



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM
UNTUK *DYNAMIC ECONOMIC DISPATCH* PADA
PEMBANGKIT JAWA-BALI DENGAN FUNGSI BIAYA
TIDAK MULUS

Joseph Danarta Sugiana
NRP 07111440000136

Dosen Pembimbing
Dr. Rony Seto Wibowo, ST., MT.
Dr. Ir. Ni Ketut Aryani, MT.

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



TUGAS AKHIR - TE 141599

SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM
UNTUK DYNAMIC ECONOMIC DISPATCH PADA
PEMBANGKIT JAWA-BALI DENGAN FUNGSI BIAYA
TIDAK MULUS

Joseph Danarta Sugiana
NRP 0711144000136

Dosen Pembimbing
Dr. Rony Seto Wibowo, ST., MT.
Dr. Ir. Ni Ketut Aryani, MT.

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



FINAL PROJECT - TE 141599

**SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM FOR
DYNAMIC ECONOMIC DISPATCH ON JAVA-BALI
GENERATION SYSTEM WITH NON-SMOOTH COST
FUNCTION**

Joseph Danarta Sugiana
NRP 07111440000136

Supervisors
Dr. Rony Seto Wibowo, ST., MT.
Dr. Ir. Ni Ketut Aryani, MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi keseluruhan Tugas akhir saya dengan judul “***SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM UNTUK DYNAMIC ECONOMIC DISPATCH PADA PEMBANGKIT JAWA-BALI DENGAN FUNGSI BIAYA TIDAK MULUS***” 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, Juli 2018



Joseph Danarta Sugiana
0711144000136

**SELF-ADAPTIVE MODIFIED FIREFLY
ALGORITHM UNTUK DYNAMIC ECONOMIC
DISPATCH PADA PEMBANGKIT JAWA-BALI
DENGAN FUNGSI BIAYA TIDAK MULUS**

TUGAS AKHIR

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

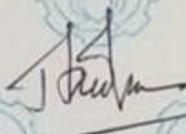
Menyetujui :

Dosen Pembimbing I



Dr. Rony S. Wibowo, ST., MT.
NIP. 19741129 200012 1 001

Dosen Pembimbing II



Ir. Ni Ketut Aryani, MT.
NIP. 19650901 199103 2 002



SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM UNTUK DYNAMIC ECONOMIC DISPATCH PADA PEMBANGKIT JAWA-BALI DENGAN FUNGSI BIAYA TIDAK MULUS

Nama mahasiswa : Joseph Danarta Sugiana
Dosen Pembimbing I : Dr. Rony Seto Wibowo, ST., MT.
Dosen Pembimbing II : Ir. Ni Ketut Aryani MT.

Abstrak:

Pada zaman sekarang, kebutuhan energi listrik sudah menjadi kebutuhan primer bagi masyarakat. Seiring dengan meningkatnya jumlah populasi dan kemajuan teknologi, kebutuhan energi listrik berbanding lurus dengan peningkatan populasi dan kemajuan teknologi. Melihat kondisi dimana kebutuhan energi listrik yang sangat banyak sedangkan ketersediaan bahan baku yang tidak dapat diberharui seperti minyak atau batu bara sangat sedikit, teknologi pengoptimalanpun mulai berkembang guna memenuhi kebutuhan energi listrik yang sangat banyak. Dengan penggunaan *Dynamic Economic Dispatch (DED)*, akan didapatkan nilai minimum (ekonomis) biaya pembangkitan dari sebuah sistem pembangkit listrik untuk jumlah energi maksimum yang dapat dihasilkan. Pada penelitian yang telah dilakukan oleh peneliti, digunakan program MATLAB untuk melakukan perhitungan dan iterasi dari *Dynamic Economic Dispatch Constraint* atau batasan yang digunakan dalam penelitian ini adalah keseimbangan daya pembangkit (*Power Balance*) dan *Inequality Constraint* serta mempertimbangkan fungsi biaya tidak mulus akibat pembukaan katup uap pada pembangkitan termal (*valve-point effect*). Untuk menunjang penyelesaian permasalahan economic dispatch tersebut, terdapat berbagai macam metode salah satunya adalah *Firefly Algorithm* atau biasa disingkat dengan *FA*. *FA* adalah salah satu algoritma yang memiliki konsep dan implementasi lebih sederhana ketimbang algoritma kawanan lainnya.

Tugas akhir ini membahas *Self-Adaptive Modified Firefly Algorithm (SAMFA)* yang diaplikasikan pada perhitungan *DED* pada sistem pembangkitan Jawa-Bali. Simulasi pada tugas akhir ini akan dibagi dua bagian, yaitu perhitungan *DED* dengan menggunakan *SAMFA* dengan mempertimbangkan fungsi biaya tidak mulus dan pada bagian kedua adalah perhitungan *DED* dengan menggunakan *FA*. Kedua bagian

tersebut akan dibandingkan untuk melihat mana yang lebih baik dalam melakukan perhitungan *DED*.

Kata kunci: *DED, constraint, self-adaptive modified firefly algorithm, valve-point effect.*

SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM FOR DYNAMIC ECONOMIC DISPATCH ON JAVA-BALI GENERATION SYSTEM WITH NON-SMOOTH COST FUNCTION

Student Name : Joseph Danarta Sugiana
Supervisor I : Dr. Rony Seto Wibowo, ST., MT.
Supervisor II : Ir. Ni Ketut Aryani MT.

Abstract:

In today's world, the need for electrical energy has become a primary need for society. Along with the increasing number of population and technological progress, the requirement of electric energy is directly proportional to the increase of population and technological progress. Seeing the conditions where the demand for electrical energy is very much while the availability of non-renewable raw materials such as oil or coal is very little, optimization technology began to develop to meet the needs of electrical energy is very much. With the use of Dynamic Economic Dispatch (DED), there will be a minimum (economic) cost of generation of a power plant system for the maximum amount of energy that can be generated. In the research that has been done by researchers, used MATLAB program to perform calculations and iterations of Dynamic Economic Dispatch. Constraints used in this study are the balance of power generation (Power Balance) and Inequality Constraints and consider the function of the cost is not smooth due to the opening of the steam valve on the generation of thermal (valve-point effect). To support the solution of economic dispatch problem, there are various methods one of them is Firefly Algorithm or commonly abbreviated as FA. FA is one algorithm that has simpler concept and implementation than other flock algorithms.

This final project discusses the Self-Adaptive Modified Firefly Algorithm (SAMFA) applied to DED calculations on Java-Bali generation system. The simulation in this final project will be divided into two parts, that is DED calculation using SAMFA by considering the cost function is not smooth and in the second part is DED calculation using FA. Both sections will be compared to see which one is better at performing DED calculations.

Key Word: *DED, constraint, self-adaptive modified firefly algorithm, valve-point effect.*

KATA PENGANTAR

Puji Syukur penulis panjatkan kehadirat Tuhan atas segala Rahmat, Karunia, dan Petunjuk yang telah dilimpahkan-Nya sehingga penulis mampu menyelesaikan tugas akhir dengan judul **“SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM UNTUK DYNAMIC ECONOMIC DISPATCH PADA PEMBANGKIT JAWA-BALI DENGAN FUNGSI BIAYA TIDAK MULUS”**.

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 mengucapkan terima kasih kepada:

1. Tuhan YME atas limpahan rahmat, karunia dan petunjuk-Nya.
2. Ibu dan Bapak penulis atas doa dan cinta yang tak henti pada penulis dalam keadaan apapun. Semoga Tuhan senantiasa melindungi dan memberi mereka tempat terbaik kelak di surgaNya.
3. Bapak Rony Seto Wibowo dan Ibu Ni Ketut Aryani selaku dosen pembimbing yang telah memberikan arahan, bimbingan dan perhatiannya selama proses penyelesaian tugas akhir ini.
4. Lidwina Aretha Quinn yang senantiasa menemani dan mendukung dari awal hingga akhir perkuliahan penulis di teknik elektro ITS.
5. Teman-teman seperjuangan e54 yang telah menemani dan memberikan dukungan selama masa kuliah sampai penyusunan tugas akhir ini.

Penulis telah berusaha maksimal dalam penyusunan tugas akhir ini. Namun tetap besar harapan penulis untuk menerima saran dan kritik untuk perbaikan dan pengembangan tugas akhir ini. Semoga tugas akhir ini dapat memberikat manfaat yang luas.

Surabaya, Juni 2018

Joseph Danarta Sugiana

Halaman ini sengaja dikosongkan

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	vii
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Penulisan	3
1.7 Relevansi	4
BAB 2 <i>DYNAMIC ECONOMIC DISPATCH</i> DENGAN MENGUNAKAN <i>FIREFLY ALGORITHM</i> DAN <i>SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM</i> MEMPERTIMBANGKAN EFEK KATUP UAP	5
2.1 Sistem Tenaga Listrik	5
2.2 Sistem Pembangkitan	5
2.3 Pembebanan Pembangkit	7
2.4 Efek Katup Uap	8
2.5 Dynamic Economic Dispatch	8
2.6 Firefly Algorithm	10
2.6.1 Dasar Biologis.....	11
2.6.2 Perilaku Kunang-kunang.....	13
2.6.3 Konsep.....	13
2.7 Self-Adaptive Modified Firefly Algorithm	14
BAB 3 IMPLEMENTASI <i>SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM</i> PADA <i>DYNAMIC ECONOMIC DISPATCH</i>	17
3.1 Sistem Pembangkitan Jawa-Bali	17
3.1.1 Data-data Pembangkit Jawa-Bali 500 kV	18

3.2	Fungsi Biaya Tidak Mulus	20
3.3	Implementasi <i>FA</i> pada <i>DED</i>	21
3.3.1	Inisiasi Awal <i>FA</i> untuk Menyelesaikan <i>DED</i>	21
3.3.2	Inisiasi Posisi Awal Partikel/Populasi	22
3.3.3	Korektor Posisi Awal	23
3.3.4	<i>Update</i> Intensitas Cahaya Kunang-kunang	23
3.3.5	<i>Update</i> Posisi Kunang-kunang	24
3.3.6	<i>Flowchart FA</i> untuk <i>DED</i>	25
3.4	Implementasi <i>SAMFA</i> pada <i>DED</i>	26
3.4.1	<i>Flowchart SAMFA</i> untuk <i>DED</i>	27
BAB 4 SIMULASI DAN ANALISIS		29
4.1	Validasi Program <i>FA</i>	29
4.2	Efek Katup Uap	35
4.3	Metode <i>FA</i> untuk menyelesaikan <i>DED</i> dengan Mempertimbangkan Fungsi Biaya Tidak Mulus	33
4.3.1	Hasil Simulasi <i>DED</i> dengan menggunakan <i>FA</i>	35
4.3.2	Biaya Total Pembangkitan (<i>FA</i>)	38
4.4	Metode <i>SAMFA</i> untuk menyelesaikan <i>DED</i> dengan Mempertimbangkan Fungsi Biaya Tidak Mulus	40
4.4.1	Hasil Simulasi <i>DED</i> dengan menggunakan <i>SAMFA</i>	44
4.4.2	Biaya Total Pembangkitan (<i>SAMFA</i>)	47
4.5	Perbandingan Hasil Perhitungan <i>FA</i> dengan <i>SAMFA</i>	48
BAB 5 PENUTUP		51
5.1	Kesimpulan	51
5.2	Saran	51
DAFTAR PUSTAKA		53
LAMPIRAN		55
BIODATA PENULIS		65

DAFTAR GAMBAR

Gambar 2.1 Sistem Tenaga Listrik.....	5
Gambar 2.2 Karakteristik Masukan-Keluaran Pembangkit Termal.....	6
Gambar 2.3 Kurva Pembangkit dengan Efek Katup Uap.....	8
Gambar 2.4 Ilustrasi Pergerakan Kunang-Kunang	11
Gambar 3.1 <i>Single Line Diagram</i> Pembangkit Jawa-Bali 500 kV	17
Gambar 3.2 Kurva <i>Cost Function</i> dengan dan tanpa Efek Katup Uap	21
Gambar 3.3 <i>Flowchart</i> Implementasi <i>FA</i> pada <i>DED</i>	25
Gambar 3.4 <i>Flowchart</i> Implementasi <i>SAMFA</i> pada <i>DED</i>	27
Gambar 4.1 Perbandingan Tanpa & Dengan Efek Katup Uap (<i>FA</i>)....	39
Gambar 4.2 Perbandingan Tanpa & Dengan Efek Katup Uap (<i>SAMFA</i>)	48
Gambar 4.3 Perbandingan Biaya Pembangkitan Antara <i>FA</i> dan <i>SAMFA</i>	50

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1	Data Pembangkitan Sistem Interkoneksi 500 kV Jawa-Bali ...	18
Tabel 3.2	Data Saluran Sistem Interkoneksi 500 kV Jawa-Bali	18
Tabel 3.3	<i>Cost Function</i> Sistem Interkoneksi 500 kV Jawa-Bali	19
Tabel 3.4	Batas Kemampuan Sistem Interkoneksi 500 kV Jawa-Bali	20
Tabel 4.1	Data Pembangkit <i>Example 3A</i>	29
Tabel 4.2	Fungsi Biaya <i>Example 3A</i>	30
Tabel 4.3	Daya Terbangkit Setiap Unit <i>Example 3A</i>	30
Tabel 4.4	Daya Terbangkit Setiap Unit <i>Example 3A (FA)</i>	30
Tabel 4.5	Fungsi Biaya <i>Example 3B</i>	31
Tabel 4.6	Daya Terbangkit Setiap Unit <i>Example 3B</i>	31
Tabel 4.7	Daya Terbangkit Setiap Unit <i>Example 3B (FA)</i>	32
Tabel 4.8	Koefisien e dan f	32
Tabel 4.9	Biaya Tidak Mulus	33
Tabel 4.10	Profil Beban untuk 24 Jam	33
Tabel 4.11	Data Koefisien Pembangkit	34
Tabel 4.12	Batasan Pembangkit	35
Tabel 4.13	Parameter awal <i>FA</i> untuk <i>DED</i>	35
Tabel 4.14	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-1 sampai 3	36
Tabel 4.15	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-4 sampai 6	36
Tabel 4.16	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-7 sampai 9	36
Tabel 4.17	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-10 sampai 12	37
Tabel 4.18	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-13 sampai 15	37
Tabel 4.19	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-16 sampai 18	37
Tabel 4.20	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-19 sampai 21	38
Tabel 4.21	Hasil Simulasi <i>FA</i> untuk <i>DED</i> Jam ke-22 sampai 24	38
Tabel 4.22	Biaya Total Pembangkitan (<i>FA</i>)	38
Tabel 4.23	Profil Beban untuk 24 Jam (<i>SAMFA</i>)	40
Tabel 4.24	Data Koefisien Pembangkit (<i>SAMFA</i>)	41
Tabel 4.25	Batasan Pembangkit (<i>SAMFA</i>)	41
Tabel 4.26	Parameter awal <i>SAMFA</i> untuk <i>DED</i>	42
Tabel 4.27	Tabel Nilai Alpha Setiap Iterasi	42
Tabel 4.28	Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-1 sampai 3	44
Tabel 4.29	Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-4 sampai 6	45
Tabel 4.30	Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-7 sampai 9	45
Tabel 4.31	Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-10 sampai 12	45
Tabel 4.32	Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-13 sampai 15	46
Tabel 4.33	Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-16 sampai 18	46

Tabel 4.34 Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-19 sampai 21 ...	46
Tabel 4.35 Hasil Simulasi <i>SAMFA</i> untuk <i>DED</i> Jam ke-22 sampai 24 ...	47
Tabel 4.36 Biaya Total Pembangkitan (<i>SAMFA</i>)	47
Tabel 4.37 Perbandingan Total Biaya Setiap Unit	49
Tabel 4.38 Perbandingan Total Biaya Pembangkitan	50

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya zaman yang berbanding lurus dengan perkembangan teknologi, kebutuhan akan energi listrik semakin meningkat. Melihat peningkatan permintaan energi listrik yang semakin besar, peluang teknologi dalam bidang energi listrik juga semakin banyak. Melihat dari kenyataan, permintaan akan energi listrik sangat banyak sedangkan ketersediaan bahan baku pembangkitan energi listrik yang terbatas, teknologi pengoptimalanpun mulai lahir dan berkembang guna memenuhi kebutuhan permintaan energi listrik dengan keterbatasan bahan baku pembangkitan.

DED merupakan salah satu masalah penting dalam operasi sistem tenaga dan kontrol. Akan tetapi perhitungan *DED* yang kompleks adalah masalah optimasi yang sulit dan tujuan dari *DED* sendiri adalah untuk mengurangi biaya bahan bakar pembangkit seoptimal mungkin, dengan penjadwalan beban guna mencari pembangkitan dari unit pembangkit. Jaringan listrik yang besar adalah sebuah sistem kompleks yang terdiri dari generator, transformer, saluran transmisi, pemutus, kapasitor, reaktor, motor, dan perangkat lainnya. Operasi, ketersediaan, dan kontinuitas dalam pelayanannya sangat tidak dapat diprediksi. Oleh karena itu, permintaan energi listrik menjadi bervariasi yang berkelanjutan sehingga sistem menjadi dinamis. Karena sistem dinamis yang tidak dapat diprediksi karakteristiknya, maka lahirlah berbagai macam metode pendekatan untuk memahami karakteristik sistem pembangkit listrik yang dinamis (algoritma optimasi).

SAMFA digunakan pada topik ini guna mencari beban yang paling tepat untuk setiap unit pembangkit pada sistem pembangkitan Jawa-Bali yang terdapat delapan unit pembangkitan yang masing-masing memiliki kapasitas dan karakteristik yang berbeda satu sama lain, ditambah permasalahan ini juga diikuti dengan pertimbangan fungsi biaya tidak mulus yang diakibatkan dari efek katup uap pada pembangkit termal.

1.2 Permasalahan

Permasalahan yang akan dibahas pada tugas akhir ini adalah :

1. Bagaimana karakteristik pembangkitan sistem pembangkit Jawa-Bali dengan mempertimbangkan fungsi biaya tidak mulus?

2. Bagaimana hasil perhitungan *DED* dengan *FA* dalam mencari biaya pembangkitan paling minimum?
3. Bagaimana hasil perhitungan *DED* dengan *SAMFA* dalam mencari biaya pembangkitan paling minimum?

1.3 Tujuan

Penelitian tugas akhir ini bertujuan untuk :

1. Mengetahui pengaruh fungsi biaya tidak mulus terhadap biaya pembangkitan.
2. Mendapatkan hasil perhitungan *DED* dengan *FA* untuk mendapatkan biaya pembangkitan minimum.
3. Mendapatkan hasil perhitungan *DED* dengan *SAMFA* untuk mendapatkan biaya pembangkitan minimum.

1.4 Batasan Masalah

Batasan masalah yang dibahas pada topik tugas akhir ini adalah sebagai berikut :

1. Simulai dilakukan pada sistem pembangkitan Jawa-Bali dengan 8 unit pembangkit.
2. Semua pembangkit dianggap menyala.
3. Analisa dilakukan dengan mempertimbangkan fungsi biaya tidak mulus, *inequality constraint*, dan kapasitas saluran.
4. Analisa dilakukan pada pembangkit selama dua puluh empat jam dengan rentang satu jam.
5. Sistem dianggap tidak mempunyai *losses*.

1.5 Metodologi

Metodologi yang akan digunakan penulis dalam proses penulisan Tugas Akhir ini adalah :

1. Pengumpulan Data dan Studi Literatur

Mempelajari perhitungan *DED* sebagai metode untuk mencari titik optimal sistem pembangkitan dari segi biaya pembangkitan. Juga dikarenakan permintaan beban setiap pembangkit berbeda-beda dan pengeluaran biaya juga berbeda sesuai dengan spesifikasi unit pembangkit, maka perlu dipelajari bagaimana cara menghitung dan melakukan simulasi. Kemudian mempelajari metode *FA* dan *SAMFA* sebagai algoritma untuk menyelesaikan perhitungan *DED* yang dapat menghasilkan hasil perhitungan yang lebih baik dan lebih mendekati titik optimal.

2. Simulasi

Pada tahap ini, perhitungan *DED* dilakukan dengan simulasi pada sistem pembangkitan Jawa-Bali. Hal ini dilakukan sebab sistem pembangkitan Jawa-Bali menggambarkan kondisi yang tepat untuk sistem kelistrikan di Indonesia.

3. Pengujian Algoritma

Kemudian, pengujian algoritma ini perlu dilakukan untuk menerapkan metode *SAMFA*. Hal ini dilakukan melalui software MATLAB. Software MATLAB akan memberikan wadah untuk melakukan pendekatan dan iterasi menggunakan kedua metode algoritma tersebut secara singkat dan lebih efisien.

4. Analisa data

Setelah melakukan simulasi dan pengujian algoritma pada sistem pembangkitan Jawa-Bali, maka data yang diperoleh akan dianalisa. Analisa mencakup bagaimana pengaruh penggunaan pendekatan metode *SAMFA* terhadap hasil perhitungan biaya pembangkitan pada sistem pembangkitan Jawa-Bali sehingga diharapkan menghasilkan hasil yang lebih optimal.

5. Kesimpulan

Pada akhir penelitian akan ditarik kesimpulan berdasarkan analisa yang telah dilakukan.

1.6 Sistematika Penulisan

Sistematika penulisan dalam tugas akhir ini terdiri atas lima bab dengan uraian sebagai berikut :

Bab 1 : Pendahuluan

Bab ini membahas tentang penjelasan mengenai latar belakang, permasalahan dan batasan masalah, tujuan, metode penelitian, sistematika pembahasan, dan relevansi.

Bab 2 : Tinjauan Pustaka

Bab ini membahas mengenai dasar teori yang digunakan untuk menunjang penyusunan tugas akhir ini.

Bab 3 : Implementasi *SAMFA* untuk *DED*

Bab ini membahas mengenai proses pengambilan data dan metode yang digunakan untuk pengolahan data.

- Bab 4 : Simulasi dan Analisis
Bab ini membahas hasil dari simulasi algoritma *SAMFA* untuk mencari biaya pembangkitan minimum dari sistem pembangkitan Jawa-Bali.
- Bab 5 : Penutup
Bab ini berisi tentang kesimpulan dari hasil tugas akhir yang telah dikerjakan dan juga saran yang dapat diberikan untuk pengembangan hasil tugas akhir ini kedepannya.

1.7 Relevansi

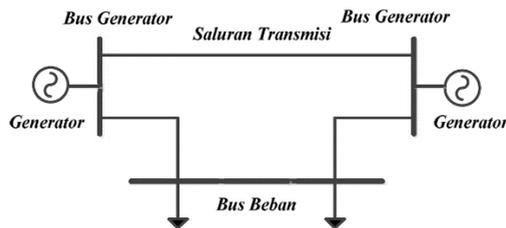
Tugas akhir ini diharapkan dapat memberikan manfaat untuk perkembangan dunia kelistrikan terutama dalam bidang pengoptimalan, dan menjadi referensi bagi mahasiswa yang melakukan penelitian dengan topik serupa.

BAB 2

DYNAMIC ECONOMIC DISPATCH DENGAN MENGGUNAKAN *FIREFLY ALGORITHM* DAN SELF-ADAPTIVE MODIFIED *FIREFLY ALGORITHM* MEMPERTIMBANGKAN EFEK KATUP UAP

2.1 Sistem Tenaga Listrik

Permintaan energi listrik sudah menjadi sebuah kebutuhan yang sangat dibutuhkan oleh manusia pada kemajuan teknologi saat ini. Sistem tenaga listrik adalah sistem yang digunakan untuk memenuhi kebutuhan permintaan energi listrik tersebut. Sistem tenaga listrik sendiri terdiri dari beberapa komponen yang kemudian akan membentuk sistem-sistem kecil dibawah sistem tenaga listrik itu sendiri. Terdapat tiga sistem yaitu sistem pembangkitan, sistem distribusi, dan sistem transmisi tenaga listrik. Sistem tenaga pembangkitan berfungsi untuk membangkitkan energi listrik dari berbagai sumber tenaga listrik yang ada. Sistem distribusi berfungsi untuk menyalurkan energi listrik ke permintaan beban oleh konsumen (industri, komersil, atau perumahan) [1]. Bentuk gambaran sistem tenaga listrik secara umum digambarkan sebagai berikut,



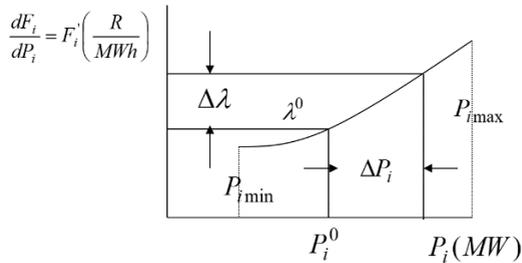
Gambar 2.1 Sistem Tenaga Listrik

2.2 Sistem Pembangkitan

Saat ini pembangkit tenaga listrik yang sangat banyak digunakan dan paling umum dikenal adalah pembangkit termal. Pembangkit termal adalah sebuah proses untuk membangkitkan energi listrik dari energi panas (termal). Terdapat beberapa parameter yang harus diperhatikan dalam menganalisa permasalahan pengoperasian sistem tenaga listrik.

Salah satu parameter yang harus diperhatikan dalam permasalahan pengoperasian sistem pembangkit tenaga listrik secara ekonomis adalah mengatur masukan dan keluaran dari sistem pembangkit termal [2]. Bahan bakar sebagai masukan dari generator termal sangat berpengaruh terhadap keluaran generator itu sendiri. Laju perubahan bahan bakar berbanding lurus dengan daya keluaran unit generator. Karakteristik masukan dan keluaran dari unit pembangkit termal yang ideal dan umum digambarkan sebagai kurva nonlinier yang kontinyu.

Gambar 2.2 menjelaskan karakteristik masukan dan keluaran pembangkit termal terlihat bahwa biaya total bahan bakar per jam dalam $F(R/h)$ atau masukan dari sistem yaitu energi panas yang dibutuhkan setiap jam dalam $H = (MBtu/h)$, dan keluaran pembangkit sendiri adalah satuan MW. Pada kurvan karakteristik pembangkit termal, dapat dilihat bahwa ada batas daya yang diizinkan dalam pengoperasian pembangkitan yaitu nilai daya minimum (P_{min}) dan daya maksimum (P_{max}) [3].



Gambar 2.2 Karakteristik Masukan-Keluaran Pembangkit Termal

Karakteristik masukan dan keluaran dari unit pembangkit pada **Gambar 2.2** adalah kurva $F(R/h)$, merupakan kurva non-linier dengan sifat fungsi kuadrat. Fungsi biaya bahan bakar pada pembangkit termal merupakan sebuah fungsi kuadrat dari daya nyata pada pembangkit yang dijabarkan dalam fungsi $f_i(P_i)$ (1).

$$f_i(P_i) = \alpha_i P_i^2 + \beta_i P_i + \gamma_i \quad (1)$$

dimana α , β , dan γ adalah koefisien masukan dan keluaran pembangkit. Nilai konstanta α akan berbanding lurus dengan konsumsi bahan bakar dari operasi unit pembangkitan tanpa keluaran [4].

2.3 Pembebanan Pembangkit

Klasifikasi pembebanan dalam sistem tenaga listrik terbagi menjadi tiga bagian berdasarkan besarnya beban, industri, komersil, dan perumahan. Biasanya untuk menganalisa perhitungan biaya dilakukan dalam interval dua belas atau dua puluh empat jam berdasarkan kurva beban harian. Pada tiap harinya beban memiliki nilai yang berbeda-beda. Beban yang paling tinggi pada suatu hari disebut dengan beban puncak (*peak load*). Pada permasalahan ini, beban yang digunakan adalah beban secara dinamis karena besarnya beban akan selalu berubah setiap jamnya dengan batasan perubahan beban naik dan turun tertentu.

Untuk skala industri dengan beban yang besar, terdapat beban komposit. Beban komposit adalah gabungan beban statis dan motor induksi yang mengkonsumsi daya reaktif dalam jumlah yang cukup besar. Biasanya dalam skala industri juga beban dianalisa secara dinamis sebab permintaan beban selalu berubah-ubah.

Faktor beban adalah perbandingan dari rata-rata beban pada suatu periode tertentu dengan beban puncak pada periode tersebut. Faktor beban biasanya diberikan dalam satu hari, satu bulan, atau satu tahun. Faktor beban tahunan adalah yang paling berguna karena menggambarkan siklus penuh dari pembangkitan. Faktor daya harian dirumuskan seperti berikut (2).

$$\text{Daily L. F.} = \frac{\text{average load}}{\text{peak load}} \quad (2)$$

Jika faktor daya harian kita kalikan selama dua puluh empat jam, maka akan didapatkan (3).

$$\text{Daily L. F.} = \frac{\text{average load}}{\text{peak load}} = \frac{\text{energy consumed during 24 hr}}{\text{peak load} \times 24 \text{ hr}} \quad (3)$$

Sedangkan faktor daya tahunan dapat dirumuskan dengan faktor daya harian yang dikalikan selama tiga ratus enam puluh lima hari, dirumuskan sebagai berikut (4) [1].

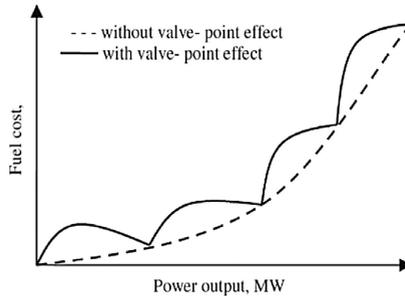
$$\text{Annual L. F.} = \frac{\text{total annual energy}}{\text{peak load} \times 8760 \text{ hr}} \quad (4)$$

2.4 Efek Katup Uap

Dalam pembangkit termal skala besar, biasanya memiliki katup untuk mengatur uap masuk yang diatur membuka secara bertahap, tujuannya adalah untuk mendapatkan keluaran daya yang semakin meningkat. Akibat pembukaan katup uap ini disebut sebagai efek katup uap atau *valve-point effect* [5] yang digambarkan dalam **Gambar 2.3**. Terdapat pengaruh empat katup pengaturan uap terhadap kurva karakteristik masukan dan keluaran dari pembangkit termal. Keadaan non-konveks ini mempengaruhi perhitungan untuk mendapatkan perhitungan optimasi dengan ketepatan yang semakin baik. Pengaruh pembukaan katup uap ini mempengaruhi persamaan karakteristik pembangkit termal sehingga menjadi (5).

$$F_i(P_i) = \alpha_i P_i^2 + \beta_i P_i + \gamma_i + |e_i \times \sin(f_i \times (P_{i,min} - P_i))| \quad (5)$$

Terdapat penambahan variabel d_i dan g_i , merupakan variabel yang menggambarkan pengaruh pembukaan katup uap pada pembangkit termal- i .



Gambar 2.3 Kurva Pembangkit dengan Efek Katup Uap

Untuk skala industri dengan beban yang besar, terdapat beban komposit. Beban komposit adalah gabungan beban statis dan motor induksi yang mengkonsumsi daya reaktif dalam jumlah yang cukup besar. Biasanya dalam skala industri juga beban dianalisa secara dinamis sebab permintaan beban selalu berubah-ubah.

2.5 *Dynamic Economic Dispatch*

DED adalah penjadwalan *dispatch* pada periode tertentu. *DED* memiliki kemampuan untuk memrediksi beban dalam satu periode

tertentu dengan mengoordinasikan prediksi perubahan beban dan kemampuan laju respon dari pembangkit itu sendiri. Sederhananya *DED* merupakan *economic dispatch* biasa yang dilakukan dalam beberapa kali sesuai dengan periode yang diminta (biasanya dua belas jam atau dua puluh empat jam).

Tujuan dasar dari *DED* adalah untuk menentukan keluaran daya yang ekonomis dari unit-unit pembangkit listrik yang nantinya akan memenuhi kebutuhan beban permintaan yang juga memenuhi permasalahan-permasalahan operasi untuk satu periode tertentu [4]. Fungsi objektif dari *DED* dinyatakan sebagai berikut (6).

$$f_T = \min \sum_{t=1}^T \sum_{i=1}^N f_i(P_i^t) \quad (6)$$

Dimana $f_i(P_i)$ adalah fungsi biaya dari pembangkit ke- i dan P_i^t adalah keluaran dari pembangkit ke- i pada periode waktu t . Sehingga $f_i(P_i^t)$ adalah fungsi biaya dari pembangkit ke- i pada periode waktu t .

Setiap pembangkit tentunya memiliki karakteristik dan kapasitas yang berbeda-beda. Tentunya terdapat persyaratan keluaran dari tiap-tiap pembangkit. Salah satunya adalah keluaran dari pembangkit tidak boleh melebihi batas daya maksimum dari unit pembangkit tersebut serta tidak boleh kurang dari batas daya minimum dari unit pembangkit tersebut, dijabarkan sebagai berikut (7).

$$P_{i,min} \leq P_i \leq P_{i,maks} \quad (7)$$

Dengan adanya batas maksimum dan minimum dari pembangkit, *DED* memiliki syarat tersendiri untuk menentukan pembebanan yang akan diberikan setiap jam dari periode tertentu tersebut. Setiap beban dalam interval waktu tertentu akan mengalami penambahan atau pengurangan beban menuju ke interval berikutnya. Namun penambahan dan pengurangan beban tersebut diatur untuk tetap menjaga karakteristik dari pembangkit dan tidak keluar dari batas maksimum dan minimum dari pembangkit tersebut (8).

$$\begin{aligned} P_i^t - P_i^{t-1} &\leq UR_i \\ P_i^{t-1} - P_i^t &\leq DR_i \end{aligned} \quad (8)$$

Dari dua persamaan diatas, terdapat UR_i dan DR_i . UR_i adalah batasan untuk penambahan beban setiap pergantian interval tertentu. Sedangkan DR_i adalah batasan untuk pengurangan beban setiap interval tertentu. UR_i dan DR_i memiliki nilai tertentu agar pembebanan setiap intervalnya tetap berada dalam batasan daya minimum dan maksimum dari pembangkit tersebut sesuai dengan persamaan (7).

Selain dari batasan penambahan dan pengurangan beban setiap pergantian interval ada lagi batasan yang harus diperhatikan. Tentunya untuk kondisi pembangkitan yang sesuai dibutuhkan batasan keseimbangan daya. Batasan keseimbangan daya adalah syarat dimana jumlah keluaran pembangkit pada waktu t harus sama dengan daya permintaan pada waktu t atau yang disebut P_i^t . Batasan ini harus ada agar daya yang dibangkitkan sesuai dengan daya yang dibutuhkan sehingga sistem kerja pembangkit mendekati titik optimal. Jika ada rugi-rugi transmisi, rugi-rugi transmisi tersebut harus dianalisa juga sehingga jumlah daya yang dibangkitkan harus sama dengan daya permintaan ditambah dengan rugi-rugi transmisinya, dijabarkan sebagai berikut (9).

$$\sum_{t=1}^N P_i^t = P_D^t + P_L \quad (9)$$

Dimana P_L adalah rugi-rugi transmisi. Rugi-rugi transmisi dapat digambarkan dengan matriks. Matriks rugi-rugi transmisi biasa dikenal sebagai matriks- β dan penjabaran perhitungan rugi-rugi transmisi adalah sebagai berikut (10).

$$P_L = \sum_{i=1}^n \sum_{j=1}^n P_i B_{ij} P_j \quad (10)$$

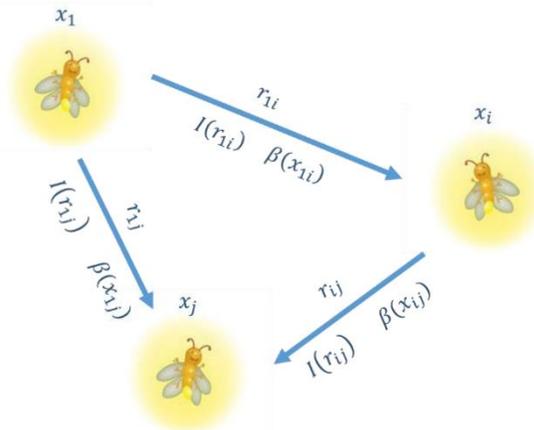
2.6 Firefly Algorithm

Algoritma Kunang-kunang atau dikenal sebagai *Firefly Algorithm (FA)* adalah salah satu penemuan metode kawanan cerdas yang dikembangkan oleh Xin She Yang pada tahun 2008. *FA* adalah betuk algoritma yang terinspirasi dari stokastik alami meta-heuristik yang diaplikasikan untuk menyelesaikan permasalahan optimasi tersulit. Algoritma stokastik artinya algoritma ini menggunakan fungsi acak untuk mencari solusi. Hal ini terinspirasi dari menyalnya cahaya pada kunang-kunang di alam. Heuristik berarti untuk mencari atau untuk menjelajah solusi dengan cara *trial and error*. Pada algoritma ini, untuk tahap yang rendah berarti heuristik, sedangkan untuk tahap yang lebih tinggi atau kompleks disebut meta-heuristik. Tahap rendah terfokus pada

pembangkitan solusi baru dengan cara mencari suatu ruang pencarian dan menentukan solusi terbaik. Berbeda dengan tahap yang lebih tinggi, meta-heuristik berarti mencari solusi terbaik dengan cara acak. Juga algoritma ini adalah algoritma yang berlandaskan populasi dimana algoritma populasi memiliki banyak kelebihan dibanding dengan algoritma yang lainnya [6][7].

2.6.1 Dasar Biologis

Kunang-kunang adalah salah satu serangga yang memikat, dan masa-masa pendekatan dari kunang-kunang telah menginspirasi penyair dan para ilmuwan. Pada masa sekarang, kunang-kunang memiliki dua ribu spesies yang tersebar di seluruh dunia. Biasanya, kunang-kunang tinggal di lingkungan yang hangat dan paling aktif saat malam hari di musim panas. Banyak peneliti yang telah mempelajari fenomena kunang-kunang di alam bebas dan sudah ada beberapa penelitian tertulis mengenai kunang-kunang. Kunang-kunang dikarakteristikan sebagai serangga yang memiliki kedipan cahaya yang dihasilkan dari proses biokimia yaitu bioluminescence. Cahaya yang berkedip seperti itu dapat berfungsi sebagai sinyal utama suatu pendekatan. Selain menarik pasangan, cahaya yang berkedip itu juga dapat berfungsi untuk memperingati potensi kedatangan predator.



Gambar 2.4 Ilustrasi Pergerakan Kunang-Kunang

Perlu diperhatikan bahwa dalam beberapa spesies kunang-kunang, beberapa kunang-kunang dewasa mampu memproduksi *bioluminescence*. Subspesies menarik pasangan mereka melalui feromon, yang mirip dengan kinerja semut. Traksi *bioluminescent* dilakukan oleh kunang-kunang dari organ yang memproduksi cahaya yang dinamakan lentera. Organisme yang paling *bioluminescent* hanya menyediakan sedikit modulasi kedipan (juga bersinar). Sebaliknya, spesies kunang-kunang dewasa mampu mengendalikan *bioluminescence* mereka untuk memancarkan kedipan cahaya yang tinggi dan berlain-lain. Produksi cahaya dari lentera diinisialisasi oleh sinyal yang berasal dari saraf pusat kunang-kunang.

Kebanyakan spesies kunang-kunang mengandalkan sinyal pendekatan *bioluminescent*. Biasanya, pemberi sinyal pertama merupakan kunang-kunang jantan yang mencoba untuk menarik kunang-kunang betina di darat. Kemudian kunang-kunang betina merespon dengan memancarkan cahaya secara berkedip atau berkelanjutan. Kedua pasangan memproduksi pola sinyal kedipan yang berbeda yang dilakukan dalam waktu yang tepat untuk mengodekan informasi seperti identitas spesies dan jenis kelamin. Kunang-kunang betina tertarik pada perbedaan perilaku saat sinyal pendekatan. Biasanya, kunang-kunang betina lebih memilih kunang-kunang jantan yang memiliki kedipan cahaya lebih terang. Hal ini sudah terkenal bahwa intensitas cahaya bervariasi, tergantung dari jarak sumbernya. Beruntungnya, dalam beberapa spesies kunang-kunang betina tidak dapat membedakan antara kedipan yang lebih jauh yang dihasilkan oleh sumber cahaya yang lebih kuat dan kedipan yang lebih dekat yang dihasilkan oleh sumber cahaya yang lebih lemah.

Dua fitur yang merupakan karakteristik dari kawanan cerdas ini adalah pengaturan mandiri dan pengambilan keputusan yang terdesentralisasi. Di sini, individu otonom hidup bersama di tempat umum, seperti misalnya lebah di dalam sarang lebah, semut di dalam sarang semut, dan lain sebagainya. Untuk hidup harmonis, dibutuhkan suatu interaksi atau komunikasi di antara anggota kelompok yang tinggal bersama. Faktanya, individu di dalam suatu kelompok tidak dapat berperilaku seolah-olah mereka hidup sendiri, tetapi harus beradaptasi dengan tujuan keseluruhan dalam suatu kelompok. Kehidupan sosial suatu kunang-kunang tidak hanya didedikasikan untuk mencari makan, namun lebih pada reproduksi. Keputusan kolektif ini terkait erat

hubungannya dengan perilaku kedipan cahaya yang berfungsi sebagai dasar biologis utama untuk mengembangkan FA [6].

2.6.2 Perilaku Kunang-Kunang

Terdapat dua ribu spesies kunang-kunang dan kebanyakan kunang-kunang memproduksi kedipan yang berirama. Pola dari kedipan, banyaknya kedipan dan rentang waktu terjadi kedipan diobservasi bersama membentuk sejenis *prototype* yang menarik kedua jenis, baik kunang-kunang jantan maupun betina kepada satu sama lain. Spesies betina bertindak sebagai respons terhadap *prototype* individu dari spesies jantan.

Intensitas cahaya pada suatu jarak tertentu(r) dari sumber cahaya sesuai dengan hukum kuadrat terbalik. Intensitas cahaya I menurun saat jarak r akan meningkat dengan kondisi $I = 1/r^2$. Selain itu, udara terus menyerap cahaya dan akan menyebabkan cahaya melemah dengan jarak yang meningkat. Dua faktor ini jika digabungkan akan membuat kunang-kunang terlihat pada jarak terbatas, biasanya hingga beberapa ratus meter di malam hari, yang cukup bagi kunang-kunang untuk berkomunikasi satu sama lain [6][7].

2.6.3 Konsep

Beberapa karakteristik kunang-kunang dapat dikhususkan untuk mengembangkan algoritma yang terinspirasi dari kunang-kunang. Karakteristik kedipan cahaya dari kunang-kunang digunakan untuk mengembangkan algoritma yang terinspirasi dari kunang-kunang. FA bekerja atas tiga aturan dan tiga faktor utama. Faktor utama meliputi intensitas cahaya dan daya tarik, jarak dan pergerakan kunang-kunang.

Terdapat beberapa aturan yaitu pertama, semua kunang-kunang adalah *unisex*, sehingga berarti bahwa satu kunang-kunang dapat tertarik pada kunang-kunang lainnya tanpa memperdulikan jenis kelamin.

Kedua, daya tarik dan kilauan memiliki timbal balik satu sama lain, yang kurang terang akan bergerak menuju ke yang lebih terang. Jika salah satu kunang-kunang lebih terang dari kunang-kunang lainnya, maka kunang-kunang tersebut akan berpindah secara acak.

Ketiga, tingkat cahaya dari suatu kunang-kunang ditentukan oleh pandangan fungsi obyektif. Untuk masalah maksimalisasi, kecerahan hanya sebanding dengan nilai fungsi obyektif. Bentuk lain dari kecerahan dapat didefinisikan dengan cara yang sama dengan kesesuaian fungsi dalam genetic algorithms [6].

Intensitas Cahaya dan Daya Tarik: Cahaya diserap oleh suatu media, sehingga kita harus memungkinkan daya tarik bervariasi dengan berbagai tingkat penyerapan. Intensitas cahaya $I(r)$ bervariasi menurut hukum kuadrat terbalik (11).

$$I(r) = \frac{1}{r^2} \quad (11)$$

Di mana I merupakan intensitas pada sumbernya. Untuk suatu medium yang dinyatakan dengan koefisien penyerapan cahaya tetap yaitu γ , intensitas cahaya I bervariasi dengan jarak r .

$$I = I_o e^{-\gamma r} \quad (12)$$

Di mana I_o merupakan intensitas cahaya awal, juga untuk daya tarik yang seimbang dengan intensitas cahaya. Sekarang dapat didefinisikan daya tarik β dari kunang-kunang sebagai berikut (13).

$$\beta = \beta_o e^{-\gamma r} \quad (13)$$

Jarak antara dua kunang-kunang i dan j pada x_i dan x_j masing-masing merupakan jarak kartesius (*Cartesian distance*).

$$r_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (14)$$

Pergerakan dari kunang-kunang yang tertarik pada yang lebih terang ditentukan oleh (15) [6].

$$x_i = x_i + \beta_o e^{\gamma r_{i,j}^2} (x_i - x_j) + \alpha \varepsilon_1 \quad (15)$$

2.7 Self-Adaptive Modified Firefly Algorithm

Terdapat tiga hal yang dapat memberikan performa *FA* yang lebih baik yaitu, faktor pergerakan acak (α). Semakin banyaknya faktor α akan membuat partikel mampu mencari solusi yang semakin luas. Kemudian faktor koefisien penyerapan (γ) yang mengatur intensitas cahaya yang mempengaruhi tingkat efektivitas dari partikel mencari solusi terbaik [12][13]. Kemudian faktor tambahan lain adalah faktor α^k yaitu faktor adaptive tuning, dimana faktor intensitas cahaya γ akan diatur secara sendirinya (*self adaptively*) untuk meningkatkan efisiensi.

Metode inilah yang disebut sebagai *Self-Adaptive Modified Firefly Algorithm*. α^k diatur secara dinamis (16).

$$\alpha^{k+1} = \left(\frac{1}{2k_{max}}\right)^{1/k_{max}} \alpha^k \quad (16)$$

SAMFA memiliki variabel tambahan dari metode *FA* yang biasa. Penambahan variabel ini berdasarkan dengan penambahan faktor adaptive tuning α^k . Faktor α^k akan mempengaruhi faktor intensitas cahaya dengan sendirinya yang dijabarkan dalam persamaan (17).

$$P_{m,t}^{k,self-adaptive} = [P_{m,t,1}^k \ P_{m,t,2}^k \ \dots \ P_{m,t,NG}^k \ \gamma_{m,t,NG+1}^k]$$

$$m = 1, \dots, N_{firefly}$$

$$t = 1, \dots, NT \quad (17)$$

Penambahan variabel baru tidak mempengaruhi cahaya tiap kunang-kunang, tapi secara sendirinya akan mengatur tingkat keterangan cahaya untuk meningkatkan efisiensi pencarian solusi terbaik [8][14].

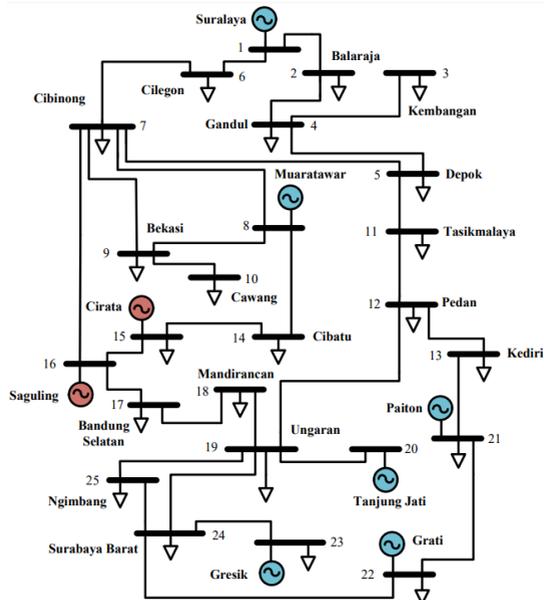
Halaman ini sengaja dikosongkan

BAB 3

IMPLEMENTASI SELF-ADAPTIVE MODIFIED FIREFLY ALGORITHM PADA DYNAMIC ECONOMIC DISPATCH

3.1 Sistem Pembangkitan Jawa-Bali

Pada tugas akhir mengenai *DED* ini menggunakan sistem pembangkitan Jawa-Bali 500 kV yang terdiri dari 23 bus dengan 28 saluran transmisi dan 8 pembangkit. Pembangkitnya adalah Suralaya, Muaratawar, Cirata, Saguling, Tanjungjati, Gresik, Paiton, dan Grati. Dari 8 pembangkit tersebut, Saguling dan Cirata adalah pembangkit listrik tenaga air (PLTA), sedangkan yang lainnya adalah pembangkit listrik tenaga uap (PLTU). Pembangkit Suralaya berfungsi sebagai *slack generator*. Data dari pembangkit Jawa-Bali diambil dari PT. PLN (Persero) Jawa-Bali, Maret 2009. *Single Line Diagram* dari Pembangkit Jawa-Bali dapat dilihat sebagai berikut.



Gambar 3.1 *Single Line Diagram* Pembangkit Jawa-Bali 500 kV

3.1.1 Data-data Pembangkit Jawa-Bali 500 kV

Berikut adalah data-data dari Pembangkit Jawa-Bali 500 kV yang diambil pada bulan Maret tahun 2009. Data yang diambil bersumber dari PT. PLN (Persero). Data yang diambil berupa data bus, data saluran, data beban, batasan kemampuan pembangkit, dan *cost function* pembangkit.

Tabel 3.1 Data Pembangkitan Sistem Interkoneksi 500 kV Jawa-Bali

No. Bus	Bus Name	Specified Bus	Volt Magnitude	Angle Degree	Load MW MVAR	
1	Suralaya	Slack	1.02	0	146	43
2	Cilegon	Load	1	0	672	217
3	Kembangan	Load	1	0	727	249
4	Gandul	Load	1	0	521	174
5	Cibinong	Load	1	0	667	206
6	Cawang	Load	1	0	727	174
7	Bekasi	Load	1	0	618	163
8	Muaratawar	Generator	1	0	0	0
9	Cibatu	Load	1	0	787	304
10	Cirata	Generator	1	0	651	234
11	Saguling	Generator	1	0	0	0
12	Bandung Sel.	Load	1	0	564	336
13	Mandiracan	Load	1	0	380	130
14	Ungaran	Load	1	0	314	347
15	Tanjungjati	Generator	1	0	0	0
16	Surabaya Barat	Load	1	0	824	304
17	Gresik	Generator	1	0	201	87
18	Depok	Load	1	0	0	0
19	Tasikmalaya	Load	1	0	265	16
20	Pedan	Load	1	0	501	233
21	Kediri	Load	1	0	343	197
22	Paiton	Generator	1	0	803	260
23	Grati	Generator	1	0	125	184

Tabel 3.2 Data Saluran Sistem Interkoneksi 500 kV Jawa-Bali

From Bus	To Bus	R (pu)	X (pu)	B (pu)
1	2	0.000626496	0.007008767	0
1	4	0.006513273	0.062576324	0.01197964
2	5	0.013133324	0.146925792	0.007061141

Tabel 3.2 Data Saluran Sistem Interkoneksi 500 kV Jawa-Bali (lanjutan)

<i>From Bus</i>	<i>To Bus</i>	<i>R (pu)</i>	<i>X (pu)</i>	<i>B (pu)</i>
3	4	0.001513179	0.016928309	0
4	5	0.001246422	0.01197501	0
4	18	0.000694176	0.006669298	0
5	7	0.00444188	0.0426754	0
5	8	0.0062116	0.059678	0
5	11	0.00411136	0.04599504	0.008841946
6	7	0.001973648	0.01896184	0
6	8	0.0056256	0.054048	0
8	9	0.002822059	0.027112954	0
9	10	0.00273996	0.026324191	0
10	11	0.001676728	0.014168458	0
11	12	0.0019578	0.0219024	0
12	13	0.00699098	0.0671659	0.01285827
13	14	0.013478	0.12949	0.024789624
14	15	0.01353392	0.15140736	0.007276522
14	16	0.01579856	0.1517848	0.007264438
14	20	0.00903612	0.0868146	0
15	16	0.037539629	0.360662304	0.017261339
16	17	0.00139468	0.0133994	0
16	23	0.003986382	0.044596656	0
18	19	0.014056	0.157248	0.030228874
19	20	0.015311	0.171288	0.032927881
20	21	0.010291	0.115128	0.022131855
21	22	0.010291	0.115128	0.022131855
22	23	0.004435823	0.049624661	0.009539693

Tabel 3.3 Cost Function Sistem Interkoneksi 500 kV Jawa-Bali

<i>no.</i>	<i>Generating Unit</i>	<i>Cost Function</i>			<i>Valve-point</i>	
		<i>a</i>	<i>b</i>	<i>c</i>	<i>e</i>	<i>f</i>
1	Suralaya	-65.94	395668.05	31630.21	100	0.063
2	Muaratawar	690.98	2478064.47	107892572.17	200	0.084
3	Cirata	0	6000	0	200	0.042
4	Saguling	0	5502	0	300	0.042

Tabel 3.3 *Cost Function* Sistem Interkoneksi 500 kV Jawa-Bali (lanjutan)

No.	Generating Unit	Cost Function			Valve-point	
		a	b	c	e	f
5	Tanjungjati	-21.88	197191.76	-1636484.18	100	0.042
6	Gresik	132.15	777148.77	13608770.96	100	0.077
7	Paiton	52.19	37370.67	8220765.38	200	0.077
8	Grati	533.92	2004960.63	86557397.40	300	0.063

Tabel 3.4 Batas Kemampuan Sistem Interkoneksi 500 kV Jawa-Bali

No.	Generating Unit	Power (MW)	
		Minimum	Maximum
1	Suralaya	1500	3400
2	Muaratawar	1040	2200
3	Cirata	600	1220
4	Saguling	400	900
5	Tanjungjati	600	1220
6	Gresik	238	1050
7	Paiton	1425	3254
8	Grati	150	827

3.2 Fungsi Biaya Tidak Mulus

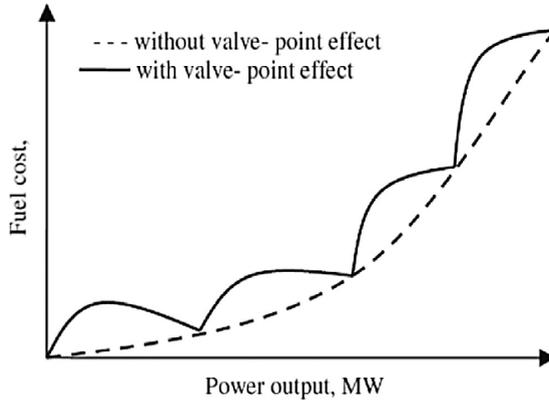
Fungsi biaya tidak mulus merupakan penambahan variabel dalam *cost function* dari sebuah pembangkit termal. Hal ini terjadi dikarenakan pengaruh dari membukanya katup uap yang berfungsi untuk menambah putaran turbin sehingga terjadi riak pada *cost function*. Pengaruhnya adalah pada *cost function* sebuah pembangkit akan ada penambahan fungsi baru yaitu fungsi *sinusoidal*. Penambahan fungsi *sinusoidal* tersebut dijabarkan sebagai berikut

$$F_i(P_i) = \alpha_i P_i^2 + \beta_i P_i + \gamma_i + |e_i x \sin(f_i x (P_{i,min} - P_i))| \quad (5)$$

Dimana, $F_i(P_i)$ adalah *cost function* pembangkit ke- i , dalam hal ini, *cost function* yang digunakan dalam satuan rupiah per jam. P_i adalah daya keluaran dari pembangkit ke- i dalam satuan MW. $P_{i,min}$ adalah daya minimum yang dibangkitkan sesuai dengan batasan kemampuan pembangkit dalam satuan MW. α_i , β_i , γ_i adalah koefisien biaya dari

unit pembangkit, e_i serta f_i adalah koefisien dari efek katup uap, yaitu koefisien fungsi biaya tidak mulus.

Secara umum, gambaran efek katup uap dibandingkan dengan *cost function* tanpa efek katup uap adalah sebagai berikut.



Gambar 3.2 Kurva *Cost Function* dengan dan tanpa Efek Katup Uap

Akibat dari efek katup uap ini, *cost function* terjadi penambahan fungsi *sinusoidal* pada fungsi kuadrat biaya, yaitu dengan koefisien e_i serta f_i [2].

3.3 Implementasi FA pada DED

FA adalah salah satu algoritma kawanan cerdas yang terinspirasi oleh kebiasaan alami kunang-kunang. FA ditemukan untuk menyelesaikan permasalahan optimasi yang sangat sulit bahkan paling sulit [6]. Penggunaan FA untuk DED dilakukan dengan program MATLAB. Ada beberapa parameter untuk melakukan optimasi dengan FA itu sendiri seperti, alpha (α), beta (β), beta min. (β_{min}), beta nol (β_0), dan gamma (γ).

3.3.1 Inisiasi Awal FA untuk menyelesaikan DED

Inisiasi awal FA diperlukan agar program yang dibangun dalam MATLAB berjalan dengan parameter-parameter yang akan digunakan. Inisiasi awal yang diperlukan adalah,

- alpha (α) = 0.5
- beta minimum (β_{min}) = 0.98

- γ = 1.0
- Jumlah populasi = 20
- Jumlah iterasi = 50

Parameter alpha (α) adalah parameter yang merupakan faktor untuk mengatur pergerakan acar dari kunang-kunang untuk mencari solusi. Umumnya parameter alpha ini besarnya antara 0 sampai 1. Parameter alpha ini sangat efektif dalam meningkatkan performa dari FA sendiri. Alpha yang besar membuat pergerakan untuk mencari solusi dalam ruang pencarian yang cukup jauh, sedangkan ukuran alpha yang kecil membuat ruang pencarian semakin kecil, cenderung memfasilitasi pencarian lokal [8].

Parameter gamma (γ) merupakan koefisien penyerapan kunang-kunang. Parameter gamma ini mengatur penurunan intensitas cahaya kunang-kunang dan biasanya nilai gamma sudah ditentukan sebesar 1 seperti pada [10].

Pengaturan pada parameter-parameter yang merupakan inisiasi awal dari FA sangat berpengaruh pada efektivitas FA untuk menyelesaikan permasalahan optimasi.

Selain inisiasi parameter awal dari FA , juga dimasukkan data-data dari sistem interkoneksi Jawa-Bali 500 kV.

3.3.2 Inisiasi Posisi Awal Partikel/Populasi

Inisiasi posisi awal partikel dilakukan untuk mencari solusi awal sebelum dilakukan iterasi. Hasil dari solusi awal ini nantinya akan masuk ke iterasi dengan pendekatan FA untuk didapatkan solusi yang lebih baik dari sebelumnya. Untuk menghitung atau mencari solusi awal digunakan rumus sebagai berikut (18).

$$PosisiAwal(i) = ((P(i)_{max} - P(i)_{min}) \times Rand) + P(i)_{min} \quad (18)$$

Dimana posisi awal merupakan solusi awal yang mewakili besarnya daya yang dibangkitkan oleh unit pembangkit. $PosisiAwal(i)$ adalah besarnya daya yang dibangkitkan oleh pembangkit ke- i . $P(i)_{max}$ adalah batas maksimum dari unit pembangkit, dan $P(i)_{min}$ adalah batas minimum dari unit pembangkit. $P(i)_{max}$ dan $P(i)_{min}$ diambil dari batasan yang diberikan sesuai dengan spesifikasi pembangkit. Kemudian $Rand$ adalah nilai acak dengan rentang 0 sampai 1.

3.3.3 Korektor Posisi Awal

Korektor posisi awal bertujuan agar jumlah daya yang dibangkitkan diawal tidak melewati atau kurang dari nilai permintaan beban yang diminta. Karena *DED* bersifat dinamis, maka korektor dilakukan sebanyak jumlah iterasi karena nilai permintaan beban juga berbeda-beda. Perhitungan korektor posisi awal jika nilai solusi awal kurang dari permintaan beban adalah sebagai berikut (19).

$$Korektor(K) = \text{Beban} - \left(\frac{\text{Jumlah Solusi Awal}}{\text{Jumlah Pembangkit}} \right) \quad (19)$$

Kemudian karena nilai solusi awal masih kurang dari jumlah permintaan beban, maka nilai solusi awal akan ditambahkan dengan nilai korektor, sehingga hasil penjumlahan tersebut akan menjadi nilai posisi awal yang valid. Kemudian, jika nilai solusi awal melebihi dari permintaan beban, perhitungan korektor adalah sebagai berikut (20).

$$Korektor(K) = \left(\frac{\text{Jumlah Solusi Awal}}{\text{Jumlah Pembangkit}} \right) - \text{Beban} \quad (20)$$

Kemudian karena nilai solusi awal yang melebihi dari jumlah permintaan beban, maka solusi awal akan dikurangi dengan nilai korektor, sehingga hasil pengurangan tersebut akan menjadi nilai posisi awal yang valid. Intinya penentuan nilai korektor adalah untuk mengoreksi nilai solusi awal agar sesuai dengan permintaan beban.

3.3.4 Update Intensitas Cahaya Kunang-kunang

Intensitas cahaya kunang-kunang diwakilkan oleh parameter koefisien penyerapan kunang-kunang yaitu γ . Tujuan *update* parameter γ ini adalah untuk menentukan arah gerak kunang-kunang. Pada dasarnya sifat alami kunang-kunang adalah bergerak menuju ke kunang-kunang yang cahayanya lebih terang. Maka dari itu *update* intensitas cahaya bertujuan untuk mencari solusi yang lebih baik daripada solusi awal. Perhitungan untuk *update* intensitas cahaya kunang-kunang dijabarkan sebagai berikut (21).

$$\text{Beta}(\beta) = (\beta_o) \times \text{exponent}(-\gamma \cdot r^2) \quad (21)$$

Dimana nilai gamma yang awalnya sebagai parameter intensitas cahaya dari kunang-kunang akan berubah menjadi beta. Beta merupakan *update* dari nilai gamma. Parameter beta menandakan kunang-kunang yang lebih terang dan lebih menarik sehingga kunang-kunang yang lain akan bergerak menuju kunang-kunang yang lebih terang.

3.3.5 Update Posisi Kunang-kunang

Setelah parameter intensitas cahaya, tentunya kunang-kunang akan bergerak menuju kunang-kunang lain yang lebih terang. Hal ini adalah sifat alami dari kunang-kunang di alam. Maka dari itu posisi dari kunang-kunang juga akan di *update* agar dapat menentukan nilai daya terbangkit yang baru yang sesuai dengan intensitas cahaya yang lebih terang (solusi yang lebih baik). *Update* posisi kunang-kunang dilakukan dengan perhitungan sebagai berikut (22).

$$Posisi_{new}(i) = Posisi_{old}(i) \times (1 - \beta) + (Posisi_{old} \times \beta) + tmpf \quad (22)$$

Dimana *tmpf* adalah *update* dari parameter α karena untuk *update* posisi dari kunang-kunang tentunya juga *update* pergerakan kunang-kunang itu sendiri. Perhitungan *tmpf* adalah (23).

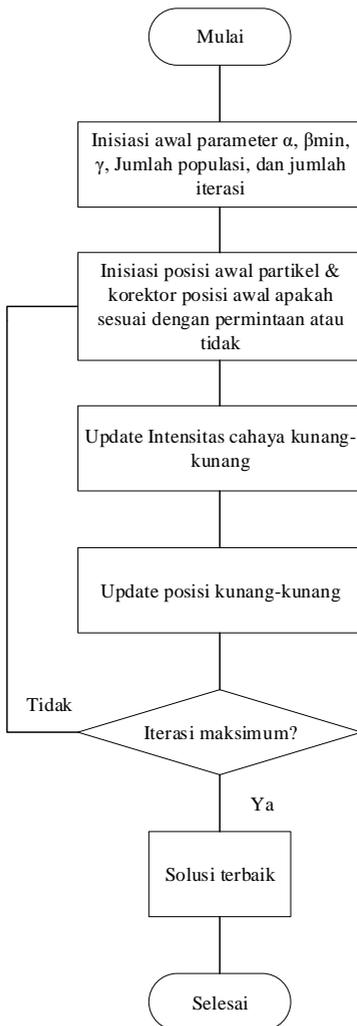
$$tmpf = \alpha \times (Rand - 0,5) \quad (23)$$

Karena nilai *Rand* adalah nilai acak dari 0 sampai 1, maka nilai *tmpf* dapat bernilai positif maupun negatif.

Proses ini berlangsung berkali-kali sesuai dengan jumlah iterasi yang diinginkan. Setelah proses ini terus dilakukan sesuai dengan iterasi yang diinginkan, nilai yang didapatkan akan semakin membaik menuju nilai solusi yang paling baik. Nilai solusi yang paling baik ini merupakan nilai optimal seberapa besar daya yang dibangkitkan oleh tiap-tiap unit pembangkit dan sesuai dengan permintaan beban pada suatu waktu tertentu.

3.3.6 Flowchart FA untuk DED

Proses untuk menyelesaikan permasalahan DED menggunakan FA dapat digambarkan dalam bentuk *flowchart* sebagai berikut.



Gambar 3.3 Flowchart Implementasi FA pada DED

3.4 Implementasi SAMFA pada DED

SAMFA adalah metode modifikasi yang dilakukan pada *FA*. Terdapat tiga parameter yang sangat mempengaruhi performa dari *FA* itu sendiri. Yaitu parameter α yang merupakan parameter pergerakan acak dari kunang-kunang, β , dan γ yang merupakan parameter penyerapan yang mengatur intensitas cahaya dari kunang-kunang. Pada *FA*, parameter pergerakan acak dari kunang-kunang atau alpha sudah diatur diawal iterasi, dan nilai alpha adalah tetap sepanjang iterasi berlangsung. Namun berbeda dengan *SAMFA*. *SAMFA* memiliki kelebihan dimana parameter pergerakan acak akan berubah-ubah setiap iterasinya. Melihat pada sifat alami kunang-kunang, kunang-kunang memiliki pergerakan yang berbeda-beda. Maka dari itu *SAMFA* memodifikasi *FA* menjadi lebih adaptif.

Pada *SAMFA*, nilai alpha akan dilakukan penyetelan secara dinamis setiap iterasinya. Dengan penyetelan nilai alpha tentunya akan mempengaruhi nilai *tmpf*. Hal ini dilakukan agar meningkatkan efisiensi dari metode pendekatan tersebut. Nilai alpha disetel setiap iterasinya dengan jабaran sebagai berikut (24) [8].

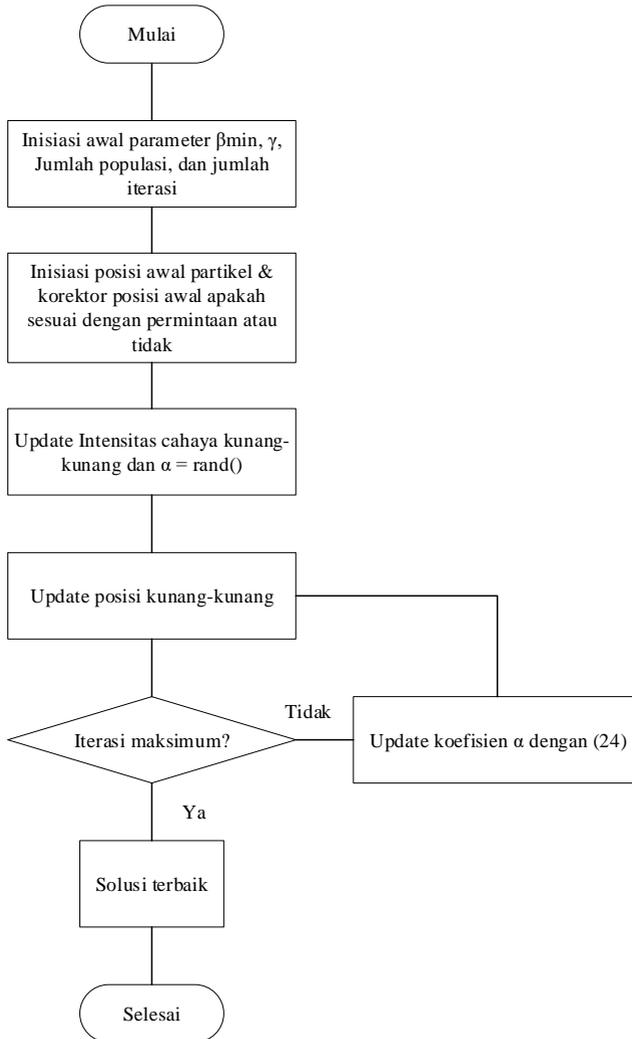
$$\alpha^{k+1} = \left(\frac{1}{2k_{max}} \right)^{1/k_{max}} x \alpha^k \quad (24)$$

Cara kerja *SAMFA* tentunya memiliki perbedaan dengan *FA* pada umumnya. Perlu dititik beratkan bahwa variabel baru tidak dimasukkan dalam perhitungan kecerahan masing-masing kunang-kunang, tetap variable baru disetel agar dapat menyesuaikan diri untuk meningkatkan karakteristik konversi kunang-kunang dalam ruang pencarian solusi dan juga mengurangi risiko jatuh dalam salah satu minima lokal.

Jadi tujuan dari penelitian *SAMFA* ini adalah untuk menambahkan operasi mutasi baru ke dalam *FA* yang biasa, sehingga akhirnya kinerja algoritma akan ditingkatkan secara efektif yang memberi dampak hasil perhitungan dan iterasi yang lebih optimal dan meningkatkan kualitas solusi [11].

3.4.1 Flowchart SAMFA untuk DED

Proses untuk menyelesaikan permasalahan DED dengan SAMFA dapat digambarkan dalam bentuk *flowchart* sebagai berikut.



Gambar 3.4 Flowchart Implementasi SAMFA pada DED

Halaman ini sengaja dikosongkan

BAB 4 SIMULASI DAN ANALISIS

Pada bab ini akan dilakukan simulasi dan analisis menggunakan dua metode yang memiliki dasar yang sama. Kedua simulasi ini dilakukan dengan tujuan membandingkan metode biasa dengan metode yang telah dimodifikasi untuk mendapatkan hasil mana yang lebih baik. Bab ini akan membahas simulasi *DED* menggunakan *FA* dan *SAMFA* yang dilakukan pada sistem interkoneksi 500 kV Jawa-Bali. Simulasi *DED* dilakukan selama 24 jam dengan interval 1 jam. *DED* yang dilakukan pada sistem interkoneksi 500 kV Jawa-Bali dianggap tidak memiliki *losses*. Pada bab ini akan dibagi menjadi dua bagian, bagian pertama adalah simulasi *DED* dengan metode *FA*, dan bagian kedua adalah simulasi *DED* dengan metode *SAMFA*. Tujuan dilakukan perbandingan metode agar dapat mengevaluasi performa dari metode *SAMFA* untuk menyelesaikan masalah optimasi.

Pada simulasi yang akan dilakukan terdapat beberapa batasan selain unit pembangkit dianggap tidak memiliki *losses*, pembangkit dipengaruhi efek katup uap sehingga fungsi biaya menjadi tidak mulus, dan juga semua unit dalam operasinya dianggap menyala semua.

Pada setiap simulasi akan menghasilkan data-data berupa daya yang akan dibangkitkan setiap unit, biaya pembangkitan setiap unit selama satu interval, biaya dari efek katup uap, dan total biaya pembangkitan semua unit yang dipengaruhi efek katup uap.

4.1 Validasi Program *FA*

Sebelum dilakukan simulasi terhadap sistem interkoneksi Jawa-Bali 500 kV, terlebih dahulu dilakukan validasi program *firefly algorithm* pada referensi buku Allan J. Wood. Diambil contoh dari buku tentang *economic dispatch* pada *example 3A* dan *example 3B*.

Tabel 4.1 Data Pembangkit *Example 3A*

Unit	Daya (MW)	
	Minimum	Maksimum
1	150	600
2	100	400
3	50	200

Dengan data daya dari setiap unit tersebut, tentunya memiliki karakteristik fungsi biaya masing-masing, Fungsi biaya ditampilkan sebagai berikut.

Tabel 4.2 Fungsi Biaya *Example 3A*

No.	Cost Function		
	a	b	c
1	0,001562	7,92	561
2	0,001940	7,85	310
3	0,004830	7,97	78

Beban permintaan pada contoh soal ini adalah sebesar 850 MW, dan permasalahan *economic dispatch* ini tidak menggunakan *losses* atau rugi-rugi pembangkitan. Sesuai dengan solusi iterasi lamda yang dilampirkan pada buku Allan J. Wood, daya masing-masing yang dibangkitkan adalah sebesar,

Tabel 4.3 Daya Terbangkit Setiap Unit *Example 3A*

Unit	Daya Terbangkit (MW)
1	392,2
2	334,6
3	122,2
Total	850

Daya yang dibangkitkan adalah 392,2 MW pada pembangkit 1, 334,6 MW pada pembangkit 2, dan 122,2 MW pada pembangkit 3. Dengan daya yang terbangkit, dimasukkan kedalam persamaan fungsi biaya, sehingga menghasilkan biaya sebesar **8194,3561 \$/h**. Lalu dilakukan simulasi dengan metode *FA* dan hasilnya adalah,

Tabel 4.4 Daya Terbangkit Setiap Unit *Example 3A (FA)*

Unit	Daya Terbangkit (MW)
1	393,5116
2	333,6609
3	122,8275
Total	850

Daya yang dibangkitkan adalah 393,5116 MW pada pembangkit 1, 333,6609 MW pada pembangkit 2, dan 122,8275 MW pada pembangkit 3. Dengan daya yang terbangkit tersebut dapat dimasukkan kedalam persamaan fungsi biayanya, sehingga menghasilkan biaya sebesar **8194,4 \$/h**. Dari hasil percobaan *economic dispatch 3A* dengan menggunakan *FA* memang memiliki perbedaan namun sangat kecil dan tidak signifikan.

Kemudian percobaan berikutnya adalah *example 3B* dengan fungsi biaya seperti berikut.

Tabel 4.5 Fungsi Biaya *Example 3B*

No.	Cost Function		
	a	b	c
1	0,001280	6,48	459
2	0,001940	7,85	310
3	0,004830	7,97	78

Beban permintaan pada contoh soal ini sama seperti *example 3A* adalah sebesar 850 MW, dan permasalahan *economic dispatch* ini tidak menggunakan *losses* atau rugi-rugi pembangkitan. Sesuai dengan solusi iterasi lamda yang dilampirkan pada buku Allan J. Wood, daya masing-masing yang dibangkitkan adalah sebesar,

Tabel 4.6 Daya Terbangkit Setiap Unit *Example 3B*

Unit	Daya Terbangkit (MW)
1	600
2	187,1
3	62,9
Total	850

Daya yang dibangkitkan adalah 704,6 MW, namun karena daya maksimum yang dapat dibangkitkan oleh pembangkit unit 1 sebesar 600 MW, sesuai dengan solusi iterasi lamda yang diberikan, pada pembangkit 1 disetting maksimum sebesar 600 MW, lalu 187,1 MW pada pembangkit 2, dan 62,9 MW pada pembangkit 3. Dengan daya yang terbangkit tersebut dapat dimasukkan kedalam persamaan fungsi biayanya, sehingga menghasilkan biaya sebesar **7252,83 \$/h**. Lalu dilakukan simulasi dengan metode *FA* dan hasilnya adalah sebagai berikut.

Tabel 4.7 Daya Terbangkit Setiap Unit *Example 3B (FA)*

Unit	Daya Terbangkit (MW)
1	599,9987
2	183,3715
3	66,6298
Total	850

Daya yang dibangkitkan adalah 599,9987 MW pada pembangkit 1, 183,3715 MW pada pembangkit 2, dan 66,6298 MW pada pembangkit 3. Dengan daya yang terbangkit tersebut dapat dimasukkan kedalam persamaan fungsi biayanya, sehingga menghasilkan biaya sebesar **7252,2 \$/h**. Dari hasil percobaan *economic dispatch 3B* dengan menggunakan *FA* memang memiliki perbedaan namun juga sangat kecil dan tidak signifikan. Sehingga dari kedua percobaan tersebut, metode *FA* yang akan dikembangkan telah tervalidasi.

4.2 Efek Katup Uap

Efek katup uap, seperti yang telah dibahas dalam bab sebelumnya, adalah efek atau pengaruh dari pembukaan katup uap pada pembangkit termal. Pengaruh dari pembukaan katup uap ini adalah membuat fungsi biaya pembangkitan yang memiliki kurva mulus menjadi tidak mulu (*sinusoidal*). Pada bagian ini, akan ditampilkan beberapa data dari efek katup uap yang mempengaruhi *DED* pada sistem interkoneksi 500 kV Jawa-Bali. Berikut adalah data-data dari pengaruh efek katup uap pada sistem interkoneksi 500 kV Jawa-Bali.

Table 4.8 Koefisien e dan f

	P1	P2	P3	P4	P5	P6	P7	P8
e	100	200	200	300	100	100	200	300
f	0,063	0,084	0,042	0,042	0,042	0,077	0,077	0,063

Dengan adanya penambahan variabel e dan f pada *cost function* pembangkit, maka terdapat perubahan *cost function* pada pembangkit itu sendiri. Perubahan *cost function* ditampilkan pada tabel berikut.

Table 4.9 Biaya Tidak Mulus

	P1	P2	P3	P4	P5	P6	P7	P8
<i>cost</i>	24,955	65,998	33,471	50,207	16,736	67,350	134,699	74,866

Dengan didapatkannya biaya tidak mulus setiap pembangkit, dapat dihitung total biaya tidak mulus untuk keseluruhan sistem. Diketahui biaya tidak mulus dari pembangkit 1 adalah sebesar 24,995, pembangkit 2 adalah 65,998, pembangkit 3 adalah 33,471, pembangkit 4 adalah 50,207, pembangkit 5 adalah 16,736, pembangkit 6 adalah 67,350, pembangkit 7 adalah 134,699, dan pembangkit 8 adalah 74,866. Sehingga jika diakumulasi, total biaya tidak mulus dari sistem interkoneksi 500 kV Jawa-Bali adalah 468,2823.

Tentunya karena biaya tidak mulus bernilai positif, maka total biaya pembangkitan akan lebih mahal jika terdapat efek katup uap dibanding dengan pembangkitan tanpa mempertimbangkan katup uap.

4.3 Metode *FA* untuk menyelesaikan *DED* dengan Mempertimbangkan Fungsi Biaya Tidak Mulus

Untuk menyelesaikan *DED* yang dipengaruhi oleh fungsi biaya tidak mulus akibat efek katup uap, diperlukan metode yang mampu menyelesaikan perhitungan *non-linear*. Metode *FA* adalah salah satu metode yang mampu menyelesaikan permasalahan *DED* tersebut. *DED* yang dilakukan menggunakan sistem interkoneksi 500 kV Jawa-Bali dalam periode 24 jam. Level beban yang digunakan adalah level beban yang diambil pada bulan Maret 2009 dari PT. PLN (Persero) yaitu.

Tabel 4.10 Profil Beban untuk 24 Jam

Jam ke-	Beban (MW)
1	10282
2	10232
3	10182
4	10132
5	10300
6	10500
7	10700
8	10750
9	10800

Tabel 4.10 Profil Beban untuk 24 Jam (lanjutan)

Jam ke-	Beban (MW)
10	10900
11	10850
12	10800
13	10750
14	10700
15	10650
16	10600
17	10700
18	10800
19	11000
20	10800
21	10700
22	10600
23	10650
24	10700

Sehingga total beban adalah 255078 MW. Koefisien dari pembangkit-pembangkit seperti a, b, c yang merupakan *cost function*, dan e, f yang merupakan efek katup uap adalah sebagai berikut.

Tabel 4.11 Data Koefisien Pembangkit

No.	Cost Function			Valve-point	
	a	b	c	e	f
P1	-65.94	395668.05	31630.21	100	0.063
P2	690.98	2478064.47	107892572.17	200	0.084
P3	0	6000	0	200	0.042
P4	0	5502	0	300	0.042
P5	-21.88	197191.76	-1636484.18	100	0.042
P6	132.15	777148.77	13608770.96	100	0.077
P7	52.19	37370.67	8220765.38	200	0.077
P8	533.92	2004960.63	86557397.40	300	0.063

Pembangkit-pembangkit tersebut juga memiliki batasan kemampuan pembangkitan. Batasan kemampuan pembangkitan ditampilkan dalam tabel seperti berikut.

Tabel 4.12 Batasan Pembangkit

Unit	Pmin (MW)	Pmax (MW)
P1	1500	3400
P2	1040	2200
P3	600	1220
P4	400	900
P5	600	1220
P6	238	1050
P7	1425	3254
P8	150	827

Dengan data-data yang diberikan dari **Tabel 4.4** dan **Tabel 4.5**, penggunaan metode *FA* tentunya juga memiliki parameter-parameter awal. Parameter-parameter awal tersebut berupa alpha (α) yang merupakan koefisien parameter pergerakan acak dari kunang-kunang, beta (β), beta minimum (β_{\min}), dan gamma (γ) yang merupakan koefisien parameter penyerapan cahaya, serta ditampilkan juga jumlah iterasi dan jumlah populasi dari kunang-kunang. Parameter-parameter tersebut adalah sebagai berikut.

Tabel 4.13 Parameter awal *FA* untuk *DED*

Parameter	Simbol	Nilai
Alpha	α	0,7574 (<i>random</i>)
Beta	β	0,5 [8]
Beta minimum	β_{\min}	0,98
Gamma	γ	1.0 [8]
Populasi	<i>firefly</i>	20
Iterasi	<i>iterasi_max</i>	50

4.3.1 Hasil Simulasi *DED* dengan menggunakan *FA*

Dengan parameter dan profil-profil pembangkit yang telah terdata di atas, maka dapat dilakukan simulasi *DED* dengan menggunakan metode *FA*. Simulasi dilakukan menggunakan aplikasi MATLAB 2014a. Hasil simulasi berupa pembebanan setiap unit dan biaya setiap unit dalam interval waktu 1 jam dalam total 24 jam akan ditampilkan sebagai berikut.

Tabel 4.14 Hasil Simulasi *FA* untuk *DED* Jam ke-1 sampai 3

Unit	Jam ke-1 (MW)	Jam ke-2 (MW)	Jam ke-3 (MW)
P1	3370,93	2583,78	3329,90
P2	1222,02	1186,77	1160,79
P3	810,57	1311,39	1089,46
P4	590,86	751,47	494,49
P5	967,69	1071,93	726,64
P6	735,42	1049,12	819,29
P7	2214,26	1977,94	2229,24
P8	370,25	299,60	332,20

Tabel 4.15 Hasil Simulasi *FA* untuk *DED* Jam ke-4 sampai 6

Unit	Jam ke-4 (MW)	Jam ke-5 (MW)	Jam ke-6 (MW)
P1	2914,56	2929,50	2642,65
P2	1126,10	1192,32	1155,42
P3	705,42	1326,68	1080,65
P4	755,54	730,97	571,13
P5	977,24	1209,28	983,35
P6	893,44	907,20	342,63
P7	2272,29	1638,64	3244,21
P8	487,41	365,39	479,96

Tabel 4.16 Hasil Simulasi *FA* untuk *DED* Jam ke-7 sampai 9

Unit	Jam ke-7 (MW)	Jam ke-8 (MW)	Jam ke-9 (MW)
P1	3073,31	2258,62	2815,16
P2	1002,18	1121,94	1194,46
P3	1022,85	1184,00	1332,03
P4	739,53	896,94	782,23
P5	959,19	1248,47	860,54
P6	226,20	772,23	987,67
P7	3068,23	2964,00	2493,28
P8	644,51	303,79	334,64

Tabel 4.17 Hasil Simulasi *FA* untuk *DED* Jam ke-10 sampai 12

Unit	Jam ke-10 (MW)	Jam ke-11 (MW)	Jam ke-12 (MW)
P1	3316,00	3325,84	2931,21
P2	1259,90	1274,11	1216,02
P3	1117,61	978,10	1060,10
P4	460,46	491,32	399,58
P5	812,72	895,58	1112,79
P6	393,17	529,57	798,29
P7	3190,16	3074,64	3007,94
P8	349,99	280,84	277,07

Tabel 4.18 Hasil Simulasi *FA* untuk *DED* Jam ke-13 sampai 15

Unit	Jam ke-13 (MW)	Jam ke-14 (MW)	Jam ke-15 (MW)
P1	3341,00	3261,49	2464,25
P2	1295,47	1118,35	1135,47
P3	899,96	901,83	1118,27
P4	924,77	455,06	817,02
P5	827,51	967,15	1091,15
P6	437,10	571,10	879,13
P7	2793,35	2973,64	2816,70
P8	230,84	451,36	328,02

Tabel 4.19 Hasil Simulasi *FA* untuk *DED* Jam ke-16 sampai 18

Unit	Jam ke-16 (MW)	Jam ke-17 (MW)	Jam ke-18 (MW)
P1	3249,86	2474,94	2855,87
P2	1148,54	1071,85	1230,35
P3	882,61	1088,57	1044,02
P4	454,65	724,90	610,33
P5	955,76	1090,64	995,32
P6	630,81	820,08	719,46
P7	2888,47	2979,49	3049,81
P8	391,31	449,53	294,85

Tabel 4.20 Hasil Simulasi *FA* untuk *DED* Jam ke-19 sampai 21

Unit	Jam ke-19 (MW)	Jam ke-20 (MW)	Jam ke-21 (MW)
P1	3150,03	3220,88	2914,56
P2	1155,58	1134,72	1151,70
P3	660,79	744,35	746,66
P4	647,84	621,28	659,68
P5	983,05	1138,46	1034,44
P6	720,18	232,43	397,02
P7	3183,68	3120,81	3354,09
P8	498,85	587,09	441,85

Tabel 4.21 Hasil Simulasi *FA* untuk *DED* Jam ke-22 sampai 24

Unit	Jam ke-22 (MW)	Jam ke-23 (MW)	Jam ke-24 (MW)
P1	3312,42	3291,92	3412,22
P2	1036,37	1237,88	1201,56
P3	864,19	1030,82	1147,85
P4	661,16	552,86	924,75
P5	777,74	858,75	1168,53
P6	582,60	565,48	541,87
P7	2780,95	2873,25	1825,88
P8	584,57	239,04	477,36

4.3.2 Biaya Total Pembangkitan (*FA*)

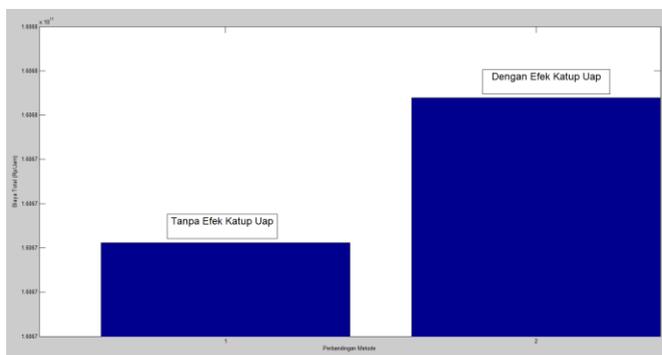
Telah dibahas sebelumnya bahwa dengan adanya efek katup uap yang membuat fungsi biaya tidak mulus akan membuat harga pembangkitan menjadi lebih mahal karena adanya penambahan biaya akibat koefisien e dan f . Biaya total setiap jam dalam total 24 jam adalah sebagai berikut.

Tabel 4.22 Biaya Total Pembangkitan (*FA*)

Jam ke-	Beban (MW)	Biaya Pembangkitan
1	10282	6234884535,44
2	10232	6296575628,84
3	10182	6242657852,72
4	10132	6213016387,99

Tabel 4.22 Biaya Total Pembangkitan (FA) (lanjutan)

Jam ke-	Beban (MW)	Biaya Pembangkitan
5	10300	6782338337,35
6	10500	6805442698,85
7	10700	6467378981,20
8	10750	6552679715,58
9	10800	6962337700,63
10	10900	6919101505,27
11	10850	6909953529,04
12	10800	6924728801,43
13	10750	6696954737,00
14	10700	6690745060,10
15	10650	6714035846,94
16	10600	6690535858,70
17	10700	6749593840,01
18	10800	6950464523,45
19 (peak)	11000	7182524216,53
20	10800	6884454828,96
21	10700	6798905106,72
22	10600	6620746431,25
23	10650	6621160209,42
24	10700	6765569351,77
Total	255078	160676785685,19



Gambar 4.1 Perbandingan Tanpa & Dengan Efek Katup Uap (FA)

Berdasarkan data-data yang telah didapatkan dari perhitungan *DED* dengan *FA* yang mempertimbangkan efek katup uap sehingga fungsi biaya menjadi tidak mulus, dapat dilihat bahwa pada beban jam ke-19 yaitu beban puncak (*peak*), menghasilkan perhitungan biaya paling tinggi sebesar 7182524216,53. Total beban sebesar 255078 MW berdasarkan perhitungan dengan *FA*, memiliki biaya pembangkitan sebesar 160676785685,19 Rp/jam.

4.4 Metode *SAMFA* untuk menyelesaikan *DED* dengan Mempertimbangkan Fungsi Biaya Tidak Mulus

Metode *SAMFA* adalah salah satu metode yang mampu menyelesaikan permasalahan *DED*. *DED* yang dilakukan menggunakan sistem yang sama dengan percobaan sebelumnya dengan menggunakan *FA* yaitu, interkoneksi 500 kV Jawa-Bali dalam periode 24 jam. Level beban yang digunakan adalah level beban yang diambil pada bulan Maret 2009 dari PT. PLN (Persero) yaitu.

Tabel 4.23 Profil Beban untuk 24 Jam (*SAMFA*)

Jam ke-	Beban (MW)
1	10282
2	10232
3	10182
4	10132
5	10300
6	10500
7	10700
8	10750
9	10800
10	10900
11	10850
12	10800
13	10750
14	10700
15	10650
16	10600
17	10700
18	10800

Tabel 4.23 Profil Beban untuk 24 Jam (*SAMFA*) (lanjutan)

Jam ke-	Beban (MW)
19	11000
20	10800
21	10700
22	10600
23	10650
24	10700

Sehingga total beban adalah 255078 MW. Koefisien dari pembangkit-pembangkit seperti a, b, c yang merupakan *cost function*, dan e, f yang merupakan efek katup uap adalah sebagai berikut.

Tabel 4.24 Data Koefisien Pembangkit (*SAMFA*)

No.	Cost Function			Valve-point	
	a	b	c	e	f
P1	-65.94	395668.05	31630.21	100	0.063
P2	690.98	2478064.47	107892572.17	200	0.084
P3	0	6000	0	200	0.042
P4	0	5502	0	300	0.042
P5	-21.88	197191.76	-1636484.18	100	0.042
P6	132.15	777148.77	13608770.96	100	0.077
P7	52.19	37370.67	8220765.38	200	0.077
P8	533.92	2004960.63	86557397.40	300	0.063

Pembangkit-pembangkit tersebut juga memiliki batasan kemampuan pembangkitan. Batasan kemampuan pembangkitan ditampilkan dalam tabel seperti berikut.

Tabel 4.25 Batasan Pembangkit (*SAMFA*)

Unit	Pmin (MW)	Pmax (MW)
P1	1500	3400
P2	1040	2200
P3	600	1220
P4	400	900
P5	600	1220
P6	238	1050

Tabel 4.25 Batasan Pembangkit (*SAMFA*) (lanjutan)

Unit	Pmin (MW)	Pmax (MW)
P7	1425	3254
P8	150	827

Berbeda dengan metode *FA* yang biasa, untuk metode *SAMFA*, koefisien alpha tidak disetel secara permanen diawal. Alpha disetel diawal sebagai parameter untuk inisiasi pembangkitan awal, kemudian alpha akan di-*update* dengan cara (24) terus-menerus setiap iterasinya. Iterasi yang digunakan adalah sebanyak 50 iterasi dengan 20 *firefly*. Data-data parameter yang dibutuhkan akan ditampilkan sebagai berikut.

Tabel 4.26 Parameter awal *SAMFA* untuk *DED*

Parameter	Simbol	Nilai
Alpha	α	0,7574 (<i>random</i>)
Beta	β	0,5 [8]
Beta minimum	β_{\min}	0,98
Gamma	γ	1.0 [8]
Populasi	<i>firefly</i>	20
Iterasi	iterasi_max	50

Perbedaannya terletak pada alpha, alpha akan di-*update* setiap iterasinya yang disimbolkan sebagai *alpha_next*. Karena iterasi yang digunakan adalah 50 kali, maka nilai alpha setiap iterasinya ditampilkan sebagai berikut.

Tabel 4.27 Tabel Nilai Alpha Setiap Iterasi

Iterasi ke-	Alpha (α)
1	0,7574
2	0,7233
3	0,6907
4	0,6596
5	0,6299
6	0,6019
7	0,5744

Tabel 4.27 Tabel Nilai Alpha Setiap Iterasi (lanjutan)

Iterasi ke-	Alpha (α)
8	0,5485
9	0,5238
10	0,5002
11	0,4777
12	0,4562
13	0,4357
14	0,4161
15	0,3974
16	0,3795
17	0,3624
18	0,3461
19	0,3305
20	0,3156
21	0,3014
22	0,2878
23	0,2748
24	0,2624
25	0,2506
26	0,2393
27	0,2285
28	0,2182
29	0,2084
30	0,1990
31	0,1900
32	0,1814
33	0,1732
34	0,1654
35	0,1580
36	0,1509
37	0,1441
38	0,1376
39	0,1314
40	0,1255
41	0,1199
42	0,1145

Tabel 4.27 Tabel Nilai Alpha Setiap Iterasi (lanjutan)

Iterasi ke-	Alpha (α)
43	0,1093
44	0,1044
45	0,0997
46	0,0952
47	0,0909
48	0,0868
49	0,0829
50	0,0792

Terlihat bahwa nilai alpha akan semakin mengecil seiring bertambahnya iterasi. Hal ini menunjukkan ruang pergerakan acak dari kunang-kunang semakin sempit seiring berjalannya iterasi. Tujuan iterasi adalah menemukan solusi terbaik, dalam hal ini biaya termurah. Semakin mendekati iterasi maksimum, ruang pergerakan kunang-kunang dipersempit agar meningkatkan efisiensi.

4.4.1 Hasil Simulasi *DED* dengan menggunakan *SAMFA*

Dengan parameter dan profil-profil pembangkit yang telah terdata di atas, maka dapat dilakukan simulasi *DED* dengan menggunakan metode *SAMFA*. Simulasi dilakukan menggunakan aplikasi MATLAB 2014a. Hasil simulasi berupa pembebanan setiap unit dan biaya setiap unit dalam interval waktu 1 jam dalam total 24 jam akan ditampilkan sebagai berikut.

Tabel 4.28 Hasil Simulasi *SAMFA* untuk *DED* Jam ke-1 sampai 3

Unit	Jam ke-1 (MW)	Jam ke-2 (MW)	Jam ke-3 (MW)
P1	2916,27	2661,83	2893,16
P2	919,36	1125,65	1128,64
P3	1020,41	944,50	831,65
P4	306,36	508,49	610,09
P5	714,56	1083,90	724,43
P6	890,12	589,11	776,78
P7	3035,02	3081,88	3078,29
P8	479,91	236,65	138,97

Tabel 4.29 Hasil Simulasi *SAMFA* untuk *DED* Jam ke-4 sampai 6

Unit	Jam ke-4 (MW)	Jam ke-5 (MW)	Jam ke-6 (MW)
P1	3103,25	3287,63	3258,00
P2	1034,40	1330,66	1247,25
P3	595,32	1080,11	1113,32
P4	336,70	572,41	551,27
P5	964,79	744,04	765,44
P6	610,50	279,73	558,69
P7	2964,53	2744,26	2727,00
P8	522,52	261,17	279,03

Tabel 4.30 Hasil Simulasi *SAMFA* untuk *DED* Jam ke-7 sampai 9

Unit	Jam ke-7 (MW)	Jam ke-8 (MW)	Jam ke-9 (MW)
P1	3119,40	3276,51	2927,82
P2	1039,12	1264,95	1236,65
P3	699,49	827,07	1363,70
P4	825,38	937,90	963,80
P5	796,59	711,03	1211,99
P6	706,87	364,54	608,78
P7	3103,30	3092,67	2096,85
P8	409,86	275,33	390,41

Tabel 4.31 Hasil Simulasi *FA* untuk *DED* Jam ke-10 sampai 12

Unit	Jam ke-10 (MW)	Jam ke-11 (MW)	Jam ke-12 (MW)
P1	2981,01	2886,50	3389,49
P2	1251,15	1227,26	1256,16
P3	1273,31	881,92	908,91
P4	684,24	754,05	670,53
P5	960,48	1171,07	1232,29
P6	488,56	428,60	980,01
P7	2907,65	3128,51	2055,98
P8	353,60	372,11	306,62

Tabel 4.32 Hasil Simulasi *FA* untuk *DED* Jam ke-13 sampai 15

Unit	Jam ke-13 (MW)	Jam ke-14 (MW)	Jam ke-15 (MW)
P1	3341,57	3452,12	3148,77
P2	1152,46	1155,91	1102,97
P3	891,66	1000,36	569,15
P4	524,88	951,41	819,68
P5	1251,82	1115,86	922,62
P6	466,27	682,21	869,35
P7	2705,90	1942,03	2704,10
P8	415,44	400,09	513,35

Tabel 4.33 Hasil Simulasi *FA* untuk *DED* Jam ke-16 sampai 18

Unit	Jam ke-16 (MW)	Jam ke-17 (MW)	Jam ke-18 (MW)
P1	3058,52	3308,60	3070,81
P2	1175,53	1090,89	1160,72
P3	1098,35	816,94	849,95
P4	612,83	501,64	600,92
P5	791,03	1162,52	1076,36
P6	955,34	311,94	886,52
P7	2643,66	2928,36	2809,87
P8	264,75	579,11	344,85

Tabel 4.34 Hasil Simulasi *FA* untuk *DED* Jam ke-19 sampai 21

Unit	Jam ke-19 (MW)	Jam ke-20 (MW)	Jam ke-21 (MW)
P1	3213,72	3233,55	2964,16
P2	1239,21	1176,47	1144,79
P3	1304,01	945,59	1177,78
P4	628,30	753,06	700,85
P5	922,33	1065,58	779,55
P6	886,99	342,86	335,69
P7	2467,22	2766,17	3091,49
P8	338,22	516,72	505,68

Tabel 4.35 Hasil Simulasi *FA* untuk *DED* Jam ke-22 sampai 24

Unit	Jam ke-22 (MW)	Jam ke-23 (MW)	Jam ke-24 (MW)
P1	3099,02	2969,40	3348,80
P2	1174,63	1144,63	1100,72
P3	731,90	1029,94	640,91
P4	664,00	775,87	768,48
P5	1048,20	954,30	723,70
P6	727,18	490,77	485,20
P7	2886,29	2832,89	3106,68
P8	268,78	452,21	525,51

4.4.2 Biaya Total Pembangkitan (*SAMFA*)

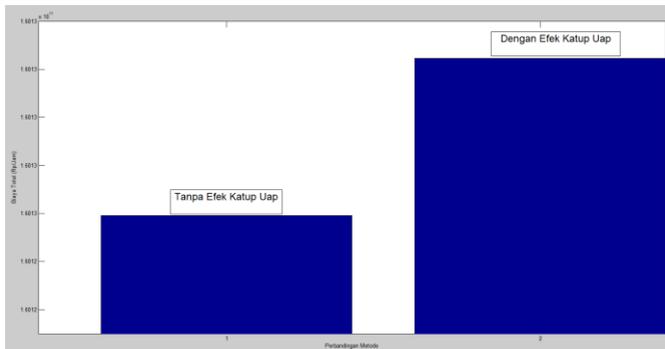
Sama seperti percobaan sebelumnya yang menggunakan *FA*, metode *SAMFA* juga terpengaruh oleh efek katup uap, sehingga penambahan koefisien *e* dan *f* juga mempengaruhi penambahan biaya pembangkitan setiap jamnya. Biaya pembangkitan setiap jam ditampilkan sebagai berikut.

Tabel 4.36 Biaya Total Pembangkitan (*SAMFA*)

Jam ke-	Beban (MW)	Biaya Pembangkitan
1	10282	6283353894,73
2	10232	6281079950,12
3	10182	6206128212,86
4	10132	6272038340,87
5	10300	6751332547,74
6	10500	6683821836,77
7	10700	6425722373,07
8	10750	6692825399,70
9	10800	6843494488,12
10	10900	6907713104,08
11	10850	6904742761,25
12	10800	6971260872,73
13	10750	6594377643,86
14	10700	6535174650,50
15	10650	6979957687,26
16	10600	6722955389,75

Tabel 4.36 Biaya Total Pembangkitan (*SAMFA*) (lanjutan)

Jam ke-	Beban (MW)	Biaya Pembangkitan
17	10700	6637482098,44
18	10800	6875245917,04
19 (<i>peak</i>)	11000	7054077488,49
20	10800	6733783235,91
21	10700	6742428145,51
22	10600	6617607549,99
23	10650	6682396459,71
24	10700	6733459858,84
Total	255078	160132459907,34



Gambar 4.2 Perbandingan Tanpa & Dengan Efek Katup Uap (*SAMFA*)

Berdasarkan data-data yang telah didapatkan dari perhitungan *DED* dengan *SAMFA* yang mempertimbangkan efek katup uap sehingga fungsi biaya menjadi tidak mulus, dapat dilihat bahwa pada beban jam ke-19 yaitu beban puncak (*peak*), menghasilkan perhitungan biaya paling tinggi sebesar 7054077488,49. Total beban sebesar 255078 MW berdasarkan perhitungan dengan *SAMFA*, memiliki biaya pembangkitan sebesar 160132459907,34 Rp/jam.

4.5 Perbandingan Hasil Perhitungan *FA* dengan *SAMFA*

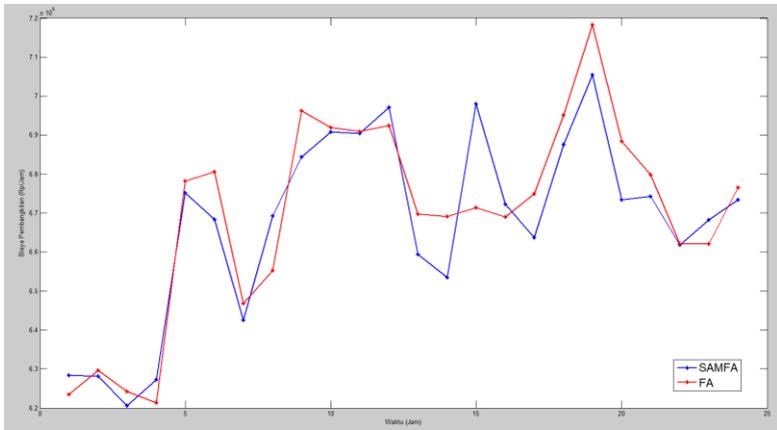
Metode *FA* dan *SAMFA* adalah salah satu metode yang mampu menyelesaikan permasalahan *DED*. Percobaan yang dilakukan memiliki

data yang sama dan parameter yang hampir sama dari jumlah *firefly*, jumlah iterasi, betamin, beta, dan gamma. Parameter yang membedakan adalah terletak pada nilai alpha. Perbedaan parameter ini ternyata membuat hasil perhitungan juga berbeda. Hasil kedua perhitungan dengan pengaruh perbedaan nilai alpha adalah sebagai berikut.

Tabel 4.37 Perbandingan Total Biaya Setiap Unit

Jam ke-	Beban (MW)	Biaya Pembangkitan	
		<i>FA</i>	<i>SAMFA</i>
1	10282	6234884535,44	6283353894,73
2	10232	6296575628,84	6281079950,12
3	10182	6242657852,72	6206128212,86
4	10132	6213016387,99	6272038340,87
5	10300	6782338337,35	6751332547,74
6	10500	6805442698,85	6683821836,77
7	10700	6467378981,20	6425722373,07
8	10750	6552679715,58	6692825399,70
9	10800	6962337700,63	6843494488,12
10	10900	6919101505,27	6907713104,08
11	10850	6909953529,04	6904742761,25
12	10800	6924728801,43	6971260872,73
13	10750	6696954737,00	6594377643,86
14	10700	6690745060,10	6535174650,50
15	10650	6714035846,94	6979957687,26
16	10600	6690535858,70	6722955389,75
17	10700	6749593840,01	6637482098,44
18	10800	6950464523,45	6875245917,04
19	11000	7182524216,53	7054077488,49
20	10800	6884454828,96	6733783235,91
21	10700	6798905106,72	6742428145,51
22	10600	6620746431,25	6617607549,99
23	10650	6621160209,42	6682396459,71
24	10700	6765569351,77	6733459858,84

Perbedaan biaya pembangkitan setiap jam dari kedua metode yaitu *FA* dan *SAMFA* dengan mempertimbangkan efek katup uap dapat dianalisis dalam bentuk grafik ditampilkan sebagai berikut.



Gambar 4.3 Perbandingan Biaya Pembangkitan Antara *FA* dan *SAMFA*

Metode *SAMFA* terlihat mayoritas lebih rendah daripada perhitungan metode *FA*. Hal ini terlihat pada interval jam ke-2 sampai jam ke-3, jam ke-5 sampai jam ke 7, jam ke-9 sampai jam ke-11, jam ke-12 sampai jam ke-14, dan jam ke17 sampai jam ke-22. Pada interval jam ke-1 sampai jam ke-2, jam ke-3 sampai jam ke-5, jam ke-5 sampai jam ke-7, jam ke-7 sampai jam ke-9, jam ke-11 sampai jam ke-12, jam ke-14 sampai jam ke-17, dan jam ke-22 sampai jam ke-24 memiliki harga sedikit lebih tinggi, namun total akumulasi semua unit ditampilkan sebagai berikut.

Tabel 4.38 Perbandingan Total Biaya Pembangkitan

Pembanding	<i>FA</i>	<i>SAMFA</i>
Total Biaya (Rp/h)	160676785685,19	160132459907,34

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil simulasi dari penelitian yang telah dibuat dengan menggunakan *self-Adaptive modified firefly algorithm* untuk menyelesaikan *dynamic economic dispatch* selama 24 jam pada sistem interkoneksi Jawa-Bali 500 kV dengan mempertimbangkan efek katup uap, dapat diperoleh kesimpulan sebagai berikut :

1. Penggunaan *SAMFA* dapat memperoleh nilai paling minimal dari harga pembangkitan dengan menentukan besar daya yang terbangkit setiap unitnya.
2. Efek katup uap pada sistem interkoneksi Jawa-Bali 500 kV memberikan dampak penambahan biaya pembangkitan sebesar 468,2823-dollar atau sekitar 6.555.952,2 Rupiah.
3. Dengan membandingkan metode *FA* dengan *SAMFA* dapat diperoleh kesimpulan bahwa, metode *SAMFA* dapat memberikan hasil lebih murah sekitar 0,4% dari metode *FA*.

5.2 Saran

Terdapat beberapa saran untuk penelitian lanjutan terkait penggunaan metode *SAMFA* untuk menyelesaikan *DED* sebagai berikut :

1. Perlu dilakukan penelitian lebih lanjut mengenai perubahan alpha disetiap iterasi, karena pengaruh alpha bukan hanya pada biaya namun pada efisiensi daripada pembangkit itu sendiri.
2. Untuk penelitian selanjutnya sebaiknya perlu menggunakan *losses* agar penelitian semakin relevan.
3. Untuk penelitian selanjutnya sebaiknya perlu memperhatikan *ramp up* dan *ramp down* serta menyempurnakan optimasi.

Halaman ini sengaja dikosongkan

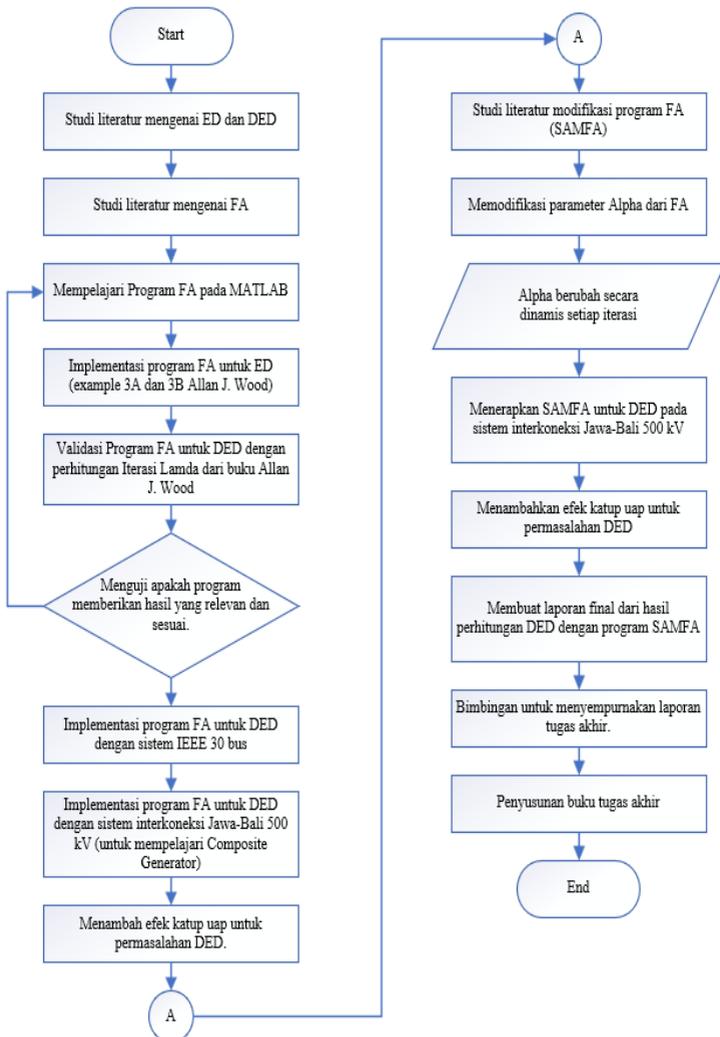
DAFTAR PUSTAKA

- [1] Saadat, H., "Power Sistem Analysis", McGraw-Hill, Inc, 1999.
- [2] A.J. Wood, "Power Generation, Operation, and Control." 2nd. New York:John Wiley & Sons, 1996.
- [3] F.S. Abu-Mouti, M.E. El-Hawary. "Optimal Dynamic Economic Dispatch Including Renewable Energy Resource Using Artificial Bee Colony Algorithm," System Conference (SysCon), 2012 IEEE International, vol.,pp.1,6, 19-22 March 2012.
- [4] D.W.Ross, S.Kim, "Dynamic Economic Dispatch of Generation," Power Apparatus and Systems, IEEE Transactions on, vol.PAS-99, no.6, pp.2060,2068, Nov.1980.
- [5] D.Karaboga. "An Idea Based on Honey Bee Swarm for Numerical Optimization," Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department. 2005.
- [6] Xin-She Yang, "A Comprehensive Review of Firefly Algorithms", Swarm and Evolutionary Computation, June 2013.
- [7] Saibal K.Pal, "Comparative Study of Firefly Algorithm and Particle Swarm Optimization for Noisy Non-Linear Optimization Problems", Intelligent Systems and Applications, September 2012.
- [8] T. Niknam, R. Azizipanah-Abarghooee, and A. Roosta, "Reserve Constrained Dynamic Economic Dispatch: A New Fast Self-Adaptive Modified Firefly Algorithm," IEEE Syst. J., vol. 6, no. 4, pp. 635–646, Dec. 2012.
- [9] I. Andi Muhammad, R. M. Natsir, "Economic Dispatch Thermal Generator Using Modified Improved Particle Swarm Optimization," TELKOMNIKA, Vol. 10, No. 3, pp. 459-470, Sept. 2012.
- [10] T. Apostolopoulos and A. Vlachos, "Application of the firefly algorithm for solving the economic emission load dispatch problem," Int. J. Combinatorics, vol. 2011, no. 523806, p. 23, 2011.

- [11] X. Yuan, L. Wang, Y. Yuan, Y. Zhang, B. Cao, and B. Yang, “A modified differential evolution approach for dynamic economic dispatch with valve point effects,” *Energy Conv. Manage.*, vol. 49, no. 12, pp. 3447–3453, Dec. 2008.
- [12] Y. Lu, J. Zhou, H. Qin, Y. Li, and Y. Zhang, “An adaptive hybrid differential evolution algorithm for dynamic economic dispatch with valve-point effects,” *Expert Systems with Applications*, vol. 37, pp. 4842–4849, 2010.
- [13] B. K. Panigrahi, V. R. Pandi, and S. Das, “Adaptive particle swarm optimization approach for static and dynamic economic load dispatch,” *Energy Conv. Manage.*, vol. 49, no. 6, pp. 1407–1415, Jun. 2008.
- [14] Y. Wang, J. Zhou, Y. Lu, H. Qin, and Y. Wang, “Chaotic self-adaptive particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects,” *Expert Syst. Appl.*, vol. 38, no. 11, pp. 14231–14237, Oct. 2011.

LAMPIRAN

I. *Flowchart* Penyusunan Tugas Akhir



II. *M-file Self-Adaptive Modified Firefly Algorithm*

```
%Joseph Danarta S.  
%07111440000136  
%Self-Adaptive Modified Firefly Algorithm  
  
clear all;  
clc;  
tic;  
  
% INISIASI AWAL  
betamin      = 0.98;      % parameter firefly ()  
gamma        = 1.0;      % parameter firefly ()  
firefly      = 20;      % jumlah firefly  
it           = 1;        % iterasi awal  
iterasi_max  = 50;      % iterasi maksimum  
MainData;    % Memanggil Data Pembangkit
```

Data Sistem Interkoneksi Jawa-Bali 500 kV (MainData)

```
pmin_max      = [1500 3400  
                1040 2200  
                600  1220  
                400  900  
                600 1220  
                238 1050  
                1425 3254  
                150  827];  
  
koef_biaya_genA = [-65.94   395668.05   31630.21   ];  
koef_biaya_genB = [690.98   2478064.47  107892572.17];  
koef_biaya_genC = [0        6000      0           ];  
koef_biaya_genD = [0        5502     0           ];  
koef_biaya_genE = [-21.88   197191.76  -1636484.18 ];  
koef_biaya_genF = [132.15   777148.77  13608770.96 ];  
koef_biaya_genG = [52.19    37370.67   8220765.38  ];  
koef_biaya_genH = [533.92   2004960.63  86557397.40 ];  
  
fuel_cost_genA = 1.0*eye(3);  
fuel_cost_genB = 1.0*eye(3);  
fuel_cost_genC = 1.0*eye(3);  
fuel_cost_genD = 1.0*eye(3);  
fuel_cost_genE = 1.0*eye(3);  
fuel_cost_genF = 1.0*eye(3);
```

```

fuel_cost_genG = 1.0*eye(3);
fuel_cost_genH = 1.0*eye(3);

```

Inisiasi Posisi Awal untuk Pembangkitan Awal

```

%===== INISIASI POSISI AWAL PARTIKEL =====
for interval = 1;
for i=1:8          % i = baris 1 sampai ke 8
    for j=1:firefly % j = kolom 1 sampai ke 20
        posisi_awal(j,i) = (pmin_max(i,2) -
pmin_max(i,1))*rand + pmin_max(i,1);
    end
end
jumlah_p = sum(posisi_awal'); % posisi awal
generator di total semua
for K = 1:firefly
    for x = ramp(1,1) %beban pada interval pertama
        if jumlah_p(K) < x          % jika posisi awal
generator kurang dari permintaan maka :
            korektor(K) = ((x - jumlah_p(K))/8);
            posisi_awal(K,:) = posisi_awal(K, :) +
korektor(K);
        elseif(jumlah_p(K) > x)
            korektor(K) = ((jumlah_p(K) - x)/8);
            posisi_awal(K,:) = (posisi_awal(K, :) -
korektor(K));
        end
    end
end
end

MainFunction;

toc;

```

Fungsi Utama *Loop SAMFA*

```

a      = rand();          % inisiasi awal random

P_1    = posisi_awal(:,1); % daya pembangkit 1
P_2    = posisi_awal(:,2); % daya pembangkit 2
P_3    = posisi_awal(:,3); % daya pembangkit 3
P_4    = posisi_awal(:,4); % daya pembangkit 4
P_5    = posisi_awal(:,5); % daya pembangkit 5

```

```

P_6 = posisi_awal(:,6); % daya pembangkit 6
P_7 = posisi_awal(:,7); % daya pembangkit 7
P_8 = posisi_awal(:,8); % daya pembangkit 8

%menemukan matriks dari setiap generator
matriks_A = koef_biaya_genA*fuel_cost_genA;
matriks_B = koef_biaya_genB*fuel_cost_genB;
matriks_C = koef_biaya_genC*fuel_cost_genC;
matriks_D = koef_biaya_genD*fuel_cost_genD;
matriks_E = koef_biaya_genE*fuel_cost_genE;
matriks_F = koef_biaya_genF*fuel_cost_genF;
matriks_G = koef_biaya_genG*fuel_cost_genG;
matriks_H = koef_biaya_genH*fuel_cost_genH;

%memasukkan rumus ax^2+bx+c
biaya_genA = (matriks_A(1,1).*(P_1).^2
+ (matriks_A(1,2).*(P_1))+matriks_A(1,3);
biaya_genB = (matriks_B(1,1).*(P_2).^2
+ (matriks_B(1,2).*(P_2))+matriks_B(1,3);
biaya_genC = (matriks_C(1,1).*(P_3).^2
+ (matriks_C(1,2).*(P_3))+matriks_C(1,3);
biaya_genD = (matriks_D(1,1).*(P_4).^2
+ (matriks_D(1,2).*(P_4))+matriks_D(1,3);
biaya_genE = (matriks_E(1,1).*(P_5).^2
+ (matriks_E(1,2).*(P_5))+matriks_E(1,3);
biaya_genF = (matriks_F(1,1).*(P_6).^2
+ (matriks_F(1,2).*(P_6))+matriks_F(1,3);
biaya_genG = (matriks_G(1,1).*(P_7).^2
+ (matriks_G(1,2).*(P_7))+matriks_G(1,3);
biaya_genH = (matriks_H(1,1).*(P_8).^2
+ (matriks_H(1,2).*(P_8))+matriks_H(1,3);
total_biaya = biaya_genA+ biaya_genB+ biaya_genC+
biaya_genD+ biaya_genE+ biaya_genF+ biaya_genG+
biaya_genH; % digunakan untuk mencari lightbest

% non-smooth cost function
e = [100 200 200 300 100 100 200 300];
f = [0.063 0.084 0.042 0.042 0.042 0.077 0.077 0.063];

pp=zeros(1,1);
cost_non=zeros(1,1);
ccc=pmin_max(1,1);
cost_non=abs(e(1,1).*sin(f(1,1)*(ccc(1)-pp(1,1)))));

```

```

pp2=zeros(1,2);
cost_non2=zeros(1,2);
ccc2=pmin_max(2,1);
cost_non2=abs(e(1,2).*sin(f(1,2)*(ccc(1)-pp2(1,2))));

pp3=zeros(1,3);
cost_non3=zeros(1,3);
ccc3=pmin_max(3,1);
cost_non3=abs(e(1,3).*sin(f(1,3)*(ccc(1)-pp3(1,3))));

pp4=zeros(1,4);
cost_non4=zeros(1,4);
ccc4=pmin_max(4,1);
cost_non4=abs(e(1,4).*sin(f(1,4)*(ccc(1)-pp4(1,4))));

pp5=zeros(1,5);
cost_non5=zeros(1,5);
ccc5=pmin_max(5,1);
cost_non5=abs(e(1,5).*sin(f(1,5)*(ccc(1)-pp5(1,5))));

pp6=zeros(1,6);
cost_non6=zeros(1,6);
ccc6=pmin_max(6,1);
cost_non6=abs(e(1,6).*sin(f(1,6)*(ccc(1)-pp6(1,6))));

pp7=zeros(1,7);
cost_non7=zeros(1,7);
ccc7=pmin_max(7,1);
cost_non7=abs(e(1,7).*sin(f(1,7)*(ccc(1)-pp7(1,7))));

pp8=zeros(1,8);
cost_non8=zeros(1,8);
ccc6=pmin_max(8,1);
cost_non8=abs(e(1,8).*sin(f(1,8)*(ccc(1)-pp8(1,8))));

cost_nonsmooth=(cost_non + cost_non2 + cost_non3 +
cost_non4 + cost_non5 + cost_non6 +cost_non7 +
cost_non8);

```

```

inequal=[P_1<2500 P_1>3400 P_2<1000 P_2>1600 P_3<400
P_3>800 P_4<400 P_4>900 P_5<700 P_5>1200 P_6<700
P_6>1200 P_7<2400 P_7>3100 P_8<150 P_8>400];
inequality = sum(inequal');
clear j
for j=1:firefly

end
[lambda,In]=min(total_biaya); % biaya minimum
lightbest(it)=min(total_biaya);
posisi_best(1,:)=posisi_awal(In,:);

clear j i
posisi_firefly=posisi_awal;

for ikj=1:firefly,
% Parameter Attractiveness Kunang-Kunang
for jkj=1:firefly,
r=sqrt(sum((posisi_firefly(ikj,:)-
posisi_firefly(jkj,:)).^2));

% MengUpdate Pergerakan kunang-kunang
if total_biaya(ikj)>total_biaya(jkj), % Apakah
lebih terang dan attractive
beta0=0.5; beta=(beta0)*exp(-gamma*r.^2);
alpha = a;
tmpf=alpha.*((rand(1,1))-0.5);

posisi_firefly(ikj,2)=posisi_firefly(ikj,2).*(1-
beta)+(posisi_firefly(jkj,2).*beta)+tmpf;

posisi_firefly(ikj,3)=posisi_firefly(ikj,3).*(1-
beta)+(posisi_firefly(jkj,3).*beta)+tmpf;

posisi_firefly(ikj,4)=posisi_firefly(ikj,4).*(1-
beta)+(posisi_firefly(jkj,4).*beta)+tmpf;

posisi_firefly(ikj,5)=posisi_firefly(ikj,5).*(1-
beta)+(posisi_firefly(jkj,5).*beta)+tmpf;

posisi_firefly(ikj,6)=posisi_firefly(ikj,6).*(1-
beta)+(posisi_firefly(jkj,6).*beta)+tmpf;

```

```

posisi_firefly(ikj,7)=posisi_firefly(ikj,7).*(1-
beta)+(posisi_firefly(jkj,7).*beta)+tmpf;

posisi_firefly(ikj,8)=posisi_firefly(ikj,8).*(1-
beta)+(posisi_firefly(jkj,8).*beta)+tmpf;
posisi_firefly(ikj,1)=x-(posisi_firefly(ikj,2)+
posisi_firefly(ikj,3)+posisi_firefly(ikj,4)+posisi_fir
efly(ikj,5)+posisi_firefly(ikj,6)+posisi_firefly(ikj,7
)+posisi_firefly(ikj,8));
    end
end
end

while it <iterasi_max
    it =it+1;
    P_1 = posisi_firefly(:,1);      % daya pembangkit 1
    P_2 = posisi_firefly(:,2);      % daya pembangkit 2
    P_3 = posisi_firefly(:,3);      % daya pembangkit 3
    P_4 = posisi_firefly(:,4);
    P_5 = posisi_firefly(:,5);
    P_6 = posisi_firefly(:,6);
    P_7 = posisi_firefly(:,7);
    P_8 = posisi_firefly(:,8);

    matriks_A = koef_biaya_genA*fuel_cost_genA; %mencari
matriks dari setiap generator
    matriks_B = koef_biaya_genB*fuel_cost_genB;
    matriks_C = koef_biaya_genC*fuel_cost_genC;
    matriks_D = koef_biaya_genD*fuel_cost_genD;
    matriks_E = koef_biaya_genE*fuel_cost_genE; %mencari
matriks dari setiap generator
    matriks_F = koef_biaya_genF*fuel_cost_genF;
    matriks_G = koef_biaya_genG*fuel_cost_genG;
    matriks_H = koef_biaya_genH*fuel_cost_genH;

    biaya_genA = (matriks_A(1,1).*(P_1).^2)
+(matriks_A(1,2).*(P_1))+matriks_A(1,3); %memasukkan
rumus  $ax^2+bx+c$ 
    biaya_genB = (matriks_B(1,1).*(P_2).^2)
+(matriks_B(1,2).*(P_2))+matriks_B(1,3);
    biaya_genC = (matriks_C(1,1).*(P_3).^2)
+(matriks_C(1,2).*(P_3))+matriks_C(1,3);

```

```

biaya_genD = (matriks_D(1,1).*(P_4).^2)
+ (matriks_D(1,2).*(P_4))+matriks_D(1,3);
biaya_genE = (matriks_E(1,1).*(P_5).^2)
+ (matriks_E(1,2).*(P_5))+matriks_E(1,3); %memasukkan
rumus ax^2+bx+c
biaya_genF = (matriks_F(1,1).*(P_6).^2)
+ (matriks_F(1,2).*(P_6))+matriks_F(1,3);
biaya_genG = (matriks_G(1,1).*(P_7).^2)
+ (matriks_G(1,2).*(P_7))+matriks_G(1,3);
biaya_genH = (matriks_H(1,1).*(P_8).^2)
+ (matriks_H(1,2).*(P_8))+matriks_H(1,3);
total_biaya = biaya_genA+ biaya_genB+ biaya_genC+
biaya_genD+ biaya_genE+ biaya_genF+ biaya_genG+
biaya_genH; % digunakan untuk mencari lightbest

inequal=[P_1<2500 P_1>3400 P_2<1000 P_2>1600 P_3<400
P_3>800 P_4<400 P_4>900 P_5<700 P_5>1200 P_6<700
P_6>1200 P_7<2400 P_7>3100 P_8<150 P_8>400];
inequality = sum(inequal');
clear j
for j=1:firefly

end

[lambda,In]=min(total_biaya); % biaya minimum
lightbest(it)=min(total_biaya)
posisi_best(1,:)=posisi_firefly(In,:);

clear j i
sebelum=posisi_firefly;
for ikj=1:firefly,
% Parameter Attractiveness Kunang-Kunang
for jkj=1:firefly,
r=sqrt(sum((posisi_firefly(ikj,:)-
posisi_firefly(jkj,:)).^2));
% MengUpdate Pergerakan kunang-kunang
if total_biaya(ikj)>total_biaya(jkj), % Apakah
lebih terang dan attractive
beta0=0.5; beta=(beta0)*exp(-
gamma*r.^2);
alpha_next =
(((1/(2*iterasi_max))^(1/(2*iterasi_max)))*alpha);
tmpf=alpha_next.*((rand(1,1))-0.5);

```

```

posisi_firefly(ikj,2)=posisi_firefly(ikj,2).*(1-
beta0)+(posisi_firefly(jkj,2).*beta0)+tmpf;

posisi_firefly(ikj,3)=posisi_firefly(ikj,3).*(1-
beta0)+(posisi_firefly(jkj,3).*beta0)+tmpf;

posisi_firefly(ikj,4)=posisi_firefly(ikj,4).*(1-
beta0)+(posisi_firefly(jkj,4).*beta0)+tmpf;

posisi_firefly(ikj,5)=posisi_firefly(ikj,5).*(1-
beta0)+(posisi_firefly(jkj,5).*beta0)+tmpf;

posisi_firefly(ikj,6)=posisi_firefly(ikj,6).*(1-
beta0)+(posisi_firefly(jkj,6).*beta0)+tmpf;

posisi_firefly(ikj,7)=posisi_firefly(ikj,7).*(1-
beta0)+(posisi_firefly(jkj,7).*beta0)+tmpf;

posisi_firefly(ikj,8)=posisi_firefly(ikj,8).*(1-
beta0)+(posisi_firefly(jkj,8).*beta0)+tmpf;
posisi_firefly(ikj,1)=x-
(posisi_firefly(ikj,2)+posisi_firefly(ikj,3)+posisi_fi
refly(ikj,4)+posisi_firefly(ikj,5)+posisi_firefly(ikj,
6)+posisi_firefly(ikj,7)+posisi_firefly(ikj,8));
    end
end
end
sesudah=posisi_firefly;
    it
end

fprintf('Jam ke- 1');
fprintf('\n');
fprintf('Pembangkit Unit 1:%0.2f',posisi_best(:,1));
fprintf(' MW\n');
fprintf('Pembangkit Unit 2:%0.2f',posisi_best(:,2));
fprintf(' MW\n');
fprintf('Pembangkit Unit 3:%0.2f',posisi_best(:,3));
fprintf(' MW\n');
fprintf('Pembangkit Unit 4:%0.2f',posisi_best(:,4));
fprintf(' MW\n');
fprintf('Pembangkit Unit 5:%0.2f',posisi_best(:,5));
fprintf(' MW\n');

```

```

fprintf('Pembangkit Unit 6:%0.2f',posisi_best(:,6));
fprintf(' MW\n')
fprintf('Pembangkit Unit 7:%0.2f',posisi_best(:,7));
fprintf(' MW\n')
fprintf('Pembangkit Unit 8:%0.2f',posisi_best(:,8));
fprintf(' MW\n')
fprintf('Total
Pembangkitan:%0.2f', (posisi_best(:,1)+posisi_best(:,2)
+posisi_best(:,3)+posisi_best(:,4)+posisi_best(:,5)+po
sisi_best(:,6)+posisi_best(:,7)+posisi_best(:,8)));
fprintf(' MW\n')
fprintf('Total Biaya Pembangkitan
:%0.2f\n',lightbest(:,50)+cost_nonsmooth);
fprintf('\n');
end

```

BIODATA PENULIS



Joseph Danarta Sugiana lahir di Jakarta tepatnya di Jakarta Barat pada tanggal 12 Februari 1996. Anak pertama dari tiga bersaudara oleh pasangan A. Sugiana dan Suzana. Penulis menempuh pendidikan dari Sekolah Dasar Penabur Gading Serpong, Sekolah Menengah Pertama Penabur Gading Serpong, Sekolah Menengah Atas Kolese Gonzaga Jakarta, hingga saat ini menempuh perkuliahan di teknik elektro Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2014.

Penulis memiliki kesenangan mencari kesibukan dengan kegiatan diluar pembelajaran mata kuliah. Penulis pernah menjabat sebagai ketua kaderisasi *Fundamental Electrical Training* 2015-2016, staff Pengembangan Sumber Daya Mahasiswa Himpunan Mahasiswa Teknik Elektro (HIMATEKTRO) 2015-2016, ketua bidang agitasi dan eskalasi opini publik divisi Kalpataru Elektro ITS, staff *public relations and communications Society of Petroleum Engineers ITS* 2016-2017, hingga kepala departemen *public relations and media communications Society of Petroleum Engineers ITS* 2017-2018. Sebagai salah satu mahasiswa yang aktif, penulis juga dianugerahi penghargaan *Future Leaders* dari XL Axiata *batch 5*. Dalam proses yang panjang ini, tentunya penulis tidak akan mampu menjadi apa adanya sekarang tanpa orang-orang disekitar, terutama teman-teman seperjuangan dan orang yang penulis kasihi.

Penulis dapat dihubungi melalui *e-mail* : josephd@xlfutureleaders.com