secara langsung dari faskes tingkat I menuju faskes tingkat lanjutan yang menyediakan pelayanan kesehatan sub-spesialis apabila hasil diagnosis menyatakan peserta harus diberikan pelayanan kesehatan sub-spesialis.

Prosedur rujukan berjenjang dapat diabaikan dalam kondisi khusus antara lain, kondisi gawat darurat, bencana, pertimbangan geografis peserta. Gambaran prosedur sistem rujukan berjenjang dapat dilihat pada Gambar 3.1

3.2 Identifikasi Permasalahan

Sistem rujukan merupakan sistem pelayanan yang terdistribusi secara spasial dalam kota. Pengguna dalam sistem rujukan berjenjang adalah pasien BPJS. Sumber kedatangan pasien dapat dipisahkan menurut lokasi geografisnya. Pasien memiliki preferensi yang berbeda terhadap rumah sakit yang akan dituju. Pertimbangan pasien dapat bergantung pada kelengkapan fasilitas, jarak tempuh dan tingkat pelayanan.



Gambar 3.1 Prosedur Sistem Rujukan Berjenjang

Pembagian pasien menurut jenis penyakitnya membuat model antrian dapat dilihat sebagai sistem M/M/n. Hal ini dikarenakan sistem yang telah terbagi ke dalam jenis penyakit ini memenuhi asumsi yang telah disebutkan pada bab II. Adapun asumsi yang terpenuhi antara lain,

1. Waktu antar-kedatangan pasien dari masing-masing area geografis memiliki distribusi eksponensial karena melalui jaringan poisson.
2. Hanya satu poli yang melayani 1 pasien dalam satu layanan.
3. Waktu pelayanan poli untuk jenis penyakit yang sama relatif sama sehingga dapat diasumsikan memiliki distribusi eksponensial dengan rata-rata yang sama ($1/\mu$).
4. Waktu pelayanan tidak dipengaruhi asal RS maupun pasien dan kondisi sebelumnya pada sistem.
5. Kapasitas antrian tak berhingga sehingga ketika semua rumah sakit sedang sibuk pasien masuk dalam antrian yang sama dan kemudian akan dilayani dengan disiplin FCFS.

3.3 Pemodelan Sistem Rujukan Berjenjang

Jaringan antrian adalah kumpulan sistem antrian yang saling terhubung. Perhatikan bahwa tiap faskes dalam sistem pelayanan kesehatan dapat ditinjau sebagai suatu sistem antrian yang terhubung melalui sistem rujukan berjenjang. Peserta JKN yang dirujuk dari faskes

I akan melanjutkan antrian pada faskes tingkat lanjutan. Berdasarkan fakta tersebut sistem rujukan berjenjang dapat dimodelkan sebagai suatu jaringan antrian. Pembahasan model jaringan antrian dari sistem rujukan berjenjang dibagi menjadi 3 sub-bagian antara lain sistem antrian M/M/n/-/P untuk faskes tingkat I, proporsi rujukan yang dimodelkan dengan proses Poisson, dan perujukan pasien dengan menggunakan model antrian *hypercube*. Hal tersebut digambarkan dalam skema di Gambar 3.2.

Gambar 3.2 menunjukkan bahwa pasien yang telah mendapatkan perawatan maupun diagnosis di faskes I akan diberi rekomendasi rujukan oleh BPJS sesuai kebutuhan penyakitnya. BPJS akan memberi rekomendasi dengan metode *routing* dinamis. BPJS memperhatikan kondisi *real-time* banyak pasien pada tiap poli penyakit milik rumah sakit.

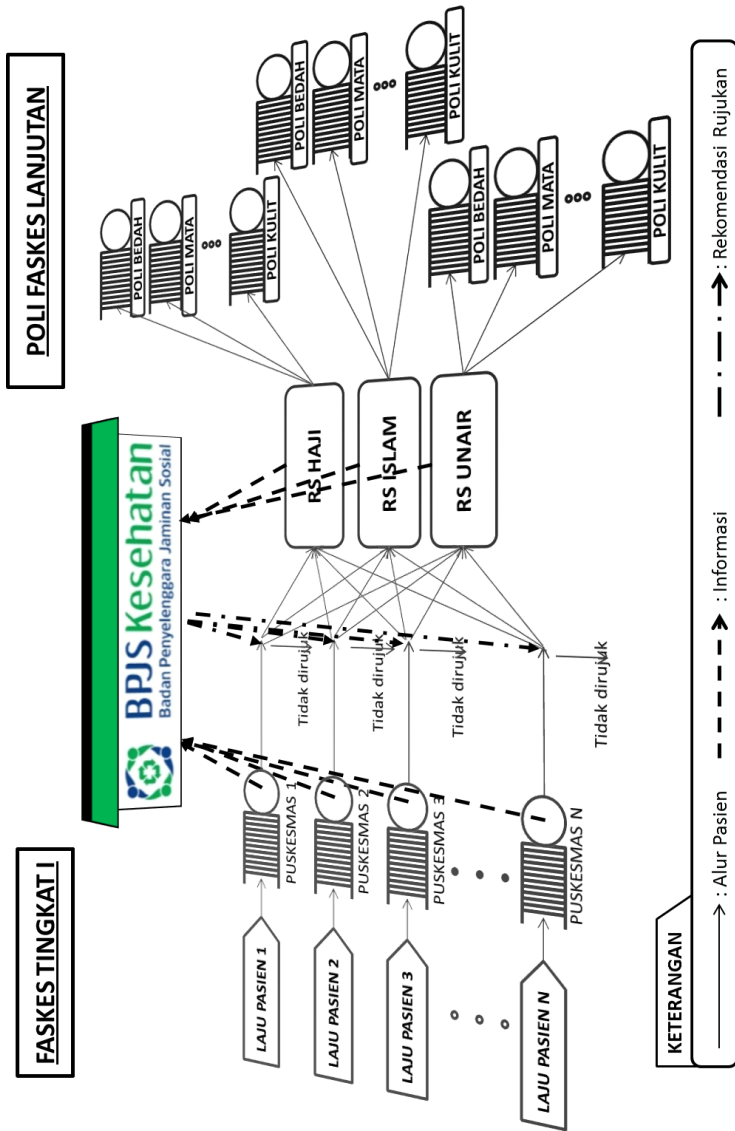
Pasien yang dirujuk datang menuju rumah sakit dan kemudian menuju poli-poli sesuai jenis penyakitnya. Antrian di loket rumah sakit untuk registrasi diabaikan karena antrian dianggap stabil sehingga laju keberangkatan sama dengan laju kedatangan. Antrian di faskes tingkat I dan antrian di poli rumah sakit dapat dianggap sebagai antrian yang independen dengan menggunakan teorema Burke.

Laju kedatangan pada antrian di poli rumah sakit merupakan laju keberangkatan dari faskes tingkat I sesuai dengan teorema Burke. Laju kedatangan pada tiap poli adalah pasien yang dirujuk dari tiap puskesmas dengan proporsi sesuai penyakitnya. Laju kedatangan tersebut dipergunakan dalam model *hypercube* dari sistem rujukan berjenjang.

3.3.1 Pembagian Kelompok Penyakit Pasien

Pasien yang ingin mendapatkan rujukan harus mendapatkan pelayanan di faskes tingkat I terlebih dahulu. Pasien akan dilayani oleh dokter di faskes tersebut. Kemudian, pasien akan mendapatkan rujukan apabila diperlukan pelayanan spesialis maupun sub-spesialis di faskes lanjutan.

Pasien datang menuju faskes tingkat I diasumsikan sebagai proses Poisson. Waktu antar pelayanan di faskes tingkat I diasumsikan terdistribusi eksponensial. Faskes tingkat I dapat dilihat sebagai suatu sistem antrian yang proses kedatangan dan pelayanannya merupakan proses Poisson. Maka, menurut Teorema Burke proses keluarnya pasien dari faskes tingkat I merupakan suatu proses Poisson.



Gambar 3.2 Skema Konsep Sistem Rujukan

Pasien keluar dari faskes tingkat I dapat dibagi menjadi dua yaitu pasien yang dirujuk dan yang tidak dirujuk. Kemudian pasien yang dirujuk dapat dikelompokkan menurut jenis penyakitnya. Pasien yang dirujuk dikelompokkan menjadi 9 kelompok sesuai jenis penyakit terbanyak di Surabaya. Adapun 9 jenis penyakit tersebut antara lain,

- A1: Jantung
- A2: Mata
- A3: THT
- A4: Penyakit Dalam
- A5: Kandungan
- A6: Paru
- A7: Gigi
- A8: Saraf
- A9: Kulit & Kelamin

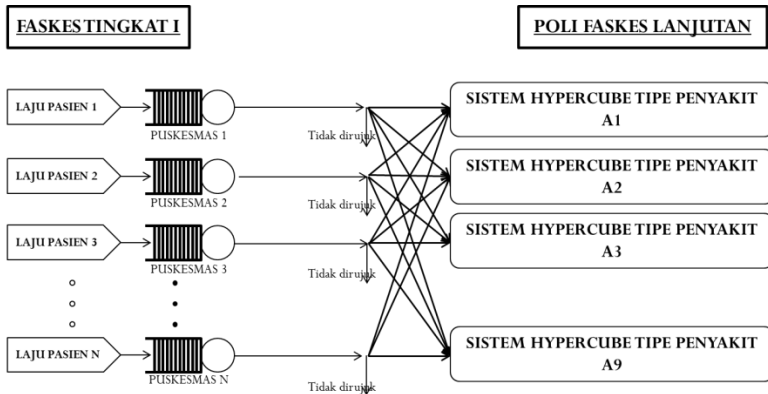
Pasien dikelompokkan dengan proporsi tertentu yang diperoleh dari data BPJS. Proses keluarnya pasien dari faskes adalah proses poisson sehingga proses keluarnya pasien sesuai pengelompokan tersebut juga merupakan proses poisson dengan laju yang mengikuti proporsi tersebut. Proses yang sudah dikelompokkan sesuai penyakit akan digunakan sebagai proses kedatangan pada sistem antrian selanjutnya (model *hypercube*) sebagai proses Poisson.

3.3.2 Pemodelan Sistem Rujukan sebagai Sistem Antrian Hypercube

Model *hypercube* dipergunakan karena dapat mengikut sertakan preferensi pengguna terhadap pilihan server yang akan melayaninya. Selain itu, model *hypercube* memiliki konsep *workload sharing*. Konsep ini cocok dengan sistem rujukan berjenjang karena pasien memiliki preferensi rujukan dan ketika rumah sakit yang paling diinginkan sedang sibuk maka pasien akan berpindah ke rumah sakit lain.

3.3.2.1 Model Hypercube Sistem Rujukan

Rujukan akan dimodelkan menjadi 9 model antrian hypercube sesuai dengan 9 jenis penyakit terbanyak di Surabaya. Pembagian pasien ke dalam 9 jenis penyakit dapat dilakukan karena preferensi pasien akan lebih beragam untuk jenis penyakit yang sama. Skema model antrian hypercube dalam jaringan antrian sistem rujukan ditunjukkan pada Gambar 3.3



Gambar 3.3 Skema Estimasi Menggunakan Model *Hypercube*

Sistem yang akan diamati merupakan sistem rujukan di Surabaya Timur. Server yang dimodelkan berasal dari 3 sumber yaitu RS Haji, RS Islam dan RS Unair. RS Haji dan RS Islam dipilih karena merupakan 2 Rumah Sakit terfavorit di Surabaya Timur. Sementara RS Unair dipilih karena paling tidak favorit sehingga dapat menyeimbangkan beban kerja dalam sistem. Server merupakan dokter yang beroperasi pada poli yang bersesuaian dengan tipe penyakit yang dalam model. Dengan demikian, banyak server dalam suatu model dapat melebihi 3 server apabila pada suatu rumah sakit terdapat lebih dari satu dokter.

Dalam model antrian yang diusulkan daerah di Surabaya Timur dibagi menjadi 13 area sejumlah puskesmas yang ada di area Surabaya Timur. Asumsi ini diambil karena 13 puskesmas ini memiliki pengaruh yang besar pada rujukan di rumah sakit Haji, Islam dan Universitas Airlangga. Adapun ke-13 puskesmas tersebut antara lain,

1. Puskesmas Gading
2. Puskesmas Gununganyar
3. Puskesmas Kalijudan
4. Puskesmas Kalirungkut
5. Puskesmas Keputih
6. Puskesmas Klampis Ngasem
7. Puskesmas Medokan Ayu
8. Puskesmas Menur
9. Puskesmas Mulyorejo
10. Puskesmas Mojo

11. Puskesmas Pacarkeling
12. Puskesmas Rangkah
13. Puskesmas Tenggilis

Skema pembagian atom geografis untuk sistem ini dapat dilihat pada Gambar 3.4



Gambar 3.4 Skema Pembagian Atom Geografis dan Posisi Server

Notasikan H_i ($i = 1, 2, 3, \dots, 9$) sebagai model hypercube untuk penyakit A_i . Misalkan λ_i adalah laju kedatangan keseluruhan Pengguna pada tiap model adalah pasien rujukan dari faskes I yang telah dikelompokkan berdasarkan jenis penyakitnya. Untuk penyakit ke- i dan

atom geografis k , laju kedatangan pasien dinotasikan sebagai $\lambda_{ik}; k = 1, 2, \dots, 18$ sehingga nilai λ_i dapat dituliskan sebagai berikut

$$\lambda_i = \sum_{k=1}^{18} \lambda_{ik} \quad (3.1)$$

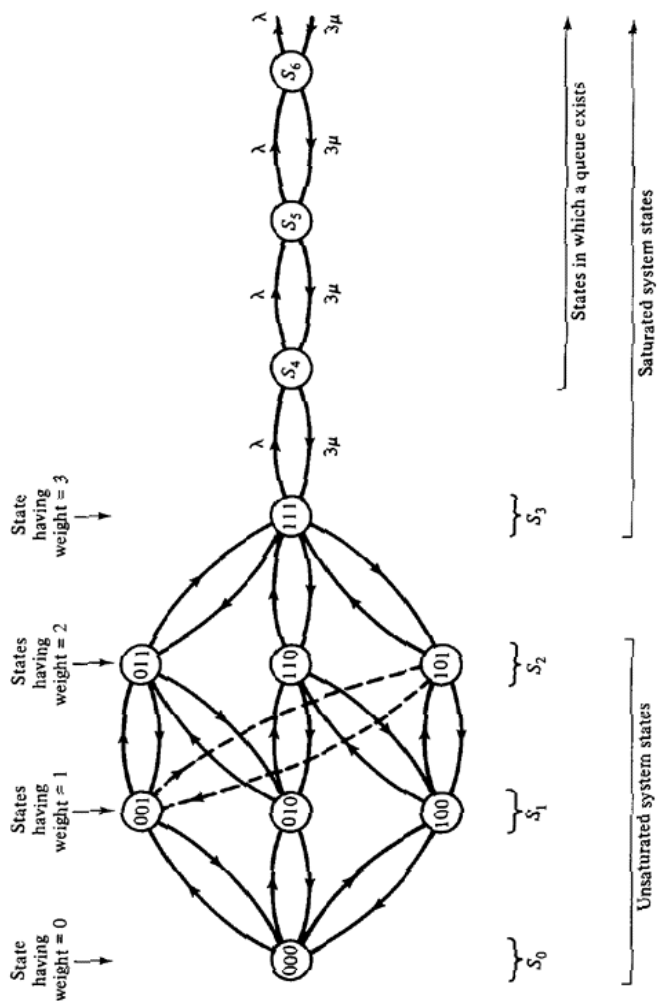
Server pada tiap model adalah dokter yang menangani jenis penyakit tersebut pada tiap poli di rumah sakit. Notasikan $n_{il}; l = 1, 2, 3$ adalah banyak dokter yang melayani penyakit i di rumah sakit l . Maka, nilai n_i yang merupakan total banyak dokter pada H_i adalah,

$$n_i = \sum_{l=1}^3 n_{il} \quad (3.2)$$

Rumah sakit yang tidak memiliki dokter yang dapat melayani jenis penyakit- i tidak dimasukkan dalam model penyakit ke- i .

Pasien dirujuk pada suatu rumah sakit berdasarkan tingkat preferensi yang dinyatakan dalam matriks preferensi. Notasikan matriks preferensi sebagai P_i . Matriks preferensi pada model H_i merupakan matriks berukuran $18 \times n_i$. Pada model hypercube matriks preferensi berisi nilai preferensi server terhadap pengguna yang akan dilayani. Namun dengan membalikkan alur penempatan pada model sistem rujukan ini, matriks preferensi berisi nilai urutan tingkat preferensi pasien terhadap dokter di rumah sakit. Entri $p_{km}; k = 1, 2, \dots, 18, m = 1, 2, \dots, n_i$ dalam matriks preferensi menunjukkan urutan preferensi pasien dari puskesmas dengan indeks k terhadap dokter dengan indeks m . Nilai matriks P_i ditentukan berdasarkan banyak pasien yang dirujuk menuju rumah sakit tersebut.

State dalam model *hypercube* menunjukkan kondisi dokter- l sedang memeriksa atau memberi pengobatan pada pasien. Transisi state menunjukkan seorang dokter selesai melayani pasien atau ada pasien datang diperiksa oleh dokter yang sedang menganggur. Diagram transisi state untuk model H_i dapat dilihat pada Gambar 3.5



S_i = State depiction of M/M/3 model, corresponding to exactly i customers on the system

Gambar 3.5 Diagram Transisi State Model *Hypcube* H_i [6]

3.3.2.2 Modifikasi Model Antrian Hypercube

Model antrian *hypercube* umumnya dipergunakan untuk antrian dalam perkotaan yang memiliki server yang bergerak dan user yang diam pada suatu tempat. Contoh antrian dengan sifat tersebut antara lain sistem layanan ambulans, sistem layanan pesan antar dan sistem patroli polisi. Sistem rujukan berjenjang merupakan antrian dengan user yang bergerak dan server yang diam. Oleh karena itu, permodelan sistem rujukan berjenjang sebagai antrian *hypercube* membutuhkan beberapa modifikasi.

Data waktu tempuh perjalanan memiliki peran penting dalam model. Waktu tempuh didefinisikan pada matriks waktu tempuh t_{ij} . Elemen t_{ij} menunjukkan rata-rata waktu tempuh dari area i ke area j . Nilai t_{ij} dengan t_{ji} tidak selalu sama. Hal ini disebabkan kompleksitas jalur seperti jalan satu arah, penutupan jalan, kemacetan dan sebagainya. Posisi dari server direpresentasikan dalam matriks lokasi L . Elemen l_{ij} menunjukkan probabilitas posisi server- i ketika sedang tidak sibuk sehingga untuk semua server i , $\sum_{j=1}^K l_{ij} = 1$. Pada kondisi server diam, l_{ij} bernilai 1 untuk area j tertentu dan bernilai 0 untuk area selain j ($l_{ik} = 0, k \neq j$).

Pada kondisi server bergerak, waktu tempuh didefinisikan sebagai waktu yang dipergunakan server untuk menuju area tertentu sehingga

$$T_{nj} = \sum_{i=1}^K l_{ni} \cdot t_{ij} \quad (3.3)$$

Pada kondisi server diam, hanya satu l_{ij} yang bernilai 1 sehingga $T_{nj} = t_{ij}$. Dimana i merupakan suatu area yang ditempati server- n . T_{nj} juga merepresentasikan waktu tempuh pasien dari area j menuju server n . Pada kondisi server bergerak terdapat waktu tempuh saat terjadi antrian yang dinotasikan sebagai T_Q . Namun, pada kondisi server diam antrian tidak menimbulkan adanya waktu tempuh tersebut. Antrian terjadi pada satu titik saja dan pasien tetap berangkat dari area yang sama yaitu server sehingga,

$$T_Q = \sum_{n=1}^N f_{nj} \cdot t_{ji} \quad (3.4)$$

Dimana i merupakan area server- n bertempat. rata-rata waktu tempuh keseluruhan adalah,

$$T = \sum_{n=1}^N \sum_{j=1}^K f_{nj} \cdot T_{nj} \quad (3.5)$$

Dengan cara yang sama maka rata-rata waktu tempuh pasien dari area i dapat dihitung dengan,

$$T_j = \frac{\sum_{n=1}^N f_{nj} T_{nj}}{\sum_{n=1}^N f_{nj}} \quad (3.6)$$

Rata-rata waktu tempuh yang dapat diamati juga adalah pasien pada suatu server n yang dapat dihitung dengan,

$$T_n = \frac{\sum_{j=1}^K f_{nj} T_{nj}}{\sum_{j=1}^K f_{nj}} \quad (3.7)$$

3.4 Perancangan Algoritma Perkiraan Performa Antrian Model Hypercube

Tiap penyakit dimodelkan menjadi sebuah model antrian hypercube. Untuk 3 jenis rumah sakit yang berbeda apabila pada suatu rumah sakit terdapat lebih dari 1 dokter yang bekerja melayani penyakit tersebut maka akan terdapat $n > 3$ dokter yang menjadi server dalam model tersebut. Kompleksitas perhitungan untuk model tersebut memerlukan algoritma penyelesaian yang mengestimasi nilai beban kerja tiap server.

Tinjau kembali Persamaan (2.28) pada bab II. Agar persamaan tersebut sesuai dengan model yang diusulkan perlu ditambahkan indeks model hypercube dari tipe penyakit. Persamaan (2.28) dapat ditulis kembali sebagai Persamaan (3.8).

$$1 - \rho_{ij} = \left[\sum_{j \in G_i^1} \lambda_{ij} + \sum_{j \in G_i^2} \lambda_{ij} \cdot Q(N, \rho, 1) \rho_{n_{j1}} + \sum_{k \in G_i^3} \lambda_{ij} \cdot Q(N, \rho, 2) \rho_{n_{j1}} \cdot \rho_{n_{j2}} + \dots \right. \\ \left. + \sum_{j \in G_i^N} \lambda_{ij} \cdot Q(N, \rho, N-1) \rho_{n_{j1}} \cdot \rho_{n_{j2}} \dots \rho_{n_{j(N-1)}} + \frac{\lambda_i}{N(1 - \rho_{ij})} \right]^{-1} \quad (3.8)$$

Algoritma untuk mengestimasi beban kerja server pada model H_i dijabarkan dalam pseudocode berikut:

Algoritma Estimasi Beban kerja

Deklarasi

$N1, N2, s := \text{integer}$

$\mu, L, u, \Gamma, e, \text{abs_e}, \bar{\rho}, \text{var_}\rho, \text{dev}, \sigma, \pi, := \text{real}$

$\text{Pref}, \text{Fd} := \text{array } 18 \times N2 \text{ of integer}$

$\text{elmb} := \text{array } 1 \times 3 \text{ of real}$

$D := \text{array } 1 \times 3 \text{ of integer}$

$\lambda, \text{Frac_L} := \text{array } 1 \times 18 \text{ of real}$

$\text{Travel} := \text{array } 18 \times 3 \text{ of real}$

$P, \text{work}, \text{pref} := \text{array } 1 \times N2 \text{ of real}$

Deskripsi

Input ($\lambda, \mu, \text{Pref}, \text{Travel}, D$);

Inisiasi Parameter

$N2 \leftarrow \sum D[i];$

$L \leftarrow \sum \lambda[i];$ (Hitung lambda keseluruhan)

Hitung utilitas dan probabilitas state untuk sistem antrian M/M/n

$P[i, u] \leftarrow \text{MMn_queue}(L, \mu, N2);$ (Hitung probabilitas state dan utilitas)

for $i=1$ to $N2$ do

$\text{frac_L}[i] \leftarrow \lambda[i]/L;$ (Hitung proporsi laju kedatangan dari masing2 atom)

endfor;

Inisialisasi Iterasi

$n \leftarrow 0;$ (Inisiasi counter iterasi)

for $i=1:N2$ do

$\text{work}[i] \leftarrow u;$ (Inisiasi beban kerja awal sebagai rata2 beban kerja)

endfor;

Proses Iterasi

while

$n \leftarrow n+1;$

$\text{prev} \leftarrow \text{work};$ (Menyimpan estimasi iterasi sebelumnya)

for $i=1$ to $N2$ do

Update load dengan Persamaan 3.3

$\text{work}[i] \leftarrow$ Hitung nilai beban kerja selanjutnya menggunakan persamaan 3.12 dengan menyubstitusikan nilai ρ_{ij} pada ruas kanan dengan $\text{prev}[i];$

endfor;


```

 $\Gamma \leftarrow \frac{N2 \cdot u}{\sum work[i]}$ ; (Hitung koefisien normalisasi)
work[i]  $\leftarrow \Gamma \cdot work[i]$ ; (Normalisasi)
e  $\leftarrow \max(|work[i] - prev[i]|)$ ;
if e <  $10^{-6}$  or n = 1000
    break;
endif;
endwhile;
Menghitung ketidakseimbangan beban kerja
abs_e  $\leftarrow \max(|work[i] - u|)$ ;
 $\bar{\rho} \leftarrow u$ ;
 $var\_p \leftarrow \frac{\sum work[i]^2}{N2} - u^2$ ; (Hitung Varians)
 $\sigma \leftarrow \sqrt{var\_p}$ ; (Standar Deviasi)
 $dev \leftarrow \frac{\max(|work[i] - u|) 100}{u}$ ; (Persentase deviasi maksimum dari mean)
Mengestimasi proporsi routing dari tiap atom menuju server
for i = 1 to 18 do
    for j = 1 to N2 do
         $\pi \leftarrow 1$ ;
        for k = 1 to j - 1 do
             $\pi \leftarrow work[pref[i, k]]$ 
        endfor;
         $Fd[i, j] \leftarrow \frac{L[i].qcf(N2, \rho, j) \cdot \pi \cdot (1 - work[pref[i, j]])}{N2}$ 
    endfor
endfor
Normalisasi
for i = 1 to 18 do
     $norm[i] \leftarrow \frac{L[i] \cdot (1 - P[N2 + 2])}{\sum Fd[i]}$ ;
endfor;
for i = 1 to 18 do
     $Fd[i, j] \leftarrow \frac{Fd[i, j]}{norm[i]}$ ;
endfor;
Menghitung performa antrian masing-masing rumah sakit
%Mendapatkan laju kedatangan pada masing-masing server
 $\lambda_s[i] \leftarrow work[i] \cdot \mu$ ;
%Menjumlahkan laju kedatangan menuju server-server dalam satu poli

```

```

K ← 1;
for i=1 to 3 do
    elmb[i] ← jumlah  $\lambda_s[i]$  dengan indeks k sampai k+D[i]-1;
    k ← k+D(i);
endfor;
for i=1 to 3 do
    Pp, Up ← MMn_queue(elmb[i],  $\mu$ , D[i]);
     $N_q[i] \leftarrow \frac{Pp(D[i]+2)}{1-Up}$ ;
    W[i] ←  $N_q[i]/elmb[i]$ ;
     $\rho_s[i] \leftarrow Up$ ;
end
end

```

3.5 Dynamic Routing Untuk Penempatan Pasien

Tiga komponen *dynamic routing* untuk sistem rujukan didefinisikan sebagai berikut

1. *Information policy*; Informasi yang diambil adalah panjang antrian pada rumah sakit tiap waktu. Estimasi laju kedatangan tiap rumah sakit diperoleh dari hasil perhitungan *hypercube*.
2. *Transfer policy*; Pasien dirujuk menuju rumah sakit preferensi utama, selanjutnya apabila preferensi utama memiliki antrian yang padat maka pasien dirujuk ke rumah sakit lain.
3. *Position policy*; Pasien dirujuk menuju rumah sakit dengan nilai fungsi preferensi paling besar.

Fungsi preferensi yang diusulkan untuk *routing* dinamis adalah ekspektasi waktu tunggu apabila pasien dirujuk menuju rumah sakit tujuan. Nilai ini didapat pada Persamaan (2.29) di bab II. Kemudian diberikan modifikasi yaitu penambahan ekspektasi waktu tunggu sesuai dengan ekspektasi bertambahnya antrian saat menempuh perjalanan menuju rumah sakit dan pengaruh preferensi dengan asumsi tiap pasien rela menunggu 6 menit lebih lama apabila berada pada faskes dengan tingkat preferensi yang lebih tinggi menurut user tersebut. Fungsi preferensi dinyatakan dalam persamaan berikut,

$$F_{ik} = \left(\frac{n_k + \lambda_k T_{ik}}{\mu_k} + P_{ik} \cdot 0.05 \right)^{-1} \quad (3.9)$$

Keterangan:

i = indeks area geografis

k = indeks rumah sakit

n_k = banyak antrian pada rumah sakit k

P_{ik} = urutan preferensi rumah sakit k menurut pasien dari area i

T_{ik} = waktu tempuh dari area i ke rumah sakit k

Ekspektasi waktu tunggu pada suatu antrian adalah n/μ namun banyaknya antrian akan bertambah saat pasien menempuh perjalanan menuju rumah sakit tersebut. Ekspektasi banyaknya pasien yang datang pada suatu rumah sakit selama perjalanan adalah $\lambda_k T_{ik}$. Banyak pasien total adalah penambahan nilai tersebut dengan n_k . Kemudian efek preferensi dinyatakan sebagai 0.05 dikalikan dengan P_{ik} karena pasien rela menunggu 6 menit (0.05 jam) kemudian dikalikan urutan preferensinya.

Penentuan *routing* dilakukan pada masing-masing puskesmas. Input yang diperlukan antara lain banyak antrian pada tiap Algoritma untuk menentukan *routing* pasien secara dinamis dijabarkan dalam pseudocode berikut,

Algoritma Routing_Dinamis

Deklarasi

$n_q, D, \text{Pref} :=$ array 1x3 of integer;

$\lambda_i, \mu :=$ real;

Travel := array 1x3 of real;

Deskripsi

for $i = 1$ to 3 do

$f_p[i] \leftarrow ((n_q + \lambda_i * \text{Travel}[i]) / (D[i] * \mu) + \text{Pref} * 0.05);$

endfor;

Cari RS- k dengan nilai f_p paling kecil

Output(k);

3.6 Konsep Pemberian Prioritas Pasien

Pasien dikelompokkan menjadi 2 kelompok prioritas sesuai tingkat kegawatan penyakit yaitu pasien kronis dan pasien non-kronis. Pasien kronis memiliki waktu pelayanan yang lebih lama daripada pasien non-kronis. Namun, pelayanan untuk pasien jenis ini harus didahulukan karena akan berbahaya bagi pasien apabila pelayanannya ditunda.

Pasien dengan penyakit kronis diberi indeks 0 dan pasien non-kronis diberi indeks 1. Notasikan p_0 sebagai proporsi pasien dengan penyakit kronis. Asumsikan $\bar{\mu}$ merupakan laju pelayanan rata-rata keseluruhan pasien maka,

$$E[S] = \frac{1}{\bar{\mu}} \quad (3.10)$$

Asumsikan proses pelayanan pasien kronis juga merupakan proses Poisson dengan laju $\mu_0 = k \cdot \bar{\mu}$. k merupakan koefisien pengali yang merupakan perbandingan laju pelayanan pasien kronis dengan laju pelayanan seluruh pasien. Nilai laju pelayanan untuk pasien non-kronis harus memiliki nilai yang membuat nilai waktu pelayanan keseluruhan tetap. Maka, nilai tersebut harus memenuhi persamaan berikut,

$$E[S] = p_0 E[S_0] + (1 - p_0) E[S_1] \quad (3.11)$$

Substitusikan nilai μ_0 pada $E[S_0]$ dan $E[S]$ dari Persamaan (3.5). Proses pelayanan pasien non-kronis juga merupakan proses Poisson sehingga $E[S_1] = 1/\mu_1$. Dengan menyubstitusikan nilai $E[S_1]$ akan diperoleh nilai μ_1 seperti pada Persamaan (3.7)

$$\mu_1 = \frac{(1 - p_0)k\bar{\mu}}{k - p_0} \quad (3.12)$$

Pasien kronis akan mendahului antrian dengan aturan *non-preemptive*. Aturan prioritas *non-preemptive* dipergunakan karena pasien yang sedang diperiksa tidak mungkin ditinggal meskipun ada pasien yang lebih gawat kondisinya datang. Akan tetapi, pasien kronis yang datang dapat menempati urutan antri setelah pasien kronis yang terakhir datang. Konsep pasien mendahului pasien lainnya ditunjukkan pada pseudocode berikut,

Algoritma Pasien Prioritas

Pasien := obyek dengan kelas prioritas dan urutan;

Urutan Pasien, Urutan Terakhir Pasien, := Integer;

Deskripsi

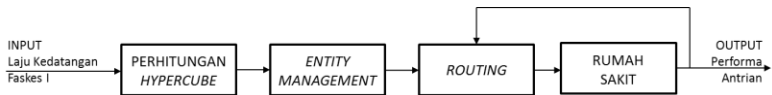
Input(Pasien);

```
Read(Kelas Prioritas);  
If Kelas Pasien=0  
    Urutan Pasien  $\leftarrow$  Urutan Terakhir Pasien Prioritas+1;  
else  
    Urutan Pasien  $\leftarrow$  Urutan Terakhir Pasien+1;  
end-if;  
Output(Urutan Pasien);
```

BAB 4

IMPLEMENTASI SISTEM

Pada bagian ini akan dibahas hasil perancangan model simulasi jaringan antrian dari sistem rujukan berjenjang menggunakan simulink. Pembahasan dalam bab ini dibagi menjadi 3 yaitu, pembuatan simulasi routing dinamis, simulasi pemberian prioritas dan pengumpulan parameter kerja model. Pembahasan pada Sub-bab 3.2 mengidentifikasi bahwa sistem rujukan dapat dibagi menjadi 3 bagian yaitu kedatangan pada faskes tingkat I, *routing* untuk tujuan pasien, kemudian proses mengantri pada poli rumah sakit. Gambar 4.1 menunjukkan skema implementasi berdasarkan pembagian tersebut.



Gambar 4.1 Skema Implementasi Model Simulasi Sistem Rujukan

Nilai laju kedatangan pada faskes tingkat I menjadi input untuk perhitungan model *hypercube* yang direalisasikan sebagai program MATLAB. Program tersebut menghasilkan nilai laju kedatangan rujukan dari area spasial tertentu sesuai jenis penyakit. Nilai tersebut dipergunakan sebagai input oleh *entity management* yang memunculkan pasien rujukan dalam simulasi. Pasien kemudian dirujuk sesuai aturan *routing* pada Sub-bab 3.5 yang direalisasikan sebagai fungsi MATLAB. Pasien yang dirujuk mengantri sesuai rujukan yang didapat pada poli rumah sakit. Nilai banyak antrian di umpan balik untuk menentukan *routing* sesuai kondisi poli.

4.1 Pembuatan Program pada MATLAB

Simulasi dibuat menggunakan *toolbox* Simulink pada MATLAB. Parameter yang dipergunakan dalam simulasi harus dikondisikan agar dapat diproses oleh model simulasi. Selain itu, estimasi laju kedatangan tiap rumah sakit melalui model *hypercube*. Penentuan *routing* dinamis juga dibuat dalam bentuk fungsi. Pembuatan *script* dilakukan menggunakan MATLAB R2011b.

4.1.1 Program Penerjemah Data

Program ini digunakan untuk menerjemahkan data sehingga dapat dijalankan oleh simulasi. Data yang akan diterjemahkan oleh program ini disimpan dalam file MS Excel dengan format .xlsx. File data excel terdiri atas 9 sheet yang berisi data untuk masing-masing penyakit. Pada tiap sheet berisi laju kedatangan pasien rujukan untuk suatu penyakit A_i dari 18 area, banyak dokter dan laju pelayanan tiap rumah sakit untuk penyakit A_i dan matriks preferensi pasien. Selain itu, program memberikan waktu tempuh dari tiap area menuju rumah sakit. *Script* MATLAB untuk program penerjemah data terdapat pada lampiran.

4.1.2 Program Estimasi Performa Hypercube

Program MATLAB untuk estimasi laju kedatangan pada model *hypercube* H_i dibuat berdasarkan *pseudocode* pada Sub-bab 3.4. Input yang diperlukan antara lain laju kedatangan tiap penyakit, laju pelayanan, jumlah dokter, waktu tempuh dan matriks preferensi yang merupakan output dari program sebelumnya. Output yang dihasilkan program ini adalah performa antrian *hypercube* seperti ketakseimbangan beban kerja, estimasi utilisasi, estimasi laju kedatangan dan distribusi pengguna. Estimasi laju kedatangan dipergunakan dalam *routing* dinamis. *Script* MATLAB untuk program estimasi performa *hypercube* terdapat pada lampiran.

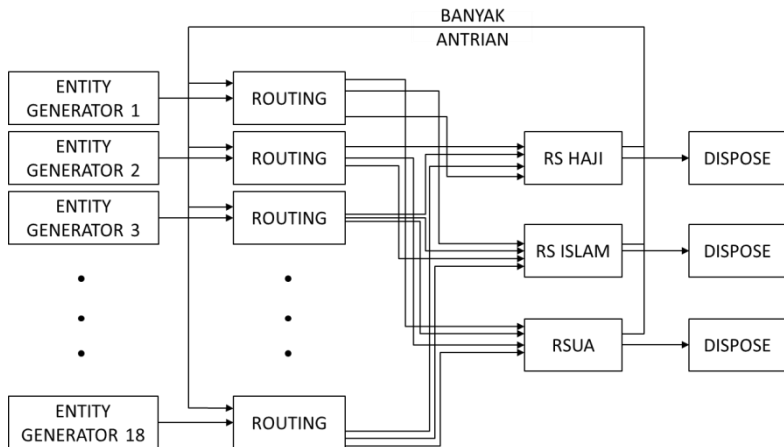
4.1.3 Fungsi Penentu Tujuan Routing Dinamis

Penentuan *routing* dinamis dibuat dalam bentuk fungsi agar dapat dijalankan oleh blok *user defined function*. Fungsi ini direalisasikan sesuai *pseudocode* pada bab 3.5. Input yang dibutuhkan antara lain banyak antrian yang diperoleh dari model simulasi, estimasi laju kedatangan rumah sakit dari program sebelumnya serta laju pelayanan, banyak dokter dan matriks preferensi. Fungsi MATLAB untuk penentuan *routing* dinamis diberi nama `dyn_pref3`.

4.2 Pembuatan Simulasi Routing Dinamis

Simulasi direalisasikan menggunakan *library* SimEvent dalam Simulink. Pada model H_i terdapat 18 *entity generator*, 18 *output switch*, 3 *path combiner*, 3 *FIFO queue* dan 3 *N-server*. *Output switch* terhubung dengan block *user defined function* yang menentukan indeks path yang dituju. Terdapat 6 *signal scope* yang digunakan untuk melihat banyak antrian dan rata-rata utilisasi server. Secara keseluruhan

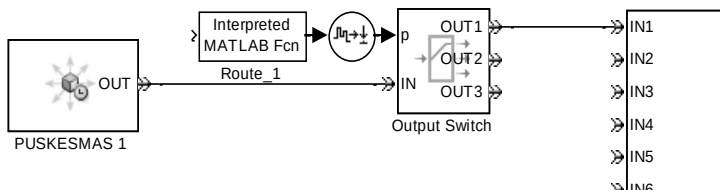
pemasangan blok untuk model simulasi ini mengikuti skema pada Gambar 4.2.



Gambar 4.2 Skema Pemasangan Blok Simulink

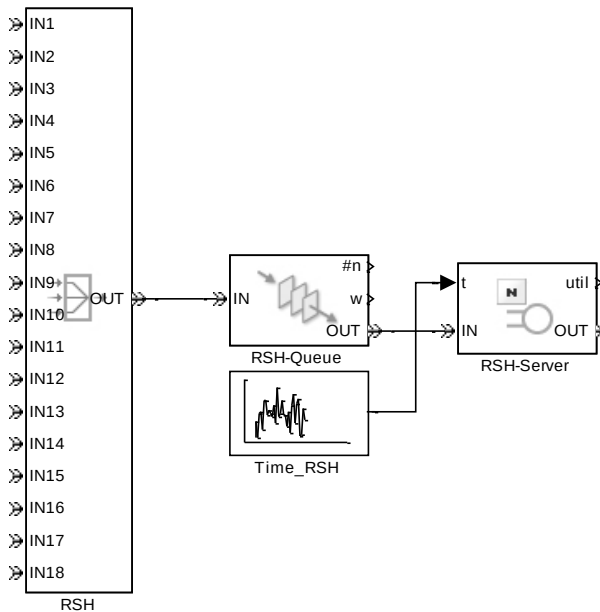
Routing diimplementasi menggunakan *output switch* yang diatur menggunakan fungsi preferensi. *Output switch* terhubung dengan masing-masing *entity generator* sesuai atom geografis. *Entity generator*-k menghasilkan pasien yang mendapatkan rujukan untuk penyakit-i dari puskesmas-k. *Output switch* mengarahkan pasien menuju tujuan rujukan yang ditentukan fungsi preferensi.

Output switch memiliki 3 output sesuai dengan jumlah rumah sakit dalam model. Setiap *output switch* dihubungkan dengan *path combiner* yang merepresentasikan poli rumah sakit tujuan *routing*. Pemasangan output switch sebagai representasi *routing* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Blok Simulink yang Merepresentasikan *Routing*

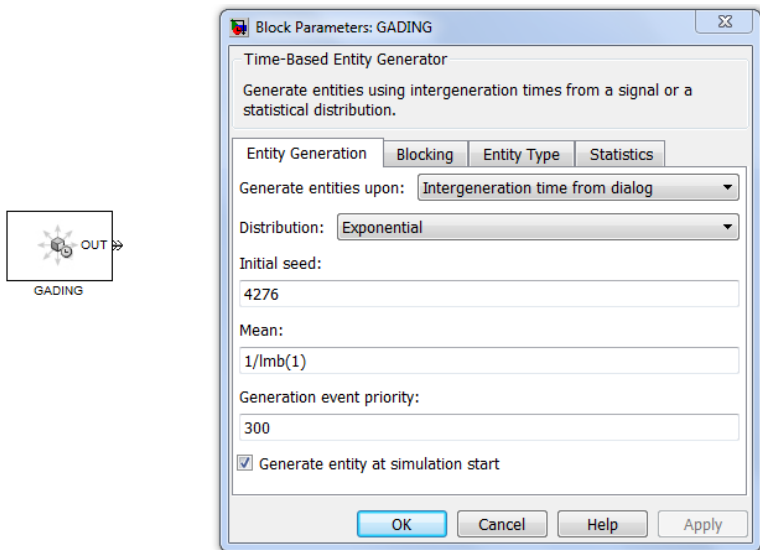
Path combiner dihungkan pada antrian yang direpresentasikan dengan blok *FIFO queue*. Blok *FIFO queue* kemudian dihubungkan pada blok *N-server* secara langsung. Blok *N-server* terhubung dengan *random number generator* pada *port p* untuk menentukan waktu pelayanan pasien yang terdistribusi secara ekponensial. *Path combiner*, *FIFO queue* dan *N-server* merepresentasikan poli penyakit A_i dari suatu rumah sakit. Pemasangan *path combiner* dan blok *queue* sebagai representasi poli RS dapat dilihat pada gambar 4.4.



Gambar 4.4 Blok Simulink yang Merepresentasikan Poli RS

Kedatangan entitas pada sistem ini mengikuti proses Poisson sehingga pada kotak dialog *entity generator* bagian *distribution* diubah menjadi *exponential*. Kemudian pada bagian *mean* diisi rata-rata waktu pelayanan dengan menggunakan laju kedatangan yang didefinisikan pada variabel *lmb*. Nilai laju kedatangan diinputkan nilai yang di estimasi pada Sub-sub-bab 4.4.1. Konfigurasi *entity generator* dapat dilihat pada Gambar 4.5

Blok *user defined function* pada gambar 4.2 diisi dengan fungsi preferensi direalisasi sesuai penjelasan pada Sub-sub-bab 4.1.3. Fungsi preferensi digunakan untuk menentukan tujuan rujukan dari suatu pasien. Input untuk fungsi tersebut antara lain banyak antrian pada rumah sakit saat ini, estimasi laju kedatangan pada tiap poli rumah sakit, preferensi pasien dari atom geografis tersebut, dan estimasi waktu tempuh. Output yang dihasilkan fungsi tersebut merupakan port yang mewakili rumah sakit tujuan.



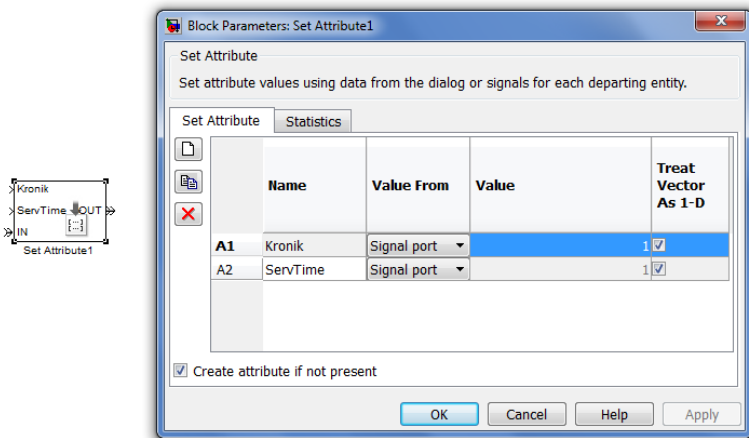
Gambar 4.5 Konfigurasi blok *entity generator*

Nilai preferensi pasien dari area geografis dan estimasi waktu tempuh merupakan nilai konstanta yang dapat dilihat pada lampiran. Nilai estimasi laju kedatangan tiap poli A_i rumah sakit didapatkan dari perkiraan performa model H_i sesuai algoritma pada Sub-bab 3.3. Sementara nilai banyak antrian dapat diambil dari statistik pada tiap blok *FIFO queue*. Kemudian dalam simulasi informasi banyak antrian diberi *delay* sebesar 0.01 jam. Hal ini menyatakan terjadinya *delay* informasi. Pemasangan keseluruhan blok simulasi dapat dilihat pada lampiran C.

4.3 Pembuatan Simulasi Pemberian Prioritas

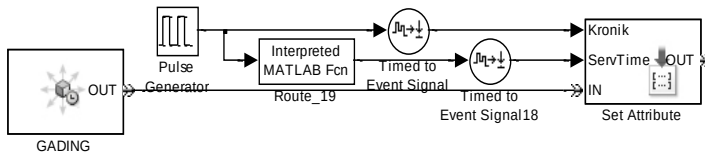
Simulasi untuk antrian dengan prioritas dibuat dengan memodifikasi simulasi pada Sub-bab 4.2 dengan menambahkan pemberian atribut pada entitas/pasien dan mengubah blok *FIFO queue* menjadi *priority queue*.

Entitas yang baru saja dibentuk langsung diberikan atribut menggunakan blok *set attribute*. Ada dua atribut yang dipergunakan yaitu, Kronik dan ServTime. Atribut Kronik menyatakan kondisi penyakit dari pasien dengan nilai biner. Nilai 1 menyatakan kondisi kronis dan nilai 0 menyatakan sebaliknya. Nilai dari atribut ServTime menyatakan nilai waktu pelayanan yang diperlukan untuk pasien tersebut. Kedua nilai tersebut didapatkan dari sinyal eksternal. Konfigurasi blok *set attribute* dapat dilihat pada Gambar 4.6



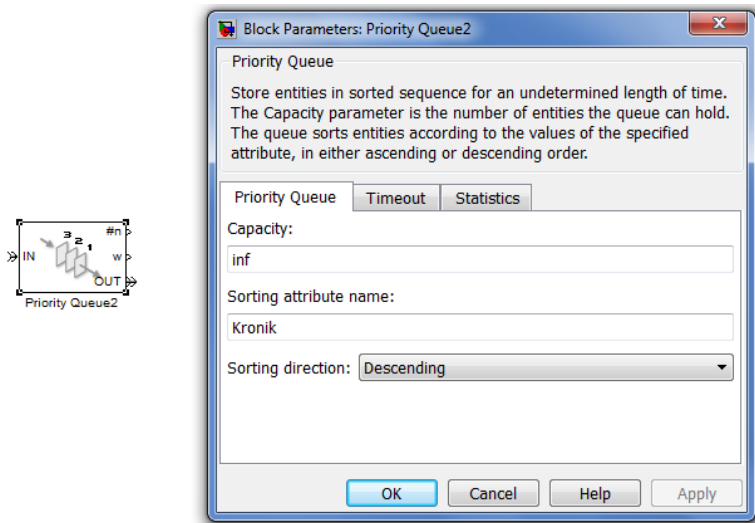
Gambar 4.6 Konfigurasi blok *set attribute*

Port kronik dihubungkan dengan *pulse generator* yang memiliki *duty cycle* sebesar jumlah populasi pasien kronis. Kemudian nilai waktu pelayanan ditentukan oleh fungsi MATLAB yang akan memberikan nilai waktu pelayanan yang terdistribusi eksponensial dengan laju yang berbeda untuk pasien kronis dan non kronis. Pemasangan blok *set attribute* yang mengelompokkan pasien pada kelas prioritas tertentu ditunjukkan pada Gambar 4.7



Gambar 4.7 Pemasangan atribut kelas prioritas pasien

Blok *FIFO queue* pada disimulasi sebelumnya digantikan dengan blok *priority queue*. Blok *priority queue* dapat mendahulukan entitas sesuai urutan nilai atributnya. Nilai atribut yang dibandingkan adalah atribut Kronik. Pasien kronis dengan atribut Kronik bernilai 1 akan didahulukan daripada pasien non-kronis dengan atribut Kronik bernilai 0 sehingga dalam kotak dialog *priority queue* urutan antrian bersifat *descending*. Konfigurasi blok *priority queue* ditunjukkan pada Gambar 4.8



Gambar 4.8 Konfigurasi blok *priority queue*

Pemasangan blok *priority queue* seperti blok *FIFO queue* pada gambar 4.3. Kemudian, pada blok *N-server* tidak dipasang sinyal

eksternal layaknya sinyal random pada Gambar 4.4. Hal ini dikarenakan nilai waktu pelayanan telah ditentukan melalui atribut *ServTime*.

Nilai atribut *ServTime* berbeda untuk tiap jenis pasien. Pasien dengan penyakit kronis diasumsikan membutuhkan waktu pelayanan dua kali rata-rata waktu pelayanan sehingga laju pelayanannya $\mu/2$. Pasien dengan penyakit non-kronis diasumsikan memiliki rata-rata waktu pelayanan yang membuat rata-rata waktu pelayanan keseluruhan tetap berada pada nilai estimasi awal.

4.4 Parameter Kerja Model

Model simulasi routing dinamis dan pemberian prioritas pasien membutuhkan nilai rata-rata waktu antar generasi dan rata-rata waktu pelayanan untuk menjalankannya. Algoritma estimasi performa model hypercube H_i membutuhkan laju kedatangan, laju pelayanan rata-rata dan banyaknya dokter serta matriks preferensi tiap pasien.

Parameter tersebut dicari menggunakan dua metode yaitu penolahan data dan perhitungan matematis. Pengolahan data menghasilkan laju keberangkatan rujukan dari tiap puskesmas, proporsi penyakit dari tiap puskesmas dan laju pelayanan tiap rumah sakit. Perhitungan matematis akan dilakukan untuk mendapatkan nilai laju kedatangan dari tiap area untuk tiap jenis penyakit dan laju pelayanan untuk dua kelas prioritas pasien.

4.4.1 Pengolahan Data

Data yang dipergunakan dalam tugas akhir ini dikumpulkan dari BPJS. BPJS memberikan data berupa daftar faskes tingkat I yang tergabung dalam BPJS, banyak pasien yang terdaftar pada masing-masing faskes tingkat I, agregasi rujukan pasien yang merincikan asal faskes rujukan, faskes tujuan, jenis penyakit dan banyak pasien yang dirujuk dengan kondisi tersebut dalam satu tahun.

Parameter pertama yang diolah adalah laju keberangkatan rujukan dari tiap puskesmas. Proses pertama dalam mendapatkan parameter tersebut adalah menjumlahkan keseluruhan kejadian rujukan sehingga didapatkan total kejadian rujukan di Surabaya. Nilai ini dibagi dengan jam kerja total fasilitas kesehatan sehingga diperoleh laju rujukan di Surabaya. Kemudian nilai laju rujukan ini diskalakan pada tiap faskes tingkat I sesuai proporsi banyak peserta yang terdaftar pada faskes tersebut. Hal ini dapat dilakukan karena laju rujukan dianggap proses

Poisson. Hasil pengolahan data laju keberangkatan tiap puskesmas dan 5 wilayah tambahan dapat dilihat pada Tabel 4.1

Tabel 4.1 Data Laju Keberangkatan dari Tiap Puskesmas (per jam)

No	Puskesmas Asal	Variabel	Laju Keberangkatan
1	Gading	L_1	2.3
2	Gunung Anyar	L_2	1.32
3	Kalijudan	L_3	0.82
4	Kalirungkut	L_4	1.79
5	Keputih	L_5	1.02
6	Klampisngasem	L_6	1.08
7	Medokan Ayu	L_7	1.62
8	Menur	L_8	1.51
9	Mojo	L_9	2.45
10	Mulyorejo	L_{10}	1.33
11	Pacarkeling	L_{11}	2.27
12	Rangkah	L_{12}	2.28
13	Tenggilis	L_{13}	1.39
14	Surabaya Pusat	L_{14}	20.41
15	Surabaya Barat	L_{15}	24.1
16	Surabaya Selatan	L_{16}	27.21
17	Surabaya Utara	L_{17}	35.61
18	Surabaya Timur	L_{18}	18.34

Tabel 4.1 menunjukkan laju keberangkatan rujukan dari tiap area yang dimodelkan setelah melalui proses pengolahan. Parameter laju keberangkatan ini dinotasikan sebagai L_i . Indeks i dengan nilai 1 sampai 13 merepresentasikan 13 puskesmas di area Surabaya Timur. Indeks i dengan nilai 14 sampai 18 merepresentasikan 5 area yang merupakan agregasi laju rujukan faskes tingkat I pada area tersebut selain 13 puskesmas yang disebutkan sebelumnya. Vektor L dengan 18 elemen merupakan vektor laju keberangkatan masing-masing area

Parameter selanjutnya adalah proporsi penyakit pada tiap puskesmas. Proses untuk mendapatkan proporsi penyakit pertama kali adalah mengelompokkan rujukan berdasarkan faskes tingkat I asal.

Kemudian, data dikelompokkan sesuai 9 jenis penyakit yang telah disebutkan pada perancangan konseptual. Banyak rujukan kemudian pada tiap kelompok penyakit dijumlahkan. Hasil yang didapatkan kemudian dibagi dengan total pasien yang dirujuk dari puskesmas tersebut.

Tabel 4.2 Proporsi Penyakit Tiap Area (dalam %)

Area	Poli Penyakit									
	Mata	THT	Paru	Kulit	Saraf	P. Dalam	Jantung	Kandungan	Bedah	Total
Gading	26.9	8.6	4.4	5.1	0.5	32.6	1.5	3.1	16.5	100
G. Anyar	13.8	11.3	3.2	1.8	0.8	26	1.7	1.6	38.2	100
Kalijudan	16.1	7.9	3.3	2.8	1.1	14.4	2.2	1.6	50.6	100
Kalirungkut	17.1	6.2	4.3	4	0.6	25.9	1.9	1.6	38	100
Keputih	20.1	9.6	6.3	4.7	0.7	15.7	0.9	0.8	40.8	100
Klampis	17.3	7	3.8	1.6	0.6	22.5	1.9	2.1	43.2	100
Medokan	17.1	7.2	3.2	2	0.8	15.6	1.3	1.8	50.8	100
Menur	15.2	6.2	4.1	1.8	1	25.5	0.9	2.2	42.6	100
Mojo	12.1	5.4	2.1	2.2	0.2	23.2	0.6	16.5	33.5	100
Mulyorejo	25.8	8.7	2.7	4.2	0.4	26.6	0.9	2	28	100
P. Keling	26.6	4.2	2.8	3.8	0.6	18.3	1.2	3.8	37.5	100
Rangkah	9.7	3.1	1.9	1.8	0.3	19.3	2.5	1	60.2	100
Tenggilis	15.6	4.1	5.3	2	0.8	43.2	1.7	2.7	24.6	100
Pusat	31.9	5	2.9	2.5	0.7	29.1	1.8	2.2	24	100
Barat	12.2	9.7	3.4	4.2	1.2	22.1	1.9	2.5	42.8	100
Selatan	8.6	11.4	3.5	2.9	0.8	21.4	2.1	2.3	47	100
Timur	19.5	9.5	3.7	1.6	0.7	19.8	1.9	1.7	41.6	100
Utara	4.2	4.1	5	1.3	1.3	16.7	1.6	1.4	64.5	100

Tabel 4.2 menunjukkan hasil pengolahan data berupa nilai proporsi dalam persen. Proporsi penyakit pada tiap puskesmas dan wilayah dapat dinotasikan sebagai p_{ik} dengan i merupakan indeks area seperti pada L dan k merupakan indeks penyakit. Nilai p_{ik} merupakan nilai pada baris- k dan kolom- i di Tabel 4.2

Parameter ketiga adalah laju pelayanan dari tiap server/dokter. Parameter diolah berdasarkan survei yang dilakukan oleh penulis. Survei dilakukan dengan menghitung lama pelayanan antar pasien pada suatu poli kemudian dicari rata-rata waktu pelayanannya. Selain itu, survei juga dilakukan untuk mendapatkan banyak dokter yang bertugas pada satu waktu untuk tiap jenis penyakit. Nilai laju pelayanan seluruh server pada suatu jenis penyakit harus sama dalam model hypercube sehingga nilai rata-rata waktu pelayanan didapatkan dari rata-rata waktu pelayanan untuk semua pasien dalam satu jenis penyakit. Hal ini dapat dilakukan karena karakteristik pasien dalam satu jenis penyakit sama sehingga perlakuan dokter dapat dianggap sama. Rata-rata laju pelayanan dan banyak dokter ditunjukkan pada Tabel 4.3

Tabel 4.3 Rata-rata laju pelayanan dan banyak dokter tiap RS

Parameter	Poli Penyakit								
	Mata	THT	Paru	Kulit	Saraf	P. Dalam	Jantung	Kandungan	Bedah
Laju pelayanan (μ_i)	8.03	8.77	6.93	8.70	8.56	10.17	11.37	5.38	6.16
Banyak dokter RSH (n_{i1})	2	2	1	2	2	2	2	1	4
Banyak dokter RSI (n_{i2})	1	1	1	1	1	2	1	1	2
Banyak dokter RSUA (n_{i3})	2	2	3	2	2	3	2	0	8
Banyak dokter total (n_i)	5	5	5	5	5	7	5	2	14

Laju pelayanan rata-rata pada Tabel 4.3 merupakan rata-rata dari laju pelayanan pada ketiga rumah sakit. Rata-rata dihitung dengan bobot sesuai banyak dokter yang bekerja pada satu shift Banyaknya dokter dalam satu shift didapatkan melalui pengamatan di rumah sakit.

Waktu tempuh yang dipergunakan adalah waktu yang diperlukan untuk bergerak dari satu area ke rumah sakit. Parameter ini dapat diestimasi setelah mendapat jarak antara kedua area tersebut. Waktu

tempuh diestimasi dengan cara membagi jarak dengan kecepatan tempuh rata-rata kedua area tersebut. Pengolahan parameter waktu tempuh pada tugas akhir ini akan dibantu dengan Google Maps.

Jarak antar dua area didapatkan dengan menghitung jarak antar titik tengah kedua area yang sedang dihitung. Kedua titik tengah dipilih terlebih dahulu pada Google Maps. Google Maps kemudian menunjukkan beberapa rute yang mungkin, jarak tempuh dan waktu tempuh tiap rute. Nilai parameter waktu tempuh diperoleh dari rata-rata waktu tempuh masing-masing rute. Tabel 4.4 menunjukkan hasil pengolahan parameter waktu tempuh.

Tabel 4.4 Waktu Tempuh menuju Rumah Sakit

Area	Waktu Tempuh ke RS (menit)		
	RS Haji	RS Islam	RSUA
Gading	13	28	13
G. Anyar	27	25	30
Kalijudan	16	21	19
Kalirungkut	17	29	18
Keputih	6	17	10
Klampis	22	26	24
Medokan	8	15	15
Menur	11	20	13
Mojo	5	21	3
Mulyorejo	14	20	15
P. Keling	21	28	14
Rangkah	19	7	21
Tenggilis	8	25	8
Pusat	16	17	18
Barat	41	31	43
Selatan	28	18	33
Timur	9	15	15
Utara	30	45	25

4.4.2 Estimasi Parameter Menggunakan Hitungan Matematis

Matriks laju keberangkatan L adalah matriks diagonal dengan nilai elemen diagonal ke- i adalah L_i seperti tertera pada Tabel 4.1. P adalah matriks proporsi penyakit dengan elemen pada baris- i dan kolom- k

bernilai p_{ik} . Matriks laju kedatangan dinotasikan sebagai Λ dan didapatkan dari Persamaan (4.1),

$$\Lambda = L.P \quad (4.1)$$

Λ merupakan matriks dengan 18 baris dan 9 kolom. Nilai pada baris ke- i dan kolom ke- k pada Λ merupakan laju kedatangan pasien dari area- i dengan penyakit- k . Nilai Λ ditampilkan pada Tabel 4.5.

Tabel 4.5 Laju Kedatangan Pasien Berdasarkan Tipe Penyakit

Area	Poli Penyakit								
	Mata	THT	Paru	Kulit	Saraf	P. Dalam	Jantung	Kandungan	Bedah
Gading	0.61	0.19	0.10	0.12	0.01	0.75	0.03	0.07	0.38
G. Anyar	0.18	0.15	0.04	0.02	0.01	0.34	0.02	0.02	0.50
Kalijudan	0.13	0.07	0.03	0.02	0.01	0.12	0.02	0.01	0.41
Kalirungkut	0.31	0.11	0.08	0.07	0.01	0.46	0.03	0.03	0.68
Keputih	0.21	0.10	0.06	0.05	0.01	0.16	0.01	0.01	0.42
Klampus	0.19	0.08	0.04	0.02	0.01	0.24	0.02	0.02	0.47
Medokan	0.28	0.12	0.05	0.03	0.01	0.25	0.02	0.03	0.82
Menur	0.23	0.09	0.06	0.03	0.02	0.39	0.01	0.03	0.64
Mojo	0.30	0.13	0.05	0.05	0.01	0.57	0.02	0.40	0.82
Mulyorejo	0.34	0.12	0.04	0.06	0.01	0.35	0.01	0.03	0.37
P. Keling	0.60	0.10	0.06	0.09	0.01	0.42	0.03	0.09	0.85
Rangkah	0.22	0.07	0.04	0.04	0.01	0.44	0.06	0.02	1.37
Tenggilis	0.22	0.06	0.07	0.02	0.01	0.60	0.02	0.04	0.34
Pusat	6.51	1.02	0.59	0.51	0.14	5.94	0.37	0.45	4.90
Barat	2.94	2.34	0.82	1.01	0.29	5.33	0.46	0.60	10.31
Selatan	2.34	3.10	0.95	0.79	0.22	5.82	0.57	0.63	12.79
Timur	6.94	3.38	1.32	0.57	0.25	7.05	0.68	0.61	14.81
Utara	0.77	0.75	0.92	0.24	0.24	3.06	0.29	0.26	11.83

Nilai laju pelayanan untuk pasien dengan penyakit kronis adalah setengah kali nilai yang tertera pada tabel 4.3 sehingga pengali k untuk μ_0 pada bab 3.6 adalah 0.5. Proporsi pasien kronis atau p_0 diasumsikan sebesar 0.05. Agar rata-rata keseluruhan waktu pelayanan tetap konstan maka nilai laju pelayanan untuk pasien non-kronis atau μ_1 harus memenuhi persamaan (3.7) sehingga $\mu_1 = 19\bar{\mu}/18$. Hasil perhitungan laju pelayanan untuk pasien berprioritas ditunjukkan pada Tabel 4.6.

Tabel 4.6 Rata-rata laju pelayanan tiap pasien kronis dan non-kronis

Parameter	Poli Penyakit								
	Mata	THT	Paru	Kulit	Saraf	P. Dalam	Jantung	Kandungan	Bedah
Laju pelayanan kronis (μ_i)	4.01	4.38	3.47	4.35	4.28	5.09	5.69	2.69	3.08
Laju pelayanan non-kronis (μ_i)	8.48	9.26	7.32	9.19	9.04	10.74	12	5.68	6.51

Nilai pada Tabel 4.6 digunakan pada model simulasi pemberian prioritas untuk menentukan waktu pelayanan pasien sesuai kelas prioritas. Proses pelayanan merupakan proses Poisson sehingga waktu pelayanannya berdistribusi ekponensial. Mean waktu pelayanan distribusi ekponensial adalah $1/\mu$ dengan μ merupakan nilai pada Tabel 4.6.

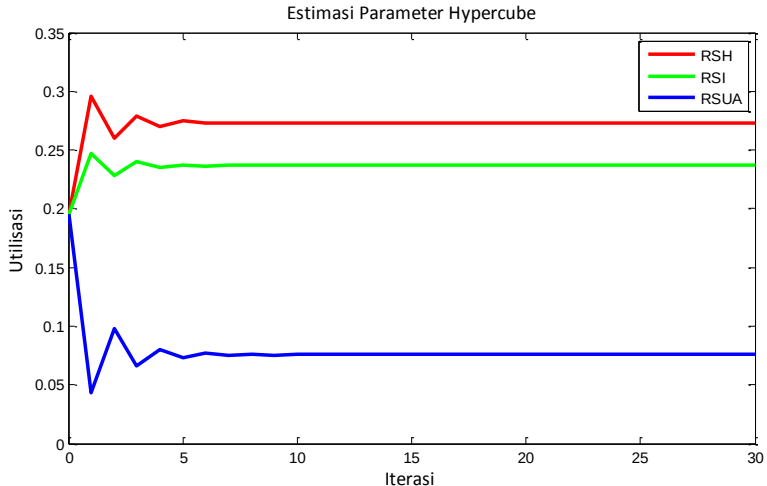
BAB 5

HASIL PENGUJIAN DAN ANALISIS

Pada bagian ini akan dibahas hasil pengujian model jaringan antrian dari sistem rujukan berjenjang BPJS Surabaya di lingkup Dinas Kesehatan Surabaya area Surabaya Timur. Pembahasan dalam bab ini dibagi menjadi 3 subbab yaitu, estimasi parameter, perbandingan kondisi awal dengan routing dinamis

5.1 Estimasi Laju Kedatangan pada Rumah Sakit

Laju kedatangan pada tiap poli rumah sakit diestimasi menggunakan program MATLAB yang disebutkan pada Sub-bab 4.1.2. Input yang dipergunakan antara lain laju kedatangan pasien tiap penyakit yang terdapat pada Tabel 4.3, banyak server dan laju pelayanan yang terdapat pada Tabel 4.4 serta matriks preferensi. Input tersebut diterjemahkan ke dalam bentuk matriks dari file excel oleh program pada Sub-bab 4.1.1. Proses iteratif estimasi beban kerja dapat dilihat pada grafik di Gambar 5.1.



Gambar 5.1 Proses Iteratif Estimasi Beban Kerja

Setelah mencapai nilai toleransi error didapatkan estimasi nilai beban kerja per dokter. Program kemudian memberikan nilai rata-rata pada tiap rumah sakit. Nilai rata-rata beban kerja dokter pada tiap rumah sakit dapat dilihat pada Tabel 5.1.

Tabel 5.1 Rata-rata beban kerja server pada masing-masing poli RS

Parameter	Poli Penyakit								
	Mata	THT	Paru	Kulit	Saraf	P. Dalam	Jantung	Kandungan	Bedah
Beban Kerja RSH (ρ_{i1})	0.58	0.44	0.37	0.20	0.08	0.56	0.14	0.31	0.82
Beban Kerja RSI (ρ_{i2})	0.50	0.20	0.30	0.02	0.01	0.50	0.01	0.28	0.78
Beban Kerja RSUA (ρ_{i3})	0.46	0.09	0.04	0.01	0.01	0.33	0.01	0	0.75

Program kemudian menghitung laju kedatangan pada tiap rumah sakit. Nilai laju kedatangan ini yang dijadikan input dalam fungsi preferensi. Hasil estimasi laju kedatangan pada tiap rumah sakit ditunjukkan pada Tabel 5.2

Tabel 5.2 Estimasi laju kedatangan poli rumah sakit

Parameter	Poli Penyakit								
	Bedah	Mata	THT	P. Dalam	Paru	Jantung	Saraf	Kulit	Kandungan
Kedatangan RSH (λ_{i1})	20.25	9.38	7.68	11.47	2.55	3.29	1.36	3.51	1.65
Kedatangan RSI (λ_{i2})	9.61	4.05	1.78	10.15	2.11	0.07	0.02	0.16	1.5
Kedatangan RSUA (λ_{i3})	36.86	7.42	1.66	9.99	0.92	0.01	0.01	0.03	0

Program tersebut kemudian menghitung performa antrian tiap rumah sakit dengan nilai laju kedatangan pada Tabel 5.2 sesuai pada Sub-bab 2.2.1. Lama antrian rata-rata ditunjukkan pada Tabel 5.3

Tabel 5.3 Estimasi waktu tunggu di poli rumah sakit (dalam menit)

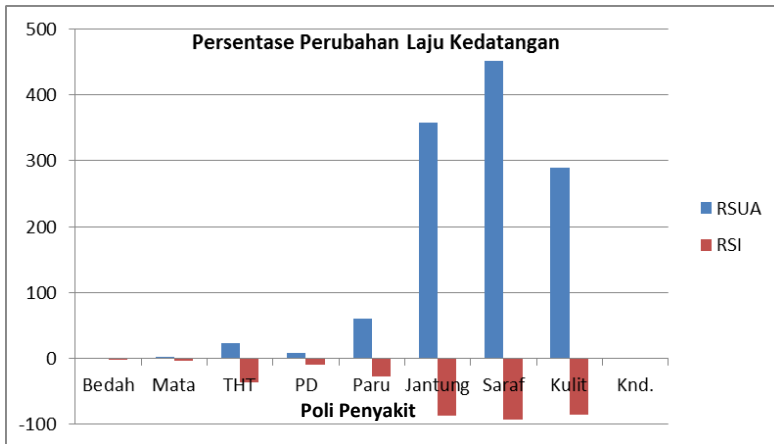
Parameter	Poli Penyakit								
	Bedah	Mata	THT	P. Dalam	Paru	Jantung	Saraf	Kulit	Kandungan
Waktu Antri RSH (W_{i1})	8.61	3.87	1.62	2.75	5.04	0.11	0.04	0.29	4.92
Waktu Antri RSI (W_{i2})	15.03	7.58	1.74	1.96	3.78	0.03	0.01	0.13	4.33
Waktu Antri RSUA (W_{i3})	1.69	2.03	0.06	0.25	0	0	0	0	0

Nilai-nilai tersebut merupakan nilai kondisi ideal penerapan routing dengan nilai matriks preferensi yang telah ditentukan. Kondisi ideal yang dimaksud adalah pasien yang dirujuk dikumpulkan pada satu antrian dan kemudian baru dirujuk setelah ada dokter yang kosong. Hasil estimasi laju kedatangan pada skenario kedua dapat dilihat pada Tabel 5.4

Tabel 5.4 Estimasi laju kedatangan poli RS pada skenario kedua

Parameter	Poli Penyakit								
	Bedah	Mata	THT	P. Dalam	Paru	Jantung	Saraf	Kulit	Kandungan
Kedatangan RSH (λ_{i1})	20.25	9.39	7.93	11.59	2.57	3.31	1.38	3.55	1.65
Kedatangan RSI (λ_{i2})	9.49	3.88	1.13	9.18	1.54	0.01	0.01	0.02	1.50
Kedatangan RSUA (λ_{i3})	36.98	7.58	2.06	10.84	1.47	0.06	0.02	0.13	0

Program ini kemudian diuji pada dua skenario preferensi yang berbeda. Kondisi pada skenario pertama menggunakan parameter yang digunakan sebelumnya dimana RSUD masih menempati urutan ketiga. Kondisi pada skenario kedua diasumsikan RSUD lebih diminati oleh pasien area Mojo, Mulyorejo, Pacar Keling, Gading dan Tambaksari sehingga menjadi urutan kedua pada area-area tersebut. Pengujian dilakukan dengan membandingkan laju kedatangan RSUD dan RSI antara dua skenario tersebut. Perubahan laju kedatangan RSUD dan RSI pada kedua skenario dapat dilihat pada Gambar 5.2



Gambar 5.2 Perbandingan Presentase Perubahan Laju Kedatangan

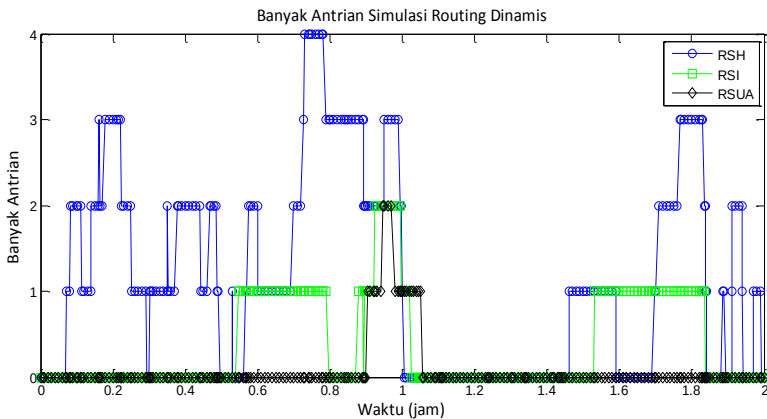
Laju kedatangan di RSUD mengalami peningkatan pada skenario kedua. Sementara itu, laju kedatangan di RSI mengalami penurunan. Perubahan preferensi pasien dari RSI menjadi RSUD mengakibatkan fenomena tersebut. Pengaruh perubahan preferensi ini berhubungan dengan utilisasi pada sistem. Perubahan laju kedatangan RSUD pada penyakit bedah sangat kecil yaitu peningkatan sebesar 0.34% dan rata-rata utilitasnya sebesar 0.75. Sementara itu, perubahan laju kedatangan RSUD pada penyakit saraf mengalami peningkatan mencapai 450% dan rata-rata utilitasnya sebesar 0.03. Dapat disimpulkan bahwa pengaruh perubahan preferensi berbanding terbalik dengan rata-rata utilisasi pada sistem. Hal ini dapat diinterpretasikan bahwa semakin ramai sistem pasien tidak dapat dengan leluasa memilih rumah sakit tujuan.

5.2 Pengaruh Routing Dinamis Pada Model Sistem Rujukan

Pada bagian ini akan dibahas hasil simulasi routing dinamis dan analisis stabilitas. Simulasi routing dinamis dilakukan menggunakan model simulasi yang dijabarkan pada Sub-bab 4.2. Analisis sensitivitas dilakukan untuk mengamati efektifitas routing terhadap perubahan parameter routing.

5.2.1 Hasil Simulasi Routing Dinamis

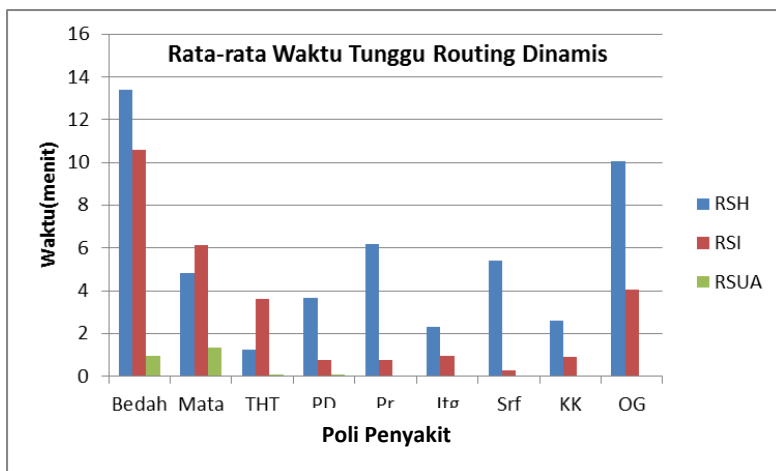
Parameter yang dipergunakan untuk menjalankan simulasi seperti pada program estimasi ditambah nilai estimasi laju kedatangan pada tiap rumah sakit pada Tabel 5.2. Fungsi preferensi yang dipergunakan adalah fungsi yang diimplementasikan sesuai penjelasan pada Sub-bab 4.1.3 dengan nama `dyn_pref3`. Gambar 5.3 menunjukkan banyak antrian saat menjalankan model simulasi untuk penyakit mata dalam durasi 2 jam.



Gambar 5.3 Banyak Antrian Simulasi Routing Dinamis

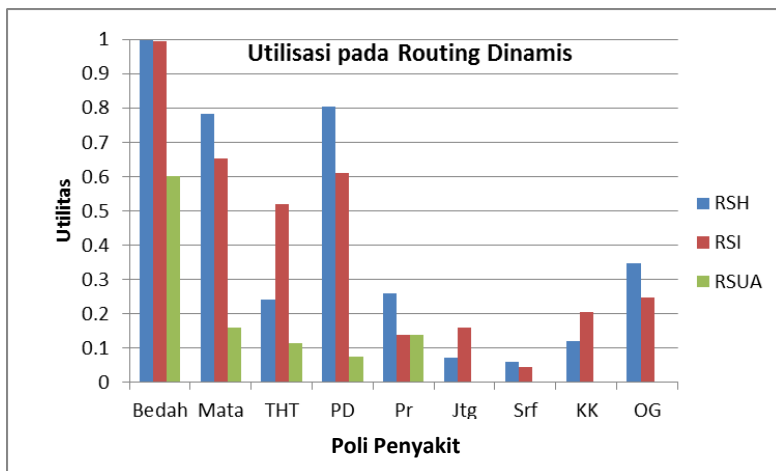
RS Haji menempati urutan preferensi pertama sehingga terlihat pada Gambar 5.3. RS Haji lebih sering terjadi antrian daripada rumah sakit lainnya. RS Islam yang menempati urutan kedua memiliki lebih banyak pasien daripada RSUA yang menempati urutan ketiga.

Output yang dihasilkan dengan menjalankan model simulasi routing dinamis antara lain rata-rata waktu tunggu pada masing-masing antrian dan nilai utilisasi server. Gambar 5.4 dan Gambar 5.5 menunjukkan waktu tunggu rata-rata dan utilisasi hasil routing dinamis.



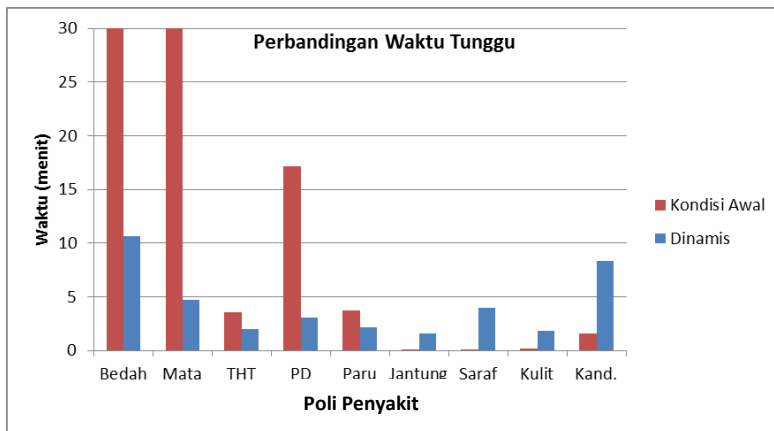
Gambar 5.4 Rata-rata Waktu Tunggu Routing Dinamis

Rata-rata waktu tunggu rumah sakit yang memiliki tingkat preferensi yang tinggi diamati pada Gambar 5.4 lebih lama daripada rumah sakit yang lebih rendah. Waktu tunggu dapat lebih rendah karena dokter yang bertugas lebih banyak seperti pada penyakit mata.



Gambar 5.5 Utilisasi pada *Routing* Dinamis

Hasil simulasi dibandingkan dengan kondisi awal untuk rata-rata waktu tunggu pada suatu penyakit dan utilisasi. Rata-rata keseluruhan dihitung dengan rata-rata berbobot laju kedatangan pasien menuju rumah sakit tersebut. Perbandingan rata-rata waktu tunggu dari keseluruhan pasien pada kedua kondisi dapat dilihat pada Gambar 5.6

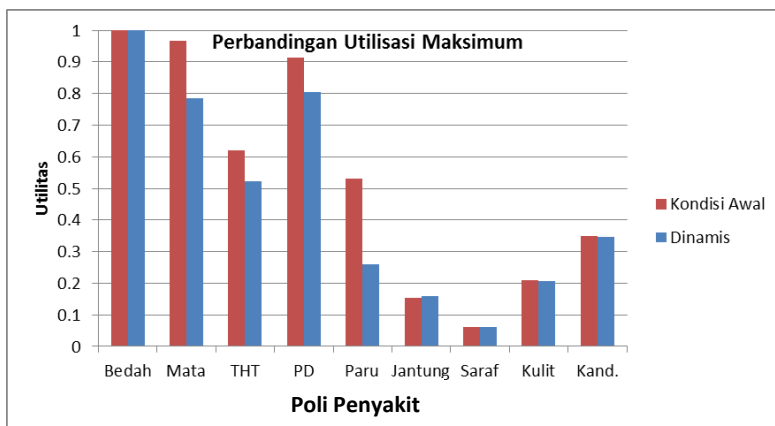


Gambar 5.6 Perbandingan waktu tunggu

Pada Gambar 5.6 dapat diamati bahwa terjadi penurunan waktu tunggu pada jenis penyakit bedah, mata, THT, paru dan penyakit dalam. Hal ini disebabkan total kedatangan pada penyakit tersebut tinggi dan persebarannya tidak merata yaitu mayoritas dirujuk menuju rumah sakit Haji dan rumah sakit Islam.

Rata-rata waktu tunggu keseluruhan justru naik pada jenis penyakit jantung, saraf, kulit dan kandungan. Hal ini disebabkan karena fungsi preferensinya memiliki toleransi waktu sebesar 6 menit pada preferensi yang lebih tinggi. Hal ini dapat diamati pada Gambar 5.6 bahwa kendati hasil routing dinamis memiliki rata-rata waktu tunggu yang lebih besar nilainya berada disekitar 6 menit.

Performa dari sisi rumah sakit dapat dilihat dengan menggunakan perbandingan dari utilisasi. Utilisasi akan dibandingkan dari utilisasi maksimumnya. Perbandingan utilisasi maksimum antara kondisi awal dan simulasi routing dinamis ditunjukkan pada Gambar 5.7



Gambar 5.7 Perbandingan utilisasi maksimum.

Pada Gambar 5.7 dapat diamati bahwa terjadi penurunan utilisasi maksimum pada jenis penyakit bedah, mata, THT, paru dan penyakit dalam. Hal ini disebabkan routing dinamis membagi pasien yang sebelumnya menumpuk pada suatu rumah sakit. Pada kondisi awal pasien hanya dirujuk berdasarkan preferensi. Penumpukan mengakibatkan lamanya waktu tunggu sehingga pengaruh routing dinamis lebih besar pada kondisi utilisasi tinggi.

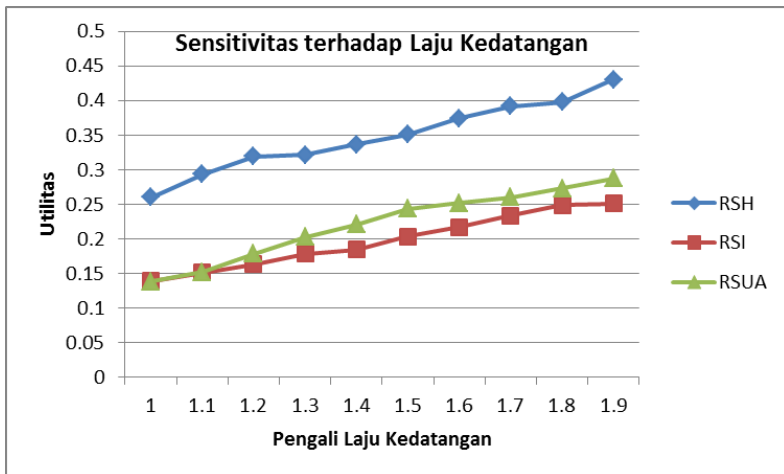
Utilisasi maksimum justru naik pada jenis penyakit jantung, saraf, kulit dan kandungan. Dapat diamati bahwa pada penyakit tersebut utilisasi rumah sakit dibawah 0.4. Utilisasi justru naik karena fungsi preferensi juga memperhatikan preferensi user. Karena beban kerja rendah, server yang diminati tidak sibuk/banyak antrian sehingga routing dinamis merujuk pasien ke server tersebut.

Kendati demikian, secara keseluruhan utilisasi maksimum pada sistem rujukan berjenjang mengalami penurunan dengan menerapkan routing dinamis. Rata-rata penurunan utilisasi maksimumnya sebesar 15.62%. Penurunan paling signifikan terjadi pada tipe penyakit bedah yang sebelumnya berada pada kondisi tidak stabil menjadi 0.9963 (penurunan 52.63%). Melalui pengamatan ini dapat disimpulkan, routing dinamis memiliki pengaruh yang besar terhadap waktu tunggu dan utilisasi pada sistem dengan utilisasi awal yang tinggi.

5.2.2 Analisis Sensitivitas Routing Dinamis

Analisis sensitivitas untuk routing dinamis dilakukan dengan mengubah laju kedatangan pasien. Pengubahan laju kedatangan mempengaruhi laju kedatangan pada rumah sakit. Hal tersebut juga dapat diamati melalui utilisasi rumah sakit. Namun, pengubahan laju kedatangan tidak mengubah rasio utilisasi untuk fungsi preferensi yang sama.

Pengubahan laju kedatangan dilakukan dengan meningkatkan laju kedatangan tiap 10% dari laju kedatangan awal hingga peningkatan 90%. Pengaruh yang diamati diambil pada penyakit paru. Nilai yang diamati adalah perubahan utilisasi terhadap laju kedatangan. Grafik pada gambar 5.8 menunjukkan pengaruh pengubahan laju kedatangan terhadap routing dinamis pada penyakit paru.

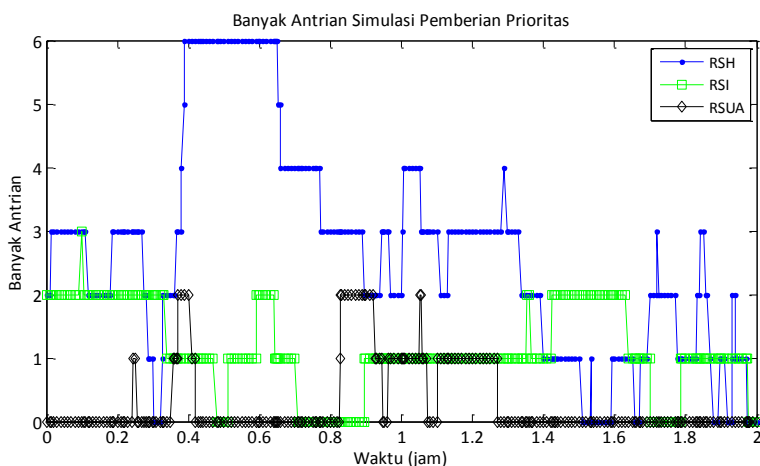


Gambar 5.8 Sensitivitas Routing Dinamis terhadap Laju Kedatangan

Pengamatan pada Gambar 5.8 menunjukkan bahwa terjadi peningkatan utilisasi seiring dengan peningkatan laju kedatangan. Namun, rasio utilitasnya tetap yaitu sekitar 45,73:25,67:28,6. RS Haji dan RS Islam memiliki 1 dokter dalam model penyakit paru Sedangkan RSUA memiliki 3 dokter. Ketika laju kedatangan bertambah, pasien yang dirujuk menuju RSUA akan bertambah meskipun RSUA berada pada urutan ketiga. Oleh karena itu, utilisasi RSUA mengalami kenaikan.

5.3 Pengaruh Prioritas Pasien Pada Model Sistem Rujukan

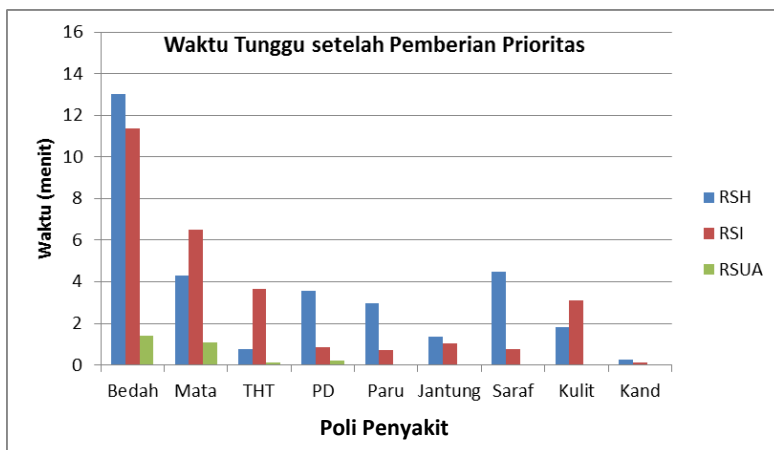
Simulasi pemberian prioritas dilakukan menggunakan model simulasi yang disebutkan pada Sub-bab 4.4. Parameter yang dipergunakan untuk menjalankan simulasi seperti pada model simulasi routing dinamis. Banyak pasien kronis diasumsikan 5% populasi dengan laju pelayanan yang tertera pada Tabel 4.5. Output yang dihasilkan dengan menjalankan simulasi antara lain rata-rata waktu tunggu pasien di tiap rumah sakit. Gambar 5.9 menunjukkan banyak antrian saat menjalankan model simulasi untuk penyakit mata.



Gambar 5.9 Banyak Antrian Simulasi Pemberian Prioritas

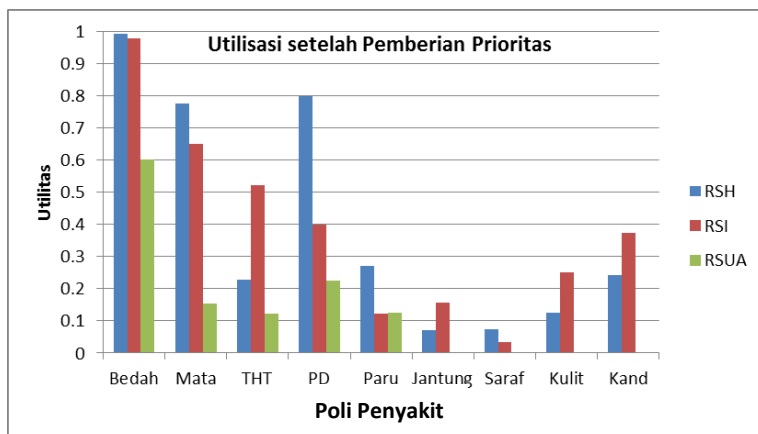
RS Haji menempati urutan preferensi pertama sehingga terlihat pada Gambar 5.9 RS Haji lebih sering terjadi antrian daripada rumah sakit lainnya. Apabila dibandingkan dengan Gambar 5.3, Banyak antrian RS Haji pada satu waktu dapat mencapai 6 orang pada Gambar 5.9. Hal ini disebabkan pasien yang memiliki prioritas didahulukan dan waktu pelayanannya lebih lama daripada pasien tanpa penyakit kronis.

Output yang dihasilkan dengan menjalankan model simulasi routing dinamis antara lain rata-rata waktu tunggu pada masing-masing antrian dan nilai utilisasi server. Gambar 5.10 dan Gambar 5.11 menunjukkan hasil simulasi pemberian prioritas.



Gambar 5.10 Rata-rata waktu tunggu pada simulasi prioritas

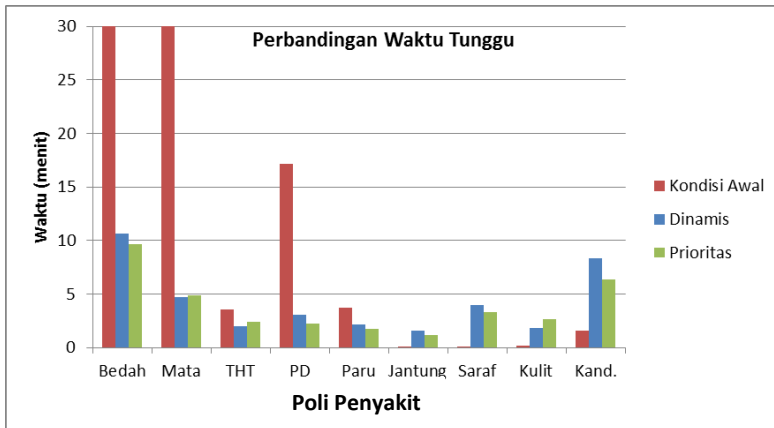
Rata-rata waktu tunggu rumah sakit yang memiliki tingkat preferensi yang tinggi diamati pada Gambar 5.10 lebih lama daripada rumah sakit yang lebih rendah. Waktu tunggu dapat lebih rendah karena dokter yang bertugas lebih banyak seperti pada penyakit mata.



Gambar 5.11 Utilisasi rumah sakit pada simulasi prioritas

Pengamatan selanjutnya dilakukan untuk melihat pengaruh pemberian prioritas pelayanan pasien dalam sistem rujukan berjenjang. routing dinamis pada sistem rujukan. Output pada Gambar 5.10 dan Gambar 5.11 dibandingkan dengan hasil dari routing dinamis dan kondisi awal. Pengaruh yang diamati antara lain pada rata-rata waktu tunggu dan utilisasi masing-masing server.

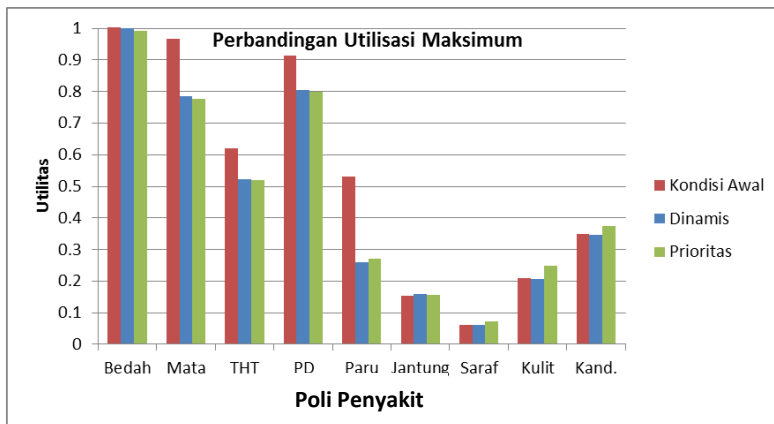
Perbandingan waktu tunggu antara hasil penambahan prioritas dengan routing dinamis dilakukan melalui waktu tunggu rata-rata dari keseluruhan pasien pada suatu jenis penyakit. Waktu tunggu rata-rata dari keseluruhan pasien pada kedua kondisi dapat dilihat pada Gambar 5.12



Gambar 5.12 Perbandingan waktu tunggu dengan adanya prioritas.

Pada Gambar 5.12 dapat diamati bahwa tidak terjadi perubahan signifikan sebelum dan setelah diberikan prioritas. Seharusnya ketika diberikan prioritas pelayanan waktu antrian meningkat karena pasien yang didahulukan memiliki waktu pelayanan yang lebih lama. Hal ini dapat disebabkan kecilnya laju kedatangan sehingga populasi non-kronis yang terbentuk lebih banyak dari proporsi yang semestinya. Waktu pelayanan non-kronis lebih kecil dari waktu pelayanan awal sehingga rata-rata waktu pelayanan keseluruhan berkurang.

Selanjutnya dibandingkan utilisasi maksimum pada kondisi awal, hasil routing dinamis dan pemberian prioritas. Grafik utilisasi maksimum dapat dilihat pada Gambar 5.13

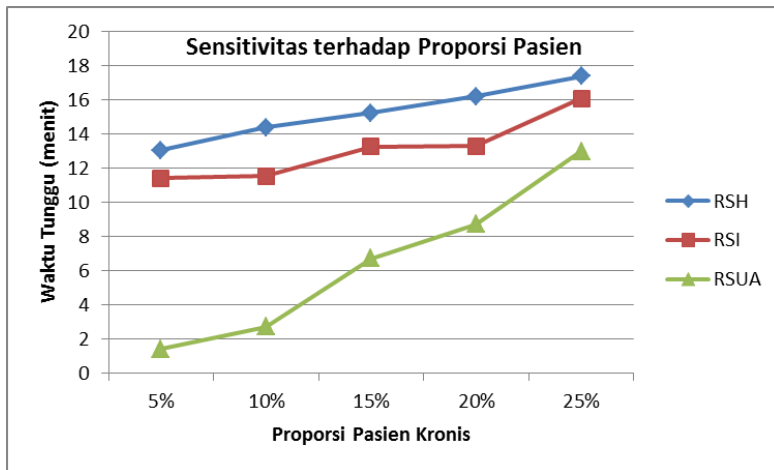


Gambar 5.13 Perbandingan utilisasi antar metode *routing*

Pada Gambar 5.13 secara keseluruhan dapat diamati bahwa tidak terjadi perubahan signifikan pada utilisasi poli dibandingkan saat hanya menggunakan *routing* dinamis. Namun, apabila diperhatikan pada poli penyakit bedah dan mata yang memiliki utilisasi yang tinggi terjadi penurunan utilisasi pada poli penyakit tersebut. Namun terjadi peningkatan utilisasi pada poli penyakit dengan utilisasi yang rendah. Menurut, pengamatan tersebut penerapan prioritas pasien dengan sistem rujukan yang dinamis akan lebih efektif jika utilisasi pada sistem tinggi.

Analisis sensitivitas pada pemberian prioritas dilakukan dengan mengubah proporsi pasien kronis. Pengubahan proporsi pasien kronis akan mengubah banyak pasien penyakit kronis. Hal tersebut akan berpengaruh pada rata-rata waktu tunggu pasien pada rumah sakit.

Proporsi diamati mulai 5% dan ditingkatkan sebesar 5% sampai peningkatan 25%. Pengamatan dilakukan dengan melihat rata-rata waktu tunggu pada rumah sakit pada penyakit beda. Grafik pada Gambar 5.14 menunjukkan sensitivitas *routing* terhadap pengubahan proporsi.



Gambar 5.14 Sensitivitas terhadap Proporsi Pasien Kronis

Gambar 5.14 menunjukkan peningkatan rata-rata waktu tunggu seiring dengan penambahan proporsi pasien kronis. Peningkatan pasien kronis yang memiliki rata-rata waktu pelayanan yang lebih lama dan didahulukan pelayanannya meningkatkan rata-rata waktu tunggu keseluruhan. Laju peningkatan waktu tunggu berbeda untuk RS Haji dan RS Islam apabila dibandingkan dengan RSUA. RSUA mengalami peningkatan yang lebih besar.

Utilisasi rumah sakit penyakit bedah pada Gambar 5.11 menunjukkan bahwa utilisasi RSUA jauh lebih kecil daripada dua rumah sakit lainnya. Dengan meningkatnya proporsi pasien kronis rentang waktu pasien dalam antrian akan lebih lama. Routing dinamis akan mengarahkan lebih banyak pasien menuju rumah sakit yang lebih sepi, dalam kondisi ini RSUA. Hal ini menjelaskan peningkatan RSUA yang besar pada Gambar 5.14

Routing dinamis bekerja saat pemberian prioritas pasien untuk membuat waktu tunggu pada tiap rumah sakit lebih merata. Apabila routing tetap statis maka rumah sakit yang ramai akan mengalami peningkatan yang sama. Hal tersebut kurang baik karena sebenarnya ada rumah sakit lain yang mampu untuk melayani pasien. Oleh karena itu, routing dinamis efektif pada kondisi pemberian prioritas pasien.

LAMPIRAN

LAMPIRAN A - DATA SURVEI

Data Pelayanan Mata

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	9:00	9:06	6
2	9:03	9:08	5
3	9:06	9:12	6
4	9:08	9:16	8
5	9:12	9:21	9
6	9:16	9:22	6
7	9:21	9:25	4
8	9:22	9:27	5
9	9:25	9:30	5
10	9:27	9:34	7
11	9:30	9:38	8
12	9:34	9:40	6
13	9:38	9:42	4
14	9:40	9:43	3
15	9:42	9:48	6
16	9:43	9:51	9

Data Pelayanan THT

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	10:00	10:04	4
2	10:02	10:09	7
3	10:04	10:12	8
4	10:09	10:16	7
5	10:12	10:17	5
6	10:16	10:19	3

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
7	10:17	10:23	6
8	10:19	10:22	3
9	10:22	10:31	9
10	10:23	10:26	3
11	10:26	10:30	4
12	10:30	10:36	6
13	10:31	10:35	4
14	10:35	10:41	6
15	10:36	10:39	3
16	10:39	10:47	8

Data Pelayanan Jantung

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	10:48	10:57	9
2	10:50	10:56	6
3	10:56	11:01	5
4	10:57	11:02	5
5	11:01	11:05	4
6	11:02	11:05	3
7	11:05	11:09	5
8	11:05	11:10	4
9	11:09	11:13	4
10	11:10	11:17	7
11	11:13	11:18	5
12	11:17	11:20	3
13	11:18	11:22	4
14	11:20	11:24	4
15	11:22	11:26	4

Data Pelayanan Kulit

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	9:34	9:39	5
2	9:35	9:41	6
3	9:39	9:45	6
4	9:41	9:44	3
5	9:44	9:49	5
6	9:45	9:58	13
7	9:49	9:54	5
8	9:54	9:58	4
9	9:58	10:00	2
10	9:58	10:03	5
11	10:00	10:05	5
12	10:03	10:14	11
13	10:05	10:12	7

Data Pelayanan Paru

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	8:12	8:21	9
2	8:21	8:29	8
3	8:29	8:34	5
4	8:34	8:38	4
5	8:38	8:45	7
6	8:45	8:55	10
7	8:55	9:06	11
8	9:06	9:11	5
9	9:11	9:16	5
10	9:16	9:23	7
11	9:23	9:30	7

Data Pelayanan Saraf

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	10:15	10:21	6
2	10:17	10:20	3
3	10:20	10:32	12
4	10:21	10:25	4
5	10:25	10:30	5
6	10:30	10:35	5
7	10:32	10:40	8
8	10:35	10:39	4
9	10:40	10:49	9
10	10:39	10:43	4
11	10:43	10:48	5
12	10:48	10:52	4
13	10:49	10:54	5

Data Pelayanan Kandungan

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	8:24	8:28	4
2	8:28	8:33	5
3	8:33	8:40	7
4	8:40	8:45	5
5	8:45	8:52	7
6	8:52	9:01	9
7	9:01	9:14	13
8	9:14	9:23	9
9	9:23	9:29	6
10	9:29	9:37	8
11	9:37	9:44	7

Data Pelayanan Dalam

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	10:55	10:59	4
2	10:57	11:01	4
3	10:59	11:06	7
4	11:01	11:06	5
5	11:06	11:08	2
6	11:06	11:11	5
7	11:08	11:11	3
8	11:11	11:15	4
9	11:11	11:15	4
10	11:15	11:19	5
11	11:15	11:20	4
12	11:19	11:22	3
13	11:20	11:27	7
14	11:22	11:28	6

Data Pelayanan Bedah

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
1	9:06	9:17	11
2	9:07	9:16	9
3	9:09	9:13	4
4	9:13	9:20	7
5	9:16	9:26	10
6	9:17	9:25	8
7	9:20	9:26	6
8	9:25	9:33	8
9	9:26	9:35	9
10	9:26	9:33	7

No	Jam Masuk	Jam Keluar	Waktu Pelayanan
11	9:33	9:37	4
12	9:33	9:40	7
13	9:35	9:44	9
14	9:37	9:47	10

LAMPIRAN B – SCRIPT MATLAB

Script Pengambil Data

```
sheet=input('No.Sheet? ');
trim=[13/60 28/60 13/60;27/60 25/60 30/60;16/60
21/60 19/60;...
      17/60 29/60 18/60;6/60 17/60 10/60;22/60
26/60 24/60;...
      8/60 15/60 15/60;11/60 20/60 13/60;5/60
21/60 3/60;...
      14/60 20/60 15/60;21/60 28/60 14/60;19/60
7/60 21/60;...
      8/60 25/60 8/60;16/60 17/60 18/60;41/60
31/60 43/60;...
      28/60 18/60 33/60;9/60 15/60 15/60;30/60
45/60 25/60];
[lmb in2 pref n
prefu3]=ambil_data('D:\MatLAB\bin\coba\RUM_BARU.
xlsx',sheet);
myu=in2*ones(size(n));
n_myu=n.*myu;
```

Script Estimasi Hypercube

```
%Model Hypercube
%% ParameterInput
%Lambda := Laju kedatangan dari tiap atom
geografis, suku ke-i menunjuk-
%          laju kedatangan dari atom geografis
i;
%Myu := Banyak server yang dimodelkan;
%Pref := Matriks preferensi user dari tiap atom,
suku indeks (i,j)
%          server pilihan atom geografis i pada
urutan j;
%% Pengambilan Input
while(1)
    lambda=input('Laju kedatangan dari tiap
atom(size 1xN_atom) ');
```



```

    myu=input('Laju pelayanan rata-rata server
    ');
    pref=input('Matriks preferensi user(size
N_atomxN_serv) ');
    sl=size(lambda);
    sp=size(pref);
    if sp(1)==sl(2)
        break
    end
end
% lambda=lmb;
% myu=in2;
% sl=size(lambda);
% sp=size(pref);
%% Inisiasi Parameter
N=size(pref);
%N(1) := banyak atom(N_atom);
%N(2) := banyak server(N_serv);
pref2=prefver2(pref); %fungsi prefver2 mengubah
representasi pref
%menjadi tipe cell
L=sum(lambda);
%L := Jumlah seluruh laju kedatangan;
%Hitung performa antrian M/M/n dengan laju
kedatangan L ...
%laju pelayanan tiap server Myu dan kapasitas
antrian tak hingga
[P u]=MMn_queue(L,myu,N(2));
%P := Array yang menunjukkan probabilitas banyak
user dalam sistem,
%     indeks i berisi probabilitas i-1 user
didalam sistem
%     untuk kapasitas antrian tak hingga suku
terakhir menunjukkan
%     probabilitas terjadinya antrian;
%u := rata-rata utilitas server;
%Hitung queue correction function untuk
kapasitas tak hingga
Q=qcf(N(2),L,myu);
%Hitung bagian laju kedatangan dari masing2 atom

```

```

frac_L=lambda/L;
%% Mencari estimasi workload
%Inisiasi iterasi
stop=0;
%Inisiasi workload
work=u*ones(1,N(2));
%work := array yang merepresentasikan estimasi
workload dari tiap server
%      suku indeks i berisi estimasi workload
server i;
buff=work;% buff menyimpan nilai workload dari
setiap iterasi
%Iterasi untuk mengestimasi workload
while(1)
    stop=stop+1; %Tambahkan iterasi
    prev=work; %Menyimpan estimasi iterasi
    sebelumnya
    %Penyelesaian persamaan matriks pada
    persamaan () di bab III

    work=Update_load_inf(Q,work,frac_L,pref,pref2,P(
    N(2)+2),L);
    %Normalisasi terhadap rata-rata workload
    gamma=(N(2)*u)/sum(work);% Koefisien
    normalisasi
    work=gamma*work;
    buff=[buff; work];% menyimpan hasil
    estimasi;
    err=max(abs(work-prev));%e=max(|rho_k-rho(k-
    1)|);
    %Iterasi berakhir apabila error < 10^-6 atau
    sudah 1000 iterasi
    if err<10^-6 || stop==1000
        break
    end
end
%% (OPTIONAL) Menampilkan hasil iterasi
%% Menghitung ketidakseimbangan workload
%Deviasi absolut
rhonxn=ones(N(2),1)*work;

```

```

abs_err=max(abs(rhonxn-rhonxn')) ;
%Varians dan Standar Deviasi
varia=sum(work.^2)/N(2)-u^2;
st_dev=varia^0.5;
%Persentase deviasi maksimum dari mean
ratio=max(abs(work-u*ones(1,N(2))))*100/u;
%% Mengestimasi bagian penempatan dari tiap atom
menuju server
%Menempatkan workload sesuai preferensi atom
prefwork=zeros(N); % Membuat kerangka solusi
for i=1:N(1)
    for j=1:N(2)
        prefwork(i,j)=work(pref(i,j));
    end
end
%Proses awal untuk mendapatkan hasil kali rho
sebelum pilihan ke j
prefwork2=ones(N);
for i=1:N(1)
    for j=2:N(2)
        prefwork2(i,j)=prefwork2(i,j-
1)*prefwork(i,j-1);
    end
end
%Persamaan pada bab III
frac_dstr=(frac_L'*Q).*(prefwork2.*(ones(N)-
prefwork));
%Normalisasi
norm=((frac_L*(1-
P(N(2)+2)))./sum(frac_dstr'))'*ones(1,N(2));
frac_dstr=frac_dstr.*(norm);
%% Menghitung performa antrian masing-masing
poli
%Mendapatkan laju kedatangan pada masing-masing
server
arrive=work*myu;
%Menjumlahkan laju kedatangan menuju server-
server dalam satu poli
k=1;
elmb=zeros(1,numel(n));

```

```

for i=1:numel(n)
    elmb(i)=sum(arrive(1,k:k+n(i)-1));
    k=k+n(i);
end
%Mendapatkan performa antrian
antri=zeros(size(n));
lama=antri;
wl=zeros(size(n));
for i=1:numel(n)
    if n(i)~=0
        [Pp Up]=MMn_queue(elmb(i),myu,n(i));
        antri(i)=Pp(n(i)+2)/(1-Up);
        lama(i)=antri(i)/elmb(i);
        wl(i)=Up;
    end
end
%% Inisialisasi Simulasi
myu=myu*ones(size(n));
pref3=prefver3(prefu3,n);
Tulis_Data;
avm=sum(myu)/3;
kron=0.05;

```

Fungsi MMn_queue

```

%M/M/n queue
%input lambda, numb server
function [prob util]= MMn_queue(l,m,n)
    prob=zeros(1,n+2);
    r=l/(n*m);
    a=1;
    for i=1:n
        prob(i)=a;
        a=a*r*n/i;
    end
    %Inf queue
    prob(n+1)=a;
    prob(n+2)=a*r/(1-r);
    GNk=sum(prob);

```

```

prob=prob/GNk;
U=zeros(1,n);
for i=1:n
    U(i)=i*prob(i+1);
end
U(n)=U(n)+n*prob(n+2);
util=sum(U)/n;
end

```

Fungsi Update load inf

```

function out =
Update_load_inf(Q,load,L,pref,pref2,busy,l)
% Update_load := This function is used to
update the estimated
% workload of each server on hypercube
queueing system based on
% equation 21 on (Lars84)
% ----INPUT----
% Q = queue correction factor
% load = previous workload of each server
% L = arrival rate of each geo atoms
% pref =
% pref2
Spf=size(pref);
Sps=size(pref2);
%Update Preprocess
Up_lod=zeros(Spf(1),Spf(2));
for i=1:Spf(1)
    Up_lod(i,1)=L(i);
    for j=2:Spf(2)
        Up_lod(i,j)=Up_lod(i,j-
1)*load(pref(i,j-1));
    end
end
%Update load
num=ones(1,Spf(2));
den=ones(1,Spf(2));
for i=1:Sps(1)

```

```

        for j=1:Sps(2)
            for k=cell2mat(pref2(i,j))
                den(i)=den(i)+Q(j)*Up_lod(k,j);
            end
        end
        den(i)=den(i)+(1*busy/Sps(1))/load(i);
    end
    load_up=num./den;
    out=ones(1,Spf(2))-load_up;
end

```

Fungsi qcf

```

function Q=qcf(N,l,m)
    Q=zeros(1,N);
    rho=l/(N*m);
    layer1=MMn_queue(l,m,N);
    final=zeros(1,N);
    for j=0:(N-1)
        for k=j:(N-1)

            final(j+1)=final(j+1)+factorial(k)*factorial(N-
            j)*(N-k)*layer1(k+1)/(factorial(N)*factorial(k-
            j)*(N-j));
        end
    end
    for i=1:N
        Q(i)=final(i)/((rho)^(i-1)*(1-rho));
    end
end

```

Fungsi Penentuan Routing Dinamis

```

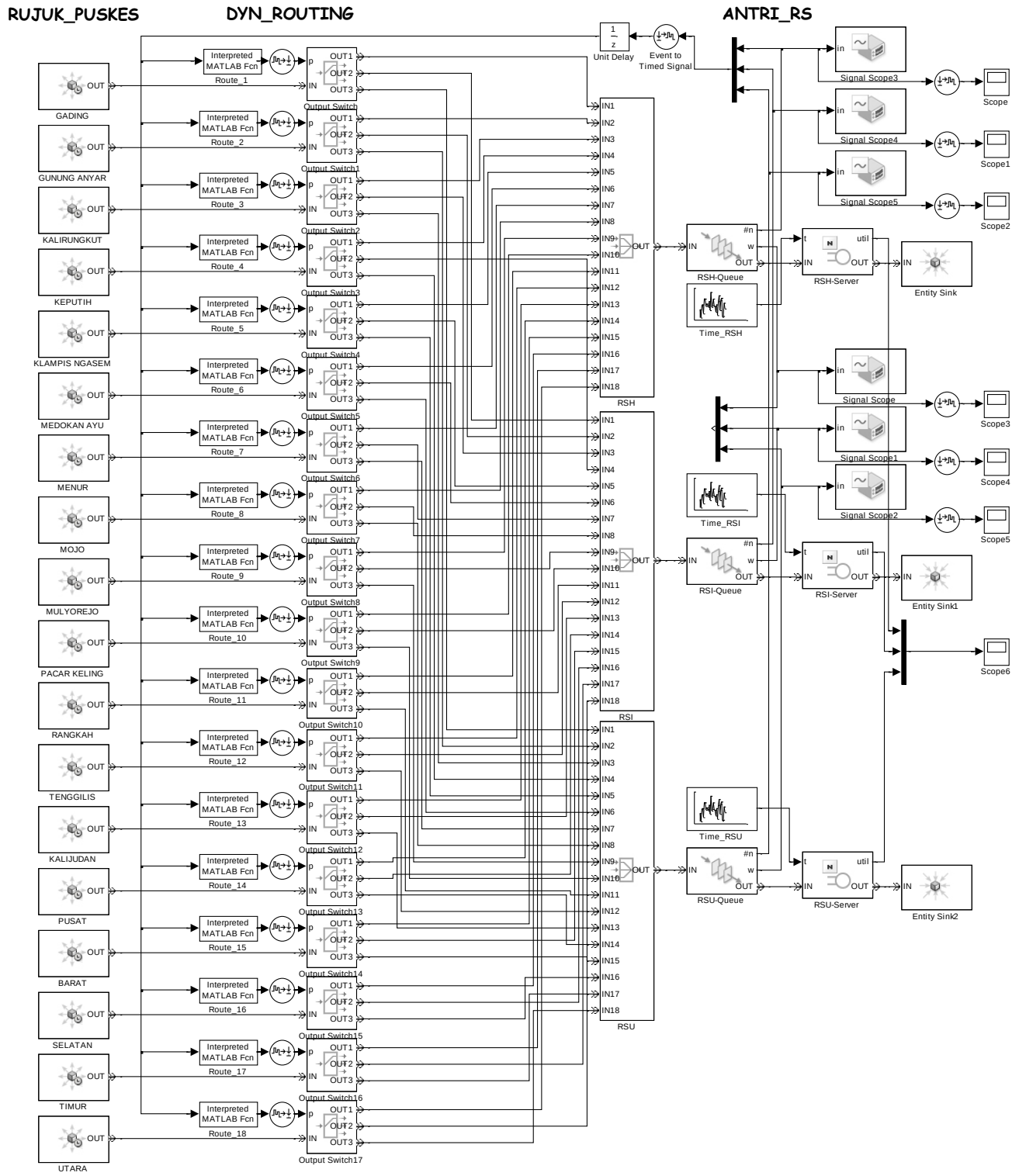
function
out=dyn_pref3(n_queue,lmb,myu,pref,trim,tom_numb
)
    %FungsiPreferensi
    f_pref=((n_queue'+lmb.*trim)./myu+pref*0.05);

```

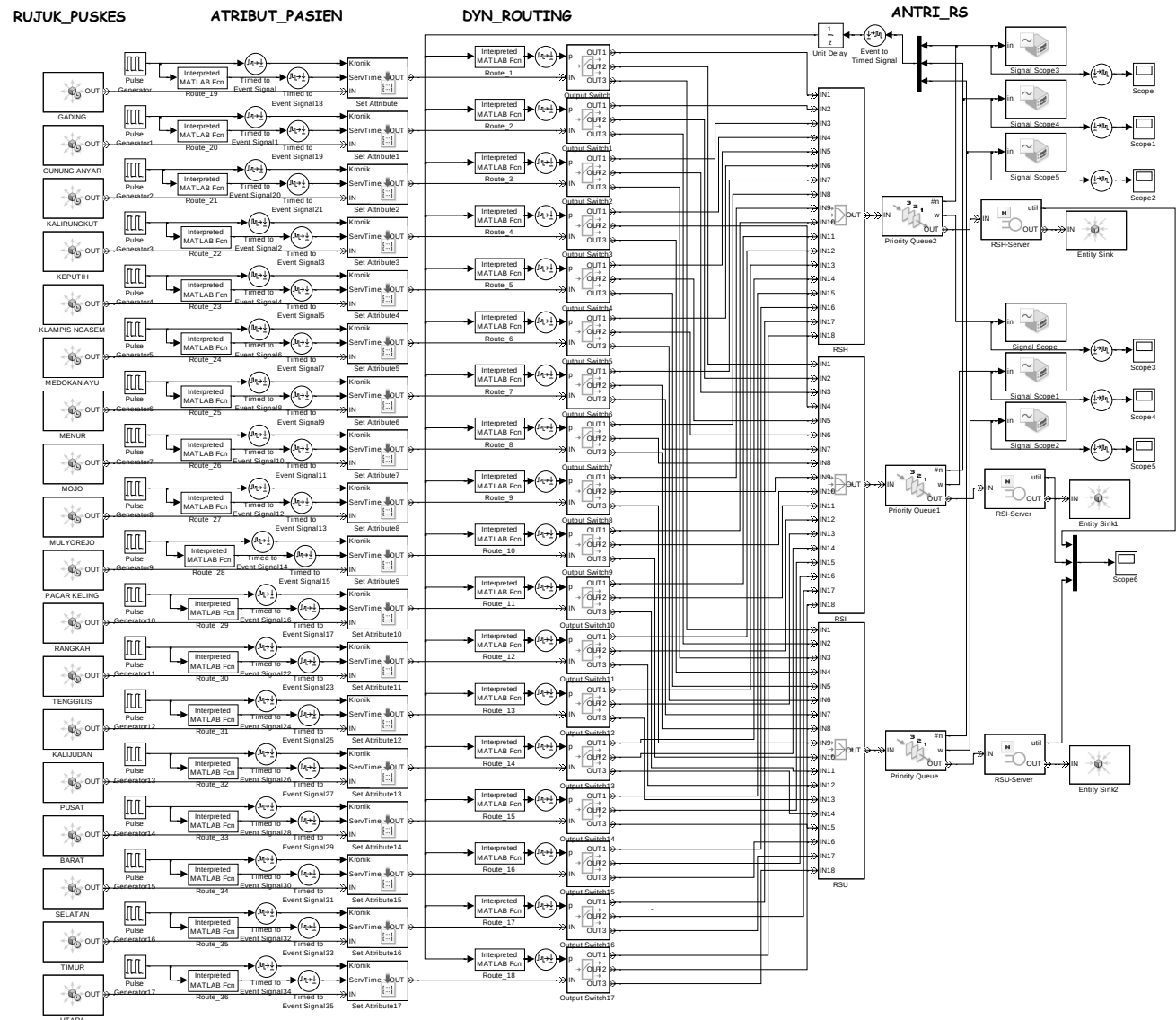
```
[nmks N]=min(f_pref);  
%out=[N f_pref];  
out=N;  
end
```

LAMPIRAN C – MODEL SIMULASI

Model Simulasi Routing Dinamis



Model Simulasi Pemberian Prioritas



BAB 6

PENUTUP

Pada bagian ini akan dibahas kesimpulan yang telah diambil dan saran- saran mengenai pengembangan tugas akhir berjudul **“Permodelan dan Analisis Performa Jaringan Antrian dari Sistem Rujukan Berjenjang BPJS di Surabaya”**.

6.1 Kesimpulan

Setelah melakukan analisis hasil simulasi model untuk routing statis, dinamis dan pemberian prioritas pelayanan pasien, penulis dapat menyimpulkan hal-hal berikut:

1. Pengubahan preferensi akan mengubah laju kedatangan menuju rumah sakit sesuai urutan preferensinya.
2. Penerapan routing dinamis dapat memperbaiki kondisi tersebut sehingga semua antrian stabil.
3. Penerapan routing dinamis memiliki pengaruh yang besar terhadap waktu tunggu dan workload pada sistem dengan workload awal yang tinggi.
4. Peningkatan laju kedatangan meningkatkan utilisasi tiap rumah sakit, pengaruh routing dinamis mengakibatkan lebih pasien banyak dirujuk menuju rumah sakit dengan dokter yang lebih banyak.
5. Terjadi perubahan waktu tunggu pada model saat pemberian prioritas pasien dibandingkan saat hanya menggunakan routing dinamis.
6. Penerapan prioritas pasien akan memperbaiki routing dinamis pada kondisi workload sistem yang tinggi.
7. Routing dinamis mendistribusi pasien sesuai waktu tunggu sehingga ketika jumlah pasien kronis bertambah rumah sakit yang lebih sepi terisi.

6.2 Saran

Adapun saran yang dapat diberikan penulis untuk pengembangan topik tugas akhir ini antara lain:

1. Mempertimbangkan pengaruh kedatangan user dengan tipe umum yang tidak menggunakan BPJS.

2. Mempertimbangkan pengaruh kedatangan user dari luar kota.
3. Penambahan tingkat pelayanan kesehatan tingkat sub-spesialis dan memperhatikan adanya rujuk balik.

DAFTAR PUSTAKA

- [1] Dede S. 2015, 'Analisis Pelaksanaan Sistem Rujukan Rawat Jalan Tingkat Pertama (RJTP) pada Peserta BPJS Kesehatan di Puskesmas 5 Ilir dan Puskesmas Merdeka', Repositori Universitas Sriwijaya.
- [2] Mengyu G., dkk. 2011, 'Effectiveness of Referral Incentive Policy: Exploring Using Queueing Network Model with Blocking', dalam *International Conference on Service Systems and Service Management (ICSSSM)*, 2011, hal. 1-6.
- [3] Cayirli, T. dkk. 2006, 'Designing Appointment Scheduling Systems for Ambulatory Care Services', *Journal of Health Care Management Sciences*, no. 9, hal. 47-58.
- [4] Trivedi, K.S. 2001, *Probability and Statistics with Reliability, Queueing, and Computer Science Applications*, 2nd Ed, Wiley, New Jersey.
- [5] Oliver, C.I. 2011, *Fundamentals of Stochastic Network*, Wiley, New Jersey.
- [6] Larson, R.C. 1981, *Urban Operation Research*, Prentice-Hall, New Jersey.
- [7] Zhang Y., dkk., "A Performance Comparison of Adaptive and Static Load Balancing in Heterogeneous Distributed Systems," *Proc. IEEE 28th Ann. Simulation Symp.*, pp. 332-340, Phoenix, Ariz., April 1995.
- [8] Watts, J. dan Taylor, S. "A Practical Approach to Dynamic Load Balancing," *IEEE Trans. Parallel and Distributed Systems*, vol. 9, no. 3, pp. 235-248, March 1998.
- [9] Larson, R.C. 1974, *Urban Emergency Service System: An Iterative Procedure for Approximating Performance Characteristics*, The Rand Institute, New York.

Halaman ini sengaja dikosongkan

RIWAYAT HIDUP



Khairurizal Alfathdyanto lahir bulan Oktober di Madiun, Jawa Timur, merupakan putra pertama dari 4 bersaudara dari pasangan Triwikantoro dan Dyah Agus Setyowati. Menamatkan pendidikan dasar di SDN Klampis Ngasem I Surabaya dan melanjutkan ke SMPN 1 Surabaya. Kemudian menamatkan tingkat SMA di SMAN SBBS Sragen hingga lulus pada tahun 2012. Penulis kemudian diterima di Jurusan Teknik Elektro, FTI-ITS. Selama menjadi mahasiswa, penulis pernah menjadi sekertaris

UKM IFLS pada tahun 2014-2015, dan asisten laboratorium teknik sistem sejak tahun 2015. Pada tahun 2016 penulis mengikuti seminar dan ujian lisan Tugas Akhir di bidang Teknik Sistem Pengaturan, Jurusan Tekni Elektro, FTI-ITS.

Email : rizal.alfath@gmail.com