



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - TE 141599

***SECURITY CONSTRAINED UNIT COMMITMENT DENGAN  
MEMPERTIMBANGKAN RAMP RATE DAN RUGI-RUGI DAYA  
SALURAN TRANSMISI MENGGUNAKAN ALGORITMA BINARY  
PARTICLE SWARM OPTIMIZATION PADA SISTEM IEEE 30 BUS***

Elpha Aulia Arifin  
NRP 07111540000116

Dosen Pembimbing

Dr. Ir. Ni Ketut Aryani, MT.  
Dr. Dimas Fajar Uman P, ST. MT.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember





**TUGAS AKHIR - TE 141599**

***SECURITY CONSTRAINED UNIT COMMITMENT DENGAN  
MEMPERTIMBANGKAN RAMP RATE DAN RUGI-RUGI DAYA  
SALURAN TRANSMISI MENGGUNAKAN ALGORITMA *BINARY  
PARTICLE SWARM OPTIMIZATION* PADA SISTEM IEEE 30  
BUS***

Elpha Aulia Arifin  
NRP 07111540000116

Dosen Pembimbing

Dr. Ir. Ni Ketut Aryani, MT.  
Dr. Dimas Fajar Uman P, ST. MT.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019





**FINAL PROJECT - TE 141599**

**SECURITY CONSTRAINED UNIT COMMITMENT  
CONSIDERING RAMP RATE AND TRANSMISSION LINE  
LOSSES USING ALGORITMA BINARY PARTICLE SWARM  
OPTIMIZATION BASED ON IEEE 30 BUS**

Elpha Aulia Arifin  
NRP 07111540000116

Supervisors

Dr. Ir. Ni Ketut Aryani, MT.  
Dr. Dimas Fajar Uman P, ST. MT.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul ***“SECURITY CONSTRAINED UNIT COMMITMENT DENGAN MEMPERTIMBANGKAN RAMP RATE DAN RUGI-RUGI DAYA SALURAN TRANSMISI MENGGUNAKAN ALGORITMA BINARY PARTICLE SWARM OPTIMIZATION PADA SISTEM IEEE 30 BUS”*** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 28 Juni 2019

A handwritten signature in black ink, appearing to read 'Elpha', with a stylized flourish underneath.

Elpha Aulia Arifin

NRP. 07111540000116

*Halaman ini sengaja dikosongkan*



**SECURITY CONSTRAINED UNIT COMMITMENT  
DENGAN MEMPERTIMBANGKAN RAMP RATE  
DAN RUGI-RUGI DAYA SALURAN TRANSMISI  
MENGUNAKAN ALGORITMA BINARY PARTICLE  
SWARM OPTIMIZATION PADA SISTEM IEEE 30  
BUS**

**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,

Dosen Pembimbing II,

**Dr. Ir. Ni Ketut Aryani, MT.**  
NIP. 196509011991032002

**Dr. Dimas Fajar Uman P. S.T., M.T.**  
NIP. 198811082012121001

**SURABAYA**  
Juni, 2019

*Halaman ini sengaja dikosongkan*

# ***SECURITY CONSTRAINED UNIT COMMITMENT DENGAN MEMPERTIMBANGKAN RAMP RATE DAN RUGI-RUGI DAYA SALURAN TRANSMISI MENGUNAKAN ALGORITMA BINARY PARTICLE SWARM OPTIMIZATION PADA SISTEM IEEE 30 BUS***

Nama : Elpha Aulia Arifin  
NRP : 07111540000116  
Dosen Pembimbing I : Dr. Ir. Ni Ketut Aryani, MT.  
NIP : 196509011991032002  
Dosen Pembimbing II : Dr. Dimas Fajar Uman P, ST. MT.  
NIP : 198811082012121001

## **ABSTRAK**

Jumlah pembangkitan daya setiap waktu nya berubah-ubah sesuai dengan kebutuhan beban. Untuk mendapatkan jumlah pembangkitan daya yang optimal dibutuhkan studi *unit commitment* dengan tujuan untuk mendapatkan kombinasi pembangkitan yang memenuhi pembebanan dengan biaya yang paling murah. Pada perhitungan *Unit Commitment*, terdapat batasan-batasan yang harus dipertimbangkan yaitu *minimum up and down time*, *spinning reserve*, *ramp rate*, *power balance*, *start up cost*, dan *power balance*. Untuk mendapatkan kombinasi pembangkit yang memenuhi batasan yang dibutuhkan, ada beberapa cara yang bisa dilakukan, salah satunya adalah dengan perhitungan algoritma *artificial intelligence*.

Pada tugas akhir ini, akan dilakukan studi *unit commitment* pada sistem IEEE 30 bus dengan mempertimbangkan batasan *ramp rate* dan rugi-rugi daya saluran transmisi menggunakan Algoritma *Binary Particle Swarm Optimization* (BPSO) untuk mendapatkan kombinasi pembangkitan yang paling optimal dan memiliki biaya pembangkitan yang paling rendah.

**Kata Kunci :** *Unit Commitment*, *Ramp Rate*, Rugi-rugi daya saluran, Penjadwalan Pembangkitan, *Binary Particle Swarm Optimization*

*Halaman ini sengaja dikosongkan*

# **SECURITY CONSTRAINED UNIT COMMITMENT CONSIDERING RAMP RATE AND TRANSMISSION LINE LOSSES USING BINARY PARTICLE SWARM OPTIMIZATION ALGORITHM BASED ON IEEE 30 BUS SYSTEM**

Name : Elpha Aulia Arifin  
NRP : 07111540000116  
Supervisor I : Dr. Ir. Ni Ketut Aryani, MT.  
NIP : 196509011991032002  
Supervisor II : Dr. Dimas Fajar Uman P, ST. MT.  
NIP : 198811082012121001

## **ABSTRACT**

The amount of power generation every time changes according to load demand. In order to get the optimal amount of power generation, a unit commitment study is needed with the aim of obtaining a combination of generation that meets the demand at the cheapest cost. In the calculation of the Unit Commitment, there are constraints that must be considered, Those are the minimum up and down time, spinning reverse, ramp rate, power balance, start-up cost, and power balance. To get a combination of generators that meet the required limits, there are several artificial intelligence algorithms approach, one of the algorithms is Binary Particle Swarm Optimization (BPSO).

In this paper, a unit commitment study will be implemented on IEEE 30 bus system by considering ramp rate and transmission line power losses using the Binary Particle Swarm Optimization (BPSO) algorithm to obtain the most optimal combination of generation with the lowest generation costs .

**Keywords** : Unit Commitment, Ramp Rate, Transmission line losses, Generation Scheduling, Binary Particle Swarm Optimization

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji Syukur penulis panjatkan kehadiran Allah SWT atas segala Rahmat, Karunia, dan Petunjuk yang telah dilimpahkan-Nya sehingga penulis mampu menyelesaikan tugas akhir dengan judul ***“Security Constrained Unit Commitment Dengan Mempertimbangkan Ramp Rate dan Rugi-Rugi Daya Saluran Transmisi Menggunakan Algoritma Binary Particle Swarm Optimization Pada Sistem IEEE 30 Bus”***.

Tugas Akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan S1 pada Bidang Studi Teknik Sistem Tenaga, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Atas selesainya penyusunan tugas akhir ini, penulis ingin mengucapkan terima kasih kepada :

1. Allah SWT atas limpahan rahmat Nya, petunjuk Nya, dan Karunia Nya.
2. Ibu, Ayah, Mbak Nadia, Mbak Della, dan Matthias yang selalu mendoakan, mendukung dan memberikan semangat.
3. Kakek dan Nenek yang tidak pernah berhenti berdoa untuk cucu-cucu nya.
4. Mbak Saini yang selalu sabar menghadapi dan membangunkan penulis tiap pagi.
5. Dr. Ir. Ni Ketut Aryani, MT dan Dr. Dimas Fajar Uman Putra, ST. MT. selaku dosen pembimbing yang telah memberikan bantuan dan bimbingan selama pengerjaan tugas akhir ini.
6. Bapak Prof. Ir. H. Ontoseno Penangsang, M.Sc., Ph.D. , Bapak Prof. Dr. Ir. Adi Soeprijanto, dan Bapak Dr. Rony Seto Wibowo, S.T., M.T. atas segala bimbingannya selama di Lab Simulasi.
7. Jajaran Beasiswa Peningkatan Prestasi Akademik 2018-2019 yang telah memberikan beasiswa kepada penulis.
8. Keluarga Asisten Bani Etap (Eunike, Gracia, Shafira, Arba, Nanang, Totek, Hotang, Baihaqi, dan Saad) yang selalu ada dan bersama di saat susah dan senang.
9. Mas dan Mbak Asisten 2013 dan 2014 yang memberikan ilmu dan hari-hari tak terlupakan. Adik-Adik Trainee 2016 dan Co-trainee, Mbak Farida, dan Mas Topik yang selalu sabar dan semangat menjadi teman bermain dan bekerja. Dan tidak lupa

Mas Aji yang selalu membantu penulis dengan sabar dalam pengerjaan tugas akhir.

10. Reza Rizki Perdana yang selalu ada dan semangat menjadi teman makan, dan teman bermain.
11. Teman – Teman Pejuang (Zhara, Denna, Tito, Cici, Yutup, Alvin, Damek, dan Ganggit) yang selalu ada sebagai booster di tiap-tiap pertemuan.
12. Alda Vionetta Jourdhhy yang mengerti dan selalu memberi dukungan kepada penulis.
13. Teman – Teman Cabu (Lina, Ilma, Dora, Sasa, dan FM) yang selalu ada untuk penulis.
14. Teman – Teman Cukok Dingding (Alief, April, Luthfan, Edo, Acong, Agra, Faishal, Nicky, Nadhif, Vito, Itang, dan lainnya yang tidak bisa saya sebutkan), terimakasih selalu ada dan siaga dari hari pertama berada di Teknik Elektro.
15. Teman-Teman angkatan e55 atas hari-hari dan pengalaman tak terlupakan.
16. Seluruh Dosen dan Karyawan Jurusan Teknik Elektro yang telah berjasa selama empat tahun membantu penulis selama kuliah dan telah memberikan banyak ilmu yang bermanfaat.
17. Semua pihak yang telah membantu selama proses pengerjaan tugas akhir ini.

Semoga tugas akhir ini bermanfaat bagi mahasiswa maupun peneliti. Penulis menyadari bahwa masih banyak kekurangan, untuk itu kritik dan saran diharapkan untuk penyempurnaan tugas akhir ini.

Surabaya, Juni 2019

Penulis

## **DAFTAR ISI**



<b>HALAMAN JUDUL</b> .....	iii
<b>PERNYATAAN KEASLIAN TUGAS AKHIR</b> .....	viii
<b>ABSTRAK</b> .....	ix
<b>ABSCTRACT</b> .....	xi
<b>KATA PENGANTAR</b> .....	xiii
<b>DAFTAR ISI</b> .....	xiv
<b>DAFTAR GAMBAR</b> .....	xvii
<b>DAFTAR TABEL</b> .....	xviii
<b>BAB I</b> .....	23
<b>1. PENDAHULUAN</b> .....	23
1.1 Latar Belakang.....	23
1.2 Permasalahan.....	24
1.3 Tujuan .....	24
1.4 Batasan .....	25
1.5 Metodologi .....	25
1.6 Sistematika Penulisan .....	26
1.7 Relevansi .....	27
<b>BAB II</b> .....	29
<b>2. <i>Security Constrained Unit Commitment</i></b> .....	29
2.1 Sistem Tenaga Listrik.....	29
2.2 Studi Aliran Daya .....	30
2.3 <i>Economic Dispatch</i> .....	33
2.4 <i>Unit Commitment</i> .....	34
2.5 Biaya Pembangkitan .....	37
2.6 Penjadwalan Pembangkitan .....	37
<b>BAB III</b> .....	43
<b>3. Implementasi Binary Particle Swarm Optimization Pada Penjadwalan Pembangkitan</b> .....	43
3.1 Security Constrained Unit Commitment (SCUC) dengan metode Binary Particle Swarm Optimization (BPSO)	43
3.2 AC Optimal Power Flow (ACOPF) Menggunakan Particle Swarm Optimization (PSO).....	46
<b>BAB IV</b> .....	51

<b>4. SIMULASI HASIL DAN PEMBAHASAN .....</b>	<b>51</b>
4.1 Sistem IEEE 30 Bus .....	51
4.2 Performa Komputasi.....	55
4.3 Hasil Simulasi.....	56
4.4 Analisa Hasil Simulasi.....	78
<b>BAB V .....</b>	<b>84</b>
<b>PENUTUP.....</b>	<b>85</b>
5.1 Kesimpulan.....	85
5.2 Saran.....	86
<b>DAFTAR PUSTAKA .....</b>	<b>87</b>
<b>RIWAYAT HIDUP .....</b>	<b>88</b>
<b>LAMPIRAN.....</b>	<b>93</b>

## DAFTAR GAMBAR

Gambar 2.1 Sistem Tenaga Listrik .....	29
Gambar 2.2. Kurva Biaya Operasi.....	33
Gambar 2.3. Kurva kenaikan beban .....	34
Gambar 3.1 Diagram alir SCUC dengan metode BPSO .....	43
Gambar 3.2. Diagram alir ACOPF dengan metode PSO .....	46
Gambar 3.3. Diagram alir ACOPF dengan metode PSO .....	47
Gambar 4.1. Sistem IEEE 30 Bus.....	51
Gambar 4.2 Kurva Pembebanan 24 Jam .....	78
Gambar 4.3 Pembangkitan Daya Tiap Generator.....	79
Gambar 4.4 Grafik pembangkitan daya .....	79
Gambar 4.5 Grafik batang perbandingan hasil BPSO dan Quadratic Programming .....	80
Gambar 4.6 Tegangan tiap bus pada kondisi normal .....	81
Gambar 4.7 Tegangan tiap bus pada kondisi kontingensi .....	82

*Halaman ini sengaja dikosongkan*

## **DAFTAR LAMPIRAN**

**Program Unit Commitment..... 82**  
**Program ACOPF ..... 105**  
**State Tiap Generator ..... 113**

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 4.1 Data rating generator .....	52
Tabel 4.2 Koefisien biaya generator .....	52
Tabel 4.3 Pembebanan 24 jam.....	52
Tabel 4.4 Data sistem IEEE 30 Bus .....	53
Tabel 4.5 Data sistem IEEE 30 bus .....	54
Tabel 4.6 Performa Komputasi .....	56
Tabel 4.7 Kombinasi pembangkitan pada kondisi normal.....	56
Tabel 4.8 Biaya pembangkitan kondisi normal .....	58
Tabel 4.9 Total biaya pembangkitan kondisi normal .....	59
Tabel 4.10 Daya keluaran generator .....	59
Tabel 4.11 Cadangan berputar generator .....	60
Tabel 4.12 Tegangan tiap bus pada jam 1 - 6.....	61
Tabel 4.13 Tegangan tiap bus pada jam 7 - 12.....	62
Tabel 4.14 Tegangan tiap bus pada jam 13 - 18.....	63
Tabel 4.15 Tegangan tiap bus pada jam 19 - 24.....	64
Tabel 4.16 Kombinasi pembangkitan pada kondisi kontingensi.....	66
Tabel 4.17 Biaya pembangkitan selama 24 jam pada kondisi kontingensi .....	67
Tabel 4.18 Total biaya pembangkitan pada kondisi kontingensi ....	68
Tabel 4.19 Keluaran daya aktif pada tiap generator pada kondisi kontingensi .....	68
Tabel 4.20 Cadangan berputar tiap generator pada kondisi kontingensi .....	69
Tabel 4.21 Tegangan tiap bus kondisi kontingensi pada jam 1 - 6..	71
Tabel 4.22 Tegangan tiap bus kondisi kontingensi pada jam 7 - 12	72
Tabel 4.23 Tegangan tiap bus kondisi kontingensi pada jam 13 - 18 .....	73
Tabel 4.24 Tegangan tiap bus kondisi kontingensi pada jam 19 - 24 .....	74
Tabel 4.25 Kombinasi pembangkitan dengan metode Quadratic Programming .....	75
Tabel 4.26 Biaya pembangkitan selama 24 jam.....	76
Tabel 4.27 Total biaya pembangkitan .....	77

<b>Tabel 4.28 Bus pada kondisi <i>under voltage</i> .....</b>	<b>82</b>
<b>Tabel 4.29 Bus pada kondisi <i>under voltage</i> .....</b>	<b>83</b>
<b>Tabel 4.30 Bus dengan nilai tegangan cenderung tetap .....</b>	<b>83</b>



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Beban pada sistem tenaga listrik berubah setiap saat, ini disebabkan karena pemakaian daya listrik yang berbeda untuk setiap pelanggan. Dimana ada saat beban rendah dan ada saat beban tinggi (beban puncak). Perubahan beban setiap harinya dapat digambarkan dalam bentuk kurva beban harian.

Jumlah daya yang dibangkitkan diharapkan dapat memenuhi seluruh kebutuhan beban. Dengan berubahnya beban maka daya yang dibangkitkan diatur sesuai dengan kebutuhan beban tersebut agar tidak ada energi yang terbuang percuma. Pada pengoperasian pembangkit tenaga listrik khususnya pembangkit thermal, jumlah pemakaian bahan bakar menjadi salah satu hal yang perlu mendapat perhatian khusus, karena sekitar 60% dari total biaya operasi yang dikeluarkan merupakan biaya bahan bakar .

Pada sistem tenaga listrik yang besar, beberapa pembangkit terinterkoneksi dalam memenuhi kebutuhan daya beban. Pada pembangkit yang terinterkoneksi dilakukan penjadwalan dan pembagian pembebanan pada unit pembangkit. Penjadwalan dan pembagian pembebanan ini dilakukan untuk mendapat biaya pembangkitan yang paling ekonomis.

Untuk mendapatkan operasi ekonomis pada sistem tenaga, terdapat dua pokok permasalahan yang harus dipecahkan dalam operasi ekonomis pembangkit pada sistem tenaga listrik yaitu : pengaturan pembebanan unit pembangkit (*economic dispatch*) dan penjadwalan ekonomis (*unit commitment*). Unit commitment bertujuan untuk menentukan jadwal (*schedule*) on/off unit pembangkit yang paling optimun dioperasikan dalam memenuhi beban untuk mendapat biaya bahan bakar minimum.

Untuk menyelesaikan masalah unit commitment terdapat beberapa metode yang dapat digunakan yaitu *Priority List (PL)*, *Dynamic Programming (DP)*, *Lagrangian Relaxation (LR)*,

*Evolutionary Programming (EP)*, *Simulated Annealing (SA)*, *Genetic Algorithm (GA)* dan *Binary Particle Swarm Optimization (BPSO)* [1]. Metode-metode perhitungan ini memiliki kekurangan dan kelebihan masing-masing. Selain memperhatikan metode yang digunakan perlu diperhatikan juga rugi-rugi daya pada saluran dikarenakan besarnya rugi-rugi daya pada saluran transmisi juga berdampak pada besarnya biaya pembangkitan.

Pada Tugas Akhir ini, akan membahas perhitungan unit commitment dengan metode *Binary Particle Swarm Optimization (BPSO)* dengan memperhatikan batasan *ramp rate* dan rugi rugi daya saluran transmisi pada sistem bus IEEE 30 bus dengan 6 pembangkit terpasang dengan tujuan mendapatkan kombinasi pembangkitan yang paling optimal dan biaya pembangkitan paling murah. Dengan adanya studi ini diharapkan efektivitas pada metode perhitungan *Binary Particle Swarm Optimization (BPSO)* dapat dibuktikan dan dapat mengetahui pengaruh batasan *ramp rate* dan rugi-rugi daya saluran transmisi terhadap besarnya biaya pembangkitan.

## 1.2 Permasalahan

Permasalahan yang akan diselesaikan dalam tugas akhir ini adalah tentang:

1. Bagaimana metode BPSO digunakan untuk menentukan penjadualan pembebanan dan menghitung pembebanan setiap pembangkit.
2. Bagaimana mendapatkan biaya pembangkitan yang paling murah, dengan memperhatikan beberapa *constrain* (seperti *ramp rate* pembangkit, kapasitas generator, *minimum up and down time*) dengan memperhatikan rugi-rugi daya dalam penyaluran.

## 1.3 Tujuan

Penelitian tugas akhir ini bertujuan untuk:

1. Mendapatkan hasil kombinasi pembangkitan yang paling optimal dengan menggunakan metode perhitungan Binary Particle Swarm Optimization (BPSO).
2. Mengetahui pengaruh batasan *ramp rate* dan rugi-rugi daya saluran transmisi terhadap besarnya biaya pembangkitan.

## 1.4 Batasan

Dalam menyelesaikan permasalahan pada tugas akhir ini terdapat batasan yang diperlukan, yaitu:

1. Generator unit 1 yang merupakan *slack bus* atau *swing bus* harus selalu menyala.
2. Sistem yang digunakan adalah sistem IEEE 30 bus dengan 6 unit generator thermal.
3. Untuk menghitung rugi-rugi daya pada saluran, digunakan metode newton raphson.
4. *Load Flow* mengabaikan beban Q dan hanya memperhatikan pengaruh P.
5. Perhitungan *unit commitment* menggunakan metode *binary particle swarm optimization*.

## 1.5 Metodologi

Tahapan pengerjaan yang dilakukan pada studi *Security Constrained Unit Commitment* Dengan Mempertimbangkan *Ramp Rate* dan Rugi-Rugi Pada Saluran Transmisi Menggunakan Algoritma *Binary Particle Swarm Optimization* (BPSO) pada Sistem IEEE 30 Bus adalah:

1. Studi Literatur  
Studi literatur berdasarkan pada jurnal terindex, buku-buku, dan laporan tugas akhir terdahulu yang berhubungan dengan *unit commitment*, dan algoritma *binary particle swarm optimization*.
2. Pengumpulan Data  
Penulis melakukan pengumpulan data sistem IEEE 30 bus. Data yang dikumpulkan adalah data koefisien *cost function* generator, *ramp up* dan *ramp down* perjam generator, *minimum up and down time* generator, rating maksimum dan minimum daya keluaran tiap unit generator, data beban selama 24 jam dan *initial state* tiap generator.
3. Perancangan Program  
Penulis melakukan perancangan program dengan data yang telah diperoleh dengan menggunakan software Matlab dan bantuan

Algoritma *Particle Swarm Optimization* (PSO) dan *Binary Particle Swarm Optimization* (BPSO) sebagai algoritma perhitungan guna mendapatkan kombinasi pembangkitan yang paling murah.

4. Analisa Hasil Simulasi Program

Setelah dilakukan simulasi pada program, penulis melakukan analisa dan membandingkan hasil simulasi dengan studi unit commitment dengan batasan yang sama walaupun menggunakan metode linear programming.

5. Kesimpulan

Dari hasil analisa program dan program pembanding dapat ditarik kesimpulan dari permasalahan yang dianalisa beserta tahapan didalamnya.

6. Penyusunan Laporan

Penulis melakukan tahap penyusunan laporan yang berisikan pendahuluan, studi literasi, metodologi penelitian, analisa dan simulasi, dan yang terakhir kesimpulan. Selain itu juga akan diberikan saran yang berkaitan dengan topik tugas akhir.

## 1.6 Sistematika Penulisan

Sistematika penulisan pada laporan tugas akhir ini adalah sebagai berikut:

BAB I           Pendahuluan.

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

BAB II           *Unit Commitment*

Pada bab ini dijelaskan mengenai teori unit commitment dan constrain-constrain yang berhubungan dengan topik penelitian tugas akhir yaitu security constrained unit commitment. Dijelaskan juga data input yang digunakan sebagai dasar pemilihan output terbaik dari penelitian ini.

- BAB III**      *Security Constrained Unit Commitment dan Binary Particle Swarm Optimization*  
Pada bab ini dijelaskan mengenai penggunaan algoritma Particle Swarm optimization pada perhitungan AC optimal power flow dan Binary particle swarm optimization pada perhitungan studi unit commitment pada sistem IEEE 30 bus.
- BAB IV**      Simulasi dan Analisis  
Pada bab ini dijelaskan hasil program unit commitment yang telah selesai disimulasikan dan analisa pada studi tersebut dan program pembanding.
- BAB V**      Penutup  
Pada bab ini penulis akan memberikan hasil dan menarik kesimpulan terhadap penelitian yang telah dilakukan dan memberikan rekomendasi serta saran yang dapat di implementasikan pada penelitian mendatang.

## **1.7 Relevansi**

Hasil dari penelitian diharapkan dapat memberikan manfaat, yaitu:

1. Diperoleh kombinasi penjadwalan pembangkitan yang menghasilkan total biaya paling murah dan dapat memenuhi batasan-batasan penelitian.
2. Menjadi masukan dan bahan pembelajaran tentang studi unit commitment bagi mahasiswa lain.
3. Dapat menambah ilmu pada bidang optimasi pembangkitan bagi penulis.

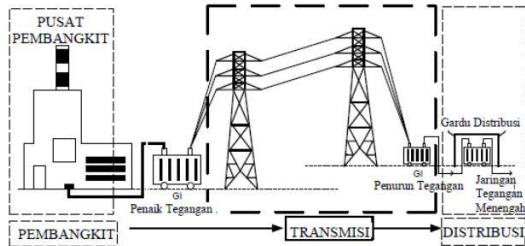
*Halaman ini sengaja dikosongkan*

## BAB II

### *Security Constrained Unit Commitment*

#### 2.1 Sistem Tenaga Listrik

Sistem tenaga listrik merupakan sistem interkoneksi jaringan yang terbagi menjadi pembangkitan, transmisi, dan distribusi.



**Gambar 2.1 Sistem Tenaga Listrik**

##### 2.1.1 Pembangkitan

Salah satu komponen terpenting bagi pembangkitan adalah generator 3 fasa atau alternator. Generator dibantu oleh *prime mover* sebagai sumber tenaga mekanik yang berfungsi untuk memutar turbin generator. Turbin yang digunakan dapat berupa hydraulic turbine dari air terjun, steam turbine dari pembakaran batu bara, gas dan tenaga nuklir. Steam turbine beroperasi pada kecepatan 3600 rpm atau 1800 rpm. *Hydraulic turbine* biasa beroperasi pada tekanan dan kecepatan yang rendah.

Generator menghasilkan daya pada tegangan yang tinggi, umumnya pada tegangan 30 kV. Pada sistem pembangkit, ukuran generator bervariasi antara 50 MW dan 1500 MW. Sebelum pembangkit di transmisi, tegangan dinaikkan dengan menggunakan trafo step up untuk mengurangi rugi-rugi daya transmisi.

### 2.1.2 Transmisi

Jaringan transmisi berfungsi untuk mengirimkan tenaga listrik dari pembangkitan ke distribusi. Jaringan transmisi juga menghubungkan antar daerah yang berbeda untuk *supply* tenaga listrik dalam keadaan normal maupun emergency. Standar tegangan transmisi di tiap negara berbeda-beda. Di Indonesia, tegangan transmisi digolongkan sesuai dengan besar tegangannya, yaitu:

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 Distribusi

Setelah daya di transmisikan, level tegangan diturunkan menggunakan trafo *step down* untuk dilanjutkan ke jaringan distribusi. Sistem distribusi merupakan jaringan yang menghubungkan konsumen dan jaringan tegangan menengah. Arus dihantarkan melalui kabel 3 fasa dengan level tegangan sesuai dengan golongannya. Untuk sistem distribusi primer berkisar antara 4 kV dan 34.5 kV melayani konsumen industri dengan kapasitas besar, sedangkan tegangan jaringan distribusi primer adalah 220/380 V yang diperuntukkan untuk konsumen residential dan komersial. Jaringan distribusi yang digunakan berupa saluran bawah tanah dan saluran udara.

## 2.2 Studi Aliran Daya

Analisa aliran daya adalah studi yang fundamental untuk dipelajari di sistem tenaga. Tujuan dari studi aliran daya adalah untuk mendapatkan tegangan dalam bentuk magnitude dan sudut pada setiap pembebanan, pembangkitan, dan kondisi jaringan. Permulaan dari penyelesaian studi aliran daya adalah untuk mengidentifikasi variable yang diketahui maupun yang tidak diketahui. Bus merupakan variable dan bus digolongkan menjadi 3 tipe yaitu: Slack, Load Bus, dan Generator Bus.



- a. *Load Bus* atau PQ Bus adalah bus yang terdapat beban yang P dan Q tetap sedangkan variable nya adalah tegangan dan sudut fasa.
- b. *Swing/ Slack Bus* merupakan bus yang terhubung dengan generator yang P dan V nya tetap dan sudut fasanya nol. Variable yang dihitung adalah daya aktif dan reaktif. Bus ini berfungsi sebagai penyeimbang pada sistem kelistrikan.
- c. *Generator bus* merupakan bus yang terhubung dengan generator yang P dan V nya tetap dan daya aktif dan sudut fasanya adalah variable yang perlu dicari.

Setiap sistem variabel yang perlu dicari adalah daya nyata, daya semu, tegangan, dan sudut fasa. Aliran daya dapat didapatkan dari formula berikut:

$$S_{ij} = V_i T_{ij}^* \quad (2.1)$$

$$S_{ij} = V_i \left[ \frac{V_i - V_s}{z_{ij}} \right] \quad (2.2)$$

Perhitungan aliran daya dapat menggunakan salah satu dari beberapa metode, metode-metode tersebut memiliki kelebihan dan kekurangannya masing-masing. Metode-metode tersebut adalah metode gauss seidel, newton raphson dan fast decouple. Pada tugas akhir ini, metode yang digunakan adalah metode newton raphson.

#### d. Metode Newton Raphson

Newton Raphson merupakan metode iterasi yang memiliki 2 metode perhitungan yaitu metode *polar coordinate* dan *rectangular coordinat*. Dari kedua metode ini, polar coordinate digunakan paling banyak sebagai metode perhitungan.

Persamaan *polar coordinate* adalah:

$$I_i = \sum_{j=1}^n |Y_{ij}| |V_j| \angle \theta_{ij} + \delta_j \quad (2.3)$$

Pada perhitungan daya aktif dan reaktif digunakan persamaan:

$$P_i = \sum_{j=1}^n |V_i| |Y_{ij}| |V_j| \cos (\theta_{ij} - \delta_i + \delta_j) \quad (2.4)$$

$$Q_i = - \sum_{j=1}^n |V_i| |Y_{ij}| |V_j| \sin (\theta_{ij} - \delta_i + \delta_j) \quad (2.5)$$

Variable yang akan di optimasi dari persamaan diatas adalah  $\delta$  dan  $|V|$  untuk PQ bus sedangkan  $\delta$  dan  $Q$  untuk PV bus. Dari persamaan diatas, dilanjutkan pada perhitungan menggunakan matriks jacobian yaitu:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix}$$

$$\Delta P_i = P_i^{\text{sch}} - P_i^{(k)} \quad (2.6)$$

Dimana  $I = 1, 2, \dots, n$  ,  $i \neq \text{slack bus}$

$$\Delta Q_i = Q_i^{\text{sch}} - Q_i^{(k)} \quad (2.7)$$

Dimana  $I = 1, 2, \dots, n$  ,  $i \neq \text{slack bus}$  ,  $I \neq \text{PV bus}$

Sedangkan untuk  $J_1$ ,  $J_2$ ,  $J_3$ , dan  $J_4$  dituliskan sebagai berikut:

Untuk elemen  $J_1$  adalah :

$$\frac{\partial P_i}{\partial \delta_i} = - |V_i| |Y_{ij}| |V_j| \sin (\theta_{ij} - \delta_i + \delta_j) \quad j \neq i \quad (2.8)$$

Untuk elemen  $J_2$  adalah :

$$\frac{\partial P_i}{\partial |V_j|} = |V_i| |Y_{ij}| \cos (\theta_{ij} - \delta_i + \delta_j) \quad j \neq i \quad (2.9)$$

Untuk elemen  $J_3$  adalah :

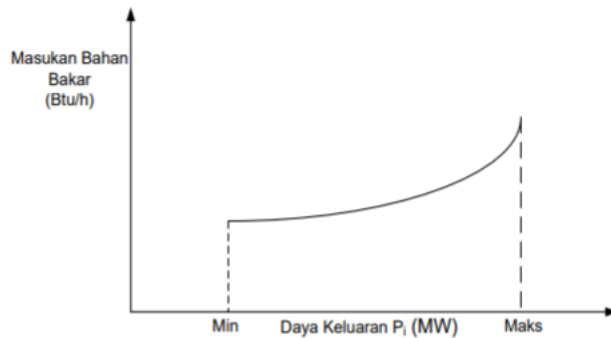
$$\frac{\partial Q_i}{\partial \delta_i} = - |V_i| |Y_{ij}| |V_j| \cos (\theta_{ij} - \delta_i + \delta_j) \quad j \neq i \quad (2.10)$$

Untuk elemen  $J_4$  adalah :

$$\frac{\partial Q_i}{\partial |V_j|} = |V_i| |Y_{ij}| \sin (-\delta_i + \delta_j) \quad j \neq i \quad (2.11)$$

## 2.3 Economic Dispatch

*Economic dispatch* adalah pengaturan daya yang dikeluarkan tiap unit generator dengan tujuan untuk mendapatkan total biaya produksi paling murah dan rugi-rugi transmisi yang paling kecil[2]. Faktor-faktor yang bisa mempengaruhi biaya produksi adalah efisiensi generator, biaya bahan bakar, dan rugi-rugi jaringan transmisi[1].



**Gambar 2.2. Kurva Biaya Operasi**

Kurva biaya operasi masukan bahan bakar pada pembangkit (Btu/h) ke biaya operasi (\$/h) dapat dilihat dari gambar diatas. Kurva berupa fungsi polynomial orde 2.

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \quad (2.12)$$

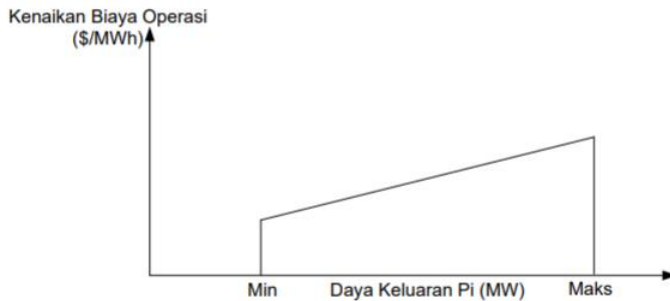
dimana  $F_i P_i$  adalah biaya operasi tiap unit pembangkit, sedangkan  $P_i$  adalah daya keluaran pembangkit dan  $a$ ,  $b$ ,  $c$  merupakan koefisien biaya operasi pembangkit.

Dengan permintaan daya / beban yang terus berubah-ubah maka hal ini mempengaruhi perubahan biaya pembangkitan. Kenaikan dari biaya pembangkitan merupakan *slope* atau kemiringan dari karakteristik masukan dan keluaran pembangkit.

*Incremental cost curve* merupakan kurva kenaikan daya beban, fungsi biaya operasi terhadap daya keluaran adalah:

$$\lambda = \frac{\Delta Fi(P_i)}{\Delta P_i} = 2 a_i P_i + b_i \quad (2.13)$$

Dimana  $\lambda$  merupakan kenaikan biaya operasi (\$/MWh). Dibawah ini merupakan kurva kenaikan beban atau *incremental cost curve*.



**Gambar 2.3. Kurva kenaikan beban**

## 2.4 Unit Commitment

Studi Unit commitment bertujuan untuk menjadwalkan pembangkitan yang optimal. Unit commitment menjadwalkan unit mana saja yang nyala dan mati dengan tujuan untuk mendapatkan biaya pembangkitan paling murah[3]. Unit commitment memperhitungkan total biaya pembangkitan yang mencakup biaya bahan bakar, biaya *start-up*, dan biaya *shut-down*. Metodologi dari unit commitment dapat di bedakan menjadi 2 tipe yaitu metode untuk optimisasi dan metode metatheuristik, unit commitment juga digambarkan dengan formulasi matematika dalam bentuk *complex non-linear* dan *mixed integer*.

Batasan pada unit commitment sangat berpengaruh pada hasil akhir kombinasi pembangkitan, batasan-batasan tersebut dibagi menjadi 2 kategori yaitu batasan level sistem dan batasan local.

### 2.4.1 Batasan Sistem

#### a. Active Power Balance

$$P_G^t = P_L^t + P_{IN}^t \quad (2.14)$$

$$P_{IN}^t = \sum \frac{1}{x_{ij}} [\delta_i - \delta_j] \quad (2.15)$$

Pada tugas akhir ini, rugi-rugi pada jaringan transmisi juga diperhitungkan.

dimana,

$P_L$	: Daya pada load bus
$P_{IN}$	: Daya yang mengalir dari bus i ke j
$\delta$	: Sudut tegangan
$x$	: Reaktansi saluran

Daya yang dibangkitkan harus memenuhi beban di tiap periode waktu.

### 2.4.2 Batasan Lokal

#### a. Generator Limit

$$P_{Gi \min} \leq P_{Gi} \leq P_{Gi \max} \quad (2.16)$$

dimana,

$P_{Gi}$	: Daya Keluaran generator i
$P_{Gi \min}$	: Batas bawah daya keluaran generator i
$P_{Gi \max}$	: Batas atas daya keluaran generator i

Tiap unit generator memiliki rating atau karakteristik masing-masing. Generator memiliki rating daya yang bisa dibangkitkan, dimana daya tersebut tidak boleh lebih dari rating daya maksimum yang bisa dibangkitkan oleh generator dan kurang dari rating daya minimum yang bisa dibangkitkan oleh generator.

a. *Minimum Up and Down Time*

$$u_i^t = 1 \text{ jika } 1 \leq Ton_i^t \leq t_i^{up} \quad (2.17)$$

$$u_i^t = 0 \text{ jika } -t_i^{down} < Toff_i^t \leq 0 \quad (2.18)$$

dimana,

- $u_i^t$  : Indikator generator on (1) atau off (0)
- $t_i^{up}$  : *Minimum up time* unit i
- $t_i^{down}$  : *Minimum down time* unit i
- $Ton_i^t$  : Durasi lama generator unit i nyala
- $Toff_i^t$  : Durasi lama generator unit i mati

*Minimum up time* adalah batas waktu minimum generator dapat dimatikan. Ketika generator dibangkitkan, generator tersebut tidak bisa dimatikan secara tiba-tiba, sehingga harus memenuhi jumlah batas *minimum up time*. *Minimum down time* adalah batas waktu minimum generator dapat dibangkitkan. Ketika generator mati, generator tersebut tidak bisa dibangkitkan secara tiba-tiba, sehingga harus memenuhi jumlah batas *minimum down time*.

b. *Ramp Rate*

$$P_i^t u_i^t \leq P_i^t \leq P_{i,max}^t u_i^t, t = 1 \dots T \quad (2.19)$$

dimana,

- $P_{i,min}^t$  : *Ramp Down* unit i saat t time
- $P_{i,max}^t$  : *Ramp Up* unit i saat t time

*Ramp rate* merupakan batas maksimum lonjakan daya yang dibangkitkan pada periode waktu setelah. Untuk ramp down yaitu lonjakan turun daya yang dibangkitkan pada periode waktu setelah dan ramp up yaitu lonjakan naik daya yang dibangkitkan pada periode waktu setelah[4].

## 2.5 Biaya Pembangkitan

Biaya pembangkitan dapat dibagi jadi beberapa golongan yaitu biaya untuk bahan bakar, *start up cost*, dan *shut down cost*.

### a. *Start Up Cost*

*Start up cost* adalah biaya pembangkitan sebelum generator tersambung pada sistem. Biaya ini tergantung pada lamanya unit selama *offline* sebelum dipanaskan. Pemanasan menggunakan bahan bakar dan biasanya prosedur *start up* membutuhkan suhu yang lebih panas dari sebelumnya.

### b. *Shut Down Cost*

*Shut Down cost* adalah biaya sebelum unit generator *shut down*. Rugi-rugi panas timbul selama proses ini karena level pembangkitan sebelum mati sudah pasti dengan kata lain *shut down cost* juga merupakan biaya konstan.

### c. Biaya bahan bakar

Biaya bahan bakar adalah biaya yang dikeluarkan tiap generator untuk memproduksi listrik. Tiap generator memiliki rating masing-masing sehingga biaya bahan bakar yang dikeluarkan tiap unit generator berbeda.

## 2.6 Penjadwalan Pembangkitan

Kombinasi penjadwalan pembangkitan didapatkan dari beberapa metode yaitu:

- a. Priority list
- b. Complete enumeration
- c. Evolutionary Program
- d. Dynamic Programming
- e. Linear Programming
- f. Simulated Annaeling
- g. Lagrange Relaxation
- h. Genetic Algorithm
- i. Binary Particle Swarm Optimization

Pada penelitian tugas akhir ini, metode yang digunakan adalah binary particle swarm optimization.

### 2.6.1 Binary Particle Swarm Optimization (BPSO)

Perhitungan dari unit commitment cukup complex sehingga memerlukan teknik optimasi terdepan diantaranya adalah Genetic Algorithm (GE), Priority List (PL), Dynamic Programming (DP), Lagrangian Relaxation (LR), Evolutionary Programming (EP), Simulated Annealing (SA), dan Particle Swarm Optimization (PSO).

Perbedaan dari Teknik optimasi yang telah disebutkan adalah dari lama waktu yang dibutuhkan, tingkat kerumitan, dan hasil optimum yang didapat. Teknik optimasi yang dipilih pada tugas akhir ini adalah Binary Particle Swarm Optimization (BPSO) dan Particle Swarm Optimization (PSO), dimana PSO digunakan untuk mendapatkan daya keluaran yang optimal dari tiap unit generator dan BPSO digunakan untuk mendapatkan kombinasi pembangkitan yang memiliki biaya pembangkitan paling murah. Kedua algoritma ini merupakan algoritma evolution search paling terkenal yang dikembangkan oleh Kennedy & Eberhart dan dipercaya perhitungannya lebih efisien[4], lebih cepat, dan tidak banyak yang harus dimodifikasi jika dibandingkan dengan algoritma artificial intelligent lainnya.

Pada program BPSO terdapat 2 hal yang harus ditetapkan terlebih dahulu selain parameter partikel, yaitu fungsi objektif dan variabel permasalahan[3]. Fungsi objektif dan variabel permasalahan pada tugas akhir ini adalah:

#### a. Fungsi Objektif

*Objective Function* pada studi unit commitment sama dengan *Economic Dispatch* dimana yang harus didapatkan adalah biaya pembangkitan paling minimum termasuk *start up cost* dan *shut down cost*.

$$\text{Minimize } F(P,u) = \sum_{t=1}^T \sum_{i=1}^N C_i(P_u, U_u) + \sum_{i=1}^N SC_i \quad (2.20)$$



$$Ci(P_u) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i \quad (2.21)$$

dimana,

- t : Jumlah rentang waktu beban
- N : Jumlah bus
- $P_G$  : Daya yang dibangkitkan
- $Ci(P_u)$  : Fungsi biaya pembangkitan generator i
- SCI : *Start-Up Cost*

Fungsi objektif yang diperhatikan adalah biaya pembangkitan, biaya operasional, dan biaya pelepasan. Pembangkitan paling optimal didapatkan dengan tujuan untuk mendapatkan biaya pembangkitan paling minimum.

#### b. Variable Permasalahan

Variable yang di optimasi adalah daya pembangkitan aktif (P). Pada tugas akhir ini yang diperhatikan hanya pembangkitan daya aktif sedangkan beban dan pembangkitan daya reaktif diabaikan. Pada system IEEE 30 bus terdapat 6 generator sehingga n adalah 6 dan ukuran matriks x adalah [1 : n]. Maka matriks variable yang dioptimasi adalah:

$$X = \begin{bmatrix} P1 \\ \vdots \\ Pn \end{bmatrix}$$

Teknik optimasi PSO dan BPSO menggunakan perumpamaan populasi yang diletakan secara random pada titik tertentu pada sebuah ruang dan pergerakan partikel dipengaruhi oleh momentum inersia, *social* dan *cognitive* komponen untuk mendapatkan nilai yang terbaik, sehingga dibutuhkan iterasi berulang kali untuk mendapatkan hasil yang optimal[5].

$$X_1(0), X_2(0), \dots, X_N(0) \quad (2.22)$$

$$V_1(0), V_2(0), \dots, V_N(0) \quad (2.23)$$

Komponen social pada PSO dan BPSO merupakan memori dari posisi terbaik partikel diantara partikel lain sedangkan komponen *cognitive* merupakan memori dari posisi terbaik partikel tersebut. Setiap partikel berpindah posisi menggunakan perhitungan kecepatan dan jarak terdekat menuju partikel yang menempati posisi terbaik.

Formula untuk memperbarui posisi partikel tersebut adalah:

$$V_{N+1} = \omega \times V_N + C_1 \times r_1 \times (P_{best\ N} - X_N) + C_2 \times r_2 \times (g_{best\ N} - X_N) \quad (2.24)$$

$$X_{N+1} = X_N + V_{N+1} \quad (2.25)$$

dimana,

$V_i$  : Kecepatan Partikel

$X_N$  : Posisi Partikel

$\omega$  : Berat Inersia

$C_1$  : Koefisien *Cognition* Inersia

$C_2$  : Koefisien *Social* Inersia

$P_{best}$  : Posisi Terbaik lokal

$g_{best}$  : Posisi Terbaik global

$r_1, r_2$ : Bilangan Random Antara 0 Sampai 1

Cara kerja algoritma BPSO terbilang cukup sama dengan cara kerja PSO tetapi perlu adanya perubahan dari posisi partikel menjadi bilangan biner.

$$S(V_i(t)) = (1 + e^{-V_i(t)})^{-1} \quad (2.26)$$

Dimana  $S(V_i(t))$  adalah fungsi simoid dan output dari fungsi simoid adalah bilangan biner yang bernilai 1 atau 0.

$$\begin{aligned} S(V_i(t)) > \text{rand}, X_{N+1} &= 1 \\ S(V_i(t)) < \text{rand}, X_{N+1} &= 0 \end{aligned}$$

Jika bilangan *random* lebih kecil dari  $S(V_i(t))$  maka output bilangan biner adalah 1, dan jika  $S(V_i(t))$  lebih besar dari bilangan *random* maka output bilangan biner akan bernilai 0. Bilangan biner yang didapatkan merupakan pembangkitan tiap unit generator dimana 1 yang berarti generator ON, dan 0 yang berarti generator OFF. Bilangan biner yang telah didapatkan nantinya akan menghasilkan matriks. Matriks yang dihasilkan berisi kombinasi pembangkitan.

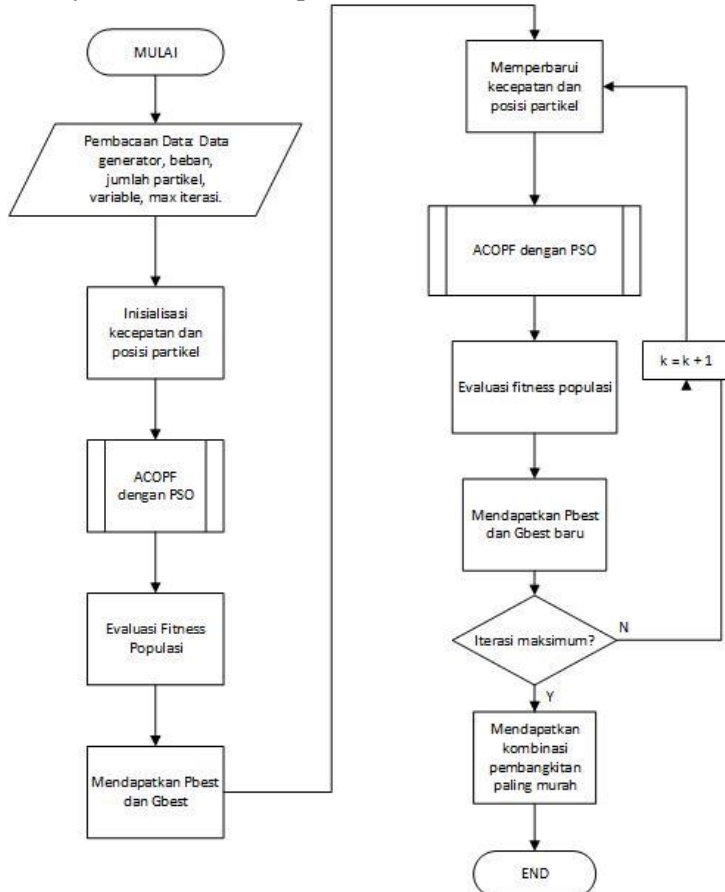
Dengan studi kombinasi pembangkitan memperhitungkan ramp rate juga rugi-rugi daya saluran transmisi menggunakan metode binary particle swarm optimization, diharapkan didapatkan kombinasi pembangkitan yang paling murah dan daya yang dibangkitkan sesuai dengan kebutuhan beban.

*Halaman ini sengaja dikosongkan*

### BAB III

## Implementasi Binary Particle Swarm Optimization Pada Penjadwalan Pembangkitan

### 3.1 Security Constrained Unit Commitment (SCUC) dengan metode Binary Particle Swarm Optimization (BPSO)



Gambar 3.1 Diagram alir SCUC dengan metode BPSO

### 3.1.1 Parameter BPSO

Parameter yang digunakan pada BPSO untuk penjadwalan pembangkitan yaitu:

Jumlah *Swarm* = Jumlah unit pembangkit (6 Generator)

Jumlah *Particle* = 50

Iterasi maksimal = 5

$C_1$  = 2

$C_2$  = 2

$\omega_1$  = 0.4

$\omega_2$  = 0.9

### 3.1.2 Inialisasi Posisi Awal Partikel

Parameter yang di inialisasi awal adalah kecepatan dan posisi partikel[6]. Dimana, posisi dan kecepatan partikel awal dibangkitkan dari bilangan random antara 0 dan 1. Berikut ini adalah posisi dan kecepatan awal dari partikel:

$$X_1(0), X_2(0), \dots, X_N(0) \quad (3.1)$$

$$V_1(0), V_2(0), \dots, V_N(0) \quad (3.2)$$

### 3.1.3 Batasan Unit Commitment

Studi Unit Commitment pada tugas akhir ini memperhatikan pembebanan selama 24 jam dan terdapat batasan-batasan perhitungan yang harus diperhatikan. Batasan tersebut terdapat pada persamaan 2.4.1 dan 2.4.2.

### 3.1.4 Evaluasi Fitness populasi

Evaluasi nilai fitness populasi didasarkan oleh fitness terbaik yang tidak melanggar batasan yang telah ditentukan. Batasan tersebut adalah maksimum dan minimum pembangkitan, permintaan beban yang dibangkitkan, dan pembangkitan dengan biaya minimum. Partikel yang melanggar

batasan-batasan diabaikan pada pemilihan nilai fitness dengan cara biaya pembangkitan di besarkan.

### 3.1.5 Penentuan Posisi Terbaik Lokal dan Global

Pemilihan partikel fitness pada PSO didapatkan dari hasil terbaik local dan terbaik global. Dimana, terbaik local adalah Pbest dan terbaik global adalah Gbest. Pbest adalah posisi terbaik yang dimiliki partikel tersebut sedangkan Gbest adalah posisi terbaik diantara partikel lain[7].

Posisi terbaik partikel didapatkan ketika partikel tersebut tidak melanggar batasan-batasan, dan partikel dapat melakukan evaluasi terhadap dirinya sendiri. Posisi terbaik diantara partikel lain atau Gbest didapatkan dari nilai minimum diantara partikel lain.

### 3.1.6 Pembaruan Kecepatan dan Posisi Partikel

Langkah selanjutnya adalah memperbarui posisi partikel dan kecepatan partikel. Pembaruan yang dilakukan BPSO dengan persamaan sebagai berikut:

$$V_{N+1} = \omega \times V_N + C_1 \times r_1 \times (P_{best\ N} - X_N) + C_2 \times r_2 \times (g_{best\ N} - X_N) \quad (3.3)$$

$$X_{N+1} = X_N + V_{N+1} \quad (3.4)$$

Untuk mendapatkan posisi partikel terbaik dalam bentuk biner perlu menggunakan fungsi simoid yaitu:

$$S(V_i(t)) = (1 + e^{-V_i(t)})^{-1} \quad (3.5)$$

Dimana  $S(V_i(t))$  adalah fungsi simoid dan output dari fungsi simoid adalah bilangan biner yang bernilai 1 atau 0.

$$\begin{aligned} S(V_i(t)) > \text{rand}, X_{N+1} &= 1 \\ S(V_i(t)) < \text{rand}, X_{N+1} &= 0 \end{aligned}$$

Jika bilangan biner bernilai 1 berarti generator tersebut nyala sedangkan jika bernilai 0 berarti generator tersebut mati.

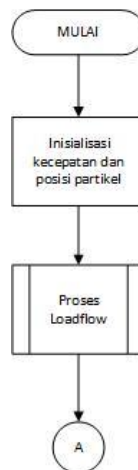
### 3.1.7 Hasil Unit Commitment

Hasil dari studi unit commitment pada tugas akhir ini adalah penjadwalan pembangkitan selama 24 jam dengan memperhatikan batasan-batasan pada 6 generator untuk mendapatkan fungsi biaya paling minimum.

Hasil kombinasi pembangkitan berupa matriks berukuran  $24 \times 6$  yang berisi bilangan biner. Selain itu pada studi unit commitment ini dapat dilihat spinning reserve tiap generator, daya yang dibangkitkan, start up cost, shut down cost, rugi-rugi jaringan transmisi dan tegangan pada tiap bus.

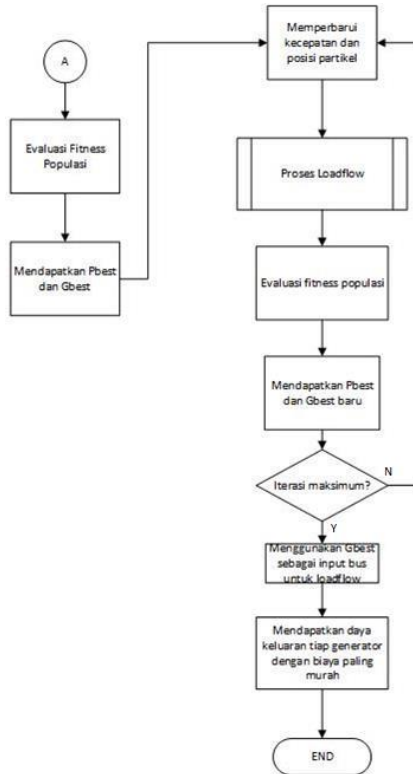
### 3.2 AC Optimal Power Flow (ACOPF) Menggunakan Particle Swarm Optimization (PSO)

Pada perhitungan aliran daya metode perhitungan yang digunakan adalah metode Newton Raphson dan untuk optimasi load flow digunakan algoritma Particle Swarm Optimization.



Gambar 3.2. Diagram alir ACOPF dengan metode PSO





**Gambar 3.3. Diagram alir ACOPF dengan metode PSO**

### 3.2.1 Parameter PSO

Parameter PSO pada tugas akhir ini untuk inialisasi awal adalah:

Jumlah *Swarm* = Jumlah unit pembangkit (6 Generator)

Jumlah *Particle* = 5

Iterasi maksimal = 2

$C_1$  = 2

$C_2$  = 2

$\omega_1$  = 0.4

$\omega_2$  = 0.9

### 3.2.2 Inialisasi Posisi dan Kecepatan Partikel

Parameter yang di inialisasi pada metode PSO sama dengan pada metode BPSO yaitu adalah kecepatan dan posisi partikel. Dimana, posisi dan kecepatan partikel awal dibangkitkan dari bilangan random antara 0 dan 1.

### 3.2.3 Metode Newton Raphson Sebagai Perhitungan Aliran Daya

Langkah-langkah perhitungan aliran daya menggunakan metode Newton Raphson adalah:

1. Membuat Y bus
2. Asumsikan nilai tegangan  $|V|$  pada bus beban dan sudut  $\delta$  pada PV bus. Biasanya nilai tegangan dan sudut fasa dianggap sama dengan slack bus dimana  $|V| = 1.0$  dan  $\delta = 0^\circ$
3. Melakukan perhitungan  $P_i$  dan  $Q_i$  untuk setiap bus menggunakan persamaan 2.4 dan 2.5.
4. Melakukan perhitungan untuk mendapatkan  $\Delta P_i$  dan  $\Delta Q_i$  pada setiap bus menggunakan persamaan 2.6 dan 2.7.
5. Melakukan perhitungan pada elemen-elemen jacobian matrix.

$$\begin{bmatrix} H & N' \\ M & L' \end{bmatrix}$$

6. Mencari nilai dari  $\Delta\delta$  dan  $\Delta|V|$ .
7. Dari nilai  $\Delta\delta$  dan  $\Delta|V|$ , tegangan dan sudut fasa diperbarui.
8. Mulai ke iterasi selanjutnya dimulai dari step 2 dengan menggunakan nilai dari tegangan dan sudut fasa yang telah diperbarui.
9. Iterasi terus berlangsung sampai di temukan error pada bus dengan toleransi:

$$|\Delta P_i^{(k)}| \leq \epsilon \quad (3.6)$$

$$|\Delta Q_i^{(k)}| \leq \epsilon \quad (3.7)$$

Dimana,  $\epsilon$  adalah level toleransi dari load bus.

10. Melakukan perhitungan untuk mendapatkan aliran daya pada tiap saluran dan aliran daya pada slack bus.

### 3.2.4 Evaluasi Fitness Populasi

Pada ACOPF evaluasi nilai fitness populasi dilakukan dengan cara yang sama dengan evaluasi fitness populasi yang dilakukan pada metode BPSO. Penentuan fitness terbaik harus didasarkan pada batasan-batasan yang tidak dilanggar. Jika terdapat partikel yang melanggar batasan-batasan maka akan diabaikan dengan cara membesarkan biaya pembangkitan sehingga partikel tersebut tidak dipilih.

### 3.2.5 Pembaruan Kecepatan dan Posisi Partikel

Setelah dilakukan evaluasi nilai fitness populasi maka yang dilanjutkan setelahnya adalah memperbarui posisi partikel dan kecepatan partikel. Pembaruan kecepatan dan partikel bertujuan untuk mendapatkan nilai konvergensi dalam waktu yang cepat.

$$V_{N+1} = \omega \times V_N + C_1 \times r_1 \times (P_{best\ N} - X_N) + C_2 \times r_2 \times (g_{best\ N} - X_N) \quad (3.8)$$

$$X_{N+1} = X_N + V_{N+1} \quad (3.9)$$

### 3.2.6 Penentuan Posisi Lokal dan Global Terbaik

Penentuan posisi terbaik local atau Pbest adalah memilih nilai paling minimum dari posisi partikel tersebut sedangkan menentukan posisi terbaik global atau Gbest adalah dengan memilih partikel dengan nilai paling minimum dari posisi partikel-partikel yang lain. Setiap partikel dapat mengevaluasi partikel itu sendiri untuk mendapatkan posisi terbaik global.

### 3.2.7 Hasil Pembangkitan

Setelah semua langkah terselesaikan didapatkan hasil besar pembangkitan daya dari tiap generator dengan biaya pembangkitan paling minimum.

*Halaman ini sengaja dikosongkan*

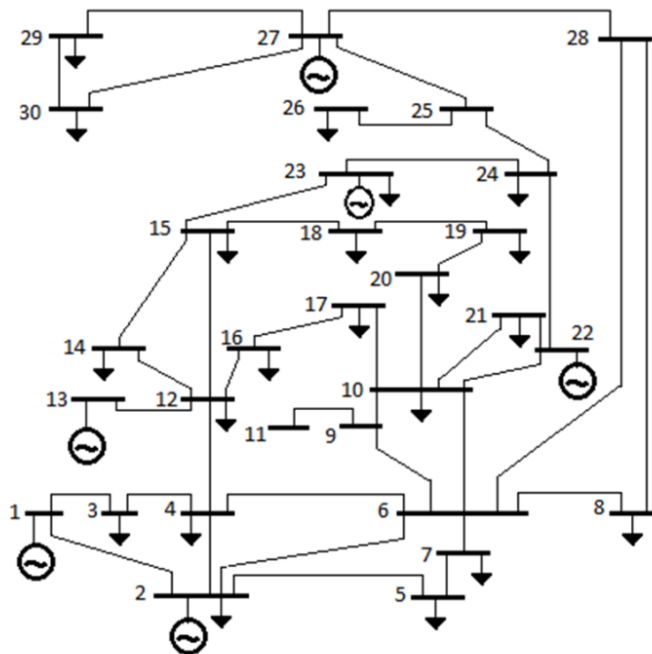
## BAB IV

### SIMULASI HASIL DAN PEMBAHASAN

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

#### 4.1 Sistem IEEE 30 Bus

Pada topik tugas akhir ini, sistem yang digunakan adalah sistem IEEE 30 bus dengan 6 pembangkit thermal. Berikut adalah *Single line Diagram* sistem:



Gambar 4.1. Sistem IEEE 30 Bus

**Tabel 4.1 Data rating generator**

<b>Gen No.</b>	<b>Pmax (MW)</b>	<b>Pmin (MW)</b>	<b>Min Up &amp; Down Time (Hour)</b>	<b>Ramp Up &amp; Down (MW)</b>	<b>Initial State</b>
1	200	50	4	80	-4
2	80	20	3	22	3
3	50	15	2	15	-3
4	35	10	2	16	-3
5	30	10	2	9	-2
6	40	12	2	16	-2

**Tabel 4.2 Koefisien biaya generator**

<b>Gen No.</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>Cost Start Hour (Hour)</b>	<b>Cold Start Cost (\$)</b>	<b>Shut Down Cost (\$)</b>	<b>Hot Start Cost (\$)</b>
1	0,00375	2	0	5	176	50	70
2	0,0175	1,75	0	5	187	60	74
3	0,625	1	0	4	113	30	110
4	0,00834	3,25	0	4	267	85	50
5	0,025	3	0	4	180	52	72
6	0,025	3	0	2	113	30	40

**Tabel 4.3 Pembebanan 24 jam**

<b>Hour</b>	<b>Load (MW)</b>	<b>Hour</b>	<b>Load (MW)</b>
1	211,904	13	274,34
2	198,66	14	270,556
3	196,768	15	266,772
4	208,12	16	259,204
5	221,364	17	287,584

**Lanjutan Tabel 4.3**

<b>6</b>	217,58	<b>18</b>	300,828
<b>7</b>	232,716	<b>19</b>	302,72
<b>8</b>	247,852	<b>20</b>	295,152
<b>9</b>	253,528	<b>21</b>	280,016
<b>10</b>	253,528	<b>22</b>	261,096
<b>11</b>	234,608	<b>23</b>	247,852
<b>12</b>	238,392	<b>24</b>	234,608

**Tabel 4.4 Data sistem IEEE 30 Bus**

<b>Bus</b>	<b>Tegangan (pu)</b>	<b>Beban (MW)</b>	<b>Generator (MW)</b>
<b>1</b>	1,04	0	23,54
<b>2</b>	1,043	21,7	60,97
<b>3</b>	1	2,4	0
<b>4</b>	1,06	7,6	0
<b>5</b>	1,01	0	0
<b>6</b>	1	0	0
<b>7</b>	1	22,8	0
<b>8</b>	1,01	30	0
<b>9</b>	1	0	0
<b>10</b>	1	5,8	0
<b>11</b>	1,042	0	0
<b>12</b>	1	11,2	0
<b>13</b>	1,041	0	37
<b>14</b>	1	6,2	0
<b>15</b>	1	8,2	0
<b>16</b>	1	3,5	0
<b>17</b>	1	9	0
<b>18</b>	1	3,2	0
<b>19</b>	1	9,5	0

**Lanjutan Tabel 4.4**

<b>Bus</b>	<b>Tegangan (pu)</b>	<b>Beban (MW)</b>	<b>Generator (MW)</b>
<b>20</b>	1	2,2	0
<b>21</b>	1	17,5	0
<b>22</b>	1	0	21,59
<b>23</b>	1	3,2	19,2
<b>24</b>	1	8,7	0
<b>25</b>	1	0	0
<b>26</b>	1	3,5	0
<b>27</b>	1	0	26,91
<b>28</b>	1	0	0
<b>29</b>	1	2,4	0
<b>30</b>	1	10,6	0

**Tabel 4.5 Data sistem IEEE 30 bus**

<b>Bus nl</b>	<b>Bus nr</b>	<b>R (ohm)</b>	<b>X (ohm)</b>	<b>1/2 B (ohm)</b>
<b>1</b>	<b>2</b>	0,0192	0,0575	0,0264
<b>1</b>	<b>3</b>	0,0452	0,1852	0,0204
<b>2</b>	<b>4</b>	0,057	0,1737	0,0184
<b>3</b>	<b>4</b>	0,0132	0,0379	0,0042
<b>2</b>	<b>5</b>	0,0472	0,1983	0,0209
<b>2</b>	<b>6</b>	0,0581	0,1763	0,0187
<b>4</b>	<b>6</b>	0,0119	0,0414	0,0045
<b>5</b>	<b>7</b>	0,046	0,116	0,0102
<b>6</b>	<b>7</b>	0,0267	0,082	0,0085
<b>6</b>	<b>8</b>	0,012	0,042	0,0045
<b>6</b>	<b>9</b>	0	0,208	0,97
<b>6</b>	<b>10</b>	0	0,556	0,96
<b>9</b>	<b>11</b>	0	0,208	0
<b>9</b>	<b>10</b>	0	0,11	0
<b>4</b>	<b>12</b>	0	0,256	0,93



**Lanjutan Tabel 4.5**

<b>Bus nl</b>	<b>Bus nr</b>	<b>R (ohm)</b>	<b>X (ohm)</b>	<b>1/2 B (ohm)</b>
12	13	0	0,14	0
12	14	0,1231	0,2559	0
12	15	0,0662	0,1304	0
12	16	0,0945	0,1987	0
14	15	0,221	0,1997	0
16	17	0,0824	0,1923	0
15	18	0,1073	0,2185	0
18	19	0,0639	0,1292	0
19	20	0,034	0,068	0
10	20	0,0936	0,209	0
10	17	0,0324	0,0845	0
10	21	0,0348	0,0749	0
10	22	0,0727	0,1499	0
21	22	0,0116	0,0236	0
15	23	0,1	0,202	0
22	24	0,115	0,179	0
23	24	0,132	0,27	0
24	25	0,1885	0,3292	0
25	26	0,2544	0,38	0
25	27	0,1093	0,2087	0
28	27	0	0,396	0
27	29	0,2198	0,4153	0
27	30	0,3202	0,6027	0
29	30	0,2399	0,4533	0
8	28	0,0636	0,2	0,0214
6	28	0,0169	0,0599	0,065

## 4.2 Performa Komputasi

Simulasi program menggunakan computer dengan spesifikasi yang terdapat pada table 4.6.

**Tabel 4.6 Performa Komputasi**

Parameter	Spesifikasi
Processor Unit	Intel Core i7 @1.80 GHz
RAM	16.0 GB
Sistem Operasi	Windows 10 64-bit
Software Simulasi	Matlab 2014

### 4.3 Hasil Simulasi

Pada subbab ini akan dibahas hasil simulasi program SCUC menggunakan metode BPSO dan SCUC menggunakan metode metode *quadratic programming* sebagai metode pembanding. Kedua program tersebut melakukan simulasi pada sistem IEEE 30 bus untuk pembebanan selama 24 jam dengan mengabaikan beban reaktif (Q).

#### 4.3.1 Hasil Simulasi SCUC Dengan Metode BPSO

Simulasi SCUC dengan metode BPSO dilakukan pada 2 kondisi yaitu, kondisi normal dan kontingensi.

##### 4.3.1.1 Kondisi Normal

1. Kombinasi Pembangkitan

Tabel 4.7 merupakan kombinasi pembangkitan selama 24 jam.

**Tabel 4.7 Kombinasi pembangkitan pada kondisi normal**

Hour	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6
1	1	1	0	0	0	0
2	1	1	0	0	0	0
3	1	1	0	0	0	0
4	1	1	0	0	0	0
5	1	1	0	0	0	0
6	1	1	0	0	0	0

**Lanjutan Tabel 4.8**

<i>Hour</i>	<b>Gen1</b>	<b>Gen2</b>	<b>Gen3</b>	<b>Gen4</b>	<b>Gen5</b>	<b>Gen6</b>
<b>7</b>	1	1	0	0	0	0
<b>8</b>	1	1	0	0	0	0
<b>9</b>	1	1	1	0	0	0
<b>10</b>	1	1	1	0	0	0
<b>11</b>	1	1	1	0	0	0
<b>12</b>	1	1	1	0	0	0
<b>13</b>	1	1	1	0	0	0
<b>14</b>	1	1	1	0	0	0
<b>15</b>	1	1	1	0	0	0
<b>16</b>	1	1	1	0	0	0
<b>17</b>	1	1	1	0	0	0
<b>18</b>	1	1	1	0	0	1
<b>19</b>	1	1	1	0	0	1
<b>20</b>	1	1	1	0	0	1
<b>21</b>	1	1	1	0	0	1
<b>22</b>	1	1	1	0	0	1
<b>23</b>	1	1	1	0	0	1
<b>24</b>	1	1	1	0	0	1

2. Biaya Pembangkitan Selama 24 Jam

Tabel 4.8 merupakan biaya pembangkitan daya untuk pembebanan 24 jam. Terdapat 6 generator pada sistem, dan pada kondisi awal hanya ada 1 generator menyala.

**Tabel 4.8 Biaya pembangkitan kondisi normal**

<i>Hour</i>	<i>Production Cost (\$)</i>	<i>Start- Up Cost (\$)</i>	<i>Shut Down Cost (\$)</i>	<i>Total Cost (\$)</i>
<b>1</b>	588,8	70	0	658,8
<b>2</b>	537,5	0	0	537,5
<b>3</b>	530,8	0	0	530,8
<b>4</b>	571,4	0	0	571,4
<b>5</b>	620,1	0	0	620,1
<b>6</b>	606	0	0	606
<b>7</b>	663	0	0	663
<b>8</b>	723,4	0	0	723,4
<b>9</b>	727,6	113	0	840,6
<b>10</b>	712,4	0	0	712,4
<b>11</b>	641,3	0	0	641,3
<b>12</b>	655,4	0	0	655,4
<b>13</b>	795,8	0	0	795,8
<b>14</b>	780,2	0	0	780,2
<b>15</b>	764,2	0	0	764,2
<b>16</b>	734,5	0	0	734,5
<b>17</b>	863,1	0	0	863,1
<b>18</b>	901,5	113	0	1014,5
<b>19</b>	909,5	0	0	909,5
<b>20</b>	869,9	0	0	869,9
<b>21</b>	808	0	0	808
<b>22</b>	735,5	0	0	735,5
<b>23</b>	686,1	0	0	686,1
<b>24</b>	638,7	0	0	638,7

**Tabel 4.9 Total biaya pembangkitan kondisi normal**

<b>Total Biaya Pembangkitan (\$)</b>	17360.7
<b>Biaya Produksi 24 jam (\$)</b>	17064.7
<b><i>Start Up Cost 24 jam (\$)</i></b>	296
<b><i>Start Down Cost 24 jam (\$)</i></b>	0

3. Pembangkitan Daya Tiap Generator

Tabel 4.10 merupakan pembangkitan daya aktif pada 6 generator selama 24 jam

**Tabel 4.10 Daya keluaran generator**

<b><i>Hour</i></b>	<b>Gen1 (MW)</b>	<b>Gen2 (MW)</b>	<b>Gen3 (MW)</b>	<b>Gen4 (MW)</b>	<b>Gen5 (MW)</b>	<b>Gen6 (MW)</b>
<b>1</b>	160,3126	60,97	0	0	0	0
<b>2</b>	161,8696	45,3838	0	0	0	0
<b>3</b>	160,2255	44,9701	0	0	0	0
<b>4</b>	170,2383	47,3998	0	0	0	0
<b>5</b>	181,9927	50,1759	0	0	0	0
<b>6</b>	178,622	49,3853	0	0	0	0
<b>7</b>	191,9812	52,7106	0	0	0	0
<b>8</b>	197,7775	63,3753	0	0	0	0
<b>9</b>	178,4669	49,2672	37	0	0	0
<b>10</b>	190,3518	52,4106	23	0	0	0
<b>11</b>	175,5411	48,9985	20,5379	0	0	0
<b>12</b>	179,894	50,1124	19,3211	0	0	0
<b>13</b>	199,3543	62,4445	26,5259	0	0	0
<b>14</b>	198,1068	59,796	26,3097	0	0	0
<b>15</b>	199,0327	58,59	22,7235	0	0	0
<b>16</b>	198,5398	50,7549	22,9081	0	0	0
<b>17</b>	195,3231	69,7782	36,9081	0	0	0

**Lanjutan Tabel 4.10**

<i>Hour</i>	<b>Gen1 (MW)</b>	<b>Gen2 (MW)</b>	<b>Gen3 (MW)</b>	<b>Gen4 (MW)</b>	<b>Gen5 (MW)</b>	<b>Gen6 (MW)</b>
<b>18</b>	197,0787	80	22,9081	0	0	15,8925
<b>19</b>	197,3934	59	35,7752	0	0	24,5423
<b>20</b>	198,5611	64,1399	23,1557	0	0	23,4303
<b>21</b>	199,3646	56,5768	22,2181	0	0	15,3982
<b>22</b>	178,9605	54,5829	21,6319	0	0	17,1971
<b>23</b>	179,1145	46,9662	18,4857	0	0	14,0086
<b>24</b>	163,0575	51,2099	16,9526	0	0	12,7921

4. Cadangan Berputar Tiap Generator Selama 24 jam

Tabel 4.11 merupakan cadangan berputar pada tiap generator yang menyala selama 24 jam.

**Tabel 4.11 Cadangan berputar generator**

<i>Hour</i>	<b>Generator (MW)</b>						<b>Total</b>
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>(MW)</b>
<b>1</b>	39,6847	19,03	0	0	0	0	58,7147
<b>2</b>	38,1304	34,6162	0	0	0	0	72,7466
<b>3</b>	39,7745	35,0299	0	0	0	0	74,8044
<b>4</b>	29,7617	32,6002	0	0	0	0	62,3619
<b>5</b>	18,0073	29,8241	0	0	0	0	47,8314
<b>6</b>	21,378	30,6147	0	0	0	0	51,9927
<b>7</b>	8,0188	27,2894	0	0	0	0	35,3082
<b>8</b>	2,2225	16,6247	0	0	0	0	18,8472
<b>9</b>	21,5331	30,3728	13	0	0	0	64,9059
<b>10</b>	9,6482	27,5894	27	0	0	0	64,2376
<b>11</b>	24,4589	31,0015	29,4621	0	0	0	84,9225
<b>12</b>	20,106	29,8876	30,6789	0	0	0	80,6725

<i>Hour</i>	<b>Generator (MW)</b>						<b>Total</b>
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>(MW)</b>
<b>13</b>	0,6457	17,555	23,4741	0	0	0	41,6748
<b>14</b>	1,8932	20,204	23,6903	0	0	0	45,7875
<b>15</b>	0,9673	21,41	27,2765	0	0	0	49,6538
<b>16</b>	1,4602	29,2451	27,0919	0	0	0	57,7972
<b>17</b>	4,6769	10,2218	13,0919	0	0	0	27,9906
<b>18</b>	2,9213	0	27,0919	0	0	24,1075	54,1207
<b>19</b>	2,6066	21	14,2248	0	0	15,4577	53,2891
<b>20</b>	1,4389	15,8601	26,8443	0	0	16,5687	60,712
<b>21</b>	0,6354	23,4232	27,7819	0	0	24,6018	76,4423
<b>22</b>	21,0395	25,4171	28,3681	0	0	22,8029	97,6276
<b>23</b>	20,8855	33,0338	31,5143	0	0	25,9914	111,425
<b>24</b>	36,9425	28,7901	33,0474	0	0	27,2079	125,9879

##### 5. Tegangan Tiap Bus

Tabel 4.12 merupakan kondisi tegangan pada tiap bus selama pembebanan 24 jam.

**Tabel 4.12 Tegangan tiap bus pada jam 1 - 6**

<b>Bus</b>	<i>Hour</i>					
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>1</b>	1,04	1,04	1,04	1,04	1,04	1,04
<b>2</b>	1,0412	1,0411	1,0411	1,0409	1,0406	1,0407
<b>3</b>	1,0101	1,0136	1,014	1,0104	1,0078	1,0086
<b>4</b>	1,0012	1,0054	1,0059	1,0015	0,9983	0,9992
<b>5</b>	1,0226	1,0253	1,0256	1,0225	1,0203	1,021
<b>6</b>	0,9987	1,0041	1,0046	0,999	0,9952	0,9963
<b>7</b>	1,0028	1,0076	1,0081	1,003	0,9994	1,0004
<b>8</b>	0,9935	0,9994	1	0,9938	0,9895	0,9908
<b>9</b>	1,015	1,0236	1,0242	1,0151	1,0106	1,0119
<b>10</b>	1,0117	1,022	1,0226	1,0117	1,0069	1,0083

**Lanjutan Tabel 4.12**

Bus	Hour					
	1	2	3	4	5	6
11	1,015	1,0236	1,0242	1,0151	1,0106	1,0119
12	1,0225	1,0275	1,0281	1,0226	1,018	1,0194
13	1,0122	1,0151	1,0156	1,0121	1,0081	1,0093
14	1,0074	1,0133	1,014	1,0076	1,0021	1,0037
15	1,0035	1,0096	1,0103	1,0037	0,9982	0,9998
16	1,0126	1,0203	1,021	1,0128	1,0077	1,0092
17	1,0084	1,0182	1,0189	1,0085	1,0033	1,0048
18	0,9943	1,0029	1,0037	0,9947	0,9884	0,9902
19	0,9922	1,0019	1,0028	0,9926	0,9861	0,988
20	0,9963	1,0063	1,0071	0,9966	0,9905	0,9923
21	0,9978	1,01	1,0107	0,9979	0,9929	0,9943
22	0,9966	1,0092	1,0098	0,9966	0,9918	0,9932
23	0,9938	0,9981	0,9988	0,9939	0,9887	0,9902
24	0,9878	0,9984	0,9991	0,988	0,9824	0,9841
25	0,98	0,992	0,9926	0,9801	0,9751	0,9765
26	0,9664	0,9795	0,9803	0,9668	0,9607	0,9624
27	0,981	0,9934	0,994	0,981	0,9768	0,978
28	0,9934	0,9998	1,0003	0,9937	0,9897	0,9908
29	0,9585	0,9729	0,9737	0,9589	0,9528	0,9546
30	0,9435	0,9591	0,9601	0,9441	0,9367	0,9389

**Tabel 4.13 Tegangan tiap bus pada jam 7 - 12**

Bus	Hour					
	7	8	9	10	11	12
1	1,04	1,04	1,04	1,04	1,04	1,04
2	1,0403	1,0402	1,0406	1,0403	1,0407	1,0406
3	1,0056	1,0027	1,0108	1,0074	1,0105	1,0095
4	0,9955	0,992	1,0021	0,9978	1,0016	1,0003
5	1,0183	1,016	1,0205	1,0184	1,0213	1,0205
6	0,9919	0,9877	0,9971	0,9929	0,9977	0,9962
7	0,9961	0,992	0,9996	0,9961	1,0009	0,9995



**Lanjutan Tabel 4.13**

Bus	<i>Hour</i>					
	7	8	9	10	11	12
8	0,9857	0,981	0,9905	0,9862	0,9917	0,9901
9	1,0066	1,0016	1,0126	1,0077	1,0134	1,0117
10	1,0025	0,9972	1,009	1,0037	1,0099	1,0081
11	1,0066	1,0016	1,0126	1,0077	1,0134	1,0117
12	1,0139	1,0089	1,0329	1,0237	1,0283	1,0263
13	1,0045	1,0002	1,0274	1,018	1,021	1,0192
14	0,9971	0,991	1,0146	1,0053	1,0113	1,009
15	0,9932	0,9869	1,0077	0,9994	1,0057	1,0034
16	1,003	0,9973	1,0161	1,0085	1,0144	1,0123
17	0,9986	0,9927	1,0067	1,0006	1,0072	1,0052
18	0,9827	0,9755	0,9936	0,986	0,9937	0,9913
19	0,9802	0,9729	0,9892	0,9821	0,9903	0,9878
20	0,985	0,978	0,9932	0,9865	0,9943	0,992
21	0,9883	0,9827	0,9947	0,9893	0,9958	0,9939
22	0,9875	0,9821	0,9941	0,9887	0,9949	0,993
23	0,984	0,9782	0,9947	0,9877	0,994	0,9919
24	0,9774	0,9712	0,9847	0,9786	0,9857	0,9836
25	0,9705	0,9649	0,9749	0,9702	0,9768	0,975
26	0,9552	0,9483	0,9584	0,9534	0,9617	0,9595
27	0,973	0,9684	0,9764	0,9725	0,9781	0,9766
28	0,986	0,9815	0,991	0,9867	0,9919	0,9904
29	0,9473	0,9405	0,9486	0,9443	0,9527	0,9506
30	0,9301	0,9218	0,9299	0,9253	0,9357	0,9331

**Tabel 4.14 Tegangan tiap bus pada jam 13 - 18**

Bus	<i>Hour</i>					
	13	14	15	16	17	18
1	1,04	1,04	1,04	1,04	1,04	1,04
2	1,0401	1,0402	1,0401	1,0401	1,0403	1,0403
3	1,004	1,0047	1,0048	1,0061	1,0037	1,001
4	0,9936	0,9944	0,9945	0,9961	0,9933	0,99

**Lanjutan Tabel 4.14**

Bus	<i>Hour</i>					
	13	14	15	16	17	18
5	1,0154	1,0159	1,0161	1,0171	1,0145	1,0132
6	0,9878	0,9888	0,989	0,991	0,9865	0,9841
7	0,991	0,9919	0,9923	0,9942	0,9894	0,987
8	0,9802	0,9813	0,9817	0,984	0,9783	0,9759
9	1,0015	1,0026	1,003	1,0053	1,0005	0,9967
10	0,9969	0,9982	0,9987	1,0011	0,9962	0,9917
11	1,0015	1,0026	1,003	1,0053	1,0005	0,9967
12	1,0178	1,0188	1,0192	1,0214	1,0209	1,0112
13	1,0158	1,0165	1,0145	1,0162	1,0206	1,0104
14	0,9966	0,9979	0,9997	1,0025	0,9988	0,9885
15	0,9895	0,9909	0,9937	0,9965	0,9908	0,9821
16	1,0015	1,0027	1,0033	1,0059	1,0025	0,9947
17	0,9932	0,9946	0,9952	0,9979	0,9928	0,987
18	0,9755	0,9772	0,9794	0,9827	0,9753	0,9669
19	0,9718	0,9735	0,9753	0,9787	0,9708	0,9629
20	0,977	0,9786	0,9801	0,9833	0,976	0,9689
21	0,9825	0,9838	0,984	0,9866	0,9817	0,9773
22	0,9823	0,9836	0,9836	0,9861	0,9817	0,9776
23	0,9733	0,9746	0,9823	0,985	0,9737	0,9682
24	0,9686	0,9701	0,9728	0,9757	0,9662	0,9633
25	0,9622	0,9635	0,9648	0,9674	0,9546	0,9593
26	0,9437	0,9454	0,9469	0,9502	0,935	0,9391
27	0,9666	0,9678	0,968	0,9701	0,9564	0,9656
28	0,9811	0,9821	0,9825	0,9846	0,9787	0,9785
29	0,9353	0,937	0,9377	0,941	0,9229	0,9315
30	0,9143	0,9164	0,9174	0,9215	0,9005	0,9086

**Tabel 4.15 Tegangan tiap bus pada jam 19 - 24**

Bus	<i>Hour</i>					
	19	20	21	22	23	24
1	1,04	1,04	1,04	1,04	1,04	1,04

Bus	<i>Hour</i>					
	19	20	21	22	23	24
2	1,0401	1,0402	1,0401	1,0407	1,0406	1,0411
3	1,0057	1,0039	1,0048	1,0092	1,0101	1,0122
4	0,9957	0,9935	0,9946	1,0001	1,0012	1,0038
5	1,016	1,0153	1,0161	1,02	1,0209	1,023
6	0,9905	0,9887	0,9898	0,9963	0,9977	1,001
7	0,9921	0,9909	0,9924	0,9988	1,0004	1,0037
8	0,9828	0,9811	0,9826	0,99	0,9917	0,9954
9	1,0032	1,0012	1,003	1,0109	1,0127	1,0167
10	0,9982	0,996	0,9982	1,0068	1,0088	1,0131
11	1,0032	1,0012	1,003	1,0109	1,0127	1,0167
12	1,0213	1,0149	1,0167	1,0254	1,0261	1,0295
13	1,0208	1,0132	1,0142	1,0193	1,0192	1,0215
14	0,9987	0,9927	0,9955	1,0069	1,0085	1,0129
15	0,9917	0,9867	0,9893	1,0017	1,0035	1,008
16	1,0033	0,9989	1,0013	1,0107	1,0124	1,0166
17	0,9942	0,9915	0,994	1,0034	1,0055	1,0102
18	0,9757	0,972	0,9755	0,9883	0,991	0,9966
19	0,9712	0,9682	0,972	0,9845	0,9876	0,9934
20	0,9767	0,974	0,9774	0,9891	0,992	0,9975
21	0,984	0,9819	0,984	0,9927	0,9948	0,9993
22	0,9844	0,9821	0,984	0,9923	0,9941	0,9985
23	0,9768	0,9731	0,9748	0,9916	0,9932	0,9976
24	0,9732	0,9706	0,9721	0,9842	0,9861	0,9909
25	0,9749	0,9731	0,9716	0,9813	0,9821	0,9859
26	0,9551	0,9538	0,9531	0,9646	0,9663	0,9712
27	0,9842	0,9827	0,9791	0,9864	0,9862	0,9889
28	0,987	0,9852	0,9855	0,9926	0,9938	0,9971
29	0,9515	0,9507	0,9484	0,9589	0,9601	0,9647
30	0,9295	0,9291	0,9278	0,9404	0,9426	0,9485

#### 4.3.1.2 Kondisi Kontingensi

Pada kondisi kontingensi, jaringan transmisi yang diputus adalah yang menghubungkan bus 14 dan bus 15. Sedangkan

untuk parameter dan data yang lain tetap sama dengan kondisi normal.

1. Kombinasi Pembangkitan

Tabel 4.16 merupakan kombinasi pembangkitan selama 24 jam dalam kondisi kontingensi.

**Tabel 4.16 Kombinasi pembangkitan pada kondisi kontingensi**

<i>Hour</i>	<b>Gen1</b>	<b>Gen2</b>	<b>Gen3</b>	<b>Gen4</b>	<b>Gen5</b>	<b>Gen6</b>
<b>1</b>	1	1	0	0	0	0
<b>2</b>	1	1	0	0	0	0
<b>3</b>	1	1	0	0	0	0
<b>4</b>	1	1	0	0	0	0
<b>5</b>	1	1	0	0	0	0
<b>6</b>	1	1	0	0	0	0
<b>7</b>	1	1	0	0	0	0
<b>8</b>	1	1	0	0	0	0
<b>9</b>	1	1	1	0	0	0
<b>10</b>	1	1	1	0	0	0
<b>11</b>	1	1	1	0	0	0
<b>12</b>	1	1	1	0	0	0
<b>13</b>	1	1	1	0	0	0
<b>14</b>	1	1	1	0	0	0
<b>15</b>	1	1	1	0	0	0
<b>16</b>	1	1	1	0	0	0
<b>17</b>	1	1	1	0	0	1
<b>18</b>	1	1	1	0	0	1
<b>19</b>	1	1	1	0	0	1
<b>20</b>	1	1	1	0	0	1
<b>21</b>	1	1	1	0	0	1
<b>22</b>	1	1	1	0	0	1

**Lanjutan Tabel 4.16**

<i>Hour</i>	<b>Gen1</b>	<b>Gen2</b>	<b>Gen3</b>	<b>Gen4</b>	<b>Gen5</b>	<b>Gen6</b>
<b>23</b>	1	1	1	0	0	1
<b>24</b>	1	1	1	0	0	1

2. Biaya Pembangkitan Selama 24 Jam

Tabel 4.17 merupakan biaya pembangkitan daya untuk pembebanan 24 jam. Terdapat 6 generator pada sistem, dan pada kondisi awal hanya ada 1 generator menyala.

**Tabel 4.17 Biaya pembangkitan selama 24 jam pada kondisi kontingensi**

<i>Hour</i>	<i>Production Cost (\$)</i>	<i>Start-Up Cost (\$)</i>	<i>Shut Down Cost (\$)</i>	<i>Total Cost (\$)</i>
<b>1</b>	588,767	70	0	658,767
<b>2</b>	537,4704	0	0	537,4704
<b>3</b>	530,8178	0	0	530,8178
<b>4</b>	571,4381	0	0	571,4381
<b>5</b>	620,0755	0	0	620,0755
<b>6</b>	606,0138	0	0	606,0138
<b>7</b>	663,0644	0	0	663,0644
<b>8</b>	722,9555	0	0	722,9555
<b>9</b>	727,6756	113	0	840,6756
<b>10</b>	712,4437	0	0	712,4437
<b>11</b>	641,3517	0	0	641,3517
<b>12</b>	655,2998	0	0	655,2998
<b>13</b>	795,9878	0	0	795,9878
<b>14</b>	779,9465	0	0	779,9465
<b>15</b>	765,6993	0	0	765,6993
<b>16</b>	734,3648	0	0	734,3648
<b>17</b>	955,6721	0	0	955,6721

**Lanjutan Tabel 4.17**

<i>Hour</i>	<i>Production Cost (\$)</i>	<i>Start-Up Cost (\$)</i>	<i>Shut Down Cost (\$)</i>	<i>Total Cost (\$)</i>
<b>18</b>	782,6506	113	0	895,6506
<b>19</b>	909,4116	0	0	909,4116
<b>20</b>	870,0434	0	0	870,0434
<b>21</b>	808,8103	0	0	808,8103
<b>22</b>	735,7568	0	0	735,7568
<b>23</b>	686,4482	0	0	686,4482
<b>24</b>	638,0642	0	0	638,0642

**Tabel 4.18 Total biaya pembangkitan pada kondisi kontingensi**

<b>Total Biaya Pembangkitan 24 jam (\$)</b>	17336,23
<b>Biaya Produksi 24 jam (\$)</b>	17040,23
<b>Start Up Cost 24 jam (\$)</b>	296
<b>Start Down Cost 24 jam (\$)</b>	0

3. Pembangkitan Daya Tiap Generator

Tabel 4.19 merupakan pembangkitan daya aktif pada 6 generator selama 24 jam

**Tabel 4.19 Keluaran daya aktif pada tiap generator pada kondisi kontingensi**

<i>Hour</i>	<b>Gen1 (MW)</b>	<b>Gen2 (MW)</b>	<b>Gen3 (MW)</b>	<b>Gen4 (MW)</b>	<b>Gen5 (MW)</b>	<b>Gen6 (MW)</b>
<b>1</b>	160,3174	60,97	0	0	0	0
<b>2</b>	161,8723	45,3838	0	0	0	0
<b>3</b>	160,228	44,9701	0	0	0	0
<b>4</b>	170,2429	47,3998	0	0	0	0
<b>5</b>	181,9017	50,2686	0	0	0	0
<b>6</b>	178,5589	49,451	0	0	0	0

<i>Hour</i>	<b>Gen1 (MW)</b>	<b>Gen2 (MW)</b>	<b>Gen3 (MW)</b>	<b>Gen4 (MW)</b>	<b>Gen5 (MW)</b>	<b>Gen6 (MW)</b>
7	192,0447	52,6563	0	0	0	0
8	199,6552	61,589	0	0	0	0
9	178,2668	49,4723	37	0	0	0
10	190,3667	52,6655	22,7528	0	0	0
11	175,1726	49,5815	20,326	0	0	0
12	178,0681	50,0326	21,0912	0	0	0
13	199,4573	61,5446	27,3039	0	0	0
14	198,2752	61,2549	24,7691	0	0	0
15	192,8697	63,8222	23,3671	0	0	0
16	196,004	54,1278	22,0121	0	0	0
17	194,7031	70,4373	20,92	0	0	15,5627
18	198,3889	69,3033	21,4986	0	0	26,1539
19	192,7399	80	24,0758	0	0	20,578
20	199,1685	59	27,2936	0	0	23,6495
21	199,0515	52,79	26,1912	0	0	15,3421
22	187,7495	46,2654	23,5798	0	0	15,1844
23	176,1799	44,0296	21,9606	0	0	16,0555
24	161,0861	50,2916	19,738	0	0	12,7255

4. Cadangan Berputar Tiap Generator Selama 24 jam

Tabel 4.20 merupakan cadangan berputar pada tiap generator yang menyala selama 24 jam.

**Tabel 4.20 Cadangan berputar tiap generator pada kondisi kontingensi**

<i>Hour</i>	<b>Generator (MW)</b>						<b>Total (MW)</b>
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	
<b>1</b>	39,6826	19,03	0	0	0	0	58,7126
<b>2</b>	38,1277	34,6162	0	0	0	0	72,7439
<b>3</b>	39,772	35,0299	0	0	0	0	74,8019

<i>Hour</i>	Generator (MW)						Total (MW)
	1	2	3	4	5	6	
4	29,7571	32,6002	0	0	0	0	62,3573
5	18,0983	29,7314	0	0	0	0	47,8297
6	21,4411	30,549	0	0	0	0	51,9901
7	7,9553	27,3437	0	0	0	0	35,299
8	0,3448	18,411	0	0	0	0	18,7558
9	21,7332	30,5277	13	0	0	0	65,2609
10	9,6333	27,3345	27,2472	0	0	0	64,215
11	24,8274	30,4185	29,674	0	0	0	84,9199
12	21,9319	29,9674	28,9088	0	0	0	80,8081
13	0,5427	18,4554	22,6961	0	0	0	41,6942
14	1,7248	18,7451	25,2309	0	0	0	45,7008
15	7,1303	16,1778	26,6329	0	0	0	49,941
16	3,996	25,8722	27,9879	0	0	0	57,8561
17	5,2969	9,5627	29,08	0	0	0	43,9396
18	1,6111	10,6967	28,5014	0	0	24,4	65,2465
19	7,2601	0	25,9242	0	0	13,8	47,0304
20	0,8315	21	22,7064	0	0	19,4	63,9599
21	0,9485	27,21	23,8088	0	0	16,4	68,3178
22	12,2505	33,7346	26,4202	0	0	24,7	97,0632
23	23,8201	35,9704	28,0394	0	0	24,8	112,6455
24	38,9139	29,7084	30,262	0	0	23,9	122,8288

## 5. Tegangan Tiap Bus

Tabel 4.21 merupakan kondisi tegangan pada tiap bus selama pembebanan 24 jam.



**Tabel 4.21 Tegangan tiap bus kondisi kontingensi pada jam 1 - 6**

Bus	Hour					
	1	2	3	4	5	6
1	1,04	1,04	1,04	1,04	1,04	1,04
2	1,0412	1,0411	1,0411	1,0409	1,0406	1,0407
3	1,0102	1,0137	1,014	1,0104	1,0079	1,0086
4	1,0013	1,0055	1,0059	1,0015	0,9983	0,9993
5	1,0226	1,0253	1,0256	1,0226	1,0203	1,021
6	0,9987	1,0041	1,0046	0,999	0,9952	0,9963
7	1,0028	1,0076	1,0081	1,003	0,9994	1,0004
8	0,9935	0,9994	1	0,9938	0,9896	0,9908
9	1,0149	1,0236	1,0242	1,0151	1,0106	1,0119
10	1,0116	1,022	1,0226	1,0117	1,0069	1,0083
11	1,0149	1,0236	1,0242	1,0151	1,0106	1,0119
12	1,0228	1,0279	1,0285	1,0229	1,0184	1,0197
13	1,0123	1,0151	1,0156	1,0122	1,0082	1,0094
14	1,0106	1,0167	1,0174	1,0109	1,0054	1,007
15	1,0027	1,0088	1,0096	1,003	0,9974	0,999
16	1,0128	1,0205	1,0211	1,0129	1,0078	1,0093
17	1,0084	1,0182	1,0189	1,0085	1,0033	1,0048
18	0,9938	1,0024	1,0032	0,9942	0,9878	0,9897
19	0,9918	1,0016	1,0024	0,9922	0,9857	0,9876
20	0,996	1,006	1,0068	0,9964	0,9902	0,992
21	0,9977	1,01	1,0106	0,9978	0,9928	0,9943
22	0,9965	1,0091	1,0097	0,9965	0,9918	0,9932
23	0,9935	0,9979	0,9986	0,9936	0,9884	0,9899
24	0,9876	0,9982	0,999	0,9878	0,9823	0,9839
25	0,9799	0,9919	0,9925	0,98	0,975	0,9764
26	0,9663	0,9794	0,9802	0,9667	0,9606	0,9624
27	0,981	0,9934	0,9939	0,9809	0,9768	0,978
28	0,9934	0,9998	1,0003	0,9937	0,9897	0,9908
29	0,9585	0,9728	0,9736	0,9588	0,9528	0,9545
30	0,9434	0,9591	0,9601	0,944	0,9367	0,9388

**Tabel 4.22 Tegangan tiap bus kondisi kontingensi pada jam 7 - 12**

Bus	Hour					
	7	8	9	10	11	12
1	1,04	1,04	1,04	1,04	1,04	1,04
2	1,0403	1,0402	1,0406	1,0403	1,0407	1,0407
3	1,0056	1,0027	1,0109	1,0074	1,0105	1,0099
4	0,9955	0,9919	1,0021	0,9977	1,0016	1,0009
5	1,0183	1,0159	1,0205	1,0183	1,0213	1,0207
6	0,9919	0,9876	0,9971	0,9928	0,9976	0,9968
7	0,9961	0,9919	0,9996	0,996	1,0009	1
8	0,9857	0,9808	0,9905	0,9861	0,9917	0,9907
9	1,0065	1,0014	1,0125	1,0076	1,0133	1,0123
10	1,0025	0,9969	1,0089	1,0035	1,0097	1,0086
11	1,0065	1,0014	1,0125	1,0076	1,0133	1,0123
12	1,0142	1,0089	1,0333	1,0239	1,0285	1,0278
13	1,0046	1,0001	1,0275	1,018	1,021	1,0206
14	1,0003	0,9938	1,0191	1,0092	1,0153	1,0143
15	0,9924	0,986	1,0065	0,9982	1,0046	1,0035
16	1,0032	0,9972	1,0162	1,0085	1,0145	1,0135
17	0,9985	0,9925	1,0066	1,0005	1,0071	1,0059
18	0,9821	0,9748	0,9928	0,9851	0,9929	0,9916
19	0,9798	0,9723	0,9886	0,9814	0,9897	0,9882
20	0,9846	0,9775	0,9927	0,986	0,9939	0,9925
21	0,9882	0,9824	0,9946	0,9891	0,9956	0,9944
22	0,9874	0,9818	0,9939	0,9885	0,9947	0,9936
23	0,9836	0,9776	0,9942	0,9871	0,9935	0,9924
24	0,9772	0,9707	0,9844	0,9782	0,9854	0,9842
25	0,9704	0,9646	0,9747	0,9699	0,9767	0,9754
26	0,955	0,948	0,9582	0,9532	0,9615	0,96
27	0,9729	0,9681	0,9763	0,9723	0,978	0,977
28	0,986	0,9813	0,991	0,9866	0,9919	0,9909
29	0,9472	0,9402	0,9485	0,9441	0,9526	0,951
30	0,93	0,9214	0,9298	0,9252	0,9356	0,9336

**Tabel 4.23 Tegangan tiap bus kondisi kontingensi pada jam 13 - 18**

Bus	<i>Hour</i>					
	13	14	15	16	17	18
1	1,04	1,04	1,04	1,04	1,04	1,04
2	1,0401	1,0402	1,0403	1,0402	1,0403	1,0402
3	1,0042	1,0044	1,0049	1,0058	1,003	1,0031
4	0,9939	0,994	0,9947	0,9957	0,9924	0,9925
5	1,0154	1,0157	1,0164	1,017	1,0148	1,0147
6	0,988	0,9883	0,9893	0,9906	0,987	0,9876
7	0,9911	0,9916	0,9926	0,9939	0,9899	0,9898
8	0,9804	0,9809	0,982	0,9836	0,9793	0,9799
9	1,0016	1,0021	1,0033	1,0047	1,0002	0,9997
10	0,997	0,9975	0,9989	1,0004	0,9954	0,9943
11	1,0016	1,0021	1,0033	1,0047	1,0002	0,9997
12	1,0187	1,0182	1,0189	1,0198	1,0143	1,0131
13	1,0165	1,0155	1,0157	1,016	1,0121	1,0113
14	1,0025	1,0023	1,0032	1,0046	0,9971	0,995
15	0,9886	0,9887	0,9898	0,9914	0,9849	0,9833
16	1,002	1,0021	1,0032	1,0046	0,9985	0,997
17	0,9935	0,9939	0,9953	0,997	0,9911	0,9895
18	0,975	0,9754	0,977	0,9791	0,971	0,9687
19	0,9714	0,9721	0,9737	0,976	0,9676	0,9651
20	0,9767	0,9774	0,979	0,9811	0,9734	0,9712
21	0,9826	0,9831	0,9845	0,9862	0,9812	0,9801
22	0,9823	0,9828	0,9842	0,9857	0,9813	0,9804
23	0,973	0,9733	0,9745	0,976	0,9715	0,9709
24	0,9685	0,9691	0,9706	0,9725	0,9673	0,9689
25	0,9622	0,9629	0,9644	0,9662	0,9629	0,9727
26	0,9437	0,9447	0,9465	0,9489	0,9437	0,953
27	0,9666	0,9673	0,9686	0,97	0,9682	0,9833
28	0,9812	0,9817	0,9828	0,9842	0,9817	0,9844
29	0,9353	0,9365	0,9384	0,9409	0,9361	0,9507
30	0,9143	0,9158	0,9182	0,9214	0,9146	0,9287

**Tabel 4.24 Tegangan tiap bus kondisi kontingensi pada jam 19 - 24**

Bus	Hour					
	19	20	21	22	23	24
1	1,04	1,04	1,04	1,04	1,04	1,04
2	1,0404	1,0401	1,0401	1,0404	1,0407	1,0411
3	1,0021	1,0049	1,0058	1,0087	1,0113	1,0129
4	0,9913	0,9947	0,9958	0,9994	1,0027	1,0047
5	1,0139	1,0158	1,0166	1,0194	1,0217	1,0234
6	0,9856	0,9898	0,9909	0,9954	0,9993	1,0017
7	0,9882	0,9918	0,9933	0,9979	1,0017	1,0043
8	0,9776	0,9823	0,9836	0,9889	0,9934	0,9962
9	0,9983	1,0024	1,0041	1,0096	1,0144	1,0175
10	0,9932	0,9973	0,9993	1,0053	1,0104	1,014
11	0,9983	1,0024	1,0041	1,0096	1,0144	1,0175
12	1,0134	1,0181	1,0198	1,0241	1,0291	1,0316
13	1,0123	1,0163	1,0172	1,0198	1,022	1,0235
14	0,9952	1,0006	1,0033	1,009	1,0152	1,0185
15	0,9828	0,9879	0,9903	0,9964	1,0051	1,0086
16	0,9966	1,0012	1,0035	1,0092	1,0148	1,0182
17	0,9886	0,993	0,9955	1,0019	1,0075	1,0112
18	0,9678	0,9733	0,9767	0,9842	0,9928	0,9973
19	0,964	0,9695	0,9731	0,9812	0,9893	0,9942
20	0,9701	0,9753	0,9786	0,9862	0,9937	0,9983
21	0,9789	0,9831	0,9852	0,9915	0,9965	1,0001
22	0,9793	0,9833	0,9851	0,9911	0,9958	0,9993
23	0,9698	0,9745	0,9761	0,982	0,9951	0,9984
24	0,9654	0,9719	0,9733	0,98	0,9881	0,9917
25	0,9629	0,9741	0,9724	0,9783	0,9843	0,9865
26	0,9427	0,9548	0,954	0,9615	0,9686	0,9718
27	0,9698	0,9834	0,9797	0,9843	0,9884	0,9894
28	0,9808	0,9863	0,9865	0,9914	0,9956	0,9978
29	0,9361	0,9515	0,9491	0,9565	0,9626	0,9652
30	0,9134	0,93	0,9285	0,9379	0,9454	0,949

#### 4.3.2 Hasil Simulasi Program SCUC Dengan Metode Quadratic Programming

Program SCUC dengan metode quadratic programming digunakan sebagai pembandingan untuk validasi hasil dari program SCUC dengan metode BPSO. Data sistem yang digunakan adalah data dalam kondisi normal (tidak ada kontingensi). Tetapi, *cold start cost* tidak diperhatikan pada metode ini.

##### 1. Kombinasi Pembangkitan

Tabel 4.25 merupakan kombinasi pembangkitan selama 24 jam dan biaya pembangkitan tiap jam.

**Tabel 4.25 Kombinasi pembangkitan dengan metode Quadratic Programming**

<i>Hour</i>	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6
1	1	1	1	0	0	0
2	1	1	1	0	0	0
3	1	1	1	1	0	0
4	1	1	1	1	0	0
5	1	1	1	1	0	0
6	1	1	1	1	0	0
7	1	1	1	1	0	0
8	1	1	1	1	0	0
9	1	1	1	1	0	0
10	1	1	1	1	0	0
11	1	1	1	1	0	0
12	1	1	1	1	0	0
13	1	1	1	1	0	0
14	1	1	1	1	0	0
15	1	1	1	1	0	0
16	1	1	1	1	0	0

**Lanjutan Tabel 4.25**

<i>Hour</i>	<i>Gen1</i>	<i>Gen2</i>	<i>Gen3</i>	<i>Gen4</i>	<i>Gen5</i>	<i>Gen6</i>
<b>17</b>	1	1	1	1	0	0
<b>18</b>	1	1	1	1	0	0
<b>19</b>	1	1	1	1	0	0
<b>20</b>	1	1	1	1	0	0
<b>21</b>	1	1	1	1	0	0
<b>22</b>	1	1	1	1	0	0
<b>23</b>	1	1	1	1	0	0
<b>24</b>	1	1	1	1	0	0

2. Biaya Pembangkitan Selama 24 Jam

Tabel 4.26 merupakan biaya pembangkitan untuk pembebanan selama 24 jam. Kondisi awal pembangkit hanya 1 generator yang menyala yaitu generator 2.

**Tabel 4.26 Biaya pembangkitan selama 24 jam**

<i>Hour</i>	<i>Production Cost (\$)</i>	<i>Start-Up Cost (\$)</i>	<i>Shut Down Cost (\$)</i>	<i>Total Cost (\$)</i>
<b>1</b>	563	180	0	743
<b>2</b>	520,9	0	0	520,9
<b>3</b>	517,4	0	0	517,4
<b>4</b>	558,5	50	0	608,5
<b>5</b>	600,77	0	0	600,77
<b>6</b>	588,5	0	0	588,5
<b>7</b>	643,9	0	0	643,9
<b>8</b>	700,9	0	0	700,9
<b>9</b>	707,3	0	0	707,3
<b>10</b>	713,8	0	0	713,8

**Lanjutan Tabel 4.26**

<i>Hour</i>	<i>Production Cost (\$)</i>	<i>Start-Up Cost (\$)</i>	<i>Shut Down Cost (\$)</i>	<i>Total Cost (\$)</i>
<b>11</b>	643,9	0	0	643,9
<b>12</b>	656,4	0	0	656,4
<b>13</b>	792,6	0	0	792,6
<b>14</b>	779,3	0	0	779,3
<b>15</b>	766,1	0	0	766,1
<b>16</b>	733,3	0	0	733,3
<b>17</b>	840,5	0	0	840,5
<b>18</b>	890,2	0	0	890,2
<b>19</b>	897,5	0	0	897,5
<b>20</b>	868,7	0	0	868,7
<b>21</b>	812,9	0	0	812,9
<b>22</b>	733,3	0	0	733,3
<b>23</b>	688,1	0	0	688,1
<b>24</b>	637,6	0	0	637,6

**Tabel 4.27 Total biaya pembangkitan**

<b>Total Biaya Pembangkitan 24 jam (\$)</b>	17085,37
<i>Biaya Produksi 24 jam (\$)</i>	16855,37
<i>Start Up Cost 24 jam (\$)</i>	230
<i>Start Down Cost 24 jam (\$)</i>	0

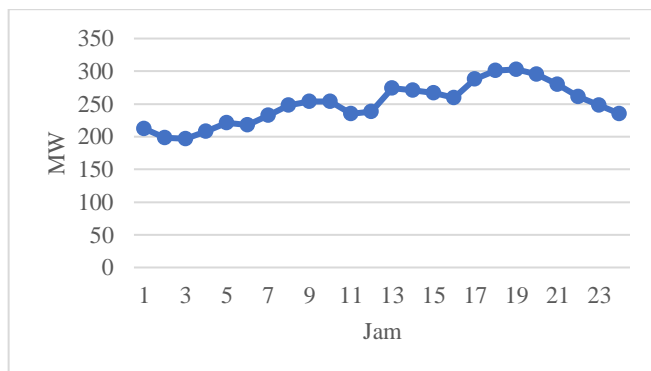
#### 4.4 Analisa Hasil Simulasi

Pada sistem IEEE 30 bus terdapat 6 generator thermal dan salah satunya berfungsi sebagai *slack / swing*. Pada kondisi awal hanya ada 1 generator yang sudah menyala selama 3 jam yaitu generator 2, sedangkan 5 generator lainnya dalam kondisi padam. Generator 1 telah padam selama 4 jam, generator 3 dan 4 telah padam selama 3 jam, dan generator 5 dan 6 telah padam selama 6 jam.

Perhitungan SCUC dilakukan dengan 5 iterasi dan 50 partikel untuk mendapatkan total biaya pembangkitan yang paling ekonomis. Dari hasil simulasi yang telah diperoleh didapatkan beberapa parameter yang dapat di analisa, yaitu:

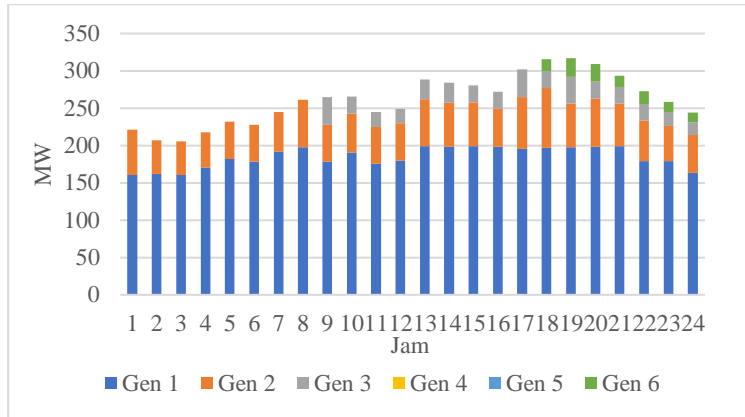
*a. Power Output*

Seperti yang sudah dijelaskan pada bab sebelumnya bahwa pembangkitan yang di perhatikan hanya pembangkitan daya aktif, sedangkan beban Q diabaikan. Grafik 4.2 merupakan beban selama 24 jam dimana beban tersebut termasuk beban rumah tangga. Grafik 4.3 merupakan pembangkitan selama 24 jam dimana hanya 4 generator yang menyala untuk memenuhi kebutuhan beban. Grafik 4.4 adalah grafik total pembangkitan daya aktif selama 24 jam yang dibangkitkan untuk memenuhi rugi-rugi daya dan total beban 24 jam.



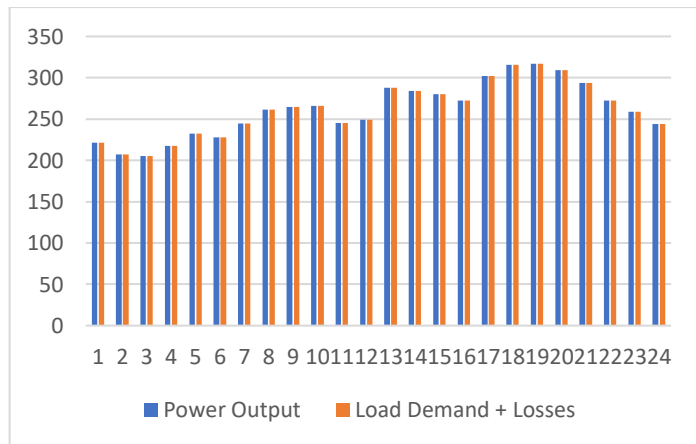
**Gambar 4.2 Kurva Pembebanan 24 Jam**





**Gambar 4.3 Pembangkitan Daya Tiap Generator**

Dari gambar 4.3 dapat dilihat bahwa hanya 4 generator yang nyala yaitu generator 1, 2, 3, dan 6. Generator 1 dan 2 mulai nyala sejak jam 1 sampai jam 24. Generator 3 mulai nyala sejak jam 9 sampai jam 24. Sedangkan, generator 6 nyala mulai dari jam 18 sampai jam 24.



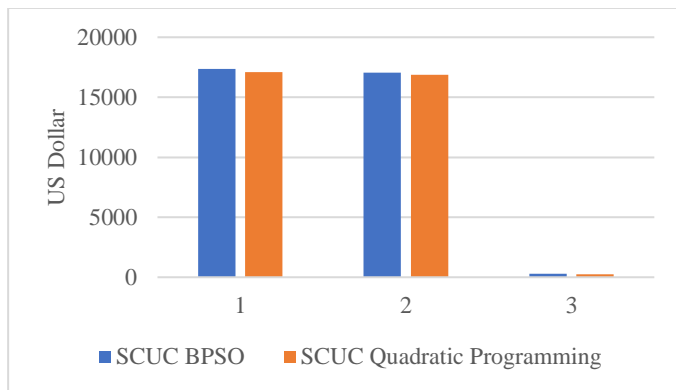
**Gambar 4.4 Grafik pembangkitan daya**

Dari gambar 4.4 dapat dilihat bahwa pembangkitan daya sudah memenuhi kebutuhan beban selama 24 jam dan total rugi-rugi jaringan transmisi. Hal tersebut membuktikan bahwa program SCUC dengan BPSO menghasilkan hasil yang benar.

b. Biaya Pembangkitan

Biaya pembangkitan merupakan penambahan dari biaya operasi, biaya penyalan generator, dan biaya pemadaman generator. Perhitungan biaya pembangkitan dilakukan oleh 2 program yaitu program utama SCUC dengan metode BPSO dan program pembandingan SCUC dengan metode Quadratic Programming.

Grafik dibawah merupakan hasil perhitungan biaya pembangkitan menggunakan metode BPSO dan Quadratic Programming.



**Gambar 4.5 Grafik batang perbandingan hasil BPSO dan Quadratic Programming**

dimana pada x axis,

1 = Total biaya pembangkitan selama 24 jam

2 = Biaya operasi selama 24 jam

3 = Biaya penyalan 24 jam

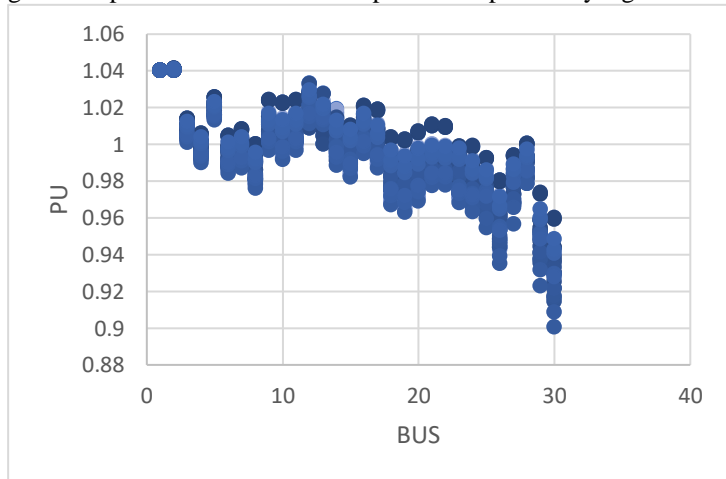
Total biaya pembangkitan selama 24 dengan metode BPSO didapatkan \$17360.7, sedangkan dengan metode Quadratic Programming didapatkan \$17085.37. Dari hasil didapati metode BPSO lebih \$275.33 dibandingkan metode Quadratic Programming. Selisih tersebut dikarenakan metode Quadratic Programming tidak mempertimbangkan *cold start cost*. Tetapi, selisih yang kecil pada total biaya pembangkitan dengan metode BPSO yaitu sebesar 1.611% dari total biaya pembangkitan pada metode quadratic programming membuktikan bahwa program SCUC dengan metode BPSO sudah benar dan sesuai.

c. Pengaruh Kontingensi Pada Tegangan Tiap Bus

Grafik 4.4 merupakan persebaran nilai tegangan pada 30 bus selama 24 jam dalam kondisi normal.

**Gambar 4.6 Tegangan tiap bus pada kondisi normal**

Titik-Titik diatas mewakili nilai tegangan pada tiap bus dan dari grafik dapat dilihat bahwa terdapat beberapa titik yang melewati



batas standar tegangan operasi (0.95 – 1.05 pu).

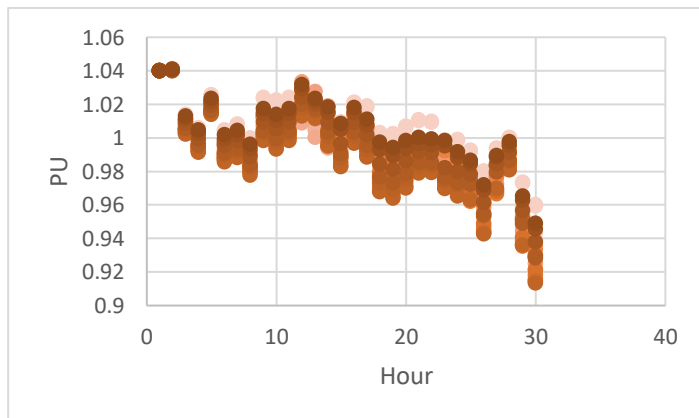
Tabel 4.28 menunjukkan nilai tegangan pada tiap bus yang melewati batas standar tegangan operasi.

**Tabel 4.28 Bus pada kondisi *under voltage***

<i>Hour</i>	<b>Bus</b>
13,14,15, 17, dan 18.	26
7 - 10, 13 - 18, dan 21	29
1, dan 3 - 24	30

Kondisi pada tiap bus yang disebutkan pada tabe 4.28 adalah kondisi *under voltage* dimana nilai tegangan dibawah 0.95 pu.

Grafik 4.5 merupakan persebaran nilai tegangan pada 30 bus selama 24 jam dalam kondisi kontingensi. Jaringan transmisi yang terputus adalah jaringan yang menghubungkan bus 14 dan 15.



**Gambar 4.7 Tegangan tiap bus pada kondisi kontingensi**

Dari grafik 4.5, didapati beberapa titik yang berada diluar dari batas standar tegangan operasi. Tabel 4.29 merupakan daftar bus mana saja yang melanggar standar tegangan batas operasi.

**Tabel 4.29 Bus pada kondisi *under voltage***

<i>Hour</i>	<b>Bus</b>
8, 13- 17, dan 19	26
7 - 10, 13 - 17, 19 dan 22	29
1, dan 4 - 30	30

Bus yang telah disebutkan pada table diatas mengalami kondisi under voltage dimana nilai tegangan dibawah 0.95 pu.

Dari tabel 4.12 – 4.15 dan tabel 4.21 – 4.24 didapati bahwa pada kondisi kontingensi nilai tegangan pada bus dengan kondisi under voltage yang telah disebutkan pada tabel 4.28 cenderung lebih rendah saat kondisi kontingensi dibandingkan dengan pada kondisi normal. Tetapi, ada beberapa bus yang tegangannya cenderung sama. Bus-bus tersebut disebutkan pada tabel 4.30.

**Tabel 4.30 Bus dengan nilai tegangan cenderung tetap**

<i>Hour</i>	<b>Bus</b>
17, 18, 21	29
17, 18, 20, 21, 23, 24	30

*Halaman ini sengaja dikosongkan*

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Pada tugas akhir yang telah saya lakukan, dapat ditarik kesimpulan yaitu:

1. Generator yang nyala adalah generator 1, generator 2, generator 3 dan generator 6 selama 24 jam pada program SCUC dengan metode BPSO.
2. Daya aktif yang dibangkitkan oleh 4 generator nyala sudah sesuai dengan jumlah pembebanan dan rugi-rugi saluran. Hal ini membuktikan bahwa program SCUC dengan metode BPSO sudah benar dan sesuai.
3. Tidak ada pelanggaran ramp-rate, kapasitas maksimum dan minimum pembangkitan, dan minimum up and down time pada program simulasi SCUC.
4. Total biaya pembangkitan SCUC dengan metode BPSO adalah \$17360.7, sedangkan total biaya pembangkitan dengan metode Quadratic Programming adalah \$17085.37. Selisih total biaya pembangkitan pada SCUC dengan metode BPSO dan metode Quadratic Programming tersebut sebesar \$275.3 atau 1.611% dari hasil biaya pembangkitan dengan metode Quadratic Programming. Hal tersebut dikarenakan SCUC dengan metode Quadratic Programming tidak mempertimbangkan *cold start cost*. Tetapi, selisih 1.611% termasuk angka yang kecil dan hal tersebut membuktikan bahwa program SCUC dengan metode BPSO sudah benar dan sesuai.
5. Tegangan tiap bus pada kondisi kontingensi cenderung lebih rendah dibandingkan pada kondisi normal. Tetapi, ada beberapa bus yang nilai tegangannya tetap.

## 5.2 Saran

Saran untuk penelitian kedepan adalah:

1. Menggunakan metode selain BPSO pada studi SCUC seperti metode Genetic Algorithm (GE), Priority List (PL), Dynamic Programming (DP), Lagrangian Relaxation (LR), Evolutionary Programming (EP), Simulated Annealing (SA), dan lain-lain.
2. Memperhitungkan batasan lain seperti contohnya batasan cadangan berputar. Dimana total cadangan berputar harus lebih besar dari kapasitas generator terbesar.
3. Mengimplementasikan studi SCUC pada pembangkit *renewable energy* seperti *hydro generator*, *wind turbine generator*, pembangkit listrik tenaga panas bumi, dan lain-lain.



## DAFTAR PUSTAKA

- [1] R. S. Wibowo, F. F. Utama, D. F. U. Putra, and N. K. Aryani, "Unit commitment with non-smooth generation cost function using binary particle swarm optimization," in *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Lombok, Indonesia, 2016, pp. 571–576.
- [2] R. S. Wibowo, K. R. Firmansyah, N. K. Aryani, and A. Soeprijanto, "Dynamic economic dispatch of hybrid microgrid with energy storage using quadratic programming," in *2016 IEEE Region 10 Conference (TENCON)*, Singapore, 2016, pp. 667–670.
- [3] R. Collett and J. Quaicoe, "Security-Constrained Unit Commitment using Particle Swarms," in *2006 Canadian Conference on Electrical and Computer Engineering*, Ottawa, Ont., 2006, pp. 1125–1129.
- [4] W. Ge, "Ramp Rate Constrained Unit Commitment by Improved Priority List and Enhanced Particle Swarm Optimization," in *2010 International Conference on Computational Intelligence and Software Engineering*, Wuhan, China, 2010, pp. 1–8.
- [5] Yuan Xiaohui, Yuan Yanbin, Wang Cheng, and Zhang Xiaopan, "An Improved PSO Approach for Profit-based Unit Commitment in Electricity Market," in *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific*, Dalian, China, 2005, pp. 1–4.
- [6] B. E. Türkay and Y. Zeybeko, "An enhanced particle swarm optimization approach for the unit commitment problem," p. 5.
- [7] A. Bikeri, P. Kihato, and C. Maina, "Profit based unit commitment using evolutionary particle swarm optimization," in *2017 IEEE AFRICON*, Cape Town, 2017, pp. 1137–1142.

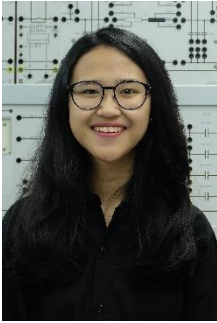
*Halaman ini sengaja dikosongkan*

## INDEX

- Binary particle swarm optimization, 27
- BPSO, ix, xi, xvi, xvii, 24, 25, 26, 38, 39, 40, 43, 44, 45, 48, 49, 51, 56, 75, 80, 81, 85, 86
- Dynamic Programming, 23, 37, 38, 86
- Economic Dispatch, xv, 33, 38
- Evolutionary Programming, 24, 38, 86
- Fitness populasi, 44
- IEEE, ix, vii, ix, xi, xiii, xvi, xvii, xxi, 24, 25, 27, 39, 51, 53, 54, 56, 78, 87
- Inialisasi, 44, 48
- Lagrangian Relaxation, 23, 38, 86
- Matlab, 25, 56
- minimum up and down, ix, xi, 24, 25, 85
- optimal power flow, 27
- optimasi, 27, 32, 38, 39, 46
- Particle Swarm Optimization, ix, xi, xiii, xv, xvi, 24, 25, 26, 27, 37, 38, 43, 46, 51, 87, 120
- Partikel, 40, 44, 45, 48, 49, 118, 120, 127, 128
- power balance, ix, xi
- Priority List, 23, 38, 86, 87
- Production Cost, 58, 67, 76
- ramp rate, ix, xi, 24, 41, 97, 111, 118, 123, 126
- random, 39, 41, 44, 48, 98
- SCUC, xvi, xvii, 43, 51, 56, 75, 78, 80, 81, 85, 86
- Shut Down, 37, 52, 58, 67, 76, 107
- Simulated Annealing, 24, 38, 86
- Software, 56, 87
- spinning reverse, ix, xi
- start up cost, ix, 37, 38, 46, 134
- Start-Up Cost, 39, 58, 67, 76
- Total Cost, 58, 67, 76
- unit commitment, ix, xi, 23, 24, 25, 26, 27, 34, 38, 46, 51, 87
- Unit Commitment, ix, xi, xiii, xv, xvi, xix, 25, 26, 27, 29, 34, 43, 44, 46, 87, 93

*Halaman ini sengaja dikosongkan*

## RIWAYAT HIDUP



Elpha Aulia Arifin, putri ketiga dari Syamsul Arifin dan Aulia Siti Aisjah yang lahir pada 21 November 1997 di Surabaya. Penulis menyelesaikan Pendidikan formal di SD Muhammadiyah 4 Surabaya pada tahun 2009, SMP Muhammadiyah 5 Surabaya pada tahun 2012, dan SMAN 2 Surabaya pada tahun 2015. Kemudian melanjutkan studi S1 Teknik Elektro di Institut Teknologi Sepuluh Nopember (ITS) pada tahun 2015. Penulis mengambil bidang studi Teknik Sistem Tenaga. Penulis aktif pada beberapa kegiatan organisasi yaitu sebagai staff Departemen Lingkar Kampus Himpunan Mahasiswa Elektro pada tahun 2016 dan melanjutkan menjadi staff ahli di tahun berikutnya. Pada tahun 2018 penulis aktif sebagai Sekertaris Departemen Dalam Negeri BEM Fakultas Teknologi Elektro. Penulis juga aktif sebagai asisten di Laboratorium Simulasi Sistem Tenaga Listrik Teknik Elektro ITS. Penulis dapat dihubungi melalui alamat email: [elva.aulia@gmail.com](mailto:elva.aulia@gmail.com)

*Halaman ini sengaja dikosongkan*

## LAMPIRAN

Program Unit Commitment :

```
clear all
clc
tic

%UNIT COMMITMENT BY PSO
%ELPHA AULIA ARIFIN

%DATA YANG DIGUNAKAN

%      No. a      b      c      Pmin      Pmaks
MU MD HSC      CSC      CSH      SDC      RU      RD      IS
Data = [ 1 0.00375      2      0      50      200
4  4  70      176      5      50      80      80      -4;
      2 0.01750      1.75      0      20      80
3  3  74      187      5      60      22      22      3;
      3 0.06250      1      0      15      50
2  3  110      113      4      30      15      15      -3;
      4 0.00834      3.25      0      10      35
2  2  50      267      4      85      16      16      -3;
      5 0.02500      3      0      10      30
2  2  72      180      4      52      9      9      -2;
      6 0.02500      3      0      12      40
2  2  40      113      2      30      16      16      -2;]
```

```
IS = Data(:,15); %Buat variable IS dan copy nilai
IS pada kolom data.
CIS =IS; %Copy nilai IS ke CIS.
fullcheckIS = CIS; %Copy nilai CIS ke fullcheckIS.
Ton = IS; %Copy nilai IS ke Ton.
Toff = IS; %Copy nilai IS ke Toff.
IS(IS<0)=0; %Set niali IS jika < 0 menjadi 0.
IS(IS>0)=1; %Set niali IS jika >0 menjadi 1.
MU = Data(:,7); %Buat variable MU dan copy nilai
MU pada kolom data.
MD = Data(:,8); %Buat variable MU dan copy nilai
MU pada kolom data.
```

```

rurd = Data(:,13);
rurdd = rurd';
Pmin = Data(:,5);
Pmaks = Data(:,6);

%Tambahan=====
multiplierPd24 = [1.12 1.05 1.04 1.1 1.17 1.15 1.23
1.31 1.34 1.34 1.24 1.26 1.45 1.43 1.41 1.37 1.52 1.59
1.6 1.56 1.48 1.38 1.31 1.24;]; %Set nilai pengali
load.
totalBiaya24(1:24) = 1000000000; %Set nilai total
biaya dari jam 1 sampai 24 dengan nilai maks.
rekap124(1:75,1:6) = 0; %Set nilai rekap1
untuk 24 jam dengan nilai 0.(kombinasi generator
paling optimal)
rekap224(1:75,1:6) = 0; %Set nilai rekap2
untuk 24 jam dengan nilai 0.(biaya dari kombinasi
paling optimal)
rekap324(1:75,1:6) = 0;
rekap424(1:75,1:6) = 0;
rekap524(1:75,1:6) = 0;

rekapkombinasi1 = [];
rekapkombinasi2 = [];
rekapkombinasi3 = [];

rekapPembangkitan1 = [];
rekapPembangkitan2 = [];
rekapPembangkitan3 = [];

rekapramprate1 = [];
rekapramprate2 = [];
rekapramprate3 = [];
state = [];
statee = [];
stateee = [];
periterasi = [];
abcdef = 0;

rekapIndex = 1; %Inisialisasi
nilai rekapIndex dengan nilai 1.
rekapbaris = 1;

```



```

baris = 1;

for hour=1:24
    multiplierPd = multiplierPd24(1, hour);

    % ----- Setting parameters of PSO -----
    -----
    dim = 6; % Number of
control variables
    N = 50; % Number of
particles
    maxit = 5; % Maximum
iterations
    rhomax = 0.9;
    rhomin = 0.4;

    Load = [189.2 189.2 189.2]; %Inisialisasi
nilai Load.

    rekap1 = []; %Inisialisasi
variable rekap1.(kombinasi generator paling optimal)
    rekap2 = []; %Inisialisasi
variable rekap2.(biaya dari kombinasi paling optimal)
    rekap3 = []; %buat Pgg
    rekap4 = []; %buat losses
    rekap5 = [];
    rekap6 = [];
    rekap7 = [];
    rekap8 = [];
    rekap9 = [];
    rekap10 = [];

    rekapp1 = []; %Inisialisasi
variable rekapp1.(kombinasi generator paling optimal)
    rekapp2 = []; %Inisialisasi
variable rekapp2.(biaya dari kombinasi paling optimal)
    rekapp3 = []; %buat Pgg
    rekapp4 = []; %buat losses
    rekapp5 = [];
    rekapp6 = [];
    rekapp7 = [];
    rekapp8 = [];
    rekapp9 = [];

```

```

rekapp10 = [];

pggsebelum = 0;
kombinasisebelum = [];
checkIS = CIS; %Copy nilai
CIS ke checkIS.
yyyyy = [];
ggggg= [];

for k=1:length(Load)

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

    r3= rand(N,dim);

    v = rand(N,dim); %
    Initial velocities of particles

    A = k;

    %Set nilai x : jika r3 < s(fungsi sigmoid dari
    velocity) maka x=1.
    %           jika r3 > s maka x=0.
    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

    aji(1:N,1)=ones;
    x=[aji x(:,2:6)]

    f = zeros(N,1); % Initial
    fitness functions

```

```

%%tambahan buat ramp rate dan initial state
state
rekapkombinasil

if hour == 1
    checkIS = CIS;
    PS = IS;
end
if hour > 1
    pggsebelum = rekapPembangkitan1(hour-1,:);
    kombinasisebelum = rekapkombinasil(hour-
1,:);
    PS = rekapkombinasil(hour-1,:);
    checkIS = state((hour-1),:); %isinya state
    tiap jam (buat state 1 jam sebelumnya )
    %yang diambil misal jam 2 load 1 berarti yg
    diambil state baris 1
end

%   Dapatkan nilai biaya untuk tiap partikel.
for i=1:N
    onofGenerator = x(i,:);           % Copy nilai
    x pada partikel ke i kedalam onofGenerator.

    % Jika semua nilai onofGenerator 0, set nilai
    f menjadi maks.
    % Jika tidak semua 0 ambil nilai f hasil dari
    fungsibiaya27.
    if onofGenerator(1)~= 0 || onofGenerator(2)~=0
    || onofGenerator(3)~=0 || onofGenerator(4)~=0 ||
onofGenerator(5)~=0 || onofGenerator(6)~=0
        f(i) = PSO_nonsmooth_ibra(multiplierPd,
onofGenerator, x(i,:), PS, checkIS, Load(k), i,
pggsebelum, kombinasisebelum, hour);           % Initial
fitness function
        load
('Pgenout','Pgg','losses','Pdt','fff','rrr','langgar',
'totalcost','startupcost','shutdowncost')
        load ('tegangan','Vreal')
    else

```

```

        f(i) = 1000000000000;
    end
    ggggg(i,:) = Pgg;    %save Pgg ke1
    loss(i,:) = losses;
    aaaa(i,:) = Pdt;
    totcost(i,:) = totalcost;
    startupcost1(i,:) = startupcost;
    shutdowncost1(i,:) = shutdowncost;
    teganganbus(i,:) = Vreal';

end

B = k;

%----- Initial iteration -----
-----
    it=1;                %nilai iterasi.
    Pbest=x;             %copy nilai x ke pbest.
    fbest=f;             %copy nilai f ke fbest.

    [minf,idk]=min(f);   %copy nilai minimum pada f ke
    minf, dan indeks nilai minimum tersebut pada idk
    Gbest=x(idk,:);      %copy nilai x pada indeks ke
    idk (x dengan f minimal) pada gbest.
    lastbest=[0 0];      %inisialisasi lastbest dengan
    nilai 0.
    minftot=[];          %inisialisasi minftot.

%% Main iteratio
while it<maxit

    r1=rand;r2=rand;r3=rand(N,dim);    %random
    nilai r1,r2,dan r3 (nantinya digunakan untuk
    mengupdate nilai v(velocity))
    c1=2;c2=2;                          %set nilai
    c1 dan c2 = 1 (digunakan mengupdate nilai v juga)
    rho=rhomax-((rhomax-rhomin)/maxit)*it; %set nilai
    rho (digunakan mengupdate nilai v juga)

    % Set nilai v dan s untuk tiap particle dan
    kontrol variabel. (s adalah nilai sigmoid dari v)

```

```

    % total size dari v dan s adalah matrix 6 x jumlah
    particle.
    % cari juga nilai x untuk (total size x = 6 x
    jumlah particle)
    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=[aji x(:,2:6)];

%% Check MD dan MU
% Cari nilai unitstart dan unitshutdown.
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

checkIS;

% Untuk particle ke 1 sampai ke N cari nilai f
(biaya).
for i=1:N
    onofGenerator = x(i,:); % parsing nilai x
    pada onofGenerator
    % Jika nilai x tidak 0 semua maka
    eksekusi ke fungsibiaya27.
    if onofGenerator(1)~= 0 || onofGenerator(2)~=0
|| onofGenerator(3)~=0 || onofGenerator(4)~=0 ||
onofGenerator(5)~=0 || onofGenerator(6)~=0
        f(i) = PSO_nonsmooth_ibra(multiplierPd,
onofGenerator, x(i,:), PS, checkIS, Load(k), i ,
pggsebelum, kombinasisebelum, hour);
        load ('Pgenout','Pgg', 'losses',
'Pdt','fff','rrr','langgar','rurd','totalcost','startu
pcost','shutdowncost')
        load ('tegangan','Vreal')
    else
        f(i) = 1000000000000;
    end
    yyyyy(i,:) = Pgg; %save pgg ke2
    loss2(i,:) = losses;
    bbbbbb(i,:) = Pdt;
    totcost2(i,:) = totalcost;
    startupcost2(i,:) = startupcost;
    shutdowncost2(i,:) = shutdowncost;
    voltase(i,:) = Vreal';
end

```

```

end

checkIS;

%-----update Pbest
ganti = f < fbest; % jika
nilai f < fbest , masukkan f pada variable ganti.
fbest=fbest.*(1-ganti)+f.*ganti; %
hitung nilai fbest baru dengan memasukkan nilai
variable ganti.
Pbest(find(ganti),:)=x(find(ganti),:); %
Hitung nilai pbest baru yaitu dengan nilai x pada f
yang lebih kecil tadi.
ggggg(find(ganti),:)= yyyyy(find(ganti),:); %ganti
pake pgg yang f nya paling min
loss(find(ganti),:) = loss2(find(ganti),:);
aaaa(find(ganti),:) = bbbbbb(find(ganti),:);
totcost(find(ganti),:) = totcost2(find(ganti),:);
startupcost1(find(ganti),:) =
startupcost2(find(ganti),:);
shutdowncost1(find(ganti),:) =
shutdowncost2(find(ganti),:);
teganganbus(find(ganti),:) =
voltase(find(ganti),:);

[minf,idk]=min(fbest); % copy
nilai minimum pada f(fbest baru) ke minf, dan indeks
nilai minimum tersebut pada idk
minftot=[minftot;minf] %
minftotal = nilai minf.
Gbest=Pbest(idk,:); %
nilai gbest adalah nilai pbest pada index ke idk.

tttt = ggggg(idk,:); %gbest
Pgg
loss3 = loss(idk,:); %gbest
losses
cccc = aaaa(idk,:); %gbest
pembangkitan

```

```

totcost3 = totcost(idk,:);
startupcost3 = startupcost1(idk,:);
shutdowncost3 = shutdowncost1(idk,:);
teganganbus2 = teganganbus(idk,:);

%      %%biaya total tiap iterasi
%      abcdef = minftot(it);
%      periterasi = [periterasi; abcdef];
%

it=it+1; %
increment iterasi
lastbest=Gbest % set
nilai lastbest = gbest.
E = k;

end

% statee = [statee; checkIS']; %ngesave state di
jam itu

% set nilai rekap1, rekap2, dan lastbest.
minftot
lastbest

rekapp1 = [rekapp1; lastbest];
rekapp2 = [rekapp2; minftot(it-1)];
rekapp3 = [rekapp3; tttt];
rekapp4 = [rekapp4; loss3];
rekapp5 = [rekapp5; cccc];
rekapp6 = [rekapp6; totcost3];
rekapp7 = [rekapp7; startupcost3];
rekapp8 = [rekapp8; shutdowncost3];

%spinning reserve

```



```

unitmati = find(rekapp1(k,:) '== 0);
spinreserve = Pmaks - rekapp3(k,:)';
spinreserve(unitmati) = 0;

rekapp9 = [rekapp9; spinreserve'];
rekapp10 = [rekapp10; teganganbus2];

end
checkIS;

statee = [statee; checkIS'];

minimumm = min(rekapp2(:,1));
if minimumm == rekapp2(1,1)
    cari = 1;
elseif minimumm == rekapp2(2,1)
    cari = 2;
else
    cari = 3;
end

rekap1 = rekapp1(cari,:);
rekap2 = rekapp2(cari,:);
rekap3 = rekapp3(cari,:);
rekap4 = rekapp4(cari,:);
rekap5 = rekapp5(cari,:);
rekap6 = rekapp6(cari,:);
rekap7 = rekapp7(cari,:);
rekap8 = rekapp8(cari,:);
rekap9 = rekapp9(cari,:);
rekap10 = rekapp10(cari,:);

% statee = stateee((find(minimumm)+(hour-1)*3),:);

% unit start and shutdown again
for intiall = 1:1

lastbest'
PS
rekapkombinasi1
%   rekapkombinasi2

```

```

%   rekapkombinasi3
statee

if hour == 1
    intial = PS;
end
if hour > 1
    intial = rekapkombinasil(hour-1,:);
end

checkIS = statee(hour,:); %define state dijam itu

unitstart = find(rekap1(intiall,:) ' > intial)
%yang start
unitshutdown = find(rekap1(intiall,:) ' < intial)
%yang baru off

% untuk state di jam setelahnya

for genn =1:dim
    genn
    if ismember(genn, unitstart) |
ismember(genn,unitshutdown)
        if ismember(genn, unitstart)
            checkIS(genn)= 1;
        end
        if ismember(genn, unitshutdown)
            checkIS(genn)= -1;
        end
    else
        if rekap1(intiall,genn) > 0
            checkIS(genn)=checkIS(genn)+1;
        end
        if rekap1(intiall,genn) == 0
            checkIS(genn)=checkIS(genn)-1;
        end
    end
end

state = [state; checkIS]; %save state dijam
setelahnya

```

```

end

rekap1
rekap2
rekap3
rekap4
rekap5
rekap6
rekap7
rekap8
rekap9
rekap10

%%buat ngeset kategori: tiap jam

% Set nilai total biaya (= jumlah dari rekap2).
Totalbiaya = sum(rekap2)

%Set nilai rekap1, rekap2 , dan rekapIndex untuk
24 jam untuk terbaik pertama.
rekap124(rekapIndex,:) = rekap1(1,:);
rekap224(rekapIndex,:) = rekap2(1,:);
rekap324(rekapIndex,:) = rekap3(1,:);
rekap424(rekapIndex,:) = rekap4(1,:);
rekap524(rekapIndex,:) = rekap5(1,:);
rekap624(rekapIndex,:) = rekap6(1,:);
rekap724(rekapIndex,:) = rekap7(1,:);
rekap824(rekapIndex,:) = rekap8(1,:);
rekap924(rekapIndex,:) = rekap9(1,:);
rekap1024(rekapIndex,:) = rekap10(1,:);

rekapIndex = rekapIndex + 1;

%Set total biaya pada jam ke-hour.
totalBiaya24(hour) = Totalbiaya;

%%buat ngeset kategori: terbaik 1 / 2 / 3

%rekap buat pembangkitan TERBAIK 1
rekapkombinasil(hour,:) = rekap124(rekapbaris,:);
rekapPembangkitan1(hour,:) =
rekap324(rekapbaris,:);

```

```

        rekapBiaya1(hour,:) = rekap224(rekapbaris,1);
        rekapTotCost1(hour,:) = rekap624(rekapbaris,1);
        rekapStartupCost1(hour,:) =
rekap724(rekapbaris,1);
        rekapShutdownCost1(hour,:) =
rekap824(rekapbaris,1);
        rekapTeganganBus(hour,:) =
rekap1024(rekapbaris,1);

```

```

        rekapbaris = rekapbaris + 1;

```

```

        plot(1:hour, rekapBiaya1, 'b-o');
        title('Biaya Pembangkitan')

```

```

end

```

```

        % Print hasil rekap1, rekap2, dan total biaya
        untuk 24 jam.
        rekapIndexPrint = 1;

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

            %Rekap1-----
            Teks = sprintf('Rekap 1 - Kombinasi
Pembangkitan:\\n');
            disp(Teks);
            Teks = sprintf('==== 1 == 2 == 3 == 4 == 5 ==
6 ==');
            disp(Teks);
            rekapIndexPrintTemp = rekapIndexPrint;
            for i = 1 : 1                                %kalo
3 diganti 3
                disp(rekap124(rekapIndexPrintTemp,:));
                rekapIndexPrintTemp = rekapIndexPrintTemp+1;
            end
            %------

```

```

%rekapIndexPrint = rekapIndexPrint + 1;

%Rekap2-----
Teks = sprintf('Rekap 2 - Biaya Pembangkitan
:\n');
disp(Teks);
%totalnya
Teks = sprintf('Total Biaya Pembangkitan :\n');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap224(rekapIndexPrintTemp,1));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
%totalcost pembangkit
Teks = sprintf('Pembangkitan :\n');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap624(rekapIndexPrintTemp,1));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
%startupcost
Teks = sprintf('Start Up Cost :\n');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap724(rekapIndexPrintTemp,1));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
%shutdowncost
Teks = sprintf('Shut Down Cost :\n');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap824(rekapIndexPrintTemp,1));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end

%-----

%Rekap3-----

```

```

Teks = sprintf('Rekap 3 - Daya Yang Dibangkitkan
Tiap Unit Generator MW :\n');
disp(Teks);
Teks = sprintf('==== 1 ===== 2 ===== 3 =====
4 ===== 5 ===== 6 =====');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap324(rekapIndexPrintTemp,:));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
Teks = sprintf('Spinning Reserve Tiap Unit
Generator MW: ');
disp(Teks);
Teks = sprintf('==== 1 ===== 2 ===== 3 =====
4 ===== 5 ===== 6 =====');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap924(rekapIndexPrintTemp,:));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
%------

%Rekap4-----
Teks = sprintf('Rekap 4 - Rugi-Rugi Daya MW :\n');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap424(rekapIndexPrintTemp,1));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
%------

%Rekap5-----
Teks = sprintf('Rekap 5 - Total Pembebanan :\n');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    disp(rekap524(rekapIndexPrintTemp,1));
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
%------

```

```

%Rekap10-----
Teks = sprintf('Rekap 6 - Tegangan Tiap Bus :\n');
disp(Teks);
rekapIndexPrintTemp = rekapIndexPrint;
for i = 1 : 1
    penomoran = [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 ...          16 17 18 19 20 21 22 23 24 25 26 27 28 29
30];
    teganganperbus = [penomoran'
rekap1024(rekapIndexPrintTemp,:)'];
    disp(teganganperbus);
    rekapIndexPrintTemp = rekapIndexPrintTemp+1;
end
%-----

rekapIndexPrint = rekapIndexPrint + 1;

Teks =
sprintf('===== \n')
;
    disp(Teks);

end

%TBiaya-----
Teks = sprintf('Total Biaya : ');
disp(Teks);
Teks = sprintf('Biaya Kombinasi Pembangkitan
Pertama: ');
disp(Teks);
rekapbiayaal = sum(rekapBiaya1);
disp(rekapbiayaal);

toc

```

Program ACOPF PSO:

```

function [f Pgg totalfm
fitness_terbaik_global_particle costv Pdt totalcost] =
PSO_nonsmooth_ibra(multiplierPd, onofGenerator, x, PS,
checkIS, Load, i, pggsebelum, kombinasisebelum, hour)
clc
tic
jum_swarm = 5; % Jumlah calon solusi
ieee30bus; %sistem data
Ra=ctranspose(mwlimits(:,2)); %batas maksimum daya
dibangkitkan
Rb=ctranspose(mwlimits(:,1)); %batas minimum daya
dibangkitkan
iterasimaks = 2; % Iterasi
jum_partikel = 6 ; % Jumlah parameter yang akan
dioptimasi
c2 = 2 ; % PSO parameter C1 (Social
Constant)
c1 = 2 ; % PSO parameter C2 (Cognitive
Constant)
%w = 0.01; % PSO momentum or inertia
wmax=0.9;
wmin=0.4;
iterasimaks=iterasimaks-1;
%onoff=[1 1 1 1 1 1];
onoff = onofGenerator;

%=====Parameter
Sistem=====

loadflow;
gencost;

%=====
=====

%Alokasi Memori-----
-----
Pgggg=Pggg;
busdata2=busdata;

```



```

beban = multiplierPd.*busdata(:,5);
busdata(:,5)=beban;
linedata2=linedata;
cost2=cost;
Data2 = Data;
hsc = Data2(:,7);
csc = Data2(:,8);
csh = Data2(:,9);
sdc = Data2(:,10);
rurd = Data2(:,12);

% inialisasi buat ramp rate
if hour > 1
    pggsebelum;           %pgg 1 jam sebelumnya
    kombinasisebelum;     %kombinasi 1 jam sebelumnya
else
    kombinasisebelum = [0 0 0 0 0 0];
    pggsebelum = [0 0 0 0 0 0];
    konstan1 = 0;
    konstan2 = 0;
    konstan3 = 0;
    konstan4 = 0;
    konstan5 = 0;
    konstan6 = 0;
    pembatas1 = 0;
    pembatas2 = 0;
    pembatas3 = 0;
    pembatas4 = 0;
    pembatas5 = 0;
    pembatas6 = 0;
end

startupcost = 0;

fitness_particle_awal=totalfm;
posisi_partikel=zeros(jum_partikel,jum_swarm);
kec_particle=zeros(jum_partikel,jum_swarm);
fitness=zeros(jum_swarm,iterasimaks);
fitness_particle=zeros(jum_swarm,1);
posisi_terbaik_global=zeros(jum_partikel,jum_swarm);
%-----
-----

```

```

%Inisialisasi Kecepatan-----
-----
v_min=zeros(1,jum_partikel);
v_max=(Ra-Rb)/iterasimaks ;

for ir=1:jum_swarm
    for k=1:jum_partikel
        kec_particle(k,ir)= v_min(1,k)+(v_max(1,k)-
v_min(1,k))*rand;
    end
end
%-----
-----

%Inisialisasi posisi partikel ----> x_ij=x_min+(x_max-
x_min)*rand(.)-----
for ir=1:jum_swarm
    for k=1:jum_partikel
        posisi_partikel(k,ir)= abs(Rb(1,k)+(Ra(1,k)-
Rb(1,k))*rand);
    end
end
%-----
-----

posisi_terbaik_lokal= abs(posisi_partikel); %
p_best=x_ij % Local Best Posisi (Pbest)

% Evaluasi Populasi-----
-----
It=1;
for ir=1:jum_swarm
    busdata=busdata2;
    linedata=linedata2;
    cost=cost2;

%% UNTUK RAMP RATE DAN ONOFF NYA PEMBANGKITAN

%Generator 2
    if onoff(1,2)==0
        busdata(2,7) = 0;
    else

```

```

        konstan2 = abs((posisi_partikel(2,ir)) -
(pggsebelum(1,2)));
        if konstan2 > rurd(2,:)
            if kombinasisebelum(:,2) == 1
                if posisi_partikel(2,ir) >
pggsebelum(1,2)
                    busdata(2,7) =
abs(pggsebelum(1,2) + rurd(2,:) - 1);
                else
                    busdata(2,7) =
abs(pggsebelum(1,2) - rurd(2,:) + 1);
                end
            end
        else
            busdata(2,7) =
abs(posisi_partikel(2,ir));
        end
    end

%Generator 3

    if onoff(1,3)==0
        busdata(13,7) = 0;
    else
        konstan3 =
abs((posisi_partikel(3,ir))- (pggsebelum(1,3)));
        if konstan3 > rurd(3,:)
            if kombinasisebelum(:,3) == 1
                if posisi_partikel(3,ir) >
pggsebelum(1,2)
                    busdata(13,7) =
abs(pggsebelum(1,3) + rurd(3,:) - 1);
                else
                    busdata(13,7) =
abs(pggsebelum(1,3) - rurd(3,:) + 1);
                end
            end
        else
            busdata(13,7) =
abs(posisi_partikel(3,ir));
        end
    end

%Generator 4

```

```

        if onoff(1,4)==0
            busdata(22,7) = 0;
        else
            konstan4 = abs((posisi_partikel(4,ir))-
(pggsebelum(1,4)));
            if konstan4 > rurd(4,:)
                if kombinasisebelum(:,4) == 1
                    if posisi_partikel(4,ir) >
pggsebelum(1,4)
                        busdata(22,7) =
abs(pggsebelum(1,4) + rurd(4,:) - 1);
                    else
                        busdata(22,7) =
abs(pggsebelum(1,4) - rurd(4,:) + 1);
                    end
                end
            else
                busdata(22,7) =
abs(posisi_partikel(4,ir));
            end
        end
    end

%Generator 5

        if onoff(1,5)==0
            busdata(23,7) = 0;
        else
            konstan5 = abs((posisi_partikel(5,ir))-
(pggsebelum(1,5)));
            if konstan5 > rurd(5,:)
                if kombinasisebelum(:,5) == 1
                    if posisi_partikel(5,ir) >
pggsebelum(1,5)
                        busdata(23,7) =
abs(pggsebelum(1,5) + rurd(5,:) - 1);
                    else
                        busdata(23,7) =
abs(pggsebelum(1,5) - rurd(5,:) + 1);
                    end
                end
            else
                busdata(23,7) =
abs(posisi_partikel(5,ir));
            end
        end
    end

```

```

        end
    end

%Generator 6

    if onoff(1,6)==0
        busdata(27,7) = 0;
    else
        konstan6 = abs((posisi_partikel(6,ir))-
(pggsebelum(1,6)));
        if konstan6 > rurd(6,:)
            if kombinasisebelum(:,6) == 1
                if posisi_partikel(6,ir) >
pggsebelum(1,6)
                    busdata(27,7) =
abs(pggsebelum(1,6) + rurd(6,:) - 1);
                else
                    busdata(27,7) =
abs(pggsebelum(1,6) - rurd(6,:) + 1);
                end
            end
        else
            busdata(27,7) =
abs(posisi_partikel(6,ir));
        end
    end

%   busdata(2,7) = posisi_partikel(2,ir); % x1
% terletak pada kolom 2 sebanyak jumlah particle dalam
% kolom (Matriks 1x50)
%   busdata(5,7) = posisi_partikel(3,ir);
%   busdata(8,7) = posisi_partikel(4,ir);
%   busdata(11,7) = posisi_partikel(5,ir);
%   busdata(13,7) = posisi_partikel(6,ir);

% Tambahan Ramp Rate Generator 1

    if hour > 1
        konstan1 = abs((posisi_partikel(1,ir))-
(pggsebelum(1,1)));
        if konstan1 > rurd(1,1)
            if posisi_partikel(1,ir) > pggsebelum(1,1)
                posisi_partikel(1,ir) = pggsebelum(1,1)+
rurd(1,1) - 1;
            end
        end
    end

```

```

        else
            posisi_partikel(1,ir) = pggsebelum(1,1) -
rurd(1,1) + 1;
        end
    end
end

%         posisi_partikel(1,ir)= Pgg(1);

if posisi_partikel(1,ir)> mwlimits(1,2)
    posisi_partikel(1,ir)= mwlimits(1,2);
elseif posisi_partikel(1,ir) < mwlimits(1,1) %biar
ga minus
    posisi_partikel(1,ir) = mwlimits(1,1);
end

if busdata(2,7)> mwlimits(2,2)
    busdata(2,7) = mwlimits(2,2);
elseif busdata(2,7) < mwlimits(2,1) && onoff(1,2)
== 1
    busdata(2,7) = mwlimits(2,1);
end

if busdata(13,7)> mwlimits(3,2)
    busdata(13,7) = mwlimits(3,2);
elseif busdata(13,7) < mwlimits(3,1) && onoff(1,3)
== 1
    busdata(13,7) = mwlimits(3,1);
end

if busdata(22,7)> mwlimits(4,2)
    busdata(22,7) = mwlimits(4,2);
elseif busdata(22,7) < mwlimits(4,1) && onoff(1,4)
== 1
    busdata(22,7) = mwlimits(4,1);
end

if busdata(23,7)> mwlimits(5,2)
    busdata(23,7) = mwlimits(5,2);
elseif busdata(23,7) < mwlimits(5,1) && onoff(1,5)
== 1
    busdata(23,7) = mwlimits(5,1);
end

```

```

    if busdata(27,7)> mwlimits(6,2)
        busdata(27,7) = mwlimits(6,2)
    elseif busdata(27,7) < mwlimits(6,1) && onoff(1,6)
== 1
        busdata(27,7) = mwlimits(6,1)
    end

%
%
busdata(1,7)    = posisi_partikel(1,ir);
loadflow;
gencost;

% Evaluasi fitness populasi ke 2 sampai ke UkPop%
fitness_particle(ir)=totalfm;
fitness_particle2(ir)=9999999999999999;
langgar=0;
M=length(batas1);
%   N=length(Vreal);
O=length(Pgg);

R=length(pggsebelum);

for iii=1:M
    if(batas1(iii,1)>batas2(iii,1))
        langgar=1;
    end
end

%   for jjj=1:N
%
if((Vreal(jjj,1)>batasV) || (Vreal(jjj,1)<batasVmin))
%       langgar=1;
%
%   end
%end

mwrealita=Pgg';
for kkk=1:O
    if(mwrealita(kkk,1)>mwlimits(kkk,2) ||
mwrealita(kkk,1)<mwlimits(kkk,1))

```

```

        if mwrealita(kkk,1)~= 0
            langgar= 1;
        end
    end
end

%%buat ramp rate
if hour > 1
for hhh = 1:6
    rrr = abs((Pgg(1,hhh))-(pggsebelum(1,hhh)));
    fff = kombinasisebelum(1,hhh);
    uuu = rurd';
    if rrr > uuu(1,hhh)
        if (fff == 1 || onofGenerator(1,hhh) == 1)
            langgar = 1;
        end
    else
        langgar = 0;
    end
end
end

if(langgar==1)
    fprintf('Iterasi ke- 0')
    fprintf(' Partikel ke- %1.0f', ir)
    fprintf(' MELANGGAR')
    fprintf('\n')
    batasan(ir,1)=1;
    batasan(1,1)=0;
else
    fprintf('Iterasi ke- 0')
    fprintf(' Partikel ke- %1.0f', ir)
    fprintf(' TDK MELANGGAR')
    fprintf('\n')
    batasan(ir,1)=0;
end

end

%-----
-----

fitness_terbaik_lokal=abs(fitness_particle2');

```







```

        konstan2 = abs((posisi_partikel(2,ir))-
(pggsebelum(1,2)));
        if konstan2 > rurd(2,:)
            if kombinasisebelum(:,2) == 1
                if posisi_partikel(2,ir) >
pggsebelum(1,2)
                    busdata(2,7) =
abs(pggsebelum(1,2) + rurd(2,:) - 1);
                else
                    busdata(2,7) =
abs(pggsebelum(1,2) - rurd(2,:) + 1);
                end
            end
        else
            busdata(2,7) =
posisi_partikel(2,ir);
        end
    end

%Generator 3

    if onoff(1,3)==0
        busdata(13,7) = 0;
    else
        konstan3 =
abs((posisi_partikel(3,ir))- (pggsebelum(1,3)));
        if konstan3 > rurd(3,:)
            if kombinasisebelum(:,3) == 1
                if posisi_partikel(3,ir) >
pggsebelum(1,2)
                    busdata(13,7) =
pggsebelum(1,3) + rurd(3,:) - 1;
                else
                    busdata(13,7) =
pggsebelum(1,3) - rurd(3,:) + 1;
                end
            end
        else
            busdata(13,7) =
posisi_partikel(2,ir);
        end
    end
end

%Generator 4

```

```

        if onoff(1,4)==0
            busdata(22,7) = 0;
        else
            konstan4 = abs((posisi_partikel(4,ir))-
(pggsebelum(1,4)));
            if konstan4 > rurd(4,:)
                if kombinasisebelum(:,4) == 1
                    if posisi_partikel(4,ir) >
pggsebelum(1,4)
                        busdata(22,7) =
pggsebelum(1,4) + rurd(4,:) - 1;
                    else
                        busdata(22,7) =
pggsebelum(1,4) - rurd(4,:) + 1;
                    end
                end
            else
                busdata(22,7) =
posisi_partikel(4,ir);
            end
        end

%Generator 5

        if onoff(1,5)==0
            busdata(23,7) = 0;
        else
            konstan5 = abs((posisi_partikel(5,ir))-
(pggsebelum(1,5)));
            if konstan5 > rurd(5,:)
                if kombinasisebelum(:,5) == 1
                    if posisi_partikel(5,ir) >
pggsebelum(1,5)
                        busdata(23,7) =
abs(pggsebelum(1,5) + rurd(5,:) - 1);
                    else
                        busdata(23,7) =
abs(pggsebelum(1,5) - rurd(5,:) + 1);
                    end
                end
            else
                busdata(23,7) =
posisi_partikel(5,ir)
            end
        end
    end
end

```

```

        end
    end

%Generator 6

    if onoff(1,6)==0
        busdata(27,7) = 0
    else
        konstan6 = abs((posisi_partikel(6,ir))-
        (pggsebelum(1,6)))
        if konstan6 > rurd(6,:)
            if kombinasisebelum(:,6) == 1
                if posisi_partikel(6,ir) >
pggsebelum(1,6)
                    busdata(27,7) =
abs(pggsebelum(1,6) + rurd(6,:) - 1);
                else
                    busdata(27,7) =
abs(pggsebelum(1,6) - rurd(6,:) + 1);
                end
            end
        else
            busdata(27,7) =
posisi_partikel(6,ir);
        end
    end

% busdata(2,7) = posisi_partikel(2,ir); % x1
% terletak pada kolom 2 sebanyak jumlah particle dalam
% kolom (Matriks 1x50)
% busdata(5,7) = posisi_partikel(3,ir);
% busdata(8,7) = posisi_partikel(4,ir);
% busdata(11,7) = posisi_partikel(5,ir);
% busdata(13,7) = posisi_partikel(6,ir);

% Tambahan ramp rate buat generator 1
%
    if hour > 1
        ccc = abs((posisi_partikel(1,ir))-
        (pggsebelum(1,1)));
    end
end

```

```

        if ccc > rurd(1,1)
            if posisi_partikel(1,ir) > pggsebelum(1,1)
                posisi_partikel(1,ir) =
abs(pggsebelum(1,1)+ rurd(1,1) - 1);
            else
                posisi_partikel(1,ir) =
abs(pggsebelum(1,1) - rurd(1,1) + 1);
            end
        end
    end

%         posisi_partikel(1,ir)= Pgg(1);

    if posisi_partikel(1,ir)> mwlimits(1,2)
        posisi_partikel(1,ir)= mwlimits(1,2);
    elseif posisi_partikel(1,ir) < mwlimits(1,1) %biar
ga minus
        posisi_partikel(1,ir) = mwlimits(1,1);
    end

    if busdata(2,7)> mwlimits(2,2)
        busdata(2,7) = mwlimits(2,2);
    elseif busdata(2,7) < mwlimits(2,1) && onoff(1,2)
== 1
        busdata(2,7) = mwlimits(2,1);
    end

    if busdata(13,7)> mwlimits(3,2)
        busdata(13,7) = mwlimits(3,2);
    elseif busdata(13,7) < mwlimits(3,1) && onoff(1,3)
== 1
        busdata(13,7) = mwlimits(3,1);
    end

    if busdata(22,7)> mwlimits(4,2)
        busdata(22,7) = mwlimits(4,2);
    elseif busdata(22,7) < mwlimits(4,1) && onoff(1,4)
== 1
        busdata(22,7) = mwlimits(4,1);
    end

    if busdata(23,7)> mwlimits(5,2)
        busdata(23,7) = mwlimits(5,2);

```

```

elseif busdata(23,7) < mwlimits(5,1) && onoff(1,5)
== 1
    busdata(23,7) = mwlimits(5,1);
end

if busdata(27,7)> mwlimits(6,2)
    busdata(27,7) = mwlimits(6,2)
elseif busdata(27,7) < mwlimits(6,1) && onoff(1,6)
== 1
    busdata(27,7) = mwlimits(6,1)
end

busdata(1,7)    = abs(posisi_partikel(1,ir));

loadflow;
gencost;

%-----
-----
    for ii = 1:jum_partikel
        fprintf('Pembangkit Unit %1.0f',ii)
        fprintf(' = %10.2f',Pg(ii))
        fprintf(' MW')
        fprintf('      Biaya Pembangkitan Unit
%1.0f',ii)
        fprintf(' = %10.2f',costv(ii))
        fprintf(' Rp/jam\n')
    end
    fprintf('\n')
    fprintf('Multiobjective = %10.2f',totalfm)
    fprintf('\n')
    fprintf('Total Pembangkitan =
%10.2f',sum(Pgg))
    fprintf(' MW')
    fprintf('\n')
    fprintf('Total Biaya Pembangkitan =
%10.2f',totalcost)
    fprintf(' $/hour\n')
    %     fprintf('Total Emisi = %10.2f',totalems)
    %     fprintf(' kg/hour\n')
    fprintf('Total Losses System =
%10.2f',sum(Pgg)-Pdt)
    fprintf(' MW\n')

```

```

        fprintf('\n')
    %-----
    -----
% Evaluasi fitness populasi ke 2 sampai ke UkPop%
    fitness_particle(ir)= totalfm;
    langgar=0;
    M=length(batas1);
    N=length(Vreal);
    O=length(Pgg);
    R = length(pggsebelum)

    for iii=1:M
        if(batas1(iii,1)>batas2(iii,1))
            langgar=1;
        end
    end
%     for jjj=1:N
%
if((Vreal(jjj,1)>batasV) || (Vreal(jjj,1)<batasVmin))
%         langgar=1;
%
%     end
%
    end
    mwrealita=Pgg';
    for kkk=1:O
        if(mwrealita(kkk,1)>mwlimits(kkk,2) ||
mwrealita(kkk,1)<mwlimits(kkk,1))
            if mwrealita(kkk,1)~=0
                langgar=1;
            end
        end
    end

%%buat ramp rate
if hour > 1
for hhh = 1:6
    rrr = abs((Pgg(1, hhh)) - (pggsebelum(1, hhh)))
    fff = kombinasisebelum(1, hhh)
    uuu = rurd'
    if rrr > uuu(1, hhh)
        if (fff == 1 || onofGenerator(1, hhh) == 1)
            langgar = 1;
        end
    else

```



```

        langgar = 0;
    end
end
end

if(langgar==1)
    fprintf('Iterasi ke- %1.0f', It)
    fprintf(' Partikel ke- %1.0f', ir)
    fprintf(' MELANGGAR')
    fprintf('\n')
    batasan(ir,1)=1;

else
    fprintf('Iterasi ke- %1.0f', It)
    fprintf(' Partikel ke- %1.0f', ir)
    fprintf(' TDK MELANGGAR')
    fprintf('\n')
    batasan(ir,1)=0;
end

end

end

%-----
%-----

%Penentuan Posisi Terbaik Lokal-----
%-----
for ir=1:jum_swarm
    if fitness_particle(ir)< fitness_terbaik_lokal(ir)
        if(batasan(ir,1)==0)

fitness_terbaik_lokal(ir)=abs(fitness_particle(ir));

posisi_terbaik_lokal(:,ir)=abs(posisi_partikel(:,ir));
        end
    end
end

%-----
%-----

% Penentuan Posisi Terbaik Global-----
%-----

```



```

% fprintf(1,'Iteration: %d,  Fitness:
%f\n',It,fitness_terbaik_global_particle)
% plotvector      = get(hbestplot,'Ydata');
% plotvector(It)= fitness_terbaik_global_particle(It);
% set (hbestplot,'Ydata',plotvector);
% drawnow

It=It+1;
end
clear busdata
% %=====Akhir Loop
PSO=====

clc
busdata=busdata2;
linedata =linedata2;
cost =cost2;

%%TAMBAHAN BUAT RAMP RATE DAN ONOFF GENERATOR

%Generator 2

    if onoff(1,2)==0
        busdata(2,7) = 0;
    else
        pembatas2 =
abs((posisi_terbaik_global(2))- (pggsebelum(1,2)));
        if pembatas2 > rurd(2,:)
            if kombinasisebelum(:,2) == 1
                if posisi_partikel(2,ir) >
pggsebelum(1,2)
                    busdata(2,7) =
abs(pggsebelum(1,2) + rurd(2,:) - 1);
                else
                    busdata(2,7) =
abs(pggsebelum(1,2) - rurd(2,:) + 1);
                end
            end
        else
            busdata(2,7) =
posisi_terbaik_global(2);
        end
    end
end

```

```

%Generator 3

    if onoff(1,3)==0
        busdata(13,7) = 0;
    else
        pembatas3 =
abs((posisi_terbaik_global(3)) - (pggsebelum(1,3)));
        if pembatas3 > rurd(3,:)
            if kombinasisebelum(:,3) == 1
                if posisi_terbaik_global(3) >
pggsebelum(1,3)
                    busdata(13,7) =
abs(pggsebelum(1,3) + rurd(3,:) - 1);
                else
                    busdata(13,7) =
abs(pggsebelum(1,3) - rurd(3,:) + 1);
                end
            end
        else
            busdata(13,7) =
posisi_terbaik_global(3);
        end
    end

%Generator 4

    if onoff(1,4)==0
        busdata(22,7) = 0;
    else
        pembatas4 =
abs((posisi_terbaik_global(4)) - (pggsebelum(1,4)));
        if pembatas4 > rurd(4,:)
            if kombinasisebelum(:,4) == 1
                if posisi_terbaik_global(4) >
pggsebelum(1,4)
                    busdata(22,7) =
abs(pggsebelum(1,4) + rurd(4,:) - 1);
                else
                    busdata(22,7) =
abs(pggsebelum(1,4) - rurd(4,:) + 1);
                end
            end
        else

```

```

                                busdata(22,7) =
posisi_terbaik_global(4);
                                end
                                end

%Generator 5

                                if onoff(1,5)==0
                                    busdata(23,7) = 0;
                                else
                                    pembatas5 =
abs((posisi_terbaik_global(5)) - (pggsebelum(1,5)));
                                    if pembatas5 > rurd(5,:)
                                        if kombinasisebelum(:,5) == 1
                                            if posisi_terbaik_global(5) >
pggsebelum(1,5)
                                                busdata(23,7) =
abs(pggsebelum(1,5) + rurd(5,:) - 1);
                                            else
                                                busdata(23,7) =
abs(pggsebelum(1,5) - rurd(5,:) + 1);
                                            end
                                        end
                                    else
                                        busdata(23,7) =
posisi_terbaik_global(5);
                                    end
                                end

%Generator 6

                                if onoff(1,6)==0
                                    busdata(27,7) = 0;
                                else
                                    pembatas6 =
abs((posisi_terbaik_global(6)) - (pggsebelum(1,6)));
                                    if pembatas6 > rurd(6,:)
                                        if kombinasisebelum(:,6) == 1
                                            if posisi_terbaik_global(6) >
pggsebelum(1,6)
                                                busdata(27,7) =
abs(pggsebelum(1,6) + rurd(6,:) - 1);
                                            else

```

```

                                busdata(27,7) =
abs(pggsebelum(1,6) - rurd(6,:) + 1);
                                end
                                end
                                else
                                busdata(27,7) =
posisi_terbaik_global(6);
                                end
                                end

%Generator 1

    pembatas1 = abs((posisi_terbaik_global(1))-
(pggsebelum(1,1)));

    if pembatas1 > rurd(1,:)
        if posisi_terbaik_global(1) > pggsebelum(1,1)
            posisi_terbaik_global(1) =
abs(pggsebelum(1,1) + rurd(1,:) - 1);
        else
            posisi_terbaik_global(1) =
abs(pggsebelum(1,1) - rurd(1,:) + 1);
        end
    else
        end

    end

    if posisi_terbaik_global(1) > mwlimits(1,2)
        posisi_terbaik_global(1) = mwlimits(1,2);

    elseif posisi_terbaik_global(1) < mwlimits(1,1)
        posisi_terbaik_global(1) = mwlimits(1,1);
    end

    if busdata(2,7)> mwlimits(2,2)
        busdata(2,7) = mwlimits(2,2);
    elseif busdata(2,7) < mwlimits(2,1) && onoff(1,2)
    == 1
        busdata(2,7) = mwlimits(2,1);
    end

    if busdata(13,7)> mwlimits(3,2)

```

```

        busdata(13,7) = mwlimits(3,2);
elseif busdata(13,7) < mwlimits(3,1) && onoff(1,3)
== 1
        busdata(13,7) = mwlimits(3,1);
end

if busdata(22,7)> mwlimits(4,2)
        busdata(22,7) = mwlimits(4,2);
elseif busdata(22,7) < mwlimits(4,1) && onoff(1,4)
== 1
        busdata(22,7) = mwlimits(4,1);
end

if busdata(23,7)> mwlimits(5,2)
        busdata(23,7) = mwlimits(5,2);
elseif busdata(23,7) < mwlimits(5,1) && onoff(1,5)
== 1
        busdata(23,7) = mwlimits(5,1);
end

if busdata(27,7)> mwlimits(6,2)
        busdata(27,7) = mwlimits(6,2);
elseif busdata(27,7) < mwlimits(6,1) && onoff(1,6)
== 1
        busdata(27,7) = mwlimits(6,1);
end

busdata(1,7) = abs(posisi_terbaik_global(1));

%      busdata(2,7)      = posisi_terbaik_global(2);
%      busdata(5,7)      = posisi_terbaik_global(3);
%      busdata(8,7)      = posisi_terbaik_global(4);
%      busdata(11,7)     = posisi_terbaik_global(5);
%      busdata(13,7)     = posisi_terbaik_global(6);
%
loadflow;
busout;
lineflow;
gencost;

```

```

toc

%% Tambahan untuk shutdown cost dan start up cost
unitstart = find(onoff' > PS);
unitshutdown = find(onoff' < 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(sdc(unitshutdown,1));

fprintf('startupcost = %10.2f', startupcost)
fprintf('shutdowncost = %10.2f', shutdowncost)

    for fgh = 1:6
        if Pgg(1,fgh) > mwlimits(fgh,2)
            totalcost = 1000000000000;

            elseif (Pgg(1,fgh) < mwlimits(fgh,1) ||
Pgg(1,fgh) < 0 ) && Pgg(1,fgh) ~= 0
                totalcost = 1000000000000;

        end
    end

    for fgh = 1:6
        ramprate(fgh) = abs((pggsebelum(1,fgh))-
(Pgg(1,fgh)));
        if (ramprate(fgh) > rurd(fgh,1)) &
kombinasisebelum(1,fgh) == 1 & onofGenerator(1,fgh) ==
1
            totalcost = 1000000000000;
        end
    end
end

```



```

if hour > 1
rampratee = abs((pggsebelum(1,1))-(Pgg(1,1)));

if rampratee > rurd(1,1)
    totalcost = 1000000000000;
end
end

for ii = 1:jum_partikel
    fprintf('Pembangkit Unit %1.0f',ii)
    fprintf(' = %10.2f',Pgg(ii))
    fprintf(' MW')
    fprintf('      Biaya Pembangkitan Unit
%1.0f',ii)
    fprintf(' = %10.2f',costv(ii))
    fprintf(' Rp/jam\n')
end
    fprintf('\n')
    %fprintf('Multiobjective = %10.2f',totalfm)
    fprintf('\n')
    fprintf('Total Pembangkitan =
%10.2f',sum(Pgg))
    fprintf(' MW')
    fprintf('\n')
    fprintf('Biaya Pembangkitan =
%10.2f',totalcost)
    fprintf(' $/hour\n')
%    fprintf('Total Emisi = %10.2f',totalems)
    fprintf(' kg/hour\n')
    fprintf('Total Losses System =
%10.2f',sum(Pgg)-Pdt)
    fprintf(' MW\n')
    fprintf('\n')
    fprintf('jam = %10.2f',hour)

%BUAT NGESAVE PEMBANGKITAN
losses = sum(Pgg)-Pdt;
save Pgenout;

f = totalcost + startupcost + shutdowncost;

```

**State generator selama 24 jam:**

<i>Hour</i>	<b>Gen 1</b>	<b>Gen 2</b>	<b>Gen 3</b>	<b>Gen 4</b>	<b>Gen 5</b>	<b>Gen 6</b>
0	-4	3	-3	-3	-2	-2
1	1	4	-4	-4	-3	-3
2	2	5	-5	-5	-4	-4
3	3	6	-6	-6	-5	-5
4	4	7	-7	-7	-6	-6
5	5	8	-8	-8	-7	-7
6	6	9	-9	-9	-8	-8
7	7	10	-10	-10	-9	-9
8	8	11	-11	-11	-10	-10
9	9	12	1	-12	-11	-11
10	10	13	2	-13	-12	-12
11	11	14	3	-14	-13	-13
12	12	15	4	-15	-14	-14
13	13	16	5	-16	-15	-15
14	14	17	6	-17	-16	-16
15	15	18	7	-18	-17	-17
16	16	19	8	-19	-18	-18
17	17	20	9	-20	-19	-19
18	18	21	10	-21	-20	1
19	19	22	11	-22	-21	2
20	20	23	12	-23	-22	3
21	21	24	13	-24	-23	4
22	22	25	14	-25	-24	5
23	23	26	15	-26	-25	6
24	24	27	16	-27	-26	7