



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE141599

***SECURITY CONSTRAINED UNIT COMMITMENT
MEMPERTIMBANGKAN CADANGAN BERPUTAR DAN
KAPASITAS SALURAN TRANSMISI MENGGUNAKAN
ALGORITMA BINARY PARTICLE SWARM OPTIMIZATION***

Anton Frian Yohanes Reynaldo
NRP 2212100165

Dosen Pembimbing
Dr. Rony Seto Wibowo, ST., MT.
Prof. Dr. Ir. Adi Soeprijanto, MT.

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



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE141599

***SECURITY CONSTRAINED UNIT COMMITMENT
CONSIDERING SPINNING RESERVE AND TRANSMISSION
LINE CAPACITY USING BINARY PARTICLE SWARM
OPTIMIZATION ALGORITHM***

Anton Frian Yohanes Reynaldo
NRP 2212100165

Supervisor
Dr. Rony Seto Wibowo, ST., MT.
Prof. Dr. Ir. Adi Soeprijanto, MT.

ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

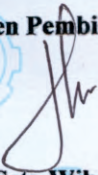
***SECURITY CONSTRAINED UNIT COMMITMENT
MEMPERTIMBANGKAN CADANGAN BERPUTAR DAN
KAPASITAS SALURAN TRANSMISI MENGGUNAKAN
ALGORITMA BINARY PARTICLE SWARM
OPTIMIZATION***

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Tenaga
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing I



Dr. Rony Seto Wibowo, S.T., M.T.
NIP. 1974 11 29 2000 12 1001

Dosen Pembimbing II



Prof. Dr. Ir. Adi Soeprijanto, MT.
NIP. 1964 04 05 1990 02 1001



SECURITY CONSTRAINED UNIT COMMITMENT MEMPERTIMBANGKAN CADANGAN BERPUTAR DAN KAPASITAS SALURAN TRANSMISI MENGUNAKAN ALGORITMA *BINARY PARTICLE SWARM OPTIMIZATION*

Nama Mahasiswa : Anton Friar Yohanes Reynaldo
NRP : 2212100165
Dosen Pembimbing I : Dr. Rony Seto Wibowo, ST., MT.
NIP : 197411292000121001
Dosen Pembimbing II : Prof. Dr. Ir. Adi Soeprijanto, MT.
NIP : 196404051990021001

ABSTRAK

Manusia modern saat ini tidak akan bisa hidup dan bekerja secara efektif dan efisien tanpa energi listrik. Energi listrik merupakan energi yang sangat mempengaruhi perkembangan teknologi dan membutuhkan suatu operasi sistem untuk pembangkitan dan penyalurannya. Operasi sistem tenaga pada sistem kelistrikan diharapkan dapat menghasilkan sistem yang handal dan optimal secara ekonomi untuk mengatasi kondisi beban yang fluktuatif dalam 24 jam. *Unit Commitment (UC)* dapat digunakan untuk melakukan penjadwalan nyala-padam unit pembangkit dalam sistem kelistrikan yang terinterkoneksi secara optimal. Namun *UC* belum mampu melakukan penjadwalan pembangkit dengan mempertimbangkan kapasitas saluran transmisi. *Security Constrained Unit Commitment (SCUC)* merupakan pengembangan dari *UC* dengan mempertimbangkan saluran transmisi sebagai batasan perhitungan. Algoritma *particle swarm optimization* digunakan untuk menyelesaikan *SCUC*. Penggunaan algoritma *particle swarm optimization* diharapkan dapat menyelesaikan permasalahan penjadwalan nyala-padam unit pembangkit jika sistem tidak memenuhi standar keamanan, dalam hal ini mempertimbangkan kapasitas saluran transmisi. Hasil penelitian ini berupa software perhitungan penjadwalan pembangkit yang lebih sesuai dan memenuhi standar sistem.

Kata kunci : SCUC, Cadangan Berputar, Transmisi, BPSO

SECURITY CONSTRAINED UNIT COMMITMENT (SCUC) CONSIDERING SPINNING RESERVE AND TRANSMISSION LINE CAPACITY USING BINARY PARTICLE SWARM OPTIMIZATION ALGORITHM

Nama Mahasiswa : Anton Friar Yohanes Reynaldo
NRP : 2212100165
Supervisor I : Dr. Rony Seto Wibowo, ST., MT.
NIP : 197411292000121001
Supervisor II : Prof. Dr. Ir. Adi Soeprijanto, MT.
NIP : 196404051990021001

ABSTRACT

Modern human nowadays couldn't live and work effectively and efficiently without electricity. Electrical energy is an energy that really affects technology's development and needs a system operation to generate and transmit the electricity. Power system operation in electrical system is expected to produce a reliable system and economically optimal to solve the fluctuative load condition in 24 hours. Unit Commitment (UC) could be used for on-off generation unit scheduling in the electrical system which is interconnected optimally. However, UC still could not do the unit scheduling by considering transmission line capacity. Security Constrained Unit Commitment (SCUC) is the development of UC with considering transmission line capacity as the calculation constraint. Particle swarm optimization algorithm is used to solve the SCUC. The application of SCUC by using particle swarm optimization is expected able to solve on-off generation unit scheduling, in case if the system couldn't fulfill the security standard, which is considering the transmission line capacity. The result of this research is a generation unit scheduling calculation software that is more appropriate and meets the system standard.

Index terms : SCUC, Transmission Capacity, Spinning Reserve, Binary Particle Swarm Optimization

DAFTAR ISI

HALAMAN JUDUL	
LEMBAR PENGESAHAN	
ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian.....	2
1.3 Permasalahan	2
1.4 Batasan Masalah.....	2
1.5 Metode Penelitian	2
1.6 Sistematika Penulisan	4
1.7 Relevansi.....	4
BAB 2 OPERASI OPTIMUM SISTEM TENAGA LISTRIK....	5
2.1 Fundamental Sistem Tenaga Listrik	5
2.1.1 Sub-sistem Pembangkitan	6
2.1.2 Sub-sistem Transmisi	7
2.1.3 Sub-sistem Distribusi	7
2.1.4 Beban	7
2.2 Perencanaan Operasi Optimum Sistem Tenaga Listrik.....	8
2.2.1 Penjadwalan Pembangkit Tenaga Listrik	8
2.2.1.1 Kombinasi	9
2.2.2 Pengaturan Daya Keluaran Pembangkit	10
2.2.2.1 Linear Programming.....	10
2.2.2.2 Quadratic Programming.....	11
2.2.3 Aliran Daya Optimal Pada Sistem Tenaga.....	11
2.2.3.1 Dynamic Optimal Power Flow Arus Searah	12
2.2.4 Security Constrained Unit Commitment.....	12

BAB 3	PENJADWALAN PEMBANGKIT MENGGUNAKAN <i>BINARY PARTICLE SWARM OPTIMIZATION</i>.....	13
3.1	Algoritma	13
3.2	Quadratic Programming.....	17
3.3	Penerapan Algoritma <i>Binary Particle Swarm Optimization</i> pada <i>Unit Commitment</i>	18
3.3.1	Inisialisasi Awal <i>BPSO</i>	19
3.3.1.1	<i>Sistem IEEE 30 Bus</i>	19
3.3.1.2	<i>Parameter-parameter BPSO</i>	20
3.3.2	Inisialisasi Posisi dan Kecepatan Awal Partikel	21
3.3.3	Perhitungan Fungsi Biaya dengan <i>Quadratic Programming</i>	21
3.3.3.1	<i>Penerapan Quadratic Programming pada SCUC</i>	21
3.3.3.2	<i>Argumen Input dari Quadprog</i>	22
3.3.3.3	<i>Inisialisasi Fungsi Objektif dan Constraints dari SCUC</i>	22
3.3.3.4	<i>Pembentukan Matrik H Simetris yang Menyatakan Bentuk Kuadrat dari Fungsi Objektif</i>	23
3.3.3.5	<i>Pembentukan Vektor Linier f yang Menyatakan Bentuk Linear dari Fungsi Objektif</i>	24
3.3.3.6	<i>Pembentukan Matrik Aeq yang Menyatakan Koefisien Linear dari Constraints $lb \leq Aeq \cdot x \leq ub$</i>	24
3.3.3.7	<i>Pembentukan Matrik Beq yang Menyatakan Koefisien Linear dari Constraints $Aeq \cdot x = Beq$</i>	24
3.3.3.8	<i>Pembentukan Vektor lb dan ub yang Menyatakan Batas Atas dan Batas Bawah dari Constraints $lb \leq Aeq \cdot x \leq ub$</i>	25
3.3.4	Evaluasi Fitness Populasi.....	25
3.3.5	Penentuan Posisi Terbaik Lokal dan Global.....	25
3.3.6	Pembaruan Kecepatan dan Posisi Partikel	26
3.3.7	Pembatasan Pergerakan Partikel.....	27
3.3.8	Hasil Penjadwalan	27

BAB 4	SIMULASI DAN ANALISIS DATA	29
4.1	Performa Komputasi	29
4.2	Hasil Simulasi	29
4.2.1	Sistem IEEE 30 Bus.....	29
4.2.1.1	<i>Penjadwalan Pembangkit Sistem IEEE 30 Bus</i>	30
4.2.1.2	<i>Pembangkitan Daya Aktif Sistem IEEE 30 Bus</i>	31
4.2.1.3	<i>Pembangkitan Daya Aktif Maksimal Sistem IEEE 30 Bus.....</i>	32
4.2.1.4	<i>Total Biaya Pembangkitan Sistem IEEE 30 Bus.....</i>	34
4.3	Analisa Hasil Simulasi.....	35
BAB 5	PENUTUP	39
5.1	Kesimpulan	39
5.2	Saran	39
DAFTAR PUSTAKA		41
BIODATA PENULIS		43

DAFTAR TABEL

Tabel 3.1	Data batasan unit pembangkit IEEE 30 Bus	19
Tabel 3.2	Data biaya operasi sistem IEEE 30 Bus	19
Tabel 3.3	Data pembebanan sistem IEEE 30 Bus 24 jam	20
Tabel 4.1	Data spesifikasi komputasi	29
Tabel 4.2	Kondisi awal unit pembangkit IEEE 30 Bus.....	29
Tabel 4.3	Penjadwalan pembangkit sistem IEEE 30 Bus	30
Tabel 4.4	Pembangkitan daya aktif sistem IEEE 30 Bus(1)...	31
Tabel 4.5	Pembangkitan daya aktif sistem IEEE 30 Bus(2)...	31
Tabel 4.6	Pembangkitan daya aktif sistem IEEE 30 Bus(3)...	31
Tabel 4.7	Pembangkitan daya aktif sistem IEEE 30 Bus(4)...	32
Tabel 4.8	Pembangkitan daya aktif maksimal sistem IEEE 30 Bus (1).....	32
Tabel 4.9	Pembangkitan daya aktif maksimal sistem IEEE 30 Bus (2).....	33
Tabel 4.10	Pembangkitan daya aktif maksimal sistem IEEE 30 Bus (3).....	33
Tabel 4.11	Pembangkitan daya aktif maksimal sistem IEEE 30 Bus (4).....	33
Tabel 4.12	Total biaya pembangkitan sistem IEEE 30 Bus	34

DAFTAR GAMBAR

Gambar 2.1	Sistem tenaga listrik secara umum.....	5
Gambar 3.1	Konsep pencarian Pbest dan Gbest dari PSO	13
Gambar 3.2	Diagram alir penyelesaian SCUC menggunakan BPSO	18
Gambar 4.1	Perbandingan daya pembangkitan dengan daya beban sistem IEEE 30 Bus	35
Gambar 4.2	Perbandingan daya pembangkitan maksimal dengan cadangan berputar sistem IEEE 30 Bus.....	36
Gambar 4.3	Perbandingan total biaya pembangkitan.....	37

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Manusia modern saat ini tidak akan bisa hidup dan bekerja secara efektif dan efisien tanpa energi listrik. Energi listrik merupakan energi yang sangat mempengaruhi perkembangan teknologi dan membutuhkan suatu operasi sistem untuk pembangkitan dan penyalurannya. Pada bidang sistem tenaga listrik, umumnya total kebutuhan energi listrik yang diperlukan konsumen lebih tinggi saat siang-sore hari, dimana beban industri tinggi, lampu dan mesin-mesin listrik banyak yang menyala [1]. Sebaliknya, kebutuhan energi listrik menurun saat malam-pagi hari, disaat kebanyakan masyarakat sedang tidur dan beristirahat. Untuk mengimbangi permintaan ketersediaan energi listrik dibutuhkan kapasitas pembangkit listrik yang mengalami peningkatan terus menerus sehingga dibutuhkan kombinasi nyala-mati unit pembangkit [2]. Jumlah beban yang harus disuplai unit-unit pembangkit selalu berubah setiap waktu. Oleh karena itu, pada penyaluran daya listrik, jumlah energi listrik terbangkitkan harus bernilai sama dengan jumlah beban yang harus disuplai untuk memperoleh biaya pembangkitan paling ekonomis. Selain itu penjadwalan unit pembangkit harus memperhatikan cadangan berputar. Jenis operasi sistem tenaga listrik area pembangkitan dapat berupa peningkatan dan penurunan daya output, serta penyalaan dan pemadaman unit-unit pembangkit untuk mengimbangi jumlah beban yang ditanggung unit-unit pembangkit [3].

Security Constrained Unit Commitment (SCUC) adalah penjadwalan nyala-mati unit-unit pembangkit yang beroperasi pada suatu sistem tenaga listrik, dengan mempertimbangkan keamanan dan kapasitas saluran. Penyalaan dan pemadaman unit pembangkit pada suatu sistem tenaga listrik memiliki banyak pertimbangan, seperti cadangan berputar (*spinning reserve*), waktu nyala minimal (*minimum up time*), waktu padam minimal (*minimum down time*), dan rentang daya keluaran unit [3].

Setiap unit pembangkit memiliki karakteristik input-output yang khas yang sangat mempengaruhi fungsi biaya pembangkitan unit pembangkit. Pada tugas akhir ini akan diusulkan algoritma BPSO (*Binary Particle Swarm Optimization*) untuk menyelesaikan permasalahan unit commitment secara optimal.

1.2 Tujuan Penelitian

Tujuan yang ingin dicapai dari tugas akhir ini adalah :

1. Membuat program penjadwalan unit pembangkit yang ekonomis dan aman dengan algoritma BPSO (*Binary Particle Swarm Optimization*).
2. Menentukan penjadwalan dengan biaya pembangkitan yang ekonomis dan aman menggunakan perhitungan DCOPF, dengan mempertimbangkan kondisi kontingensi.

1.3 Permasalahan

Permasalahan yang akan diselesaikan dalam tugas akhir ini adalah :

1. Bagaimana membuat program penjadwalan ekonomis pembangkit yang aman dan sesuai standar dengan mempertimbangkan fungsi biaya pembangkitan menggunakan algoritma BPSO (*Binary Particle Swarm Optimization*).
2. Bagaimana menentukan penjadwalan unit pembangkit dengan biaya pembangkitan yang ekonomis dan aman menggunakan perhitungan DCOPF, dengan mempertimbangkan kondisi kontingensi.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah :

1. *Losses* pada saluran transmisi diabaikan untuk DCOPF.
2. Menggunakan sistem uji IEEE 30 bus.
3. Simulasi dilakukan dengan menggunakan *Matlab Matpower*.
4. Metode yang digunakan untuk menyelesaikan permasalahan adalah algoritma BPSO (*Binary Particle Swarm Optimization*) dan *quadratic programming* untuk DCOPF.
5. Profil beban sama untuk ketiga kondisi: normal, kontingensi I, kontingensi II.

1.5 Metode Penelitian

Pada tugas akhir ini dilakukan penelitian tentang penjadwalan *security constrained unit commitment* dengan profil beban sama untuk tiga kondisi: normal, kontingensi I, kontingensi II. Untuk mendapatkan penjadwalan dengan biaya pembangkitan yang ekonomis akan digunakan

algoritma BPSO (*Binary Particle Swarm Optimization*) dan *quadratic programming* untuk DCOPF sebagai metode penyelesaian.

Data yang digunakan dalam tugas akhir ini meliputi waktu penyalan dan pemadaman minimal, biaya penyalan, fungsi biaya pembangkitan dan batasan-batasan unit pembangkit, serta profil beban sama untuk ketiga kondisi berbeda: normal, kontingensi I, kontingensi II. Perhitungan operasi *security constrained unit commitment* akan disimulasikan dalam pada *M-File* dalam Matlab. Tahapan dalam pengerjaan tugas akhir ini adalah :

1. Studi literatur
Literatur-literatur yang digunakan berasal dari jurnal, prosiding dan *text book*.
2. Pengumpulan data
Data yang dibutuhkan berupa waktu penyalan dan pemadaman minimal, biaya penyalan, biaya pemadaman, fungsi biaya pembangkitan, dan rentang daya pembangkitan untuk masing-masing pembangkit, serta penentuan tiga kondisi berbeda: normal, kontingensi I, kontingensi II.
3. Pembuatan program dalam *M-File* pada Matlab serta penerapan algoritma BPSO (*Binary Particle Swarm Optimization*) pada operasi *unit commitment* dan penggunaan *quadratic programming* untuk DCOPF.
4. Simulasi dan analisis
Simulasi dilakukan dengan sistem uji IEEE 30 Bus dalam penjadwalan *security constrained unit commitment*. Data yang diperoleh dari hasil simulasi tersebut akan dianalisis untuk mengetahui keberhasilan dan keefektifan algoritma yang diusulkan terhadap penjadwalan dan biaya total pembangkitan.
5. Penulisan buku
Penulisan laporan dilakukan sebagai penggambaran kesimpulan dari tugas akhir ini. Kesimpulan tersebut merupakan jawaban dari permasalahan yang dianalisis. Selain itu juga akan diberikan saran sebagai masukan berkaitan dengan apa yang telah dilakukan.

1.6 Sistematika Penulisan

Laporan tugas akhir ini disusun dalam suatu sistematika sebagai berikut :

1. Bab 1 : PENDAHULUAN
Bab ini membahas mengenai latar belakang, tujuan penelitian, permasalahan, batasan masalah, metode penelitian, sistematika penulisan dan relevansi dari tugas akhir.
2. Bab 2 : SECURITYCONSTRAINED UNIT COMMITMENT
Bab ini berisi tentang dasar teori yang merupakan penjelasan mengenai *security constrained unit commitment*, *optimal power flow*, karakteristik *input-output* unit pembangkit, fungsi biaya.
3. Bab 3 : SCUC MENGGUNAKAN BPSO (*Binary Particle Swarm Optimization*) dan *Quadratic Programming* DCOPF.
Bab ini berisi tentang penjelasan mengenai penerapan algoritma BPSO (*Binary Particle Swarm Optimization*) dan penggunaan fungsi biaya pembangkitan pada penjadwalan *security constrained unit commitment*.
4. Bab 4 : SIMULASI DAN ANALISIS
Bab ini berisi hasil simulasi penjadwalan unit commitment menggunakan algoritma BPSO (*Binary Particle Swarm Optimization*) serta analisis biaya pembangkitan.
5. Bab 5 : PENUTUP
Kesimpulan yang diperoleh dari hasil pembahasan dan saran-saran yang berhubungan dengan pokok-pokok pembahasan.

1.7 Relevansi

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

1. Dapat memberikan manfaat terhadap perkembangan sistem tenaga listrik khususnya pada permasalahan yang mengenai *security constrained unit commitment* dan algoritma BPSO (*Binary Particle Swarm Optimization*).
2. Dapat digunakan sebagai referensi bagi mahasiswa lain yang akan mengambil tugas akhir dengan permasalahan yang serupa.
3. Dapat menambah ilmu pengetahuan di bidang operasi sistem tenaga listrik bagi penulis.

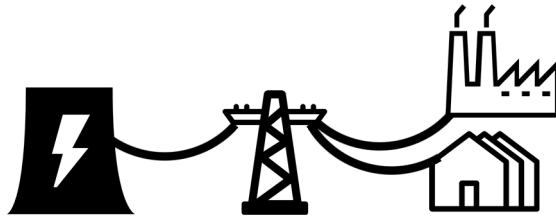
BAB 2

OPERASI OPTIMUM SISTEM TENAGA LISTRIK

2.1 Fundamental Sistem Tenaga Listrik

Sistem tenaga listrik adalah komponen-komponen tenaga listrik yang terbagi menjadi tiga sub-sistem yaitu sub-sistem pembangkitan, sub-sistem transmisi dan sub-sistem distribusi. Sistem tenaga listrik secara umum digunakan untuk memenuhi kebutuhan energi listrik. Sistem pembangkitan terdiri dari kumpulan pembangkit tenaga listrik yang terhubung dengan sistem transmisi dan sistem distribusi. Sistem transmisi digunakan dalam penyaluran daya listrik dari sistem pembangkitan menuju sistem distribusi dengan menggunakan tegangan tinggi untuk mengurangi rugi-rugi saluran. Sedangkan untuk sistem distribusi terdiri dari gardu induk dan beban.

Sistem tenaga listrik yang terdiri dari beberapa pembangkit yang terhubung disebut juga dengan sistem tenaga listrik terinterkoneksi. Keuntungan dari penggunaan sistem tenaga listrik terinterkoneksi adalah dapat meningkatkan keandalan sistem, efisiensi pembangkit tenaga listrik, dapat menyalurkan daya listrik ke daerah yang jauh dari sistem pembangkitan, dan mempermudah penjadwalan pembangkit.



Gambar 2.1 Sistem tenaga listrik secara umum

Sebuah sistem tenaga listrik menjadi sebuah unit usaha yang sangat memperhatikan faktor teknis dan ekonomis dalam hal pengoperasian. Begitu pula dengan keandalan dan kontinuitas operasinya.

Kondisi seimbang antara pemasukan dan pengeluaran pada sistem tenaga listrik harus selalu dijaga agar dapat diperoleh keuntungan yang diharapkan, sehingga kualitas serta kapabilitas unit usaha dapat meningkat dan kelangsungan unit usaha dapat dijaga.

Sistem interkoneksi memberikan kemungkinan pengaturan daya listrik yang terbangkitkan pada setiap unit pembangkit dengan terhubungnya banyak pembangkit ke dalam sistem, dan biaya pembangkitan dapat diatur pada tingkat yang ekonomis [4].

2.1.1 Sub-sistem Pembangkitan

Secara umum sistem pembangkitan merupakan kumpulan dari pembangkit tenaga listrik yang terdiri dari beberapa komponen utama, seperti turbin dan generator. Pembangkit tenaga listrik digunakan untuk membangkitkan daya listrik yang kemudian didistribusikan kepada konsumen. Di dalam sebuah sistem pembangkit beberapa generator dioperasikan secara parallel dan dihubungkan dengan bus dalam sebuah sistem tenaga listrik guna menyediakan total daya yang diperlukan.

Pembangkit tenaga listrik dapat dibedakan menjadi beberapa jenis sesuai dengan bahan bakar yang digunakan, salah satu diantaranya adalah pembangkit listrik tenaga panas atau *thermal*. Pembangkit tipe ini merupakan pembangkit listrik yang mayoritas digunakan untuk memenuhi beban harian atau base *load*.

Setiap pembangkit memiliki karakteristik unit pembangkit masing – masing. Karakteristik unit pembangkit meliputi karakteristik *input – output* pembangkit, dan karakteristik *incremental rate*. Karakteristik tersebut diperoleh dari data – data seperti, desain generator dari pabrik pembuat generator, data histori pengoperasian generator maupun data percobaan. Karakteristik unit pembangkit digunakan dalam perhitungan biaya pembangkitan dari tiap unit pembangkit sehingga dapat dicapai nilai ekonomis atau optimum.

Karakteristik input – output dari pembangkit *thermal* merupakan hubungan antara input berupa bahan bakar yang digunakan dan output berupa daya yang dibangkitkan tiap pembangkit. Input bahan bakar dinyatakan dalam bentuk MBtu/h atau konsumsi energi sedangkan output daya dinyatakan dalam bentuk MW atau daya yang dibangkitkan.

Karakteristik *incremental rate* pembangkit termal merupakan hubungan antara perubahan daya pembangkitan yang dihasilkan dengan konsumsi bahan bakar yang dibutuhkan. *Incremental rate* atau $\Delta H/\Delta P$ biasanya dinyatakan dengan satuan Btu/kWh.

2.1.2 Sub-sistem Transmisi

Transmisi pada sistem tenaga listrik merupakan jaringan listrik yang berfungsi untuk menyalurkan daya listrik yang dibangkitkan melalui unit pembangkit tenaga listrik menuju sistem distribusi. Pada sistem transmisi tegangan yang digunakan adalah tegangan tinggi guna mengurangi rugi – rugi atau losses transmisi yang disebabkan oleh panas penghantar akibat adanya arus yang mengalir. Selain itu transmisi pada sistem tenaga listrik terinterkoneksi digunakan untuk meningkatkan keandalan dan efektivitas sistem. Saluran pada sistem transmisi dapat dibedakan menurut besar tegangannya yaitu sebagai berikut:

1. Saluran udara tegangan ekstra tinggi (SUTET): 200 kV – 500 kV.
2. Saluran udara tegangan tinggi (SUTT): 30 kV – 150 kV.
3. Saluran kabel tegangan tinggi (SKTT): 30 kV – 150 kV.
4. Saluran udara tegangan menengah (SUTM): 6 kV – 30 kV.
5. Saluran kabel tegangan menengah (SKTM): 6 kV – 30 kV.

2.1.3 Sub-sistem Distribusi

Sistem distribusi berfungsi untuk menyalurkan daya listrik dari sistem transmisi melalui gardu induk distribusi ke peralatan konsumen. Sistem distribusi dapat dibagi berdasarkan jenis pelanggannya yaitu sistem distribusi primer dan distribusi sekunder.

Sistem distribusi primer digunakan untuk melayani pelanggan dengan kapasitas daya besar seperti sektor industri. Tegangan yang digunakan pada sistem distribusi primer berkisar antara 4 kV sampai dengan 34.5 kV. Sedangkan untuk sistem distribusi sekunder digunakan untuk melayani pelanggan residential dan komersial. Tegangan yang digunakan pada sistem distribusi sekunder adalah tegangan satu fasa 240/120 V dengan tiga kawat, tegangan tiga fasa 208/120 V dengan empat kawat, tegangan tiga fasa 480/277 V dengan empat kawat.

2.1.4 Beban

Beban dalam sistem tenaga listrik dibagi ke dalam beberapa jenis yaitu beban industry, beban perumahan, dan beban komersil. Beban industri pada umumnya terdiri dari beban komposit berupa motor induksi yang mengkonsumsi daya reaktif dalam jumlah besar. Beban komposit bergantung pada tegangan dan frekuensi sehingga kualitas tegangan dan frekuensi dipertahankan pada batas yang ditentukan.

Beban perumahan dan beban komersil terdiri dari beban untuk penerangan, pemanasan, dan pendinginan. Beban tersebut tidak bergantung pada frekuensi dan hanya mengkonsumsi daya reaktif dengan nilai yang kecil.

Pada umumnya kurva beban harian dapat digolongkan menjadi beban dasar, beban rata – rata dan beban puncak. Golongan beban tersebut kemudian digunakan untuk mengatur pola daya yang dibangkitkan pada tiap pembangkit. Pengaturan daya yang dibangkitkan pada tiap pembangkit untuk menyuplai beban didasarkan pada biaya pembangkitan, fleksibilitas operasi, dan kapasitas pembangkitan.

2.2 Perencanaan Operasi Optimum Sistem Tenaga Listrik

Perencanaan operasi optimum sistem tenaga listrik secara umum adalah proses yang digunakan dalam merencanakan operasi dari sistem pembangkitan, transmisi, dan distribusi sistem tenaga listrik. Perencanaan operasi optimum sistem tenaga listrik dapat dibagi menjadi beberapa tahap, yaitu:

2.2.1 Penjadwalan Pembangkit Tenaga Listrik

Pada sistem tenaga listrik penjadwalan pembangkit tenaga listrik disebut juga dengan *unit commitment*, yaitu proses untuk menentukan jadwal pembangkit nyala dan mati dalam suatu sistem di setiap jam. Penjadwalan pembangkit dilakukan untuk mengurangi biaya pembangkit dan menjaga umur dari pembangkit. Parameter – parameter penjadwalan pembangkit adalah sebagai berikut:

1. Minimum Up Time Down Time:

Minimum up time down time merupakan parameter perhitungan penjadwalan pembangkit terhadap waktu pembangkit beroperasi atau tidak beroperasi. Minimum up time merupakan waktu minimal pembangkit beroperasi. Minimum down time merupakan waktu minimal pembangkit tidak beroperasi. Parameter tersebut harus dipenuhi dalam penjadwalan pembangkit sehingga pembangkit tidak dapat operasi atau tidak beroperasi pada setiap jam.

2. Start Up Cost:

Biaya tambahan pada saat penyalaan pembangkit baru yang terdiri menjadi dua tipe yaitu *cold start up* dan *hot/cold start up*. Cold start up adalah kondisi saat pembangkit dilepas dari sistem dan temperatur *boiler* dibiarkan turun dari temperatur kerjanya,

sehingga pada saat akan beroperasi kembali harus dilakukan pemanasan kembali. Sedangkan *hot/cold start up* adalah kondisi saat pembangkit dilepas dari sistem dan temperatur *boiler* dijaga pada temperatur kerjanya.

3. *Ramp Rate*:

Batas kerja aman suatu pembangkit berdasarkan thermal gradient dan pressure gradient untuk menjaga umur pembangkit tenaga listrik. Ramp rate juga berarti maksimum penambahan atau pengurangan daya yang dikeluarkan pembangkit di tiap jam.

3. *Spinning Reserve*:

Merupakan cadangan daya yang harus diperhitungkan dari unit-unit pembangkit yang beroperasi, dimana apabila ada salah satu unit yang mengalami kegagalan operasi (jatuh/trip) maka daya yang berkurang akibat kegagalan operasi dari unit tersebut dapat ditanggulangi oleh cadangan daya dari semua unit yang beroperasi. Biasanya sebesar kapasitas 1 pembangkit terbesar pada sistem.

2.2.1.1 **Kombinasi**

Terdapat dua macam kombinasi dalam perhitungan penjadwalan pembangkit tenaga listrik, yaitu *priority list* dan *complete enumeration*. Kombinasi dalam perhitungan penjadwalan pembangkit digunakan sebagai acuan untuk mencari kombinasi dari unit pembangkit yang beroperasi dan tidak beroperasi dalam suatu sistem yang memiliki biaya pembangkitan yang paling murah. Pengertian kombinasi – kombinasi tersebut adalah sebagai berikut:

1. *Priority list* atau urutan prioritas:

Priority list merupakan kombinasi dalam perhitungan penjadwalan pembangkit dengan jumlah kombinasi sama dengan jumlah pembangkit. *Priority list* dapat didasarkan pada biaya pembangkit pada saat berbeban penuh, sehingga pembangkit yang memiliki biaya paling mahal akan lebih diprioritaskan untuk tidak beroperasi.

2. *Complete enumeration*:

Complete enumeration merupakan kombinasi dalam perhitungan penjadwalan pembangkit dengan memperhitungkan seluruh kombinasi yang mungkin terjadi.

Complete enumeration dapat dinyatakan sebagai berikut:

$$2^n - 1$$

Dengan n merupakan jumlah pembangkit pada sistem. Complete enumeration akan menghasilkan nilai lebih akurat daripada priority list karena jumlah kombinasi yang diperhitungkan lebih banyak.

2.2.2 Pengaturan Daya Keluaran Pembangkit

Selain penjadwalan pembangkit, hal yang penting untuk dilakukan dalam perencanaan operasi optimum sistem tenaga listrik adalah pengaturan daya keluaran tiap pembangkit pada suatu sistem atau yang dikenal dengan economic dispatch. Pengaturan daya keluaran pembangkit yang optimal dilakukan untuk mengurangi biaya pembangkitan dan menjaga agar jumlah daya terbangkit sama dengan beban tersambung pada sistem sehingga frekuensi sistem stabil pada level 50 Hz.

Pengaturan daya keluaran pembangkit dapat dilakukan dengan beberapa cara tergantung dengan data pembangkit yang tersedia. Penyelesaian perhitungan untuk data pembangkit dengan orde 1 dapat diselesaikan dengan linear programming (LP) sedangkan untuk data pembangkit dengan orde 2 dapat diselesaikan dengan quadratic programming (QP).

2.2.2.1 Linear Programming

Linear programming (LP) merupakan program optimasi dari fungsi objektif berupa persamaan linear atau orde 1. LP akan melakukan perhitungan dengan mempertimbangkan batasan linear. Secara umum algoritma LP dapat dinyatakan dalam persamaan (2.2),

$$F(x) = f + g^T x + \frac{1}{2} x^T H x \quad (2.2)$$

Dengan batasan linear pada persamaan (2.3) dan (2.4).

$$lb \leq Ax \leq ub \quad (2.3)$$

$$x_{min} \leq x \leq x_{max} \quad (2.4)$$

Batasan linear dalam LP dapat dikelompokkan menjadi equality constraint dan inequality constraint, sesuai dengan kebutuhan perhitungan. Batasan equality constraint dapat dilihat pada persamaan (2.5) sedangkan untuk inequality constraint dapat dilihat pada persamaan (2.8).

- Equality constraint

$$\text{Set nilai untuk equality constraint: } lb = ub = b_{eq} \quad (2.5)$$

Maka:

$$b_{eq} \leq Ax \leq b_{eq} \quad (2.6)$$

$$Ax = b_{eq} \quad (2.7)$$

- Inequality Constraint

$$\text{Set nilai untuk inequality constraint: } lb < ub \quad (2.8)$$

Maka:

$$lb \leq Ax \leq ub \quad (2.9)$$

2.2.2.2 Quadratic Programming

Quadratic programming (QP) merupakan program optimasi dari fungsi objektif berupa persamaan kuadrat. QP akan melakukan perhitungan dengan mempertimbangkan batasan linear. Batasan linear digunakan untuk membatasi nilai variable yang dioptimasi dalam fungsi objektif. Algoritma QP dapat dinyatakan dalam persamaan (2.10).

$$F(x) = f + g^T x + \frac{1}{2} x^T H x \quad (2.10)$$

QP memiliki batasan linear yang sama dengan LP.

2.2.3 Aliran Daya Optimal Pada Sistem Tenaga

Studi aliran daya atau *power flow* (PF) merupakan salah satu bagian dari perencanaan sistem tenaga listrik yaitu perhitungan magnitude dan sudut fasa dari tegangan setiap bus serta aliran daya aktif dan reaktif pada setiap saluran di sistem tenaga listrik. Pada perhitungan aliran daya akan diperhitungkan rugi – rugi atau losses jaringan sehingga nilai daya yang dibangkitkan akan bernilai rugi – rugi daya ditambah dengan beban sistem tenaga listrik.

Sedangkan aliran daya optimal atau optimal power flow (OPF) merupakan perhitungan aliran daya yang mempertimbangkan biaya

pembangkitan sehingga didapatkan nilai pembangkitan yang murah dengan memperhitungkan rugi – rugi atau losses jaringan. PF dan OPF merupakan perhitungan aliran daya yang terbagi menjadi AC dan DC. AC pada PF dan OPF akan memperhitungkan daya aktif dan reaktif, sedangkan untuk DC hanya memperhitungkan daya aktif saja.

2.2.3.1 *Dynamic Optimal Power Flow Arus Searah*

Pada kondisi real, beban sistem berubah ubah secara dinamis selama 24 jam. Pada penelitian ini diasumsikan beban berubah dengan selang waktu 1 jam. Dengan beban yang dinamis, penyelesaian *OPF* disebut juga dengan *Dynamic Optimal Power Flow (DOPF)*.

Untuk menyelesaikan permasalahan *DOPF*, perlu diperhatikan pula mengenai *ramp rate* pembangkit dan batasan lainnya. *Ramp rate* merupakan batasan kenaikan atau penurunan nilai pembangkitan pada generator. Hal ini untuk menjaga gradien temperatur dan tekanan generator dalam batasan aman sehingga mencegah kerusakan dan menjaga masa hidup generator [5].

Selain batasan generator, batasan saluran, batasan cadangan berputar, dengan adanya *ramp rate* maka batasan pada sistem akan bertambah.

2.2.4 *Security Constrained Unit Commitment*

Security Constrained Unit Commitment (SCUC) adalah penjadwalan pembangkit tenaga listrik yang tidak hanya handal secara ekonomis tetapi juga memiliki sistem yang aman. Pada SCUC kondisi kontingensi juga dipertimbangkan, yaitu kondisi dimana adanya kemungkinan putusnya saluran transmisi sehingga kapasitas saluran juga berkurang dan impedansi sistem juga berubah. Kontingensi adalah suatu kejadian yang disebabkan oleh kegagalan atau pelepasan dari satu atau lebih generator dan/atau transmisi (Ditjen LPE, 2004). Kontingensi N-1 adalah kontingensi yang dihasilkan dari terlepasnya satu komponen sistem yaitu satu saluran transmisi atau satu generator. Kontingensi N-k adalah kontingensi yang dihasilkan dari terlepasnya sejumlah k komponen sistem. Dengan melakukan analisa SCUC, diharapkan sistem tenaga listrik dapat dioperasikan secara optimal dengan handal dan aman. Saat ini penggunaan SCUC sudah marak di berbagai negara, terutama di daerah dimana secara geografis menggunakan saluran transmisi dalam skala besar.

BAB 3

PENJADWALAN PEMBANGKIT MENGGUNAKAN BINARY PARTICLE SWARM OPTIMIZATION

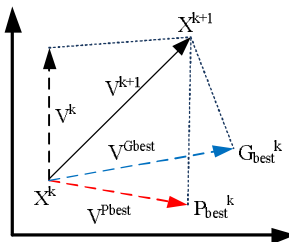
Pada bab ini akan dijelaskan bahwa penyelesaian penjadwalan nyala-mati unit pembangkit dalam operasi *security constrained unit commitment* menggunakan algoritma *Binary Particle Swarm Optimization* (BPSO) dan untuk menentukan biaya pembangkitan *direct current optimal power flow* menggunakan *quadratic programming*. Program *direct current optimal power flow* merupakan subprogram *unit commitment*.

3.1 Algoritma PSO (*Particle Swarm Optimization*)

Particle Swarm Optimization (PSO) adalah metode optimasi berbasis populasi yang pertama kali dikembangkan oleh Kennedy dan Eberhart pada tahun 1995, terinspirasi oleh perilaku sosial sekumpulan burung dan ikan. PSO memiliki alat optimasi yang menyediakan prosedur pencarian berbasis populasi yang mana setiap individu disebut partikel. Partikel tersebut mengganti posisinya setiap waktu. Kumpulan partikel yang merupakan calon solusi dinamakan *swarm*. Dalam sistem PSO, partikel melayang-layang di sekitar ruang pencarian multidimensi.

Selama proses penerbangan, setiap partikel menentukan posisinya terbaiknya sendiri berdasarkan pengalamannya sendiri (nilai ini yang disebut P_{best}) dan posisinya terbaiknya berdasarkan pengalaman dari partikel tetangganya (nilai ini disebut G_{best})

Proses pencarian P_{best} dan G_{best} dapat diilustrasikan dalam gambar 3.1 berikut.



Gambar 3.1 Konsep pencarian P_{best} dan G_{best} dari PSO

Pada algoritma PSO ini, pencarian solusi dilakukan oleh suatu populasi yang terdiri dari beberapa partikel. Populasi dibangkitkan secara random dengan batasan nilai terkecil dan terbesar.

Setiap partikel merepresentasikan posisi atau solusi dari permasalahan yang dihadapi. Setiap partikel melakukan pencarian solusi yang optimal dengan melintasi ruang pencarian (*search space*). Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari partikel tersebut (*local best*) dan penyesuaian terhadap posisi partikel terbaik dari seluruh kawanan (*global best*) selama melintasi ruang pencarian. Jadi, penyebaran pengalaman atau informasi terjadi di dalam partikel itu sendiri dan antara suatu partikel dengan partikel terbaik dari seluruh kawanan selama proses pencarian solusi. Setelah itu, dilakukan proses pencarian untuk mencari posisi terbaik setiap partikel dalam sejumlah iterasi tertentu sampai didapatkan posisi yang relatif steady atau mencapai batas iterasi yang telah ditetapkan. Pada setiap iterasi, setiap solusi yang direpresentasikan oleh posisi partikel, dievaluasi performansinya dengan cara memasukkan solusi tersebut kedalam fitness function.

Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel [Kennedy and Eberhart, 1995].

Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel. Berikut ini merupakan formulasi matematika yang menggambarkan posisi dan kecepatan partikel pada suatu dimensi ruang tertentu [11] :

$$x_1(0), x_2(0), \dots, x_N(0) \quad (3.1)$$

$$v_1(0), v_2(0), \dots, v_N(0) \quad (3.2)$$

Modifikasi ini dapat diwakili sebagai sebuah konsep kecepatan. Kecepatan dari setiap partikel dapat dirumuskan dari persamaan berikut.

$$v_i(t) = wv_i(t-1) + c_1r_1(P_{best_i} - x_i(t-1)) + c_2r_2(G_{best} - x_i(t-1)) \quad (3.3)$$

Menggunakan persamaan 3.3, kecepatan tertentu yang secara bertahap akan semakin mendekati Pbest dan Gbest bisa dihitung. Posisi saat ini (pencarian dalam ruang solusi) bisa didapatkan dari persamaan 3.4 seperti berikut.

$$x_i(t) = v_i(t) + x_i(t - 1) \quad (3.4)$$

Bobot inersia ini diusulkan oleh Shi and Eberhart [1998] untuk meredam kecepatan selama iterasi menuju titik target secara lebih akurat dan efisien. Untuk mencapai itu dan mempercepat konvergensi, suatu bobot inersia yang mengecil nilainya dengan bertambahnya iterasi digunakan dengan rumus dari persamaan 3.5 berikut.

$$w(t) = w_{max} - \left(\frac{w_{max} - w_{min}}{max.iter} \right) \times t \quad (3.5)$$

Dimana,

x	: posisi partikel
v	: kecepatan partikel
i	: indeks partikel
t	: iterasi ke- t ($t = 1, 2, \dots, n$)
P_{best}	: posisi terbaik lokal
G_{best}	: posisi terbaik global
w	: bobot inersia
c_1, c_2	: konstanta positif = 2
r_1, r_2	: bilangan random yang berkisar antara 0 sampai 1
w_{max}	: bobot inersia maksimal = 0.9
w_{min}	: bobot inersia minimal = 0.4
$max.iter$	: iterasi maksimal

Biasanya algoritma PSO hanya mampu menyelesaikan permasalahan optimasi dengan representasi solusi bilangan real. Oleh karena itu konsep PSO yang telah ada dikembangkan oleh penemunya agar mampu menyelesaikan permasalahan optimasi dengan representasi solusi biner, yang dikenal dengan algoritma *Binary Particle Swarm Optimization* (BPSO). Perbedaan paling signifikan pada *Binary Particle Swarm Optimization* (BPSO) selain pada representasi solusinya adalah pada proses *update* vektor posisi partikel yang tidak menggunakan persamaan 3.4 melainkan menggunakan persamaan 3.6 [8].

$$x_i(t) = \begin{cases} 1 & \text{jika } rand(1.0) < s(v_i(t)) \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

Dimana $S(v_i(t))$ adalah fungsi sigmoid

$$S(v_i(t)) = (1 + e^{-v_i(t)})^{-1} \quad (3.7)$$

Model ini akan disimulasikan dalam ruang dengan dimensi tertentu dengan sejumlah iterasi sehingga di setiap iterasi, posisi partikel akan semakin mengarah ke target yang dituju (minimasi atau maksimasi fungsi). Ini dilakukan hingga maksimum iterasi dicapai atau bisa juga digunakan kriteria penghentian yang lain.

Algoritma PSO meliputi langkah berikut

- Bangkitkan posisi awal sejumlah partikel sekaligus kecepatan awalnya secara random.
- Evaluasi *fitness* dari masing-masing partikel berdasarkan posisinya.
- Tentukan partikel dengan *fitness* terbaik, dan tetapkan sebagai *Gbest*. Untuk setiap partikel, *Pbest* awal akan sama dengan posisi awal.

Ulangi langkah berikut sampai maksimal iterasi dipenuhi

1. Menggunakan *Pbest* dan *Gbest* yang ada, perbarui kecepatan setiap partikel menggunakan persamaan 3.3. Lalu dengan kecepatan baru yang didapat, perbarui posisi setiap partikel menggunakan persamaan 3.4.
2. Evaluasi *fitness* dari setiap partikel.
3. Tentukan partikel dengan *fitness* terbaik, dan tetapkan sebagai *Gbest*. Untuk setiap partikel, tentukan *Pbest* dengan membandingkan posisi sekarang dengan *Pbest* dari iterasi sebelumnya.
4. Cek maksimal iterasi. Jika dipenuhi, berhenti. Jika tidak, kembali ke 1.

3.2 Quadratic Programming

Quadratic programming (QP) merupakan masalah optimasi dari fungsi objektif berupa persamaan kuadrat dengan *constraints* linear [6]. Batasan linear digunakan untuk membatasi nilai variabel yang dioptimasi.

Quadratic programming telah menjadi dasar dari beberapa algoritma pemrograman umum nonlinier lain dan sangat banyak diaplikasikan dalam berbagai bidang [6]. Secara umum algoritma *quadratic programming* dapat ditulis sebagai :

Meminimalkan fungsi :

$$F(x) = \frac{1}{2}x^T Hx + f^T x \quad (3.8)$$

Sesuai dengan *constraint* sistem linier

$$lb \leq x \leq ub \quad (3.9)$$

Dimana :

$$\begin{aligned} f &= \text{konstan scalar} & ub, lb &= \text{konstan } m - \text{vector} \\ H &= \text{matrik } n \times n & A_{eq} &= \text{matrik } m \times n \text{ dengan } m < n \\ x &= n - \text{vector yang dioptimasi} \end{aligned}$$

Constraints linear dalam *quadratic programming* biasanya dikelompokkan menjadi *equality constraint* dan *inequality constraints* dapat diuraikan sebagai berikut :

equality constraint

$$\text{set nilai} \quad lb = ub = b_{eq} \quad (3.10)$$

$$\text{maka :} \quad lb \leq A_{eq} * x \leq ub \quad (3.11)$$

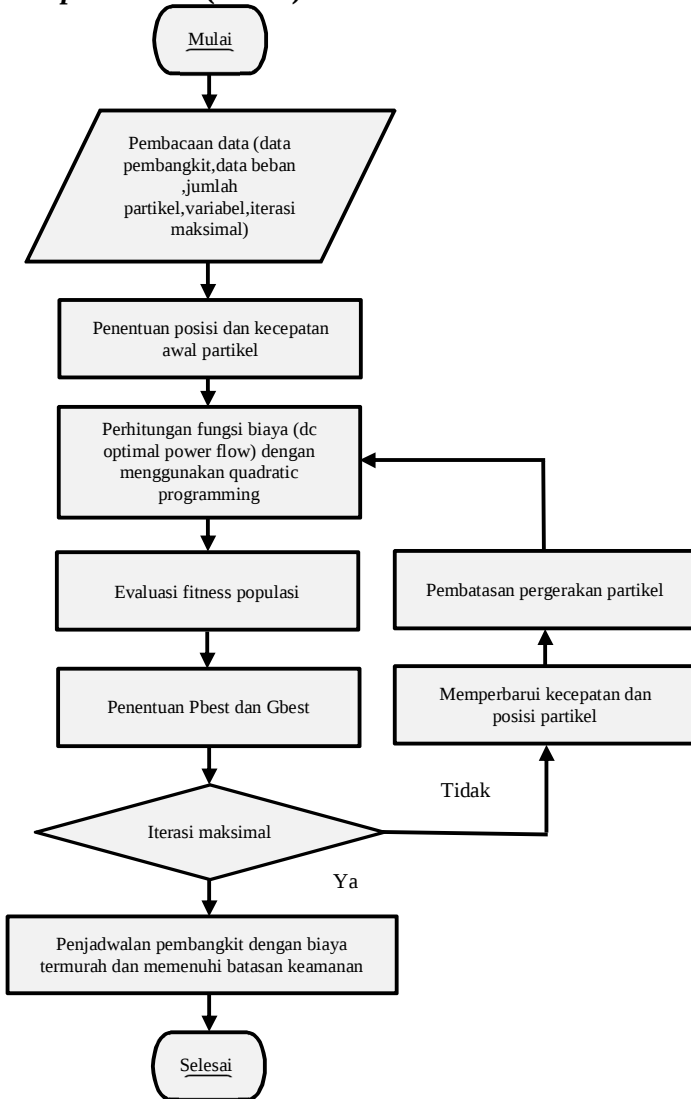
$$A_{eq} * x = b_{eq} \quad (3.12)$$

inequality constraints

$$\text{set nilai} \quad lb < ub \quad (3.13)$$

$$\text{maka :} \quad lb \leq A_{eq} * x \leq ub \quad (3.14)$$

3.3 Penerapan Algoritma *Binary Particle Swarm Optimization* (BPSO) Pada Unit Commitment



Gambar 3.2 Diagram alir penyelesaian SCUC menggunakan BPSO

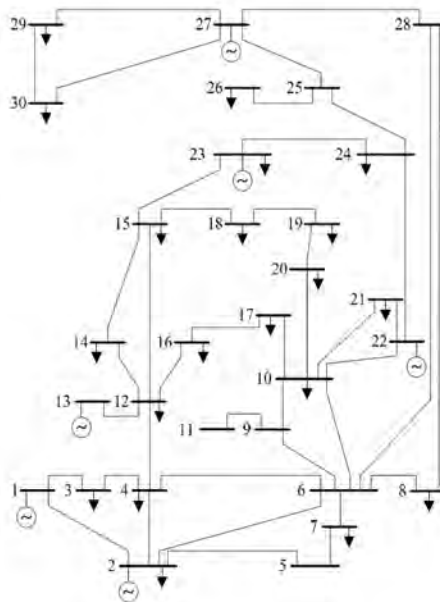
Algoritma *Binary Particle Swarm Optimization* (BPSO) merupakan metode yang akan digunakan pada program utama dalam menyelesaikan penjadwalan SCUC. BPSO akan menentukan kombinasi nyala-mati unit pembangkit dalam beberapa rentang waktu tertentu untuk memenuhi permintaan beban dan untuk menentukan pembiayaan pembangkitan *direct current optimal power flow* menggunakan *quadratic programming*. Langkah-langkah yang disajikan dalam tugas akhir ini terdapat pada gambar 3.2.

3.3.1 Inisialisasi Awal BPSO

Pada proses inisialisasi, memuat sistem yang akan dioptimisasi. Pada tugas akhir ini mengusulkan sistem IEEE 30 bus dengan 6 unit pembangkit.

3.3.1.1 Sistem IEEE 30 Bus

Pada Sistem IEEE 30 bus dengan 6 unit pembangkit termal. Berikut ini adalah data yang akan menjadi input program utama.



Gambar 3.1 Single Line Diagram Sistem IEEE 30 bus

Tabel 3.1 Data batasan unit pembangkit sistem IEEE 30 bus

Unit	Pmax (MW)	Pmin (MW)	Nyala min (jam)	Padam min (jam)
1	100	50	4	4
2	100	20	3	3
3	90	15	2	3
4	30	10	2	2
5	30	10	2	2
6	30	12	2	2

Tabel 3.2 Data biaya operasi sistem IEEE 30 bus

Unit	Koefisien biaya operasi			Biaya penyalaaan (\$)		Biaya padam (\$)
	A	b	c	Panas	Dingin	
1	0,00375	2,00	0	70	176	50
2	0,01750	1,75	0	74	187	60
3	0,06250	1,00	0	110	113	30
4	0,00834	3,25	0	50	267	85
5	0,02500	3,00	0	72	180	52
6	0,02500	3,00	0	40	113	30

Tabel 3.3 Data pembebanan sistem IEEE 30 bus 24 jam

Jam ke	Beban (MW)	Jam ke	Beban (MW)
1	163,00	13	210,00
2	152,50	14	208,00
3	150,00	15	205,00
4	158,80	16	198,00
5	170,00	17	220,00
6	167,00	18	230,00
7	178,00	19	232,00
8	190,00	20	226,00
9	195,00	21	214,00

10	194,00	22	200,00
11	180,00	23	190,00
12	183,20	24	180,00

3.3.1.2 Parameter-parameter BPSO

Parameter-parameter yang digunakan pada BPSO tugas akhir ini merupakan inisialisasi awal operasi algoritma. Parameter-parameter tersebut antara lain.

Jumlah swarm = Jumlah unit pembangkit

Jumlah variabel = 250

Iterasi maksimal = 60

Social constant = 2

Cognitive constant = 2

Inertia (w) = 0.4 – 0.9

Partikel adalah sebuah himpunan yang terdiri dari variabel. Variabel adalah parameter yang akan dioptimalkan dalam tugas akhir ini. Fungsi objektif dari tugas akhir ini adalah dengan mencari biaya pembangkitan termurah dengan memperhatikan batasan-batasan yang telah ditentukan dalam *security constrained unit commitment*. Iterasi maksimal adalah banyaknya proses pengulangan yang dilakukan dalam tugas akhir ini. *Social* dan *Cognitive constant* adalah konstanta yang telah ditetapkan dalam metode pengoptimalan menggunakan algoritma *particle swarm optimization*.

3.3.2 Inisialisasi Posisi dan Kecepatan Awal Partikel

Parameter kedua yang diinisialisasi adalah posisi dan kecepatan awal partikel. Bankitkan populasi awal x dengan rentang $x^{(B)}$ dan $x^{(A)}$ secara random sehingga didapat $x_1, x_2, \dots, x_N$ partikel i dan kecepatan pada iterasi t dinotasikan sebagai $x_i(t)$ dan $v_i(t)$, sehingga partikel awal dinotasikan [11].

$$x_1(0), x_2(0), \dots, x_N(0) \quad (3.15)$$

$$v_1(0), v_2(0), \dots, v_N(0) \quad (3.16)$$

3.3.3 Perhitungan Fungsi Biaya Dengan Quadratic Programming

Quadratic programming merupakan program yang digunakan untuk meminimalkan fungsi kuadratik dengan batasan tertentu. Batasan

ini biasanya disebut dengan *equality* dan *inequality constraint*. Salah satu algoritma *quadratic programming* yang sudah *include* dalam Matlab adalah quadprog [6].

Quadprog dapat digunakan untuk meminimalkan fungsi kuadratik dengan batasan tertentu, baik batasan nilai variabel yang digunakan dalam fungsi tersebut maupun batasan dari permasalahan yang kita definisikan dalam fungsi variabel.

3.3.3.1 Penerapan *Quadratic Programming* pada SCUC

Untuk dapat menerapkan *Quadratic Programming* untuk menyelesaikan perhitungan *security constrained unit commitment*, maka perlu dipelajari terlebih dahulu sintaksis program, argumen input-ouput program serta perumusan masalah berupa *objective function* dan *constraints* permasalahan. Sintaksis program adalah perintah yang digunakan untuk melakukan pemanggilan program dengan argumen input yang kita masukkan. Sintaksis program yang dapat digunakan dalam *Quadratic Programming* adalah sebagai berikut.

X, = quadprog (H,f,A,b,Aeq,beq,lb,ub,x0)

Menyelesaikan permasalahan yang dimulai dari vektor awal x0. x0 dapat digunakan untuk mempercepat proses pencarian nilai x yang diinginkan, namun nilai x0 bersifat juga dapat diabaikan.

3.3.3.2 Argumen Input dari *Quadratic Programming*

Argumen input adalah variabel yang akan kita libatkan sebagai data input dalam program. Argumen input yang ada pada *Quadratic Programming* adalah :

- H : Matrik simetris yang menyatakan bentuk kuadratik dari persamaan $\frac{1}{2}x'Hx + f'x$.
- F : Vector yang menyatakan bentuk linier dari persamaan $\frac{1}{2}x'Hx + f'x$.
- Aeq : Matrik yang menyatakan koefisien linier dari equality dan inequality constraints $lb \leq Aeq*x \leq ub$.
- Beq : Matrik yang menyatakan koefisien linier dari equality $Aeq*x = Beq$.

- Lb :Vektor yang menyatakan batas bawah dari equality dan inequality constraints $lb \leq Aeq \cdot x \leq ub$.
- Ub :Vektor yang menyatakan batas bawah dari equality dan inequality constraints $lb \leq Aeq \cdot x \leq ub$.

3.3.3.3 Inisialisasi Fungsi Objektif dan Constraints dari Security Constrained Unit Commitment

Dalam kasus *unit commitment* pada tugas akhir ini, yang menjadi fungsi objektif adalah total biaya pembangkitan dari semua unit pembangkit yang ada selama rentang waktu tertentu. Sedangkan batasan terdiri dari batasan generator. Secara detail dapat dijelaskan sebagai berikut.

- a. Fungsi objektif : Fungsi biaya unit pembangkit
Fungsi biaya unit pembangkit biasanya dinyatakan dalam bentuk persamaan kuadrat 3.17 berikut.

$$F_{nh}(P_{nh}) = a_n(P_{nh})^2 + b_n(P_{nh}) + C_n \quad (3.17)$$

- b. Variabel permasalahan : variabel yang akan dioptimasi dalam *unit commitment* ini adalah pembangkitan daya aktif (P). Jika n adalah jumlah generator, Sehingga ukuran matrik x adalah $[n : 1]$ maka variabel yang akan dioptimasi dapat ditulis dengan persamaan 3.18 berikut.

$$x = \begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} \quad (3.18)$$

- c. *Constraints* : batasan generator
Batasan maksimum-minimum pembangkitan generator. Semua batasan diatas dapat dikelompokkan sebagai *equality constraint* dan *inequality constraint*.
equality constraint : active power balance

$$\sum_{n=1}^N P_{nh} = D_h$$

$$(3.19)$$

inequality constraint

$$U_{nh}P_n(max) \geq P_{nh} \geq U_{nh}P_n(min) \quad (3.20)$$

3.3.3.4 Pembentukan Matrik H Simetris yang Menyatakan Bentuk Kuadrat dari Fungsi Objektif

Pengambilan data *cost function* generator pada koefisien a dari semua generator pembangkit yang ada. Matrik dimanipulasi menjadi matrik $[1 : n]$ dengan n adalah jumlah generator maka dapat ditulis dengan persamaan 3.21 berikut.

$$[a_1 \quad \dots \quad a_n] \quad (3.21)$$

$$H = 2 * diag[a_1 \quad \dots \quad a_n] \quad (3.22)$$

$$H = 2 * \begin{bmatrix} a_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & a_n \end{bmatrix} \quad (3.23)$$

Nilai $[a_1 \quad \dots \quad a_n]$ = koefisien kuadrat a pada *cost function* generator.

3.3.3.5 Pembentukan Vektor Linier f yang Menyatakan Bentuk Linear dari Fungsi Objektif

Pembacaan data koefisien b pada *cost function* dari semua generator pembangkit yang ada. Koefisien b inilah yang akan dijadikan sebagai vektor linier f yang menyatakan bentuk linear dari fungsi objektif. Matrik dimanipulasi menjadi matrik $[n : 1]$ dengan n adalah jumlah generator maka dapat ditulis dengan persamaan 3.24 berikut.

$$f = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad (3.24)$$

3.3.3.6 Pembentukan Matrik A_{eq} yang Menyatakan Koefisien Linear dari Constraints $lb \leq A_{eq} * x \leq ub$

Dalam *security constrained unit commitment* matrik A_{eq} menyatakan jumlah unit pembangkit yang beroperasi. Ukuran matrik A_{eq} adalah $[1 : n]$ dengan n adalah jumlah generator maka dapat ditulis dengan persamaan 3.25 berikut.

$$A_{eq} = [A_{eq1} \dots A_{eqn}] \quad (3.25)$$

3.3.3.7 Pembentukan Matrik B_{eq} yang Menyatakan Koefisien Linear dari Constraints $A_{eq} * x = B_{eq}$

Dalam *unit commitment* matrik B_{eq} menyatakan permintaan beban dengan persamaan 3.26 berikut.

$$B_{eq} = [D_1 \dots D_h] \quad (3.26)$$

3.3.3.8 Pembentukan Vektor lb dan ub yang Menyatakan Batas Atas dan Batas Bawah dari Constraints $lb \leq A_{eq} * x \leq ub$

Vektor ub dan vektor lb menyatakan batasan pembangkitan daya aktif (P) setiap unit generator dengan persamaan 3.27 berikut.

$$lb = \begin{bmatrix} P_{\min 1} \\ \vdots \\ P_{\min n} \end{bmatrix} \quad ub = \begin{bmatrix} P_{\max 1} \\ \vdots \\ P_{\max n} \end{bmatrix} \quad (3.27)$$

3.3.4 Evaluasi Fitness Populasi

Evaluasi nilai *fitness* populasi adalah proses menentukan nilai *fitness* terbaik dengan memperhatikan batasan-batasan yang telah ditentukan. Batasan tersebut adalah permintaan beban puncak, harga pembangkitan minimal. Partikel yang melanggar batasan yang telah

ditentukan akan diberi pinalti sehingga partikel tersebut tidak termasuk dalam pemilihan calon solusi.

Cara yang dilakukan untuk pinalti partikel yang melanggar batas adalah dengan merubah nilai *fitness* partikel tersebut menjadi nilai yang sangat besar sehingga partikel tersebut sehingga tidak termasuk dalam pemilihan calon solusi.

3.3.5 Penentuan Posisi Terbaik Lokal dan Global

Parameter pemilihan *fitness* pada metode PSO ini didasarkan pada dua hal yaitu posisi terbaik lokal dan posisi terbaik global. Posisi terbaik lokal adalah posisi terbaik yang dimiliki oleh masing-masing partikel sedangkan posisi terbaik global adalah posisi terbaik yang dimiliki oleh kumpulan partikel tersebut.

Fungsi objektif dari tugas akhir ini adalah mencari biaya minimum pembangkitan dengan memperhatikan batasan-batasan. Oleh karena itu, posisi terbaik lokal tiap partikel adalah berdasar pada total biaya pembangkitan. Jika partikel tersebut tidak melanggar batasan maka harga nilai *fitness* partikel tersebut akan menjadi posisi terbaik partikel itu sendiri.

Posisi terbaik global didapat dengan cara mencari nilai paling rendah dari kumpulan partikel yang memiliki posisi terbaik lokal sendiri. Setiap partikel akan mengevaluasi dirinya sendiri dan partikel lain dalam pencarian posisi terbaik global.

3.3.6 Pembaruan Kecepatan dan Posisi Partikel

Proses selanjutnya dari metode ini adalah proses pembaruan kecepatan masing-masing partikel. Proses pembaruan kecepatan ini bertujuan agar metode ini dapat mencapai nilai konvergensi dalam waktu yang cepat. Setiap partikel memiliki kecenderungan tersendiri dalam menentukan pembaruan kecepatannya. Pembaruan kecepatan setiap partikel mengacu terhadap dua hal yaitu posisi terbaik lokal partikel itu sendiri atau posisi terbaik global dari keseluruhan partikel. Partikel tersebut dengan konstanta pergerakan yang diatur acak akan cenderung mendekati posisi terbaik global atau cenderung menjauhi posisi terbaik global karena posisi terbaik global pada iterasi tertentu belum tentu akan menjadi nilai dengan fungsi objektif terbaik. Partikel yang mengalami perubahan kecepatan akan mengalami perubahan posisi baru. Pembaruan kecepatan partikel dirumuskan pada persamaan 3.28 sebagai berikut.

$$v_i(t) = wv_i(t-1) + c_1r_1(P_{best_i} - x_i(t-1)) + c_2r_2(G_{best} - x_i(t-1)) \quad (3.28)$$

Dimana x adalah posisi partikel, v adalah kecepatan partikel, i adalah indeks partikel, t adalah iterasi ke- t , P_{best} adalah posisi terbaik lokal, G_{best} adalah posisi terbaik global, w adalah bobot inersia.

c_1 dan c_2 masing masing adalah learning rate untuk kemampuan individu (cognitive) dan pengaruh sosial (kawanan). r_1 dan r_2 bilangan random dalam interval 0 dan 1. c_1 dan c_2 menunjukkan bobot dari memory (position) sebuah partikel terhadap memory (posisi) dari kelompok. Nilai c_1 dan c_2 biasanya 2 sehingga perkalian c_1r_1 dan c_2r_2 memastikan bahwa partikel akan mendekati target setengah selisihnya.

Untuk menentukan posisi partikel dalam bentuk bilangan biner dengan menggunakan fungsi sigmoid dirumuskan pada persamaan 3.29 seperti berikut.

$$S(v_i(t)) = (1 + e^{-v_i(t)})^{-1} \quad (3.29)$$

$$x_i(t) = \begin{cases} 1 & \text{jika } rand(1.0) < S(v_i(t)) \\ 0 & \text{otherwise} \end{cases} \quad (3.30)$$

3.3.7 Pembatasan Pergerakan Partikel

Pada setiap transisi antara akhir iterasi dan kembali ke tahap awal iterasi partikel akan bergerak acak sehingga ada kemungkinan partikel tersebut melaju ke luar dari daerah pencarian nilai objektif terbaik. Cara yang dilakukan untuk menghindari hal tersebut adalah dengan membatasi pergerakan partikel.

3.3.8 Hasil Penjadwalan

Pada bagian akhir ini, akan ditampilkan hasil penjadwalan setiap jam yang berupa kondisi nyala-mati unit, biaya operasi, serta total biaya pembangkitan.

Halaman ini sengaja dikosongkan

BAB 4

SIMULASI DAN ANALISIS DATA

Pada bab ini akan dibahas tentang simulasi yang dihasilkan oleh program *Binary Particle Swarm Optimization* (BPSO) untuk menyelesaikan operasi *security constrained unit commitment* (SCUC) dan hasil perhitungan total biaya pembangkitan merupakan biaya paling ekonomis.

4.1 Performa Komputasi

Program simulasi tugas akhir ini menggunakan komputer dengan spesifikasi yang tercantum pada tabel 4.1.

Tabel 4.1 Data spesifikasi komputasi

Parameter	Spesifikasi
Processor unit	Intel Core i3 @ 3,07 GHz
RAM	2,00 GB
GPU	Mobile Intel 286 MB
Sistem operasi	Windows 7 32-bit
Software simulasi	Matlab

4.2 Hasil Simulasi

Pada bab ini akan dibahas mengenai hasil simulasi dari program PSO. Hasil dari simulasi berupa penyelesaian operasi *security constrained unit commitment* berupa penjadwalan pembangkitan selama 24 jam, pembangkitan tiap generator, dan biaya operasi unit pembangkit.

4.2.1 Sistem IEEE 30 Bus

Pada sistem IEEE 30 bus dengan 6 unit pembangkit dilakukan simulasi dengan kondisi awal semua pembangkit padam seperti pada tabel 4.2 sebagai berikut .

Tabel 4.2 Kondisi awal unit pembangkit sistem IEEE 30 bus

Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
-4	-3	-3	-3	-2	-2

4.2.1.1 Penjadwalan Pembangkit Sistem IEEE 30 Bus

Pada tabel 4.3 merupakan hasil penjadwalan unit pembangkit pada sistem IEEE 30 bus dengan 6 unit pembangkit dengan kondisi awal semua unit pembangkit padam.

Tabel 4.3 Penjadwalan pembangkit sistem IEEE 30 bus

Jam Ke	Unit					
	1	2	3	4	5	6
1	1	1	1	0	0	0
2	1	1	1	0	0	0
3	1	1	1	0	0	0
4	1	1	1	0	0	0
5	1	1	1	0	0	0
6	1	1	1	0	0	0
7	1	1	1	0	0	0
8	1	1	1	0	0	0
9	1	1	1	0	0	1
10	1	1	1	0	0	1
11	1	1	1	0	0	1
12	1	1	1	0	0	1
13	1	1	1	0	0	1
14	1	1	1	0	0	1
15	1	1	1	0	0	1
16	1	1	1	0	0	1
17	1	1	1	1	0	1
18	1	1	1	1	0	1
19	1	1	1	1	0	1
20	1	1	1	1	0	1
21	1	1	1	1	0	1
22	1	1	1	1	0	1
23	1	1	1	1	0	1
24	1	1	1	1	0	1

4.2.1.2Pembangkitan Daya Aktif Sistem IEEE 30 Bus

Pada tabel 4.4, tabel 4.5, dan tabel 4.6 menunjukkan daya aktif yang dibangkitkan oleh setiap unit pembangkit selama 24 jam.

Tabel 4.4 Pembangkitan daya aktif sistem IEEE 30 bus (1)

Unit Pembangkit	Daya (MW) Jam ke					
	1	2	3	4	5	6
1	60,03	55,77	54,76	58,33	62,88	61,66
2	75,75	70,88	69,72	73,81	79,00	77,61
3	27,21	25,85	25,52	26,67	28,12	27,73
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
Daya Total	163,0	152,5	150,0	158,8	170,0	167,0

Tabel 4.5 Pembangkitan daya aktif sistem IEEE 30 bus (2)

Unit Pembangkit	Daya (MW) Jam ke					
	7	8	9	10	11	12
1	68,18	77,58	61,25	60,95	56,65	57,64
2	80,00	80,00	77,15	76,80	71,89	73,01
3	29,82	32,42	27,60	27,50	26,13	26,44
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	29,00	28,76	25,32	26,11
Daya Total	178,0	190,0	195,0	194,0	180,0	183,2

Tabel 4.6 Pembangkitan daya aktif sistem IEEE 30 bus (3)

Unit Pembangkit	Daya (MW) Jam ke					
	13	14	15	16	17	18
1	66,98	66,04	64,62	62,17	63,68	66,09
2	80,00	80,00	80,00	78,20	79,87	80,00
3	29,43	29,13	28,68	27,89	24,53	26,58
4	0	0	0	0	18,45	19,29
5	0	0	0	0	0	0

6	33,58	32,83	31,70	29,74	33,47	36,04
Daya Total	210,0	208,0	205,0	198,0	220,0	230,0

Tabel 4.7 Pembangkitan daya aktif sistem IEEE 30 bus (3)

Unit Pembangkit	Daya (MW) Jam ke					
	19	20	21	22	23	24
1	67,69	64,58	60,40	55,93	53,39	50,56
2	80,00	80,00	76,12	71,01	68,11	64,89
3	26,87	27,19	27,64	27,86	26,81	25,58
4	19,99	19,34	18,36	17,44	25,36	16,37
5	0	0	0	0	0	0
6	37,45	34,89	31,48	27,76	16,32	22,60
Daya Total	232,0	226,0	214,0	200,0	190,0	180,0

4.2.1.3 Pembangkitan Daya Aktif Maksimal Sistem IEEE 30 Bus

Pada tabel 4.8, tabel 4.9, tabel 4.10, dan tabel 4.11 menunjukan daya aktif masimal yang mampu dibangkitkan oleh setiap unit pembangkit selama 24 jam. Selisih antara daya maksimal dengan daya beban digunakan sebagai cadangan berputar yang akan digunakan ketika ada permintaan daya beban secara tiba-tiba. Dengan adanya cadangan berputar keandalan sistem tenaga listrik menjadi lebih baik.

Tabel 4.8 Pembangkitan daya aktif maksimal sistem IEEE 30 bus (1)

Unit Pembangkit	Daya Maksimal (MW) Jam ke					
	1	2	3	4	5	6
1	100	100	100	100	100	100
2	100	100	100	100	100	100
3	90	90	90	90	90	90
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
Daya Total	290	290	290	290	290	290

Tabel 4.9 Pembangkitan daya aktif maksimal sistem IEEE 30 bus (2)

Unit Pembangkit	Daya Maksimal (MW) Jam ke					
	7	8	9	10	11	12
1	100	100	100	100	100	100
2	100	100	100	100	100	100
3	90	90	90	90	90	90
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	30	30	30	30
Daya Total	290	290	320	320	320	320

Tabel 4.10 Pembangkitan daya aktif maksimal sistem IEEE 30 bus (3)

Unit Pembangkit	Daya Maksimal (MW) Jam ke					
	13	14	15	16	17	18
1	100	100	100	100	100	100
2	100	100	100	100	100	100
3	90	90	90	90	90	90
4	0	0	0	0	30	30
5	0	0	0	0	0	0
6	30	30	30	30	30	30
Daya Total	320	320	320	320	350	350

Tabel 4.11 Pembangkitan daya aktif maksimal sistem IEEE 30 bus (4)

Unit Pembangkit	Daya Maksimal (MW) Jam ke					
	19	20	21	22	23	24
1	100	100	100	100	100	100
2	100	100	100	100	100	100
3	90	90	90	90	90	90
4	30	30	30	30	30	30
5	0	0	0	0	0	0
6	30	30	30	30	30	30
Daya Total	350	350	350	350	350	350

4.2.1.4 Total Biaya Pembangkitan Sistem IEEE 30 Bus

Pada tabel 4.12 menunjukan biaya pembangkit pada sistem IEEE 30 bus dengan 6 unit pembangkit selama 24 jam. Pada biaya pembangkitan terdapat biaya penyalan karena semua unit dalam kondisi awal padam.

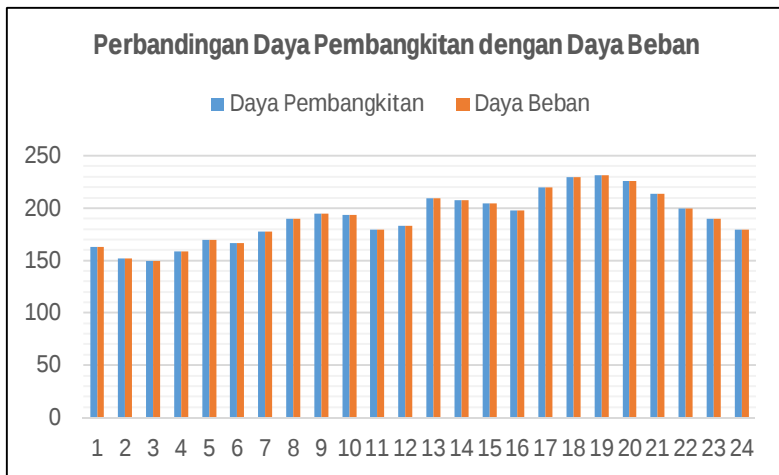
Tabel 4.12 Total biaya pembangkitan sistem IEEE 30 bus

Jam Ke	Biaya (\$)			
	Operasi	Penyalan	Pemadaman	Pembangkitan
1	497,3717	476	0	973,8477
2	456,5055	0	0	456,5055
3	440,5660	0	0	440,5660
4	480,8508	0	0	480,8508
5	529,4021	0	0	529,4021
6	516,6582	0	0	516,6582
7	566,9031	0	0	566,9031
8	625,4243	0	0	625,4243
9	620,2311	113	0	733,4111
10	615,1863	0	0	615,1863
11	554,2667	0	0	554,2667
12	568,0491	0	0	568,0491
13	678,5845	0	0	678,5845
14	677,7897	0	0	677,7897
15	665,4916	0	0	665,4916
16	637,2948	0	0	637,2948
17	711,7481	267	0	978,8501
18	741,5799	0	0	741,5799
19	750,2232	0	0	750,2232
20	726,5976	0	0	726,5976
21	684,9454	0	0	684,9454
22	623,1116	0	0	623,1116
23	579,8995	0	0	579,8995
24	539,0516	0	0	539,0516
Total Biaya Pembangkitan (\$)				14488,4904

4.3 Analisa Hasil Simulasi

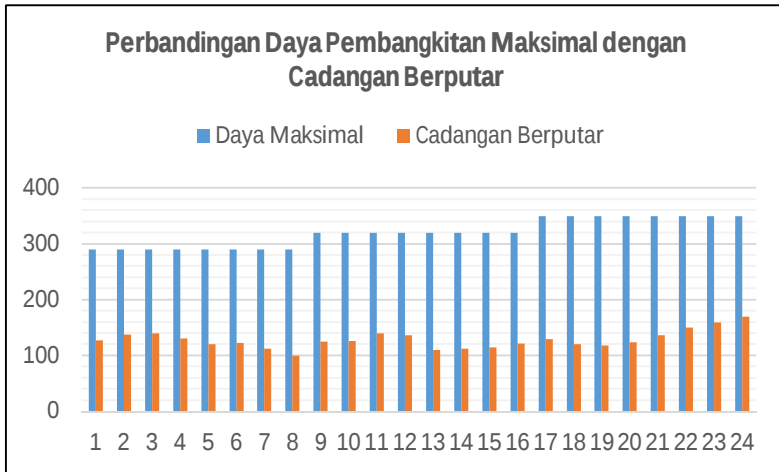
Pada sistem IEEE 30 bus dengan 6 unit pembangkit, kondisi awal semua unit pembangkit dalam keadaan padam. Unit pembangkit pertama telah padam selama 4 jam, unit pembangkit kedua telah padam selama 3 jam, unit pembangkit ketiga telah padam selama 3 jam, unit pembangkit ke empat telah padam selama 3 jam, unit pembangkit kelima telah padam selama 2 jam, dan unit pembangkit keenam telah padam selama 2 jam

Simulasi dijalankan dengan enam puluh iterasi pada sistem IEEE 30 bus dengan 6 unit pembangkit menghasilkan biaya yang optimal. Hasil penjadwalan menunjukkan bahwa terjadi kombinasi nyala padam pembangkit sesuai permintaan beban dengan mempertimbangkan biaya penyalan dan biaya operasi paling ekonomis. Dilihat dari hasil penjadwalan pada tabel 4.3 tidak ada unit pembangkit yang melanggar batasan nyala minimal dan padam minimal sesuai batasan pada tabel 3.1.



Gambar 4.1 Perbandingan daya pembangkitan dengan daya beban sistem IEEE 30 bus

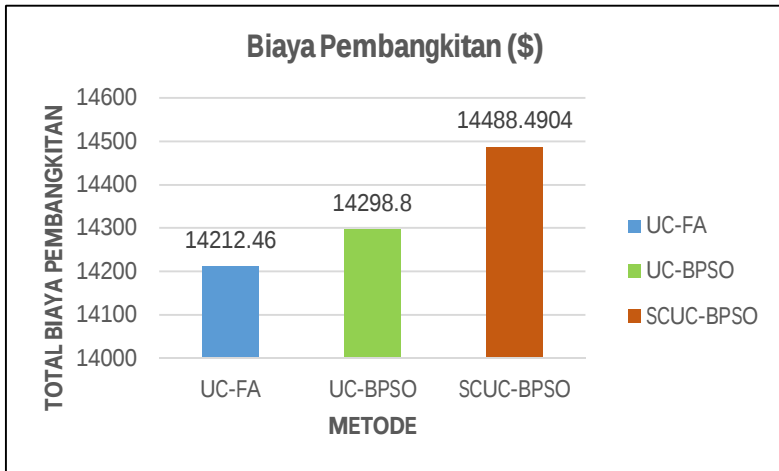
Dari gambar 4.1, terlihat pembangkitan daya aktif selama 24 jam dengan perbedaan permintaan daya beban tiap jam, kombinasi unit pembangkit mampu memenuhi permintaan daya beban sesuai dengan tabel 3.6 tanpa melanggar batasan P_{min} dan P_{max} pada tabel 3.1.



Gambar 4.2 Perbandingan daya pembangkitan maksimal dengan cadangan berputar sistem IEEE 30 bus

Dari gambar 4.2 di atas, terlihat pembangkitan daya aktif maksimal selama 24 jam dengan perbedaan permintaan cadangan berputar tiap jam, kombinasi unit pembangkit mampu memenuhi permintaan cadangan berputar sebesar kapasitas 1 pembangkit terbesar dari kapasitas pembangkit yang terpasang pada sistem.

Dilihat dari tabel 4.12 total biaya pembangkitan selama 24 jam, kombinasi unit pembangkit yang beragam telah memenuhi biaya operasi paling ekonomis. sesuai dengan batasan nyala minimal, padam minimal, daya pembangkitan minimal, daya pembangkitan maksimal, permintaan beban, dan permintaan cadangan berputar.



Gambar 4.3 Perbandingan total biaya pembangkitan

Dari gambar 4.3 di atas, terlihat bahwa terdapat selisih biaya pembangkitan sebesar 1,3% atau sebesar \$ 189,69 antara metode SCUC-BPSO dengan metode UC-BPSO dan sebesar 1,9% atau sebesar \$ 276.03 antara metode SCUC-BPSO dengan metode UC-FA yang disimulasikan pada sistem IEEE 30 bus dengan 6 unit pembangkit. Metode SCUC-BPSO menghasilkan total biaya pembangkitan yang lebih mahal yaitu sebesar \$ 14488,49 hal ini disebabkan metode SCUC-BPSO menggunakan 2 jenis penyalan, yaitu penyalan dingin dan penyalan panas. Pada metode BPSO juga mempertimbangkan cadangan berputar dan kontingensi. Metode FA menghasilkan total biaya yang lebih murah yaitu sebesar \$ 14212,46 hal ini disebabkan metode FA hanya menggunakan 1 jenis penyalan, yaitu penyalan panas. Pada metode FA juga tidak mempertimbangkan cadangan berputar dan kontingensi.

Security Constrained Unit Commitment mempertimbangkan cadangan berputar dan kontingensi menghasilkan total biaya pembangkitan yang lebih mahal karena *security constrained unit commitment* tidak hanya mencari kombinasi unit pembangkit yang memenuhi permintaan beban saja tetapi juga harus mempertimbangkan permintaan cadangan berputar serta kondisi kontingensi.

Kondisi awal unit pembangkit mempengaruhi total biaya pembangkitan. Kondisi awal pembangkit nyala menghasilkan total biaya

yang lebih murah dibandingkan dengan kondisi awal pembangkit padam, karena kondisi awal pembangkit nyala tidak memerlukan biaya penyalaan dan cenderung mempertahankan operasi untuk menghindari biaya penyalaan dan biaya pemadaman. Kondisi awal pembangkit padam memerlukan biaya penyalaan sehingga mengakibatkan total biaya pembangkitan yang lebih mahal.

Lampiran

```
% SECURITY CONSTRAINED UNIT COMMITMENT
% Rony Seto Wibowo & Anton Friar Yohanes
Reynaldo=====
=====

% UC_PSO_27
multiplierPd24 = [0.86 0.81 0.79 0.84 0.898
0.883 0.941 1.004 1.031 1.025 0.951 0.968 1.1099
1.099 1.084 1.575 1.163 1.216 1.226 1.946 1.132
1.058 1.004 0.951];
totalBiaya24(1:24) = 10000000000;
rekap124(1:75,1:6) = 0;
rekap224(1:75,1:6) = 0;
rekapIndex = 1;
for hour=1:24
    multiplierPd = multiplierPd24(hour);
    % ----- Setting parameters of PSO --
    -----
        dim = 6;                                % Number
of control variables
        N    = 10;                                %
Number of particles
        maxit = 7;                                %
Maximum iterations
        rhomax = 0.9;
        rhomin = 0.4;

    Load = [189.2 189.2 189.2];

    rekap1 = [];
    rekap2 = [];

    checkIS = CIS;

    for k=1:length(Load)
```

```

% ----- Initializing swarm and
velocities -----
x = rand(N,dim); %
Initial positions of particles
r3= rand(N,dim);
v = x; %
Initial velocities of particles

A = k;

for i=1:dim
    for j=1:N
        s(j,i)= 1/(1+exp(-v(j,i)));

        if r3(j,i)<s(j,i)
            x(j,i)=1;
        else
            x(j,i)=0;
        end
    end
end

f = zeros(N,1); %
Initial fitness functions
for i=1:N
    onofGenerator = x(i,:);
    if onofGenerator(1)~= 0 ||
onofGenerator(2)~=0 || onofGenerator(3)~=0 ||
onofGenerator(4)~=0 || onofGenerator(5)~=0 ||
onofGenerator(6)~=0
        f(i) = fungsibiaya27(onofGenerator,
multiplierPd, x(i,:), PS, Load(k), checkIS);
% Initial fitness functions
    else
        f(i) = 10000000000;
    end
end

B = k;

```

```

%----- Initial iteration -----
-----

it=1;
Pbest=x;
fbest=f;
[minf,idk]=min(f);
Gbest=x(idk,:);
lastbest=[0 0];
minftot=[];

%% Main iteratio
while it<maxit

    r1=rand;r2=rand;r3=rand(N,dim);
    c1=1;c2=1;
    rho=rhmax-((rhmax-rhomin)/maxit)*it;

    for i=1:dim
        for j=1:N

            v(j,i)=rho*v(j,i)+r1*c1*(Pbest(j,i)-
x(j,i))+r2*c2*(Gbest(i)-x(j,i));
            s(j,i)= 1/(1+exp(-v(j,i)));

            if r3(j,i)<s(j,i)
                x(j,i)=1;
            else
                x(j,i)=0;
            end
        end
    end

    C = k;

    checkIS;
    x

```

```

%% Check MD dan MU
for j=1:N
    y = x(j,:);
    unitstart = find(y' > PS);
    unitshutdown = find(y' < PS);

    F = k;

    for i = 1:length(unitstart)
        if checkIS(unitstart(i)) > -
MD(unitstart(i));
            y(unitstart(i)) = 0;
        end
    end

    G = k;

    for i=1:length(unitshutdown)
        if checkIS(unitshutdown(i)) <
MU(unitshutdown(i));
            y(unitshutdown(i)) = 1;
        end
    end

    H = k;

    x(j,:) = y;
end

D = k;

x

for i=1:N
    onofGenerator = x(i,:);
    if onofGenerator(1)~= 0 ||
onofGenerator(2)~=0 || onofGenerator(3)~=0 ||

```

```

onofGenerator(4)~=0 || onofGenerator(5)~=0 ||
onofGenerator(6)~=0
        f(i)=fungsibiaya27(onofGenerator,
multiplierPd, x(i,:), PS, Load(k), checkIS);
    end
end

%-----update Pbest
ganti = f < fbest;
fbest=fbest.*(1-ganti)+f.*ganti;
Pbest(find(ganti),:)=x(find(ganti),:);
[minf,idk]=min(fbest);
minftot=[minftot;minf]
Gbest=Pbest(idk,:);

it=it+1;
lastbest=Gbest
E = k;

end

%% unit start and shutdown again

lastbest'
PS

unitstart = find(lastbest' > PS)
unitshutdown = find(lastbest' < PS)

for k=1:dim
    k
    if ismember(k, unitstart) |
ismember(k,unitshutdown)
        if ismember(k, unitstart)
            checkIS(k)= 1;
        end
        if ismember(k, unitshutdown)
            checkIS(k)= -1;

```

```

        end
    else
        if checkIS(k) > 0
            checkIS(k)=checkIS(k)+1;
        end
        if checkIS(k) < 0
            checkIS(k)=checkIS(k)-1;
        end
    end
end

minftot
lastbest
plot(minftot)
rekap1 = [rekap1; lastbest];
rekap2 = [rekap2; minftot(it-1)];

fullcheckIS = [fullcheckIS checkIS]

checkIS
PS = lastbest'

end

rekap1
rekap2
Totalbiaya = sum(rekap2)

rekap124(rekapIndex,:) = rekap1(1,:);
rekap224(rekapIndex,:) = rekap2(1,:);
rekapIndex = rekapIndex + 1;

rekap124(rekapIndex,:) = rekap1(2,:);
rekap224(rekapIndex,:) = rekap2(2,:);
rekapIndex = rekapIndex + 1;

rekap124(rekapIndex,:) = rekap1(3,:);
rekap224(rekapIndex,:) = rekap2(3,:);

```

```

        rekapIndex = rekapIndex + 1;

        totalBiaya24(hour) = Totalbiaya;
end

rekapIndexPrint = 1;
for hour = 1 : 24
    Teks =
    sprintf('=====
==\nJam ke-%d :\n', hour);
    disp(Teks);

    %Rekap1-----
    ---
    Teks = sprintf('Rekap1 :\n');
    disp(Teks);
    rekapIndexPrintTemp = rekapIndexPrint;
    for i = 1 : 3
        disp(rekap124(rekapIndexPrintTemp,:));
        rekapIndexPrintTemp =
rekapIndexPrintTemp+1;
    end
    %-----
    ---

    %Rekap2-----
    ---
    Teks = sprintf('Rekap2 :\n');
    disp(Teks);
    rekapIndexPrintTemp = rekapIndexPrint;
    for i = 1 : 3
        disp(rekap224(rekapIndexPrintTemp,1));
        rekapIndexPrintTemp =
rekapIndexPrintTemp+1;
    end
    %-----
    ---
    rekapIndexPrint = rekapIndexPrint + 3;

```

```

%Tbiaya-----
---
Teks = sprintf('Total Biaya : ');
disp(Teks);
disp(totalBiaya24(hour));
%-----
---
Teks =
sprintf('=====
==\n');
disp(Teks);

% fungsibiaya27
function f = fungsibiaya27(onofGenerator,
multiplierPd, x,PS, Load, checkIS)

%% Data pembangkit
% %      No. a      b      c      Pmin
Pmaks   Harga BB   Fmaksrata
% Data =[1  1.7    1245    274025    225
370     120        0        % PLTU
Batubara H(Mkal/h), HBB (Rp/Mkal)
%      2  1      2372    15865    43
80     150        0        % PLTU Minyak
(MFO) H(Mkal/h), HBB (Rp/Mkal) or 800 Rp/Mkal
%      3  8      1829    63484    40
99     132.5      0];      % PLTG
H(Mkal/h), HBB (Rp/Mkal)

%      No. a      b      c      Pmin
Pmaks   MU  MD  HSC      CSC      CSH      IS
Data =[ 1  0.00048    16.19    1000    150
455     8   8   4500    9000     5     8;
      2  0.00031    17.26     970    150
455     8   8   5000   10000     5     8;
      3  0.002     16.6     700    20
130     5   5   550    1100     4    -5; %5
      4  0.00211    16.5     680    20
130     5   5   560    1120     4    -5; %5

```

	5	0.00398	19.7	450	25
162	6	6	900	1800	4
	6	0.00712	22.26	370	20
80	3	3	170	340	2

-6; %6
-3;]

```

define_constants;

upbnd = Data(:,6)';
lwbnd = Data(:,5)';
hsc = Data(:,9);
csc = Data(:,10);
csh = Data(:,11);
startupcost = 0;

%[Totalcost,P] = economicdispatch27(x, Load);
rundcopfResult = rundcopf('case30',
onofGenerator, multiplierPd);
if rundcopfResult.raw.info>=0
    Totalcost = rundcopfResult.f;
else
    Totalcost = 10000000000;
end
%P = rundcopfResult.gen(:,PG);

unitstart = find(x' > PS);
unitshutdown = find(x' < PS);

for i=1:length(unitstart)

    if checkIS(unitstart(i)) <= -
csh(unitstart(i))
        startupcost = startupcost +
csc(unitstart(i));
    else
        startupcost = startupcost +
hsc(unitstart(i));
    end
end

```

```

shutdowncost = sum(Data(unitshutdown, 9))*0;
if sum(x.*upbnd)<= Load*1.1
    Totalcost = 1000000000;
end

f = Totalcost + startupcost + shutdowncost;

% dcof_solver
function [results, success, raw] =
dcof_solver(om, om01, om02, mpopt)
%DCOF_SOLVER Solves a DC optimal power flow.
%
% [RESULTS, SUCCESS, RAW] = DCOF_SOLVER(OM,
MPOPT)
%
% Inputs are an OPF model object and a
MATPOWER options struct.
%
% Outputs are a RESULTS struct, SUCCESS flag
and RAW output struct.
%
% RESULTS is a MATPOWER case struct (mpc) with
the usual baseMVA, bus
% branch, gen, gencost fields, along with the
following additional
% fields:
% .order      see 'help ext2int' for
details of this field
% .x          final value of optimization
variables (internal order)
% .f          final objective function
value
% .mu         shadow prices on ...
% .var
% .l          lower bounds on variables
% .u          upper bounds on variables
% .lin

```

```

%          .l  lower bounds on linear
constraints
%          .u  upper bounds on linear
constraints
%
%  SUCCESS    1 if solver converged
successfully, 0 otherwise
%
%  RAW        raw output in form returned by
MINOS
%          .xr  final value of optimization
variables
%          .pimul  constraint multipliers
%          .info  solver specific termination code
%          .output solver specific output
information
%
%  See also OPF, QPS_MATPOWER.

%  MATPOWER
%  $Id: dcopf_solver.m 2338 2014-06-27
18:34:09Z ray $
%  by Ray Zimmerman, PSERC Cornell
%  and Carlos E. Murillo-Sanchez, PSERC Cornell
& Universidad Autonoma de Manizales
%  Copyright (c) 2000-2010 by Power System
Engineering Research Center (PSERC)
%
%  This file is part of MATPOWER.
%  See http://www.pserc.cornell.edu/matpower/
for more info.
%
%  MATPOWER is free software: you can
redistribute it and/or modify
%  it under the terms of the GNU General Public
License as published
%  by the Free Software Foundation, either
version 3 of the License,
%  or (at your option) any later version.

```

```

%
% MATPOWER is distributed in the hope that it
% will be useful,
% but WITHOUT ANY WARRANTY; without even the
% implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR
% PURPOSE. See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU
% General Public License
% along with MATPOWER. If not, see
% <http://www.gnu.org/licenses/>.
%
% Additional permission under GNU GPL version
% 3 section 7
%
% If you modify MATPOWER, or any covered work,
% to interface with
% other modules (such as MATLAB code and MEX-
% files) available in a
% MATLAB(R) or comparable environment
% containing parts covered
% under other licensing terms, the licensors
% of MATPOWER grant
% you additional permission to convey the
% resulting work.

%%----- initialization -----
%% define named indices into data matrices
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS,
BS, BUS_AREA, VM, ...
VA, BASE_KV, ZONE, VMAX, VMIN, LAM_P, LAM_Q,
MU_VMAX, MU_VMIN] = idx_bus;
[GEN_BUS, PG, ~, QMAX, QMIN, VG, MBASE,
GEN_STATUS, PMAX, PMIN, ...
MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1,
PC2, QC1MIN, QC1MAX, ...

```

```

        QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30,
        RAMP_Q, APF] = idx_gen;
    [F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B,
    RATE_C, ...
    TAP, SHIFT, BR_STATUS, PF, QF, PT, QT,
    MU_SF, MU_ST, ...
    ANGMIN, ANGMAX, MU_ANGMIN, MU_ANGMAX] =
    idx_brch;
    [PW_LINEAR, POLYNOMIAL, MODEL, STARTUP,
    SHUTDOWN, NCOST, COST] = idx_cost;

```

```

%% options

```

```

alg = upper(mpopf.dc.solver);

```

```

%% unpack data

```

```

mpc = get_mpc(om);
[baseMVA, bus, gen, branch, gencost] = ...
    deal(mpc.baseMVA, mpc.bus, mpc.gen,
    mpc.branch, mpc.gencost);
cp = get_cost_params(om);
[N, H, Cw] = deal(cp.N, cp.H, cp.Cw);
fparm = [cp.dd cp.rh cp.kk cp.mm];
Bf = userdata(om, 'Bf');
Pfinj = userdata(om, 'Pfinj');
[vv, ll] = get_idx(om);

```

```

%% problem dimensions

```

```

ipol = find(gencost(:, MODEL) == POLYNOMIAL); %%
polynomial costs
ipwl = find(gencost(:, MODEL) == PW_LINEAR); %%
piece-wise linear costs
itop = find(gen(:, APF) == 1) %%
generator with take or pay
nb = size(bus, 1); %% number of buses
nl = size(branch, 1); %% number of
branches
ng = size(gen, 1); %% number of
generator

```

```

nw = size(N, 1);           %% number of general
cost vars, w
ny = getN(om, 'var', 'y'); %% number of piece-
wise linear costs
nxyz = getN(om, 'var');    %% total number of
control vars of all types

%% linear constraints & variable bounds
[A, l, u] = linear_constraints(om);
[A1, l1, u1] = linear_constraints(om01);
[A2, l2, u2] = linear_constraints(om02);
[x0, xmin, xmax] = getv(om);

%% set up objective function of the form: f =
1/2 * X'*HH*X + CC'*X
%% where X = [x;y;z]. First set up as quadratic
function of w,
%% f = 1/2 * w'*HHw*w + CCw'*w, where w =
diag(M) * (N*X - Rhat). We
%% will be building on the (optionally present)
user supplied parameters.

%% piece-wise linear costs
any_pwl = (ny > 0);
if any_pwl
    Npwl = sparse(ones(ny,1), vv.i1.y:vv.iN.y,
1, 1, nxyz);    %% sum of y vars
    Hpwl = 0;
    Cpwl = 1;
    fparam_pwl = [1 0 0 1];
else
    Npwl = sparse(0, nxyz);
    Hpwl = [];
    Cpwl = [];
    fparam_pwl = [];
end

%% quadratic costs
npol = length(ipol);

```

```

if any(find(gencost(ipol, NCOST) > 3))
    error('DC opf cannot handle polynomial costs
with higher than quadratic order.');
```

end

```

iqdr = find(gencost(ipol, NCOST) == 3);
ilin = find(gencost(ipol, NCOST) == 2);
polycf = zeros(npol, 3);
%% quadratic coeffs for Pg
if ~isempty(iqdr)
    polycf(iqdr, :) = gencost(ipol(iqdr),
COST:COST+2);
end
polycf(ilin, 2:3) = gencost(ipol(ilin),
COST:COST+1);
polycf = polycf * diag([ baseMVA^2 baseMVA 1]);
%% convert to p.u.
Npol = sparse(1:npol, vv.i1.Pg-1+ipol, 1, npol,
nxyz);           %% Pg vars
Hpol = sparse(1:npol, 1:npol, 2*polycf(:, 1),
npol, npol);
Cpol = polycf(:, 2);
fparm_pol = ones(npol,1) * [ 1 0 0 1 ];

%% combine with user costs
NN = [ Npwl; Npol; N ];
HHw = [ Hpwl, sparse(any_pwl, npol+nw);
        sparse(npol, any_pwl), Hpol,
        sparse(npol, nw);
        sparse(nw, any_pwl+npol), H    ];
CCw = [ Cpwl; Cpol; Cw];
ffparm = [ fparm_pwl; fparm_pol; fparm ];

%% transform quadratic coefficients for w into
coefficients for X
nnw = any_pwl+npol+nw;
M = sparse(1:nnw, 1:nnw, ffparm(:, 4), nnw,
nnw);
MR = M * ffparm(:, 2);
HMR = HHw * MR;
```

```

MN = M * NN;
HH = MN' * HHw * MN;
CC = full(MN' * (CCw - HMR));
C0 = 1/2 * MR' * HMR + sum(polycf(:, 3));    %%
constant term of cost

%% default solver
alg = 'OT';

%% set up input for QP solver
opt = struct('alg', alg, 'verbose',
mpopt.verbose);

HHH = HH
CCC = CC
AAA = A
lll = l
uuu = u
xminnn = xmin
xmaxxx = xmax
x000 = x0

% %% Time interval
% T = 3;

%% To expand matrixes due to considered
contingencies
HH = blkdiag(HH, HH*0, HH*0);
CC = [CC; CC*0; CC*0];
A = blkdiag(A, A1, A2);
l = [l; l1; l2];
u = [u; u1; u2];
xmin = [xmin; xmin; xmin];
xmax = [xmax; xmax; xmax];
x0 = [x0; x0; x0];

T = 2    %% number considered contingencies

```

```

%% To expand matrixes due to ramp rate
constraint
add01 = zeros(ng*(T), length(A(1,:)));
ramprate = gen(:, RAMP_10)/baseMVA*0;

for j=1:T
for i=1:ng
    add01((j-1)*ng + i, nb + i) = 1;
    add01((j-1)*ng + i, (j*(nb+ng))+ nb + i) = -
1;
end
    l = [1; -ramprate];
    u = [u; ramprate];
end

%% To add ramp rate constraint into main matrix A
A = [A; add01];

%%----- run opf -----
[x, f, info, output, lambda] = qps_matpower(HH,
CC, A, l, u, xmin, xmax, x0, opt)
success = (info == 1);

xdummy = x
info

%%----- calculate return values -----
if ~any(isnan(x))
    %% update solution data
    Va = x(vv.i1.Va:vv.iN.Va);
    Pg = x(vv.i1.Pg:vv.iN.Pg);
    f = f + C0;

    %% update voltages & generator outputs
    bus(:, VM) = ones(nb, 1);
    bus(:, VA) = Va * 180/pi;
    gen(:, PG) = Pg * baseMVA;

```

```

    %% compute branch flows
    branch(:, [QF, QT]) = zeros(nl, 2);
    branch(:, PF) = (Bf * Va + Pfinj) * baseMVA;
    branch(:, PT) = -branch(:, PF);
end

%% package up results
mu_l = lambda.mu_l;
mu_u = lambda.mu_u;
muLB = lambda.lower;
muUB = lambda.upper;

%% update Lagrange multipliers
il = find(branch(:, RATE_A) ~= 0 & branch(:,
RATE_A) < 1e10);
bus(:, [LAM_P, LAM_Q, MU_VMIN, MU_VMAX]) =
zeros(nb, 4);
gen(:, [MU_PMIN, MU_PMAX, MU_QMIN, MU_QMAX]) =
zeros(size(gen, 1), 4);
branch(:, [MU_SF, MU_ST]) = zeros(nl, 2);
bus(:, LAM_P) =
(mu_u(ll.i1.Pmis:ll.iN.Pmis) -
mu_l(ll.i1.Pmis:ll.iN.Pmis)) / baseMVA;
branch(il, MU_SF) = mu_u(ll.i1.Pf:ll.iN.Pf) /
baseMVA;
branch(il, MU_ST) = mu_u(ll.i1.Pt:ll.iN.Pt) /
baseMVA;
gen(:, MU_PMIN) = muLB(vv.i1.Pg:vv.iN.Pg) /
baseMVA;
gen(:, MU_PMAX) = muUB(vv.i1.Pg:vv.iN.Pg) /
baseMVA;
pimul = [
    mu_l - mu_u;
    -ones(ny>0, 1);    %% dummy entry corresponding
to linear cost row in A (in MINOS)
    muLB - muUB
];

```

```

mu = struct( ...
    'var', struct('l', muLB, 'u', muUB), ...
    'lin', struct('l', mu_l, 'u', mu_u) );

results = mpc;
[results.bus, results.branch, results.gen, ...
    results.om, results.x, results.mu,
results.f] = ...
    deal(bus, branch, gen, om, x, mu, f);

raw = struct('xr', x, 'pimul', pimul, 'info',
info, 'output', output);

```

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil simulasi diperoleh beberapa kesimpulan sebagai berikut :

1. *Security Constrained Unit Commitment* mempertimbangkan cadangan berputar dan kontingensi menghasilkan total biaya pembangkitan yang lebih mahal karena *security constrained unit commitment* tidak hanya mencari kombinasi unit pembangkit yang memenuhi permintaan beban saja tetapi juga harus mempertimbangkan permintaan cadangan berputar serta kondisi kontingensi.
2. Selisih biaya pembangkitan sebesar 1,3% atau sebesar \$ 189,69 antara metode SCUC-BPSO dengan metode UC-BPSO dan sebesar 1,9% atau sebesar \$ 276.03 antara metode SCUC-BPSO dengan metode UC-FA yang disimulasikan pada sistem IEEE 30 bus dengan 6 unit pembangkit. Metode SCUC-BPSO menghasilkan total biaya pembangkitan yang lebih mahal yaitu sebesar \$ 14488,49 hal ini disebabkan metode SCUC-BPSO menggunakan 2 jenis penyalan, yaitu penyalan dingin dan penyalan panas. Pada metode BPSO juga mempertimbangkan cadangan berputar dan kontingensi. Metode FA menghasilkan total biaya yang lebih murah yaitu sebesar \$ 14212,46 hal ini disebabkan metode FA hanya menggunakan 1 jenis penyalan, yaitu penyalan panas. Pada metode FA juga tidak mempertimbangkan cadangan berputar dan kontingensi.

5.2 Saran

Saran untuk penelitian ke depan adalah :

1. Penyelesaian *security constrained unit commitment* dapat dilakukan dengan algoritma cerdas yang lain, seperti DE, GA, ICA, dan lain-lain.
2. Penjadwalan *security constrained unit commitment* dapat dilakukan dengan mengadopsi kondisi yang nyata, tanpa ada pengabaian batasan-batasan operasi unit pembangkit.

DAFTAR PUSTAKA

- [1] Allen J. Wood, Bruce F. Wollenberg, “*Power, Generation, Operation, and Control*”, John Willey & Sons Inc, America, 1996.
- [2] R.Reddy Gaddam, “*Optimal Unit Commitment Using Swarm Intelligence for Secure Operation of Solar Energy Integrated Smart Grid*”.Thesis of intenasional Institute Of Information Technologi,India,2013.
- [3] Saadat, Hadi, “*Power System Analysis 2nd Edition*”, McGrowHill, Ch.1, 1999.
- [4] Aris Heri Andriawan, “Analisa Aplikasi Penjadwalan Unit-Unit Pembangkit Pada Sistem Kelistrikan Jawa-Bali dengan Menggunakan *Unit Commitment, Unit Decommitment*, dan *Modified Unit Decommitment*”, Thesis Jurusan Teknik Elektro FTI-ITS,Surabaya,2010.
- [5] Nursidi, “*Dynamic Optimal Power Flow Arus Search Menggunakan Quadratic Programming* ”, Skripsi Jurusan Teknik Elektro FTI-ITS,Surabaya,2013.
- [6] Benny Prastikha Hadhi, “*Optimisasi Unit Commitment Mempertimbangkan Fungsi Biaya Tidak Mulus Dengan Firefly Algorithm*”,Skripsi Jurusan Teknik Elektro FTI-ITS,Surabaya,2014.
- [7] Xin-she Yang, “*Nature Inspired Optimization Algorithm*”, School of Science and Technology Middlesex University London, London,2014.
- [8] Han Yu, Kit Po Wong, “*An Advanced Quantun-Inspired Evolutionary Algorithm for Unit Commitment*”, IEEE Trans. on Power Systems, vol. 26, no. 2, May. 2011.
- [9] M. V. C. Rao, and C. K. Loo, “*A Novel Approach for Unit Commitment Problem via an Effective Hybrid Particle Swarm Optimization*”, IEEE trans on power systems, vol. 21, no. 1, February 2006.
- [10] Budi Santosa dan Paul Willy, *Metoda Metaheuristik, Konsep dan Implementasi*, Graha Ilmu, Surabaya, 2011.
- [11] Penangsang, O., “*Analisis Aliran Daya*”, ITS Press Surabaya, 2012

RIWAYAT HIDUP PENULIS



Anton Friar Yohanes Reynaldo, putra pertama dari pasangan Frins Mulia Ramsey dan Mariance Elfrida ini lahir di Jakarta, 5 Agustus 1994. Penulis tamat dari bangku sekolah dasar di SD PSKD Kwitang 8 Depok pada tahun 2006. Pada tahun 2006 melanjutkan sekolah menengah pertama di SMPN 2 Depok. Pada tahun 2009 melanjutkan sekolah menengah atas di SMAN 1 Jakarta. Pada tahun 2012 melanjutkan studi S1 di Institut Teknologi Sepuluh Nopember (ITS) Surabaya jurusan Teknik Elektro. Konsentrasi yang diambil adalah bidang studi Teknik Sistem Tenaga. Penulis aktif dalam mengikuti kegiatan kemahasiswaan seperti lomba, konferensi, kepanitiaan, dan organisasi. Selain itu penulis juga terdaftar menjadi asisten di Laboratorium Simulasi Sistem Tenaga Listrik B103 ITS dan pernah menjadi volunteer di ITS International Office. Penulis dapat dihubungi melalui email: anton.frian@gmail.com.