

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO
PROGRAM STUDI TEKNIK ENERGI LISTRIK**



**ANALISIS OPTIMASI *ECONOMIC DISPATCH COMBINED CYCLE*
POWER PLANT DENGAN METODE EVOLUTIONARY
PROGRAMMING**

SKRIPSI

**Disusun Oleh :
AGUS YULIANTO
NIM: 99.12.068**

SEPTEMBER 2005

INSTITUTE OF TECHNOLOGY
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING
PROGRAM STUDI TEKNIK ELEKTRO

ANALISIS OPTIMALISASI EKONOMI SISTEM tenaga listrik
POWER PLANT DENGAN METODE EVOLUSIONARY
PROGRAMMING

SKRIPSI

Disusun oleh :

ABDUL KADIR

NIM. 02.12.001

SEPTEMBER 2002

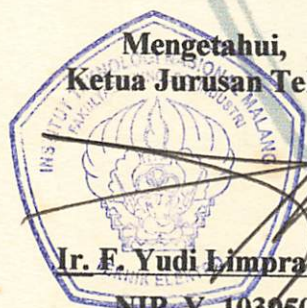
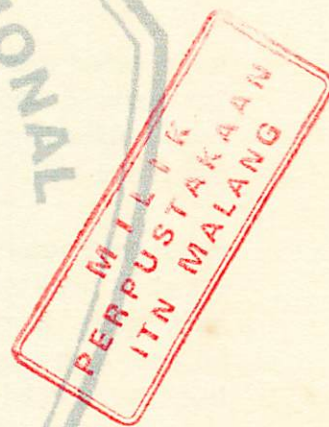
LEMBAR PERSETUJUAN

**ANALISIS OPTIMASI *ECONOMIC DISPATCH COMBINED CYCLE POWER*
PLANT DENGAN METODE EVOLUTIONARY PROGRAMMING**

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat-Syarat
Guna Mencapai Gelar Sarjana Teknik*

**Disusun Oleh :
AGUS YULIANTO
NIM. 99.12.068**



**Mengetahui,
Ketua Jurusan Teknik Elektro S-1**

Ir. F. Yudi Limprptomo, MT
NIP. Y. 1039500274

**Menyetujui,
Dosen Pembimbing**

Ir. Choirul Saleh, MT
NIP.P. 101880 0190

**KONSENTRASI TEKNIK ENERGI LISTRIK
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

ABSTRAKSI

ANALISIS OPTIMASI *ECONOMIC DISPATCH* COMBINED CYCLE POWER PLANT DENGAN METODE *EVOLUTIONARY PROGRAMMING*

Agus Yulianto, 99.12.068, Jurusan Teknik Elektro/Teknik Energi Listrik S-1, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang, 2005, Dosen Pembimbing Ir. Choirul Saleh, MT.

Kata Kunci: *Economic Dispatch* (Pembebanan Ekonomis), *Combine Cycle Power Plant*, *Evolutionary Programming*.

Dalam skripsi ini membahas masalah optimasi untuk biaya bahan bakar pada suatu sistem tenaga listrik dengan menggunakan metode *Evolutionary Programming* pada Pembangkit Listrik Tenaga Termal . Pembangkit Listrik Tenaga Termal tersebut adalah *Combined Cycle Power Plant* (CCPP) atau dalam bahasa Indonesia disebut Pusat Listrik Tenaga Gas Uap (PLTGU).

Pada CCPP gas panas yang dibuang hasil pembakaran yang dibuang oleh turbin gas tersebut dimanfaatkan kembali melalui peralatan yang disebut *Heat Recovery Steam Generator* (HRSG) untuk menghasilkan siklus uap-air yang akan digunakan memutar sebuah turbin uap. Pembangkit listrik tenaga termal jenis ini memiliki keuntungan karena efisiensi siklus termalnya menjadi lebih tinggi Hasil dari analisis tersebut nantinya dapat digunakan sebagai salah satu acuan dalam mengefisienkan biaya bahan bakar yang bisa digunakan oleh PT. PLN. PJB II Unit Pembangkitan Gresik. Sehingga nilai kerugian yang sementara ini diderita oleh PT. PLN. PJB UP Gresik bisa berkurang, yang akhirnya nanti akan bisa menambah keuntungan bagi PT. PLN sebagai perusahaan penyedia energi listrik di Indonesia.

Analisa dilakukan dengan bantuan program komputer menggunakan bahasa pemrograman *Borland Delphi* versi 7.0 dicoba pada PT. PLN. PJB II Unit Pembangkitan Gresik yang terdiri dari 3 Blok *Combined Cycle Power Plant*. Dengan menggunakan *Economic Dispatch* dengan metode *Evolutionary Programming* maka akan didapatkan optimasi biaya bahan bakar yang lebih efisien.

KATA PENGANTAR

Dengan Rahmat Allah SWT dan mengucapkan syukur kehadirat-Nya atas karunia yang telah dilimpahkan kepada penulis sehingga dapat menyelesaikan skripsi dengan lancar.

Penulis menyadari bahwa skripsi ini jauh dari sempurna, karena itu saran dan kritik yang bersifat membangun penulis sangat harapkan.

Atas tersusunya skripsi ini tidak lupa pula penulis mengucapkan terima kasih kepada:

1. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. FX Yudi Limprptomo, MT, selaku Ketua Jurusan Teknik Elektro, Institut Teknologi Nasional Malang.
3. Ir. Choirul Saleh, MT, selaku dosen pembimbing yang memberikan ilmu, dorongan semangat dan bantuanya.
4. Rekan-rekan Elektro angkatan 1999 khususnya kelas 1-2 atas segala kebersamaanya selama ini.
5. Serta semua pihak yang turut membantu dalam penyelesaian skripsi ini.

Akhir kata semoga skripsi ini dapat bermanfaat bagi kita semua dalam memperkaya ilmu pengetahuan.

Malang, September 2005

Penyusun



LEMBAR PERSEMBAHAN

Allah SWT

Atas segala berkat dan rahmatnya Sehingga dapat menyelesaikan Kuliah ini dengan sukses dan lancar

Nabi Muhammad SAW

Atas segala ajaran-ajaran dan tuntunanya sehingga Aku dapat menjalankan Ajaranmu tanpa ada keraguan sedikitpun

Ibunda Tercinta

Hanya Engkau Wanita di muka bumi ini yang aku kagumi, Kesabaranmu tak ada yang bisa menandingi
Baktiku untukmu selalu...

Ayahanda Tercinta

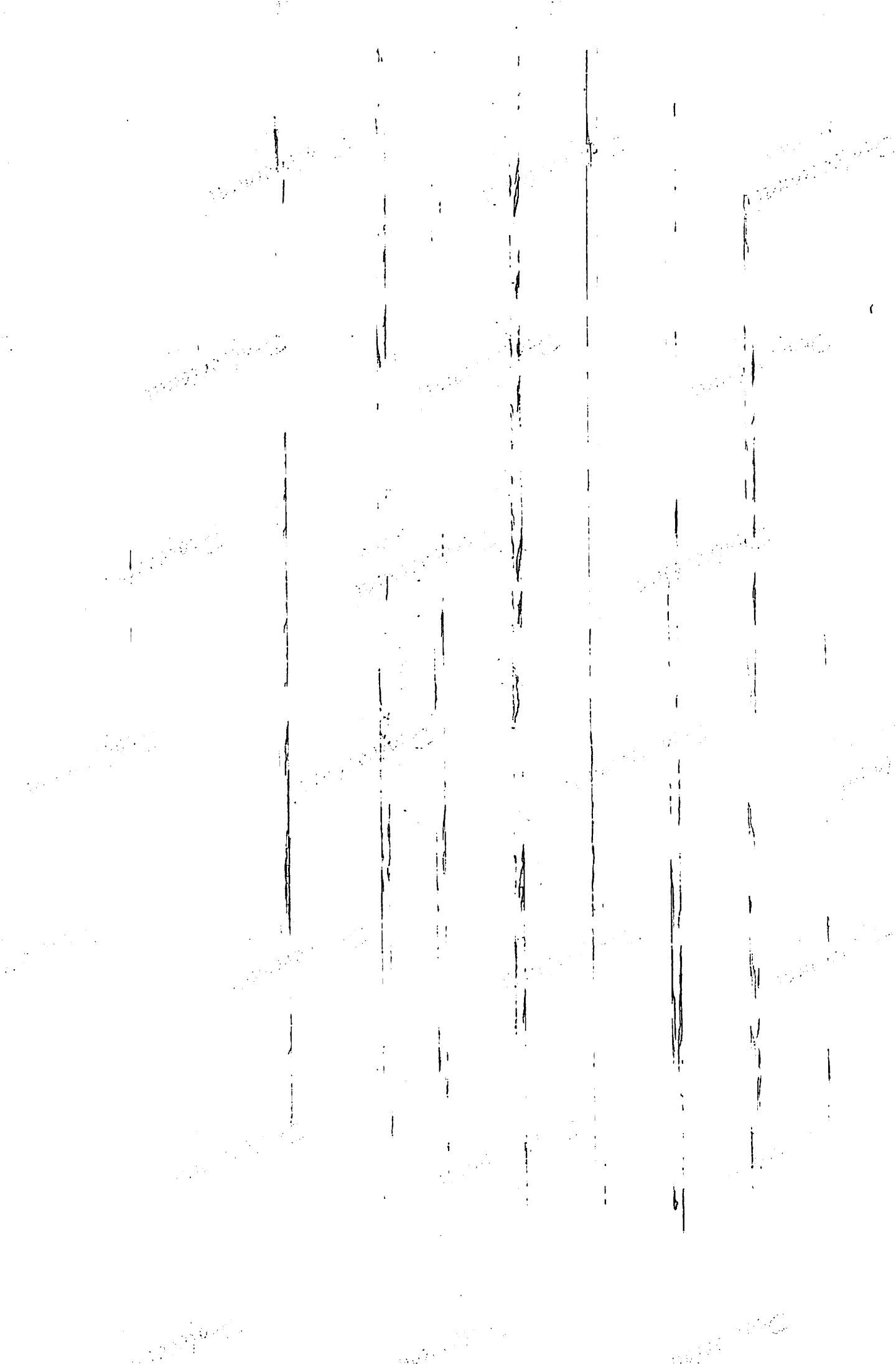
Cucuran keringatmu yang tanpa lelah membuatku bangga.
Tanpa pengorbananmu tak mungkin aku bisa begini
Saya hanya bisa berdoa, semoga diberi kesehatan jasmani dan rohani
Hormatku untukmu selalu...

Adik-adikku Tercinta

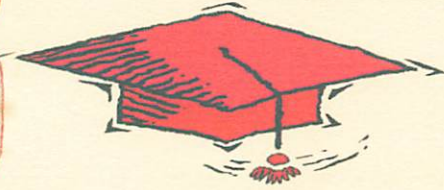
Adik cewek Rinda (Jangan maen terus n nakal ya...!klo disuruh ibu yang nurut dong!),
adik cowokku Dwi Hasan Kuncoro (Aku doakan Sukses terus di Jayapura,
sory ya aku baru bisa lulus sekarang).
Hanya Tawa, canda, dan tangisan kalian menghiasi hidupku.
Tanpa kalian hidupku terasa hampa
I love u for all..



Adikku tersayang Tantri Citra. Makasih atas doa n supportnya dorongan selama ini,
dengan sindiran-siindirinya supaya cepet lulus yang diberikan kepadaku sehingga dapat menyelesaikan kuliah dengan baik. Akhirnya aku bisa lulus to..!



SPECIAL THANKS TO: SPECIAL THANKS TO:



.....Bapak Ir. Choirul Saleh, MT. yang cakep

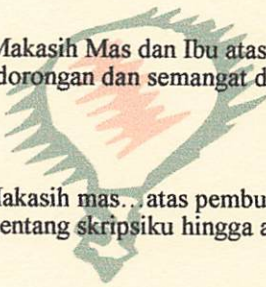
Terima kasih buanyak, atas dorongan semangat, bimbingan yang setia mendampingi dari seminar proposal, seminar hasil sampai komprehensif! Mohon maaf kalau sedikit g bisa jawab. Makasih juga ilmunya sehingga saya bisa mengerti tentang pembebanan ekonomis, biaya bahan bakar dalam pembangkit thermal dll, dalam bidang ilmu elektro dan mampu dalam menyelesaikan Tugas Akhir ini.

.....Bu Ida Barita, SH (Dosen Waliku)

Ibu baik deh! makasih selama ini sudah menganggap saya sebagai anak sendiri di kampus dan satu yang selalu kuingat dari nasihat ibu "terus belajar, berusaha dan berdoa". Maafin saya kalau selama ini saya sering minta tolong ibu minjamen buku di perpustakaan.

.....Mas Jayeng dan Bu Puji

Makasih Mas dan Ibu atas kesabaranmu kami akhirnya dapat lulus. Spesial Mas Jayeng bantuan, dorongan dan semangat dalam menyelesaikan kasus P. Made hingga tuntas dan bantuan selama persiapan seminar sampai yudisium tak akan pernah kulupakan.



.....Mas Ugro Wiseno (programmer)

Makasih mas... atas pembuatan programnya dan bimbingan teorinya sehingga aku bisa mengerti tentang skripsiku hingga aku bias lulus. Maaf kalo selama ini saya terlalu menuntut Mas Sugro untuk cepat menyelesaikan programku.

.....Mas Nurtanto (Dosen pembimbing 2)

To, kamu temen seperjuangan makasih bimbingane mungkin dak ada kamu skripsi ini dak selesai n aku mengerti tentang metode skripsiku dan makasih atas kebersamaannya, gue g nyesel kenal kamu (kapan nyari STMJ lagi).

.....KOST PAPAGABE

Ardian (Sang Pejuang Cinta tanpa tanda jasa) Semoga terkabul jadi Bupati, Rahmad (Aku ikut senang u udah bebas) pertahankan ibadahmu sukses dunia akhirat moy...!, Andi Mamak "alumni" (ingat tanggung jawabmu tambah besar sekarang) makasih support n flassdisnya, cepet gede aj buat Bintang!, Dwi Plencung "alumni" (awakmu tak susul wi, ojo lali lowongane) salam "I my Place", Pyman (maju terus pantang mundur u emang pejuang sejati), Lola "mama" tetep mesra aj ya sama papa (u bisa kok, percaya diri aj lagi), Cempluk (Terusin kuliah S1 nya dengan cepat, jgn banyak mainan yang macem-macem he...3x!), Mas Adi (jgn sering sakit ya, enter kuliahnya ketinggalan lo), Ghofur "papa" tambah mesra aj sama mama biar kelihatan cute gitu...!, Mas Tom (tetep Rajin aj ya) tolong jahain papagabe Ya!, Dany "Bajul" (jgn pindah-pindah kerja trus, klo ada lowongan kasih kabar aku ya), Yazid (klo cari kesibukan jgn bikin brisik papagabe dong), David "magrifat" (jagalah hati jgn sampe terlena oleh wanita, wanita itu setan), Dhek Henry (rukun trus ma Uun), Andik Brekele (ternyata kita wisuda bareng) semoga tetep cakep aj di depan kaca!, Verby, dicky, modin (cepat cari gebetan, keburu telat lo...!) ee.. sampai lupa buat susilo (sukses aj buat djarumnya) Enjoy aj Lagi...!

Terma kasih buat kalian semua,kebersamaan, kekeluargaan, keharmonis, keakrapan, dan saling menghargai merupakan isi dari kehidupanku sehari-hari bersama kalian. Aku betah disini karena kalian semua

.....Kost PAPAGACE

Indah, Inong, Uun, Iva, wardah, diana, Vinta, Kristin, Rossi, Pepy, Vell, Emil, Apit
Tawa canda kalian membuat keceriaan hidup buatku, walau dulu aku sering
nggodain! kalian semua aku anggap bagian keluarga dimalang, seneng banget deh
kenal kalian !

.....Laskar generasi biru Elektro

Rendra (sukses trus untuk asmaranya, tapi jgn lupa skripsinya tetep dikerjakan
lo..) ojo lali karo aku Boss!, Wahyu "Bom-Bom" makasih banget
jurnalnya, akhirnya aku bisa lulus. tak doain banyak harta, wanita, mati masuk
surga amin...!, Mas Rimo (sory mas aku nyonto sedikit...!). Didit (makasih tip n
trik nya serta wejangan-wejangan boneknya, akhire aku iso nututi awakmu),
Asrofi (Akhire awakke dewe iso wisuda bareng Pak De), Choky (jangan banyak
bikin dosa sama cewek apalagi TKW, ingat AZAB), Rahmad Y (makasih doa n
supportnya waktu kompre, g nyangka aku bisa lulus), Mince (hati-hati klo naek
motor jangan liat cewek trus), Farid (kapan mbakar jagung bareng), Gusti
andri, mandra, dany Fly, Ndemo, Eko tetelan, Diana (satu-satunya cewek yang
tersisa), gandhi, mukti, Dony prob, Rully, bambang degan, ngeb, jefri, boby (roy).

*Kenangan bersama kalian takan terlupakan tuk menempuh perkuliahan ini, Tk's
kekompakannya dan kerjasamanya selama kita kuliah.
Tang's semua meski aku dak bisa nyebutin satu-satu, aku bangga kuliah bersama kalian
semua di jurusan Elektro ini*

.....Barang-barangku

Motor Grand 3194 HM & Grand 3395 HM yang telah setia menemaniku dari
awal sampai akhir kuliah, yang membantu perjuanganku dalam urusan akademik
n organisasi sampek urusan cewek. Perangkat komputer Pentium IV yang selalu
menemaniku nglembur sampai rusak memorinya, entar klo ada rejeki tk upgrade
deh... biar g le mood bgt mikirnya! Serta HP Siemen C45 alat komunikasiku
yang membuatku kenal banyak orang n juga kugunakan konsultasi tentang
skripsiku! apalagi ada disampingmu setiap saat djamin tidak ketinggalan berita
deh...! sory ya klo slama ini kamu jarang ad pulsanya.

... Dan terima kasih pada semua pihak yang tidak bisa saya sebutkan satu-
persatu, atas bantuannya dalm menyelesaikan skripsi ini

Malang.....
Aku Merindukanmu

Blitar.....
Aku Menantangmu



DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
DAFTAR GRAFIK	ix

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi Penelitian.....	4
1.6 Kontribusi Pembahasan	4
1.7 Sistematika Penulisan	5

BAB II TEORI DASAR

2.1 Sistem Tenaga Listrik	7
2.2 Karakteristik Pembangkit Listrik Termal	8
2.2.1 Karakteristik <i>Masukan Keluaran</i>	9
2.2.2 Karakteristik Laju Tambahan Biaya Bahan Bakar	11

2.2.3	Pembebanan Ekonomis Pembangkit Listrik.....	13
2.2.4	Penyelesaian <i>Economic Dispatch</i> Dengan Metode Iterasi Lamda	16
2.3	Kendala Operasi PLTGU.....	16
2.4	Metode Evolutionary Programming	17
2.4.1	Langkah-langkah dalam Evolutionary Programming.....	18
2.4.2	<i>Economic Dispatch</i> dengan Metode <i>Evolutionary Programming</i>	18
2.4.3	Mekanisme Evolutionary Programming dengan CCP.....	19

BAB III PEMBANGKIT LISTRIK TENAGA GAS UAP

3.1	PLTGU Gresik.....	26
3.2	Konfigurasi dan Kapasitas Unit Pembangkitan Gresik	27
3.3	Produksi Energi Listrik pada PLTGU UP. Gresik.....	29
3.4	Pola Pembebanan pada PLTGU Unit Pembangkitan Gresik.....	31

BAB IV HASIL DAN ANALISA HASIL

4.1.	Pendahuluan.....	36
4.2	Data Pembangkitan.....	36
4.3	Perhitungan Persamaan Karakteristik Masukan Keluaran (Bahan Bakar) masing-masing Blok Unit Pembangkitan PLTGU Gresik.....	45
4.4	Perhitungan Distribusi Beban Blok	52
4.4.1	Penggunaan Program Komputer Dengan Metode Evolutionary Programming.....	53
4.4.2	Tampilan Program	54

BAB V KESIMPULAN

5.1. Kesimpulan	63
5.2. Saran	63

DAFTAR GAMBAR

Gambar 2-1. Elemen Pokok Sistem Tenaga Listrik.....	8
Gambar 2-2. Blok Pembangkit Listrik Tenaga Uap	9
Gambar 2-3. Kurva Karakteristik <i>Masukan Keluaran</i> Unit Pembangkit Listrik Termal	11
Gambar 2-4. Kurva Karakteristik <i>Incremental Fuel Cost Rate</i>	12
Gambar 2-5. Diagram Segaris CCPP Konfigurasi 3.3.1.....	13
Gambar 2-6. N Unit Pembangkit Thermal Melayani Beban P_D	14
Gambar 2-7. Ilustrasi Proses Mutasi.....	21
Gambar2-6.1 Flowchart Economic Dispatch menggunakan Metode Evolutionary Programming.....	24
Gambar 2-6.2. Flowchart Fungsi <i>Fitness</i>	25
Gambar 4-1. Tampilan Utama Program.....	54
Gambar 4-2. Input Data General.....	54
Gambar 4-3. Input Data Generator	55
Gambar 4-4. Input Data Tingkatan Pembebanan Pola Gresik.....	56
Gambar 4-5. Parameter <i>Evolutionary Programming</i>	57
Gambar 4-6. Hasil Optimasi Pembebanan Ekonomis dengan EP	58

DAFTAR TABEL

Tabel 3.1.	Tipe Pembangkit Unit Pembangkitan Gresik	27
Tabel 3.2.	Tabel Distribusi Beban Blok Pada Konfigurasi 3.3.1 - 3.3.1 - 3.3.1.....	33
Tabel 4.1.	Total Aliran BBG per tingkat beban Pada Blok 1	37
Tabel 4.2.	Total Aliran BBG per tingkat beban Pada Blok 2	38
Tabel 4.3.	Total Aliran BBG per tingkat beban Pada Blok 3	39
Tabel 4.4.	Fungsi Biaya Bahan Bakar Masing-masing Blok Terhadap Tingkat Bebab.....	48
Tabel 4.5.	Fungsi Laju Tambahan Bahan Bakar Masing-masing Blok Terhadap Tingkat Beban.....	49
Tabel 4.6.	Tabel fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing Blok Pada PLTGU Gresik.....	52
Tabel 4.10.	Tabel fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing Dengan Metode Evolutionary Programming ..	62
Tabel 4.6.	Tabel Hasil Penghematan Biaya Bahan Bakar Dengan Metode Evolutionary Programming dibanding PLTGU Gresik.....	59

DAFTAR GRAFIK

Grafik 4.1. Konsumsi Bahan Bakar Blok 1	41
Grafik 4.2. Konsumsi Bahan Bakar Blok 2	42
Grafik 4.3. Konsumsi Bahan Bakar Blok 3	43
Grafik 4.4. Konsumsi Bahan Bakar Pada Kombinasi 3.3.1.....	44
Grafik 4.5. Fungsi Biaya Bahan Bakar Pada Konfigurasi 3.3.1	50
Grafik 4.6. Fungsi Laju Tambahan Biaya Bahan Bakar Pada Konfigurasi 3.3.1.....	51

BAB I

PENDAHULUAN

1.1 Latar Belakang^[3]

Pembangkitan tenaga listrik merupakan bagian dari permasalahan energi dan lingkungan yang dihadapi oleh Indonesia sebagai negara berkembang. Secara garis besar, suatu sistem tenaga listrik dibagi menjadi 3 bagian yaitu : sisi pembangkit tenaga listrik, jaringan transmisi dan beban. Untuk suatu operasi pada beban tertentu, perhitungan ekonomis harus tetap merupakan suatu prioritas atau nilai yang harus diperhitungkan disamping hal-hal yang lain, sehingga nantinya diperlukan suatu rencana operasi yang optimum dengan tetap memenuhi beberapa persyaratan pengoperasian sistem tenaga listrik yaitu antara lain daya yang dibangkitkan cukup untuk memasok beban. Dalam pembangkitan tenaga listrik dilakukan usaha agar biaya pembangkitannya semurah mungkin. Usaha untuk mengoptimalkan biaya operasi ini , salah satunya dilakukan dengan penerapan *Economic Dispatch*. Didalam operasi sistem tenaga listrik *Economic Dispatch* adalah hal yang sangat perlu diperhatikan untuk mendapatkan biaya bahan bakar yang ekonomis (murah) dalam suatu sistem pembangkit.

Dalam pemenuhan kebutuhan akan energi listrik dari para pelanggan maka diperlukan perencanaan penggunaan pembangkit listrik yang ada secara efisien dan seoptimal mungkin. Namun kendala utama yang dihadapi oleh pembangkit listrik tenaga termal adalah tingginya biaya yang harus

dikeluarkan untuk membeli bahan bakar. Upaya menekan biaya dan pertimbangan akan besarnya kebutuhan energi dalam negeri serta kelestarian sumber daya energi untuk masa mendatang mendorong dilakukannya usaha-usaha konversi energi.

Bertitik tolak dari masalah konservasi energi tersebut, khususnya dalam tahap pengoprasian pembangkit listrik tenaga termal, telah ditempuh cara-cara tertentu antara lain dengan melakukan pengaturan pembebanan yang optimum (ekonomis) pada sebuah pembangkit listrik termal. Sebuah tipe baru Pembangkit Listrik Tenaga Termal mulai dikembangkan dan menjanjikan prospek yang demikian luas. Pembangkit Listrik Tenaga Termal tersebut adalah Combined Cycle Power Plant (CCPP) atau dalam bahasa Indonesia disebut Pusat Listrik Tenaga Uap (PLTGU).

1.2 Rumusan Masalah

Pada pembebanan di PLTGU Unit Pembangkitan Gresik pengaturan distribusi beban perlu dipertimbangkan.

Untuk memecahkan permasalahan di atas, akan digunakan metode Evolutionary Programming. Maka dari itu muncul beberapa pertanyaan, antara lain:

1. Apakah dengan metode pengaturan pembebanan yang dipakai pada PLTGU Gresik telah tercapai pembebanan ekonomis (economic dispatch).

2. Apakah dengan metode Evolutionary Programming dapat dicapai pembebanan ekonomis (economic dispatch) yang lebih baik.

Berdasar pada apa yang ada pada permasalahan di atas maka kami menyusun skripsi dengan judul :

**“ANALISIS OPTIMASI ECONOMIC DISPATCH COMBINED
CYCLE POWER PLANT DENGAN METODE EVOLUTIONARY
PROGRAMMING”**

1.3 Tujuan Pembahasan

Adapun tujuan dari skripsi ini adalah :

- Menentukan pola pembebanan bagi masing-masing blok yang terdapat di *Combine Cycle Power Plant* (CCPP) PT. PLN. PJB II Unit Pembangkitan Gresik dengan Metode Evolutionary Programming agar dapat dioperasikan lebih optimum (ekonomis).
- Membandingkan hasil yang diperoleh dengan pola pembebanan yang telah diterapkan di *Combine Cycle Power Plant* PT. PLN. PJB II Unit Pembangkitan Gresik.

1.4 Batasan Masalah

Agar skripsi ini mencapai sasaran, maka pembahasan masalah dibatasi:

- Menganalisa Optimasi pembebanan hanya pada PLTGU Gresik saja.
- Menggunakan Metode Evolutionary Programming.
- Menganalisa Unit Pembangkitan Termal berbahan bakar gas.

- Pengamatan dilakukan terhadap CCPP Unit Pembangkitan Gresik yang beroperasi pada pola konfigurasi 3.3.1-3.3.1-3.3.1, yaitu kapasitas maksimum pembangkit.
- Rugi-rugi transmisi diabaikan.

1.5 Metodologi Pembahasan

Metodologi yang digunakan dalam pembahasan dilaksanakan dengan langkah-langkah sebagai berikut :

- Studi kepustakaan mengenai hal-hal yang berhubungan dengan pembahasan masalah.
- Studi lapangan untuk mendapatkan data-data yang dibutuhkan serta mencari data dari objek yang diperlukan sesuai dengan pedoman teori yang diperoleh dari studi kepustakaan.
- Melakukan perhitungan dari data lapangan yang berkaitan dengan optimasi pembebanan dengan metode Evolutionary Programming kemudian membandingkannya dengan pencapaian optimasi Pembebanan pada PLTGU Gresik.
- Membuat evaluasi, sehingga dapat disimpulkan apakah dengan metode Evolutionary Programming yang diterapkan lebih efisien (ekonomis).

1.6 Kontribusi Pembahasan

Dengan adanya analisa ini diharapkan nantinya dapat memberikan alternatif terbaik dalam pemecahan yang ada pada sistem Combine Cycle

Power Plant (CCPP) untuk mendapatkan pembebanan yang optimum (ekonomis) pada masing-masing blok serta harus memperhatikan kendala-kendala pengoperasian pembangkit listrik tenaga temal yang bersangkutan sehingga produksi tenaga listrik dapat dilakukan dengan biaya operasi serendah mungkin.

1.7 Sistematika Penulisan

Penulisan skripsi ini terdiri atas beberapa bab dan sub bab dan tersusun dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Terdiri atas latar belakang penulisan, rumusan masalah, maksud dan tujuan penulisan skripsi, lingkup bahasan dan sistematika penulisan.

BAB II TEORI DASAR

Berisi antara lain teori dasar tentang sistem tenaga listrik dan karakteristik pembangkitan termal, pembebanan ekonomis pembangkitan termal, economic dispatch menggunakan metode Evolutionary Programming, kendala operasi PLTGU.

BAB III PEMBANGKIT LISTRIK TENAGA GAS UAP

Berisi data konsumsi bahan bakar masing-masing blok, konfigurasi PLTGU Unit Pembangkitan Gresik.

BAB IV ANALISA DATA

Berisi tentang hasil perhitungan dengan metode Evolutionary Programming dibandingkan dengan pola pembebanan yang sekarang digunakan di PLTGU Gresik, dengan menggunakan data yang diperoleh selama survey lapangan sebagai variabelnya.

BAB V KESIMPULAN

Berisi tentang kesimpulan akhir, dan saran-saran.

BAB II

TEORI DASAR

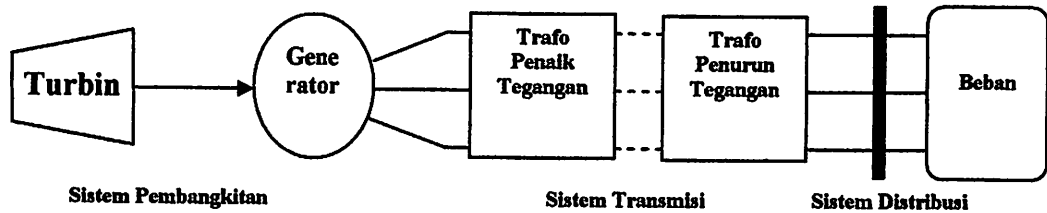
2.1 Sistem Tenaga Listrik^[3]

Performa dari sebuah pusat pembangkit tenaga listrik pada prinsip ditentukan oleh apa yang dinamakan lengkung masukan-keluaran (Input-Output).. Lengkung ini memberikan gambaran tentang efisiensi termis pusat pembangkit tersebut. Selain tergantung pada sifat-sifat pusat tenaga listrik itu sendiri, seperti keadaan air, pendingin kualitas bahan bakar, kecakapan para operator pusat pembangkit dan bentuk lengkung beban.

Berikut ini macam-macam karakteristik pembangkit yang berhubungan dengan penjadwalan operasi pembangkit untuk memperjelas keterangan diatas.

Jaringan setelah keluar dari gardu induk umumnya disebut jaringan distribusi, sedangkan jaringan antara pusat listrik dengan gardu induk disebut jaringan transmisi. Setelah itu disalurkan melalui jaringan distribusi primer maka tenaga listrik kemudian diturunkan tegangannya oleh gardu-gardu distribusi menjadi tegangan 380/220 volt atau 220/127 volt, dan baru kemudian disalurkan ke konsumen.

Dari uraian diatas kiranya dapat dimengerti bahwa besar kecilnya tenaga listrik ditentukan sepenuhnya oleh konsumen, yaitu tergantung bagaimana para konsumen akan menggunakan peralatan listriknya, kemudian PT. PLN (Persero) harus mengimbangi kebutuhan tenaga listrik tersebut dalam arti selalu menyesuaikan daya listrik yang dibangkitkan dari waktu ke waktu.



Gambar 2-1 : Elemen Pokok Sistem Tenaga Listrik

Untuk memenuhi kebutuhan tenaga listrik bagi para konsumen, diperlukan berbagai peralatan listrik. Berbagai peralatan listrik ini dihubungkan satu sama yang lain secara keseluruhan membentuk suatu sistem tenaga listrik. Sehingga yang dimaksud dengan sistem tenaga listrik di sini adalah sekumpulan pusat listrik dan gardu induk yang satu dengan yang lain dihubungkan oleh jaringan transmisi sehingga merupakan sebuah kesatuan interkoneksi. Oleh karena itu daya listrik yang dibangkitkan harus sama dengan tenaga listrik yang dibutuhkan oleh konsumen

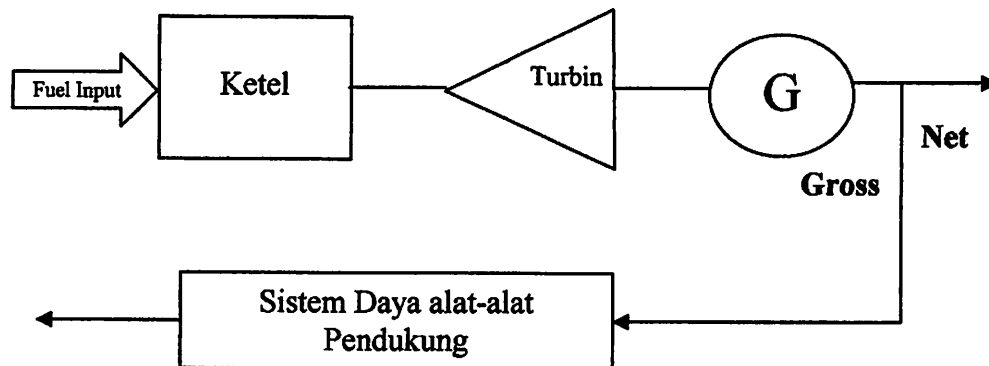
2.2 Karakteristik Pembangkit Listrik Termal

Dasar untuk menyelesaikan masalah pembebanan ekonomis (economic dispatch) dari sebuah pembangkit listrik tenaga termal sangat ditentukan oleh karakteristik masukan-keluaran (input-output characteristics) pusat listrik tersebut.

Tipe dari sebuah blok pembangkit listrik tenaga termal tampak pada gambar 2-2. Blok tersebut terdiri atas sebuah ketel yang menghasilkan uap untuk menggerakkan turbin uap yang akan dikopel dengan sebuah generator listrik. Daya listrik yang dihasilkan tidak seluruhnya disalurkan ke sistem

tetapi sebagian kecil digunakan untuk mengoperasikan peralatan yang terdapat pada pusat listrik tersebut. Sebuah PLTU kira-kira memerlukan 2% sampai 6% dari daya listrik yang dihasilkan oleh generator untuk mengoperasikan berbagai peralatan seperti ketel, pompa, kompresor, dan sebagainya, serta untuk mencatu peralatan kontrol, telemetri, komunikasi, penerangan dan komputer.

Dalam mendefinisikan karakteristik masukan-keluaran sebuah pembangkit listrik tenaga termal, akan dibiarkan tentang gross input dan net output yang dihasilkan pusat listrik tersebut. Gross input pembangkit listrik termal menyatakan jumlah keseluruhan bahan bakar yang diperlukan, sedangkan net output adalah daya nyata (real power) yang dihasilkan generator setelah dikurangi untuk keperluan sendiri.



Gambar 2-2 : Blok Pembangkit listrik tenaga uap

2.2.1 Karakteristik Masukan Keluaran

Masukan sebuah Pembangkit Listrik Termal umumnya dinyatakan sebagai banyaknya energi persatuan waktu dari bahan bakar yang diberikan ke

ketel untuk menghasilkan daya listrik yang merupakan keluaran dari pusat listrik tersebut. Terdapat dua notasi yang umum digunakan :

H dengan satuan [MBTU/hour]

F dengan satuan [\$US/hour],

dimana $F = H \times \$US/BTU$, dan $\$US/BTU$ menyatakan harga bahan bakar persatuan energi yang dikandung oleh bahan bakar tersebut. Sedangkan keluaran dari pembangkit listrik termal adalah daya nyata yang dihasilkan oleh generator dikurangi dengan daya nyata yang dipakai oleh pusat listrik tersebut.

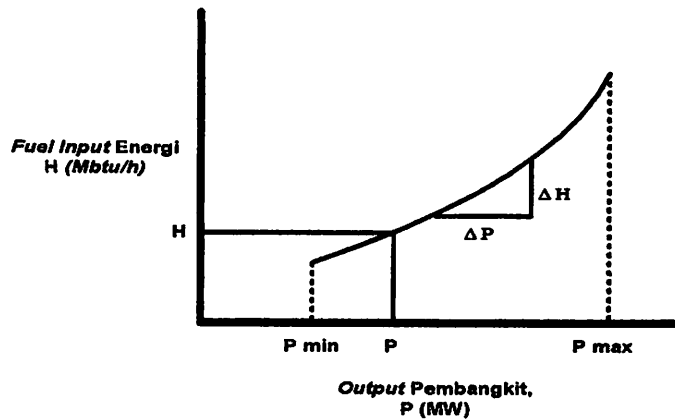
Notasi yang digunakan adalah :

P dengan satuan [MW],

jika dinyatakan bahwa masukan pusat listrik merupakan fungsi terhadap keluarannya, maka hubungan tersebut dapat ditulis sebagai berikut :

$$H = f(P) \text{ [MBTU/H] atau } F = f(P) \text{ [\$US/H]}$$

Pembahasan selanjutnya akan berpedoman atas dasar fungsi biaya bahan bakar ($F = f(P) \text{ [\$US/H]}$) pada keadaan tertentu biaya-biaya lain yang merupakan fungsi terhadap keluaran pusat listrik dimasukkan ke dalam perhitungan biaya bahan bakar, misalnya biaya perawata (maintenance cost), upah pekerja (labour cost), dan biaya pengoprasian (operational cost).



**Gambar 2-3 : Diagram Karakteristik Masukan-Keluaran
Pembangkit Listrik Termal**

Data yang diperlukan untuk menggambarkan diagram fungsi karakteristik masukan-keluaran dapat diperoleh dari perhitungan pada saat perencanaan atau tes yang telah dilakukan terhadap unit pembangkit yang bersangkutan.

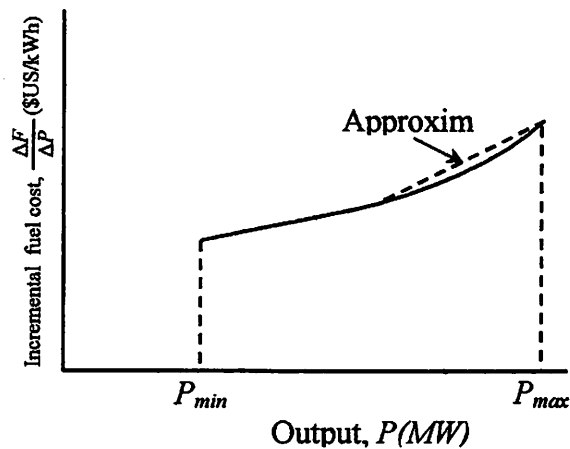
2.2.2 Karakteristik Laju Tambahan Biaya Bahan Bakar

Karakteristik laju tambahan biaya bahan bakar atau *Incremental Fuel Cost Characteristic* adalah turunan pertama dari fungsi biaya bahan bakar F [\$/h] terhadap tingkat pembebanan P [MW] dari pusat listrik yang bersangkutan. Fungsi ini menunjukkan besarnya kenaikan atau penurunan biaya bahan bakar untuk setiap satu satuan perubahan beban.

Secara luas fungsi biaya bahan bakar akan digunakan untuk menentukan pembebanan ekonomis dari sebuah pembangkit listrik tenaga termal. Tampak pada gambar 2 - 4 kurva laju tambahan biaya bahan bakar

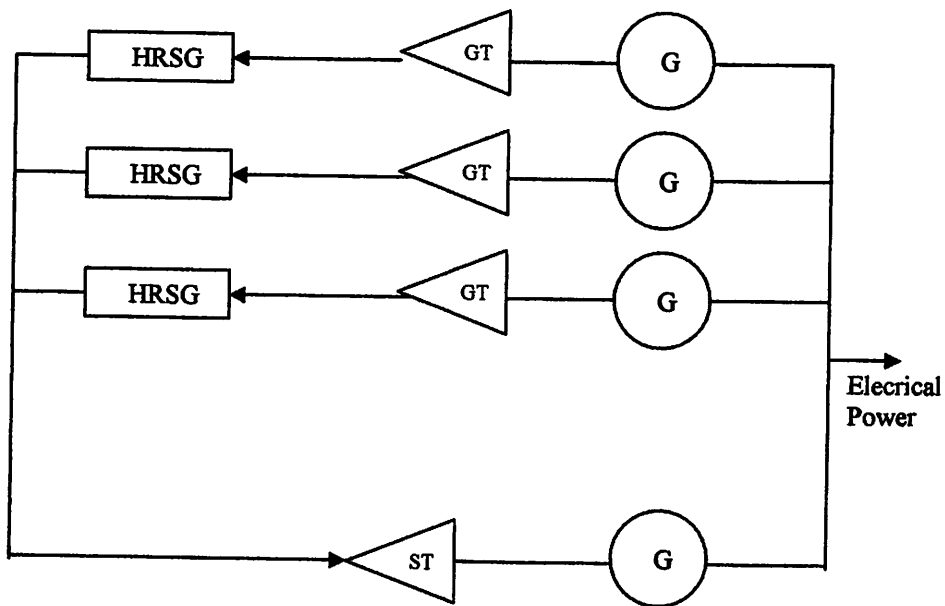
yang telah diidealkan melalui pendekatan linier dari sebuah pembangkit listrik termal.

Gambar 2 -4
Kurva Karakteristik Laju Tambahan Biaya Bahan Bakar



Pembangkit Listrik Tenaga Termal mulai dikembangkan dan menjanjikan prospek yang demikian luas. Pembangkit Listrik Tenaga Termal tersebut adalah Combined Cycle Power Plant (CCPP) atau dalam bahasa Indonesia disebut Pusat Listrik Tenaga Gas Uap (PLTGU).

Pada CCPP gas panas yang dibuang hasil pembakaran yang dibuang oleh turbin gas tersebut dimanfaatkan kembali melalui peralatan yang disebut *Heat Recovery Steam Generator* (HRSG) untuk menghasilkan siklus uap-air yang akan digunakan memutar sebuah turbin uap. Pembangkit listrik tenaga termal jenis ini memiliki keuntungan karena efisiensi siklus termalnya menjadi lebih tinggi. Pada gambar 2-5 tampak diagram segaris dari sebuah blok CCPP yang terdiri atas tiga turbin gas, tiga HRSG dengan satu turbin uap.



Gambar 2-5 : Diagram Segaris CCPP konfigurasi 3.3.1

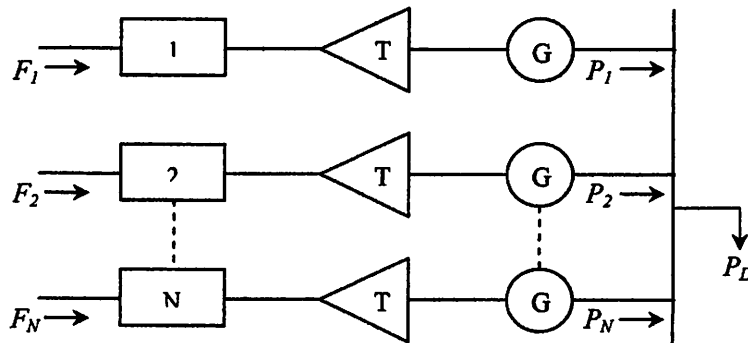
2.2.3 Pembebanan Ekonomis Pembangkit Listrik

Pembebanan ekonomis atau *economic dispatch* adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam sistem tenaga listrik, secara optimal dan ekonomis pada harga beban tertentu. Komponen terbesar dari biaya pembangkitan adalah biaya bahan bakar. Oleh sebab itu dengan dilakukannya *economic dispatch* berarti pula didapatkan biaya bahan bakar pembangkitan yang paling murah.

Sistem tenaga listrik dengan mengabaikan rugi-rugi transmisi dapat dilihat pada gambar 2-6. Sistem ini memperlihatkan pembangkit termal yang terdiri atas N buah unit yang dihubungkan pada sebuah bus bar untuk melayani total beban sebesar P_D . Masukan untuk setiap unit ke- i adalah F_i yang menyatakan tingkat biaya bahan bakar dari masing-masing unit, dan daya

keluaran dari masing-masing unit P_i adalah daya listrik yang dibangkitkan oleh tiap-tiap unit.

Biaya total F_T yang ditanggung sistem adalah jumlah biaya dari tiap-tiap unit pembangkit. Dan batasan yang paling penting dari pengoperasian pembangkit termal tersebut adalah daya listrik yang dihasilkan harus sama dengan besarnya beban konsumen.



Gambar 2-6 : N Unit Pembangkit Termal Melayani Beban P_D

Fungsi obyektif (objective function) F_T sama dengan biaya bahan bakar total yang dibutuhkan untuk melayani beban dari pusat listrik tersebut. Dengan mengabaikan rugi-rugi yang terjadi pada jaringan transmisi, gambaran diatas dapat pula dijabarkan secara matematis seperti dibawah ini.

Biaya bahan bakar total pada Pembangkit Termal dengan N Blok adalah :

$$F_T = F_1 + F_2 + \dots + F_N$$

$$F_T = \sum_{i=1}^N F_i(P_i) \dots\dots\dots (2.1)$$

Dan daya listrik yang dihasilkan oleh setiap Blok untuk melayani beban total adalah:

$$P_R = \sum_{i=1}^N P_i \dots\dots\dots(2.2)$$

$$P_R - \sum_{i=1}^N P_i = \phi = 0 \dots\dots\dots(2.3)$$

dimana P_R = kebutuhan beban, dan

P_i = jumlah daya yang dihasilkan.

Penyelesaian permasalahan optimasi terkendala seperti ini dapat diselesaikan dengan metode yang menyangkut Fungsi Lagrange:

$$L = F_i + \lambda \phi \dots\dots\dots(2.4)$$

atau

$$L = \sum_{i=1}^N F_i(P_i) + \lambda \left(P_R - \sum_{i=1}^N P_i \right) \dots\dots\dots(2.5)$$

dimana λ = Lagrange Mutiplier.

Bila L diturunkan terhadap λ maka diperoleh :

$$\left(\partial L / \partial P_i \right) = \frac{dF_i(P_i)}{dP_i} - \lambda \dots\dots\dots(2.6)$$

Untuk mencari harga optimal fungsi Lagrange terhadap P_i , persyaratan yang harus dipenuhi adalah $(\partial L / \partial P_i) = 0$

$$\text{atau } \frac{dF_i(P_i)}{dP_i} - \lambda = 0$$

$$\text{atau } \frac{dF_i(P_i)}{dP_i} = \lambda \dots\dots\dots(2.7)$$

2.2.4 Penyelesaian *Economic Dispatch* Dengan Metode Iterasi Lamda.

Pada metode iterasi lamda (λ), lamda ditentukan dahulu kemudian dihitung harga keluaran untukn tiap-tiap unit pembangkit. Bila jumlah dari daya keluaran tiap-tiap unit pembangkit tidak sama dengan daya beban maka dilakukan perhitungan kembali harga lamda berikutnya untuk mendapatkan lamda berikutnya yang mendekati solusi sebenarnya, setelah beberapa kali iterasi.

2.3 Kendala Operasi PLTGU

PLTGU pada prinsipnya merupakan gabungan antara PLTG dan PLTU. Pembangunan PLTGU dimaksudkan untuk mengurangi kendala-kendala operasi yang dihadapi PLTG dan PLTU apabila masing-masing beroperasi sendiri-sendiri (independent) namun karena turbin uap pada PLTGU masih dapat dioperasikan meskipun turbin uapnya dimatikan, maka dalam konfigurasi operasi yang demikian kendala-kendala oprasi PLTGU pada dasarnya sama dengan kendala-kendala operasi PLTG.

Hal yang diuntungkan dari pengoperasian PLTGU adalah kapasitas maksimum yang meningkat dan kecepatan melakukan perubahan beban. Untuk tingkat bahan bakar yang sama, tentunya PLTGU akan dapat membangkitkan daya listrik yang lebih besar daripada PLTU dan PLTG.

Ciri-ciri spesifik dari operasi PLTGU adalah:

- PLTU hanya dapat membangkitkan daya listrik bila PLTG beroperasi.

- Kurva masukan-keluaran PLTGU bukan merupakan penjumlahan kurva masukan-keluaran dari PLTG dan PLTU, jadi sifatnya spesifik.

Sedangkan mengenai kecepatan perubahan beban lebih disebabkan adanya blok PLTG, sehingga apabila terjadi kenaikan beban pada Blok PLTG, Blok tersebut dapat merespon perubahan dengan cepat tanpa mengganggu operasional blok PLTU-nya.

Sistem yang demikian ini sangat menguntungkan bagi sebuah pembangkit termal. Selain karena efisiensi total pusat listrik tersebut meningkat, juga pusat listrik tersebut dapat dioperasikan untuk memikul beban dasar, selain juga memikul beban menengah, dan sewaktu-waktu dapat dioperasikan untuk memikul beban puncak yang perubahan bebannya cukup fluktuatif. Tetapi bagaimanapun hal ini tetap tergantung pada biaya bahan bakar yang diperlukan.

2.4 Metode Evolutionary Programming

Evolutionary Programming diperkenalkan oleh Lawrence J. Fogel pada tahun 1960. Merupakan strategi optimasi yang meniru prinsip evolusi alam sebagai metode untuk memecahkan masalah optimasi parameter. Individu yang kuat mempunyai kemungkinan untuk menjadi pemenang dan mempunyai kesempatan hidup yang lebih besar.

2.4.1 Langkah – langkah dalam Evolutionary Programming

Pada dasarnya Evolutionary Programming terdiri dari tiga langkah. Terjadi pengulangan sampai diperoleh solusi hasil yang diinginkan (maksimum).

Langkah – langkah tersebut adalah:

1. Pemilihan inisial populasi dari solusi percobaan secara acak. Banyaknya solusi pada populasi yang relevan pada kecepatan optimasi, tapi tidak ada batasan yang digunakan.
2. Masing – masing solusi diulang sampai diperoleh populasi baru. Masing – masing solusi offspring dimutasi berdasarkan distribusi jenis – jenis mutasi, berkisar antara kecil keadaan ekstrim dengan suatu jenis rangkaian mutasi. Mutasi ini ditentukan oleh parentnya.
3. Masing – masing solusi offspring ditentukan oleh perhitungan fitnessnya. Yang perlu digarisbawahi EP tidak menggunakan *CROSSOVER* seperti pada operator GA melainkan operasi *competition* (kompetisi).

2.4.2 Economic Dispatch dengan Metode Evolutionary Programming

Evolutionary Programming secara konseptual didasarkan pada genetika alami dan mekanisme evolusi yang bekerja pada populasi solusi dan berlawanan dengan teknik penelitian lainnya yang bekerja secara tunggal. Penelitian bukan pada ruang solusi parameter biner, tetapi pada bit string

dekodernya, mereka meniru genetika kromosom alami dengan mengaplikasikan operator seperti genetika dalam penelitian untuk memperoleh optimum global.

Aspek paling menarik dari Evolutionary Programming adalah walaupun mereka tidak membutuhkan pengetahuan sebelumnya dan mereka tidak membutuhkan ruang terbatas seperti kehalusan, kecembungan dari fungsi untuk dioptimumkan, mereka menunjukkan kinerja yang sangat baik untuk mayoritas problem yang diaplikasikan. Mereka hanya membutuhkan fungsi evaluasi untuk menentukan nilai kualitas (nilai kesesuaian) untuk setiap solusi yang dihasilkan. Ciri menarik lainnya adalah mereka secara inheren paralel (solusi secara individual tidak berhubungan satu sama lain), oleh karena itu implementasi mereka pada mesin paralel secara signifikan mengurangi waktu CPU yang dibutuhkan.

2.4.3. Mekanisme Evolutionary Programming dengan combined cycle Power Plant

A. Pengkodean atau Representasi

Langkah pertama kali yang dilakukan dalam penggunaan EP adalah melakukan pengkodean atau representasi terhadap permasalahan yang akan dilakukan.

Secara umum EP dibentuk oleh serangkaian kromosom yang ditandai dengan x_i ($i=1,2,\dots,N$). Setiap elemen dalam kromosom ini adalah variabel string yang disebut gen., berisi nilai-nilai allele. Variabel-variabel ini dapat dinyatakan dalam bentuk bilangan real (*floating point*).

Selanjutnya beberapa kromosom dibentuk dan berkumpul membentuk populasi. Populasi inilah populasi awal bagi EP untuk awal melakukan pencarian.

B. Fungsi *Fitness* (Fungsi Evaluasi)

Dalam EP, sebuah fungsi *fitness* $f(x)$ harus dirancang untuk masing-masing permasalahan yang akan diselesaikan. Dengan menggunakan kromosom tertentu, fungsi obyektif atau fungsi evaluasi akan mengevaluasi status masing –masing kromosom. Setiap gen x_i ($i = 1, 2, \dots, N$) dipergunakan untuk menghitung $f_k(x)$ ($k = 1, 2, \dots, \text{POPSIZE}$)

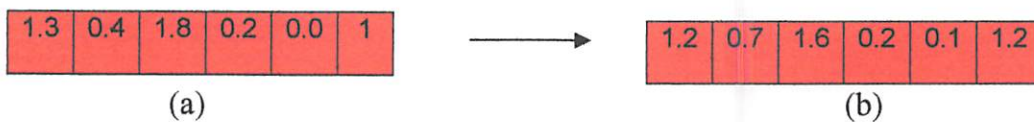
Pada permulaan optimasi, biasanya nilai *fitness* masing-masing individu masih mempunyai rentang yang lebar. Seiring dengan bertambah besar generasi, beberapa kromosom mendominasi populasi dan mengakibatkan rentang nilai *fitness* semakin kecil.

C. Seleksi

Masalah yang paling mendasar pada proses ini adalah proses penyeleksiannya. Menurut teori Darwin proses seleksi individu adalah : “individu terbaik akan tetap hidup dan akan menghasilkan keturunan”.

D. *Mutation* (Mutasi)

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam individu. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana $k = 1, 2, \dots, NVAR$ (panjang kromosom). Probabilitas mutasi (P_m) ditentukan dan digunakan untuk mengendalikan frekuensi operator mutasi. Apabila nilai random r_k, P_m maka gen ke-k kromosom tersebut terpilih untuk mengalami mutasi. Proses mutasi dalam *Evolutionary Strategies* yaitu menggunakan operator *Gaussian mutation*, dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak Gaussian untuk untuk menciptakan individu baru (*offspring*).



Gambar 3.1. Ilustrasi Proses Mutasi

a). Mutasi Gaussian dari Induk (*parent*) b). Menghasilkan Anak (*offspring*)

Fungsi dari operator mutasi adalah untuk menghindari agar solusi permasalahan yang diperoleh bukan merupakan solusi optimum lokal. Tipe dan implementasi dari operator mutasi bergantung dari pada jenis pengkodean dan permasalahan yang dihadapi. Seberapa sering mutasi dilakukan dinyatakan dengan suatu probabilitas mutasi, P_m . Posisi elemen pada kromosom yang akan mutasi ditentukan secara random. Mutasi dikerjakan dengan cara melakukan perubahan pada elemen tersebut.

E. *Competition* (kompetisi)

Dalam tahap kompetisi, mekanisme seleksi dipakai untuk menghasilkan populasi baru dari populasi yang ada. Melalui penggunaan skema kompetisi setiap individu dalam populasi baik orang tua (*parent*) maupun anak (*offspring*) akan dikompetisi/bersaing satu dengan yang lainnya. Kompetisi setiap individu dengan lawannya didasarkan pada nilai *fitness* dari setiap individu tersebut. Agar optimal, solusi yang lebih pas atau lebih optimal seharusnya memiliki peluang seleksi yang lebih besar. Individu yang memenangkan dari kompetisi akan digunakan sebagai individu yang baru bagi pembangkitan selanjutnya.

2.5 Algoritma Program

2.5.1. Algoritma Penyelesaian *Economic Dispatch* menggunakan Metode *Evolutionary Programming*

Urutan langkah-langkah dalam program komputer yang menggunakan metode *Evolutionary Programming* adalah sebagai berikut :

1. Memasukkan inputan data parameter unit pembangkit termal dan data tiap-tiap blok pembangkit yaitu jumlah unit pembangkit, daya maksimum, daya minimum, konstanta persamaan biaya bahan bakar.
2. Memasukkan inputan parameter *Evolutionary programming* yang meliputi jumlah populasi, jumlah generasi.

3. Menentukan untuk populasi awal ($pop = 0$) dan menentukan untuk generasi awal ($gen = 0$).
4. Menghitung nilai *fitness* populasi awal atau populasi *parent*..
5. Melakukan proses statistik.
6. Melakukan proses seleksi.
7. Melakukan proses mutasi.
8. Apakah *offspring* sama dengan jumlah *parent*.
 - a. Jika “Ya” maka dilanjutkan ke langkah 9
 - b. Jika “Tidak” maka kembali ke langkah 6
9. Melakukan proses kompetisi.
10. Diperiksa, apakah generasi sudah maksimum.
 - a. Jika “Ya” maka perhitungan selesai.
 - b. Jika “Tidak” maka $gen = gen + 1$ dan kembali ke langkah 6

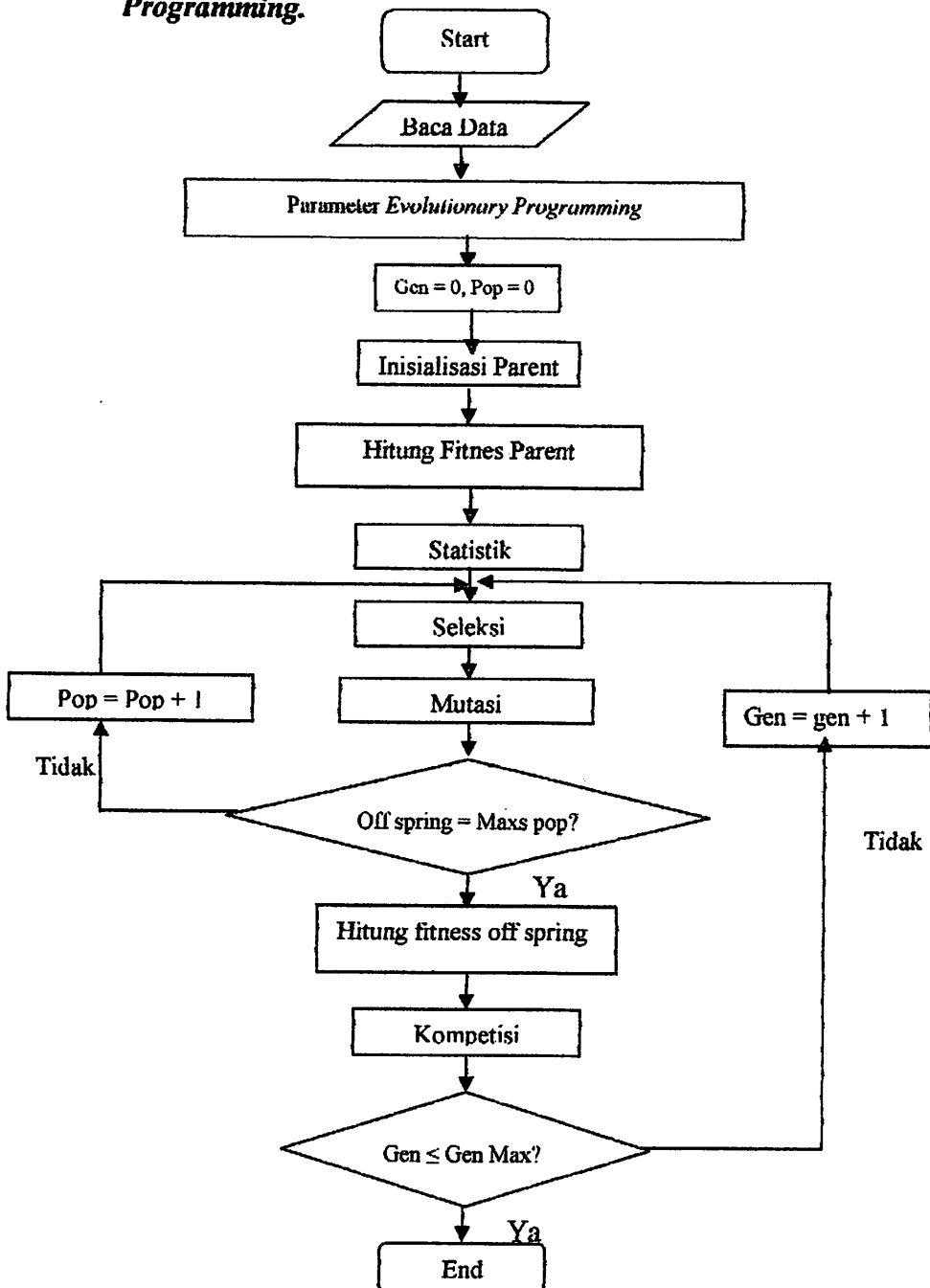
2.5.2 Algoritma Fungsi *Fitness*

1. Memasukkan input kromosom yang dilakukan secara random (acak) menggunakan metode *Evolutionary Programming*
2. Menghitung objective function total fuel cost
3. Selesai

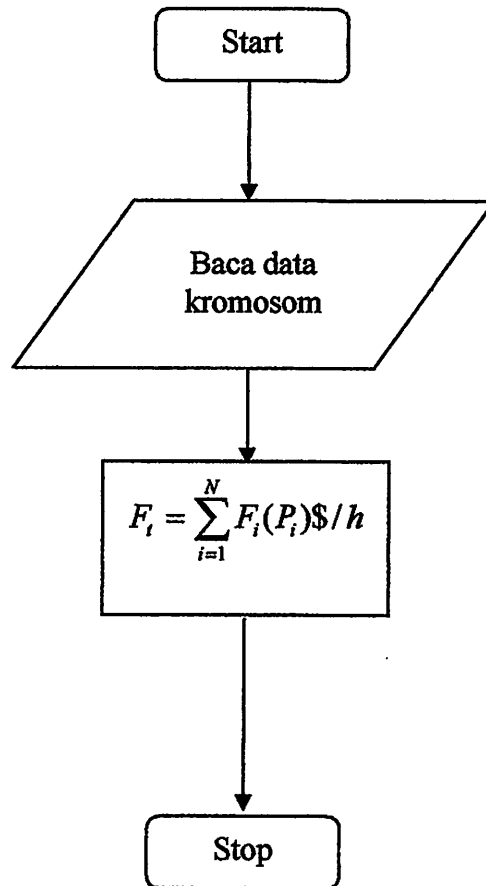
2.6 Flowchart

2.6.1 Flowchart Economic Dispatch menggunakan Metode *Evolutionary Programming*

Programming.



2.6.2 Flowchart Fungsi *Fitness*



BAB III

Pembangkit Listrik Tenaga Gas Uap

3.1. PLTGU Gresik

Pada pertengahan tahun 1978 untuk pertama kali di kota Gresik dibangun Pusat Listrik Tenaga Gas (PLTG) dengan total kapasitas 40 MW yang terdiri dari 2 mesin pembangkit masing-masing berkapasitas 20 MW, dalam wilayah unit kerja PLN sektor Perak. Awal tahun 1981 di lokasi yang sama dibangun lagi 2 pusat Listrik Tenaga Uap (PLTU) yang berkapasitas masing-masing 100 MW. Berdasarkan SK direksi PLN no. 023 / DIR / 1981 tgl 16 Maret 1981, kedua pusat listrik tersebut di jadikan satu wilayah kerja tersendiri dengan nama PLN sektor Gresik. Penambahan mesin pembangkit pun terus dilaksanakan sejalan dengan tuntutan kebutuhan tenaga listrik hingga pada awal tahun 1989 total kapasitas PLN sektor Gresik menjadi ± 702 MW.

Pada pertengahan tahun 1996 berdasarkan SK Dirut PT PLN PJB II no.023.K/Dir/1992, tanggal 4 Pebruari 1992, dibentuk sektor Gresik Baru (SGRB) dengan total kapasitas 1578,78 MW yang terdiri atas 3 blok daur ganda (combined cycle) yang masing-masing blok terdiri atas 3 unit GT 9 (per GT berkapasitas 112,45 MW) dan 1 unit ST yang berkapasitas 188,98 MW.

Pada tahun 1997 terjadi penggabungan antara sektor Gresik baru dengan nama baru yaitu PT PLN PJB II Unit Pembangkitan Gresik.

3.2 Konfigurasi dan Kapasitas Unit Pembangkitan Gresik

PLTGU Unit pembangkitan Gresik terdiri dari 3 blok, masing-masing blok terdiri atas 3 buah Turbin gas dengan 3 Generator Turbo, 3 HRSG dan 1 buah Turbin uap dengan 1 generator turbo. Blok 1 dan blok 2 dapat menggunakan bahan bakar berupa High Speed Diesel (HSD), dan Gas Alam atau Natural Gas (NG), sedangkan Blok 3 hanya berbahan bakar gas.

Terdapat beberapa kemungkinan konfigurasi untuk mengoperasikan Blok PLTGU Unit Pembangkitan Gresik, yaitu:

- Konfigurasi 1 GT, 1 HRSG, dan 1 ST
- Konfigurasi 2 GT, 2 HRSG, dan 1 ST
- Konfigurasi 3 GT, 3 HRSG, dan 1 ST

Tabel 3.1

Tipe Pembangkit di Unit Pembangkitan Gresik

NO	BLOK	FUEL	KAPASITAS	PABRIKAN
	Gas Turbin Power Plant			
I	GT 1	HSD/NG	20 MW	Alsthom-France
	GT 2	HSD/NG	20 MW	Alsthom-France
	GT 3	HSD	21 MW	Alsthom-France
	GT 4	HSD/NG	20 MW	GE- USA
	GT 5	HSD/NG	21 MW	GE- USA
	Steam Turbin Power Plant			
II	ST 1	HSD	100 MW	Toshiba-Japan
	ST 2	HSD	100 MW	Toshiba-Japan
	ST 3	HSD/NG	200 MW	Toshiba-Japan
	ST 4	HSD/NG	200 MW	Toshiba-Japan

Combined Cycle Power Plant				
III	Blok 1			
	GT 1.1	NG	112,45 MW	Mitsubishi
		HSD	100,98 MW	
	GT 1.2	NG	112,45 MW	Mitsubishi
		HSD	100,98 MW	
	GT 1.3	NG	112,45 MW	Mitsubishi
		HSD	100,98 MW	
	ST 1.0	-	188,92 MW	Mitsubishi
	Blok 2			
	GT 2.1	NG	112,45 MW	Mitsubishi
		HSD	100,98 MW	
	GT 2.2	NG	112,45 MW	Mitsubishi
		HSD	100,98 MW	
	GT 2.3	NG	112,45 MW	Mitsubishi
		HSD	100,98 MW	
	ST 2.0	-	188,92 MW	Mitsubishi
Blok 3				
GT 3.1	NG	112,45 MW	Mitsubishi	
GT 3.2	NG	112,45 MW	Mitsubishi	
GT 3.3	NG	112,45 MW	Mitsubishi	
ST 3.0	-	188,92 MW	Mitsubishi	
TOTAL KAPASITAS		2280 MW		

Sumber data operational PLTGU Gresik

Kapasitas terpasang PLTGU Unit Pembangkitan Gresik :

Turbin Gas : 112,45 MW (BBG) dan 100,98 MW (BBM)

Turbin Uap : 188,92 MW (BBG)

Kapasitas total per Blok, Konfigurasi 3.3.1 : 526,26 MW

Kapasitas total PLTGU (BBG) : 1578,78 MW

Kapasitas Minimum :

Kapasitas total per Blok, konfigurasi 3.3.1 : 125 MW

Kapasitas total PLTGU : 375 MW

Kapasitas Maksimum :

Turbin Gas : 115 MW (BBG)

Turbin Uap : 189 MW (BBG), Konfigurasi 3.3.1

Kapasitas total per Blok, konfigurasi 3.3.1 : 538, 2 MW

Kapasitas total PLTGU (BBG) : 1614,6 MW

3.3 Produksi Energi Listrik pada Pusat Listrik Tenaga Gas-Uap Unit Pembangkitan Gresik

Konsep dari sebuah PLTGU adalah pendayagunaan udara panas (500-550 °C) yang telah dihasilkan oleh sebuah atau beberapa turbin gas untuk digunakan sebagai sumber energi (kalor) guna memanaskan air di ketel agar dapat dihasilkan uap untuk mengoperasikan turbin uap. Pemanfaatan kembali uap panas yang sebelumnya dibuang ke atmosfer tersebut dapat meningkatkan efisiensi Pembangkitan Listrik Tenaga Termal.

Komponen-komponen utama yang terdapat di Unit Pembangkitan Gresik meliputi:

- Blok pengolahan bahan bakar

- Blok Turbin Gas
- Blok Turbin Uap
- Blok kendali operasi

Berikut ini akan dijelaskan secara umum tentang peralatan-peralatan utama yang terdapat di PLTGU Unit Pembangkitan Gresik.

a) Blok pengolahan bahan bakar PLTGU Unit Pembangkitan Gresik.

Berfungsi untuk memperbaiki mutu gas alam yang akan digunakan sebagai bahan bakar utama PLTGU, karena mutu bahan bakar tersebut mungkin menurun selama proses pengiriman.

b) Blok Turbin Gas PLTGU Unit Pembangkitan Gresik.

Selain berfungsi sebagai peralatan konversi energi listrik dari bahan bakar, juga merupakan sumber energi panas yang akan dimanfaatkan oleh sirkit HRSG.

c) Blok kendali operasi PLTGU Unit Pembangkitan Gresik

Berfungsi sebagai pusat sistem kontrol bagi seluruh Blok PLTGU. Pengendalian operasi PLTGU berlangsung secara otomatis yang dipantau melalui terminal komputer yang terpasang di Control Centre Room (CCR). Didalam CCR terdapat empat buah Block Control Desk (BCD) yang pada prinsipnya merupakan terminal input-output dari komputer tersebut untuk memantau operasi PLTGU. Tugas utama

komputer adalah untuk menyelenggarakan supervisi dan pengendalian operasi PLTGU. Untuk menyelenggarakan tugas supervisi dan pengendalian tersebut, komputer mengumpulkan data dan informasi dari sistem yang kemudian diolah menurut prosedur tertentu, dimana prosedur ini diatur oleh perangkat lunak (software) komputer. Informasi yang dikumpulkan komputer berasal dari peralatan-peralatan telemetri yang menghasilkan sinyal analog, melalui Direct Digital Controler (DDC) informasi tersebut diterima oleh komputer sebagai sinyal digital untuk kemudian diolah dan ditampilkan sebagai informasi real time yang perlu diketahui operator yang bertugas. Selain informasi real time, komputer juga menyimpan data-data masa lalu (history) yang direkam dalam suatu media penyimpanan informasi tertentu.

3.4 Pola Pembebanan Pada PLTGU Unit Pembangkitan Gresik

Pola pembebanan di PLTGU Unit Pembangkitan Gresik dibentuk dengan menggunakan asumsi bahwa blok yang beroperasi dengan konfigurasi sama. Secara fisik hal ini disebabkan oleh peralatan yang digunakan, pabrikan, standart material, kapasitas terpasang, bahkan tahun pembuatan dan tahun pengoperasian masing-masing Blok tidak jauh berbeda. Dengan asumsi diatas maka diadakan test untuk mendapatkan data aliran bahan bakar pada setiap tingkat beban masing-masing blok.

Berikut ini adalah tabel dari pola pembebanan pada konfigurasi 3.3.1 yang mana pada periode tertentu telah diterapkan di PLTGU Unit Pembangkitan Gresik. Tabel ini merupakan acuan bagi operator operasi yang bertugas pada Control Centre Room (CCR) untuk membagi beban per blok untuk memenuhi permintaan P3B.

Tabel 3-2

**Tabel Distribusi Beban Blok
Pada Konfigurasi 3.3.1 – 3.3.1 – 3.3.1**

Beban Total 375 MW – 725 MW

No	Beban Total	Beban Blok-1	Beban Blok-2	Beban Blok-3	BBG Blok-1	BBG Blok-2	BBG Blok-3	Total Aliran BBG
	(MW)				(KNM ³ /H)			
1	375	125	125	125	39.21	39.21	39.21	117.63
2	400	125	125	150	39.21	39.21	44.78	123.20
3	425	125	125	175	39.21	39.21	47.98	126.40
4	450	125	125	200	39.21	39.21	52.31	130.73
5	475	125	125	225	39.21	39.21	55.62	134.04
6	500	125	125	250	39.21	39.21	58.65	137.07
7	525	125	125	275	39.21	39.21	62.98	141.40
8	550	125	125	300	39.21	39.21	65.54	143.96
9	575	125	125	325	39.21	39.21	70.12	148.54
10	600	125	125	350	39.21	39.21	74.03	152.45
11	625	125	125	375	39.21	39.21	77.96	156.38
12	650	125	225	300	39.21	55.62	65.54	10.37
13	675	125	250	300	39.21	58.65	65.54	163.40
14	700	125	275	300	39.21	62.98	65.54	167.73
15	725	125	300	300	39.21	65.54	65.54	170.29

Sumber : Data operasi PLTGU Gresik

Tabel 3-3

Tabel Distribusi Beban Blok
 Pada Konfigurasi 3.3.1 – 3.3.1 – 3.3.1

Beban Total 750 MW – 1100 MW

No	Beban Total	Beban Blok-1	Beban Blok-2	Beban Blok-3	BBG Blok-1	BBG Blok-2	BBG Blok-3	Total Aliran BBG
1	750	125	300	325	39.21	65.54	70.12	174.87
2	775	125	300	350	39.21	65.54	74.03	178.78
3	800	125	300	375	39.21	65.54	77.96	182.71
4	825	225	300	300	55.62	65.54	65.54	186.70
5	850	250	300	300	58.65	65.54	65.54	189.73
6	875	275	300	300	62.98	65.54	65.54	194.06
7	900	300	300	300	65.54	65.54	65.54	196.62
8	925	300	300	325	65.54	65.54	70.12	201.20
9	950	300	300	350	65.54	65.54	74.03	205.11
10	975	300	300	375	65.54	65.54	77.96	209.04
11	1000	300	350	350	65.54	74.03	74.03	213.60
12	1025	300	350	375	65.54	74.03	77.96	217.53
13	1050	300	375	375	65.54	77.96	77.96	221.46
14	1075	350	350	375	74.03	74.03	77.96	226.02
15	1100	375	375	350	77.96	77.96	74.03	229.95

Sumber : Data operasi PLTGU Gresik

Tabel 3-4

**Tabel Distribusi Beban Blok
Pada Konfigurasi 3.3.1 – 3.3.1 – 3.3.1**

Beban Total 1125 MW – 1500 MW

No	Beban Total	Beban Blok-1	Beban Blok-2	Beban Blok-3	BBG Blok-1	BBG Blok-2	BBG Blok-3	Total Aliran BBG
	(MW)				(KNM ³ /H)			
1	1125	375	375	375	77.96	77.96	77.96	233.88
2	1150	375	375	400	77.96	77.96	83.04	238.96
3	1175	300	375	500	65.54	77.96	83.04	243.56
4	1200	350	375	475	74.03	77.96	95.81	247.80
5	1225	375	375	475	77.96	77.96	95.81	251.73
6	1250	375	375	500	77.96	77.96	100.06	255.98
7	1275	375	400	500	77.96	83.04	100.06	261.06
8	1300	350	475	475	74.03	95.81	95.81	265.65
9	1325	375	475	475	77.96	95.81	95.81	269.58
10	1350	375	475	500	77.96	95.81	100.06	273.83
11	1375	375	500	500	77.96	100.06	100.06	278.08
12	1400	400	500	500	83.04	100.06	100.06	283.16
13	1425	475	475	475	95.81	95.81	95.81	287.43
14	1450	475	475	500	95.81	95.81	100.06	291.68
15	1475	475	500	500	95.81	100.06	100.06	295.93
16	1500	500	500	500	100.06	100.06	100.06	300.18

Sumber : Data operasi PLTGU Gresik

BAB IV

ANALISA DATA

4.1. Pendahuluan

PT PLN PJB II Unit Pembangkitan Gresik.dalam penyelenggaraan usaha ketenagalistrikan berdasarkan prinsip industri dan perniagaan yang sehat, dituntut mampu bersaing dan mampu memanfaatkan sebesar-besarnya peluang pasar dalam bidang tenaga listrik. Dalam hal tersebut, PT PLN PJB II Unit Pembangkitan Gresik harus menjaga efisiensi dan keadalan operasional penyediaan tenaga listrik dari pembangkit-pembangkit yang dimilikinya.

Dengan demikian merupakan suatu keharusan bagi seluruh jajaran PT PLN PJB II Unit Pembangkitan Gresik agar selalu berupaya untuk meningkatkan kondisi penyediaan tenaga listrik dari pembangkit agar lebih ekonomis, bermutu dan didukung keandalan yang tinggi.

4.2 Data Pembangkitan

PLTGU Unit pembangkitan Gresik terdiri dari 3 blok, masing-masing blok terdiri atas 3 buah Turbin gas dengan 3 Generator Turbo, 3 HRSG dan 1 buah Turbin uap dengan 1 generator turbo.

Tabel 4.1

Total Aliran BBG per tingkat beban
Pada Blok 1

No	Tingkat Beban	Total Aliran BBG	
		KNM ³ /Hr	MBTU/Hr
1	150	46,55	1838,33
2	175	49,37	1939,69
3	200	53,11	2097,39
4	225	56,83	2244,30
5	250	59,67	2356,45
6	275	63,31	2460,71
7	300	65,25	2576,82
8	325	69,70	2752,55
9	350	73,77	2913,28
10	375	78,82	3112,72
11	400	83,59	3301,09
12	425	88,06	3477,62
13	450	92,69	3660,46
14	475	96,05	3793,15
15	500	99,98	3948,35

Sumber : Data test PLTGU Gresik

Catatan :

- 1 KNM³ = 37,3258 MSCF
- HHV Bahan Bakar Gas = 1058,02 BTU/SCF

Tabel 4.2

**Total Aliran BBG per tingkat beban
Pada Blok 2**

No	Tingkat Beban	Total Aliran BBG	
		KNM ³ /Hr	MBTU/Hr
1	150	43,56	1720,44
2	175	47,42	1872,92
3	200	51,28	2025,44
4	225	55,15	2177,91
5	250	59,01	2330,43
6	275	62,87	2482,91
7	300	66,73	2635,42
8	325	70,59	2787,94
9	350	73,92	2919,21
10	375	78,23	3089,53
11	400	82,73	3267,44
12	425	87,28	3447,13
13	450	91,83	3626,82
14	475	96,38	3806,46
15	500	100,93	3986,15

Sumber : Data test PLTGU Gresik

Catatan :

- 1 KNM³ = 37,3258 MSCF
- HHV Bahan Bakar Gas = 1058,02 BTU/SCF

Tabel 4.3

Total Aliran BBG per tingkat beban
Pada Blok 3

No	Tingkat Beban	Total Aliran BBG	
		KNM ³ /Hr	MBTU/Hr
1	150	41,44	1636,84
2	175	45,27	1787,78
3	200	49,09	1938,67
4	225	52,91	2089,61
5	250	56,73	2240,55
6	275	60,55	2391,48
7	300	64,37	2542,38
8	325	68,20	2693,32
9	350	72,31	2855,98
10	375	75,88	2996,73
11	400	80,22	3168,08
12	425	84,70	3344,96
13	450	89,18	3521,85
14	475	93,66	3698,77
15	500	98,13	3875,65

Sumber : Data test PLTGU Gresik

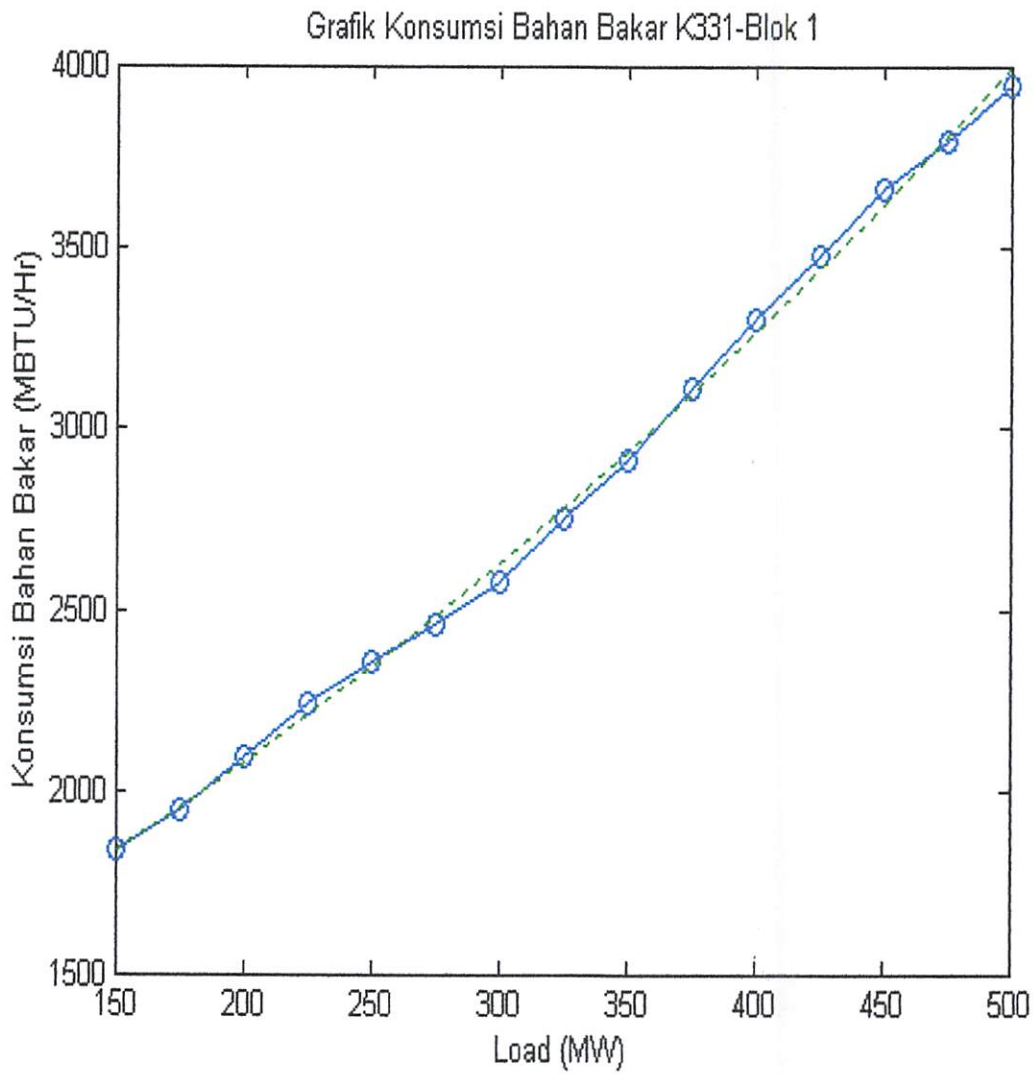
Catatan :

- 1 KNM³ = 37,3258 MSCF
- HHV Bahan Bakar Gas = 1058,02 BTU/SCF

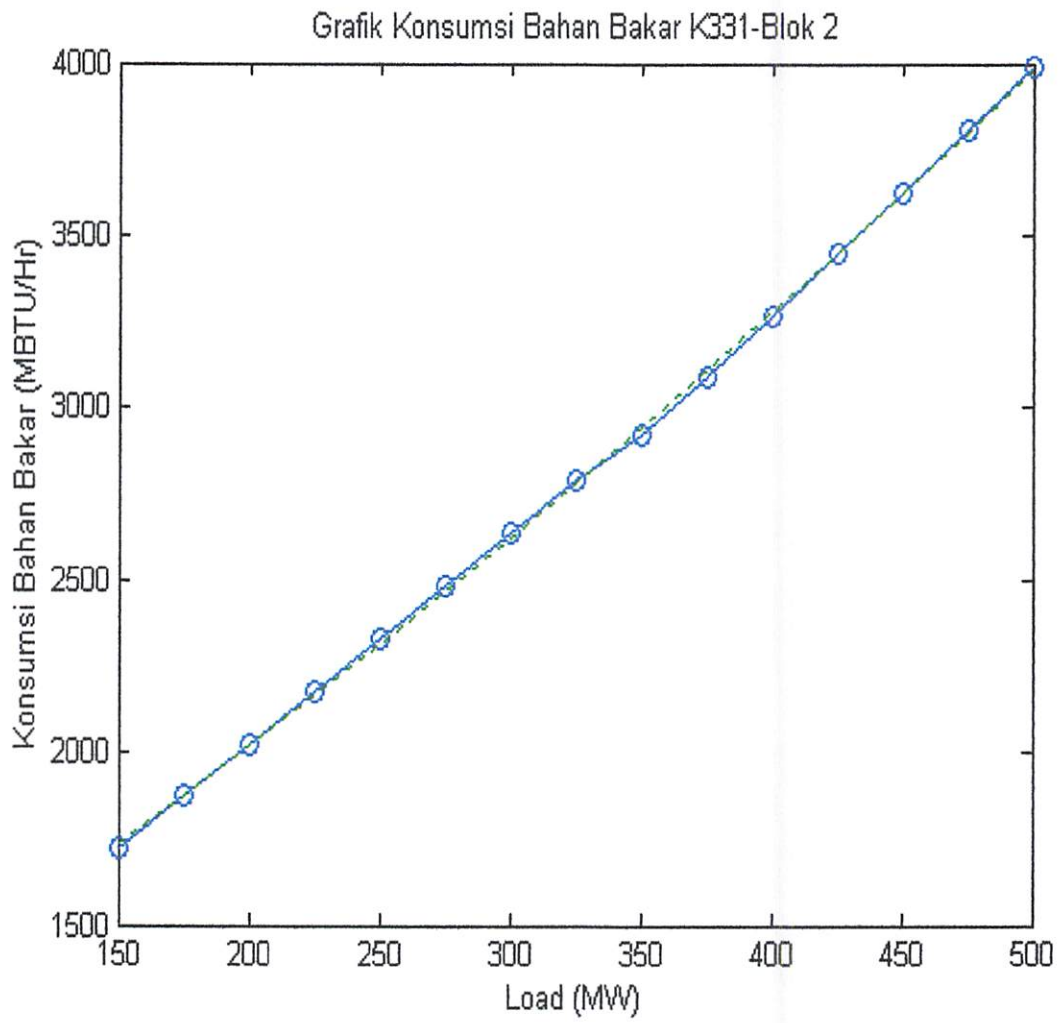
Data hasil perhitungan dilakukan sesuai dengan data dikumpulkan dan diolah. Misalnya nilai HHV bahan bakar gas rata-rata yang digunakan yaitu sebesar 1058,02 (BTU/SCF), adalah sesuai dengan hasil pengukuran nilai HHV rata-rata bahan bakar gas yang dilakukan oleh operator lapangan pada saat data dikumpulkan.

Selanjutnya data aliran bahan bakar gas masing-masing blok yang telah ditabulasikan diatas, diplot pada diagram pancar.

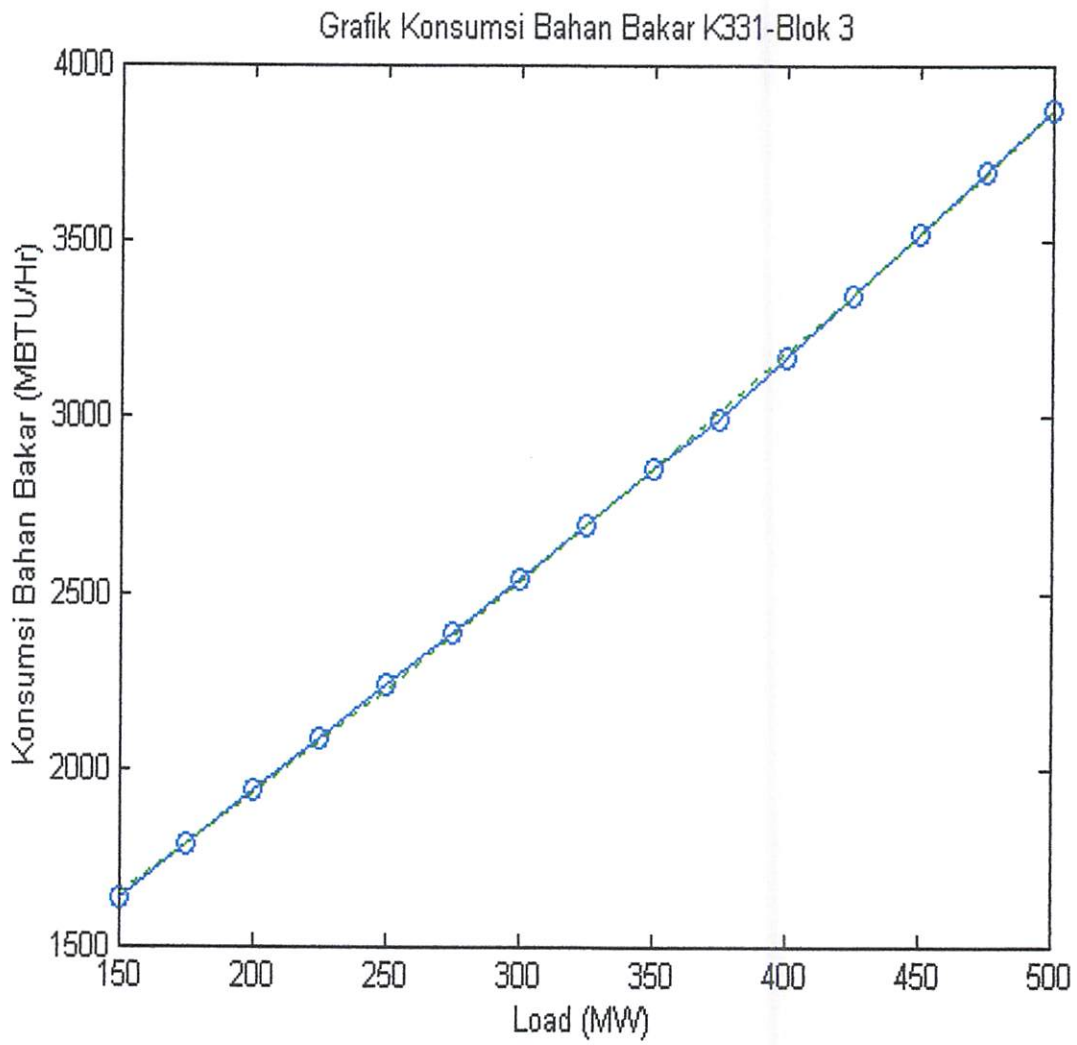
Grafik 4.1 :
Total aliran BBG per tingkat beban
Pada Blok 1



Grafik 4.2 :
Total aliran BBG per tingkat beban
Pada Blok 1



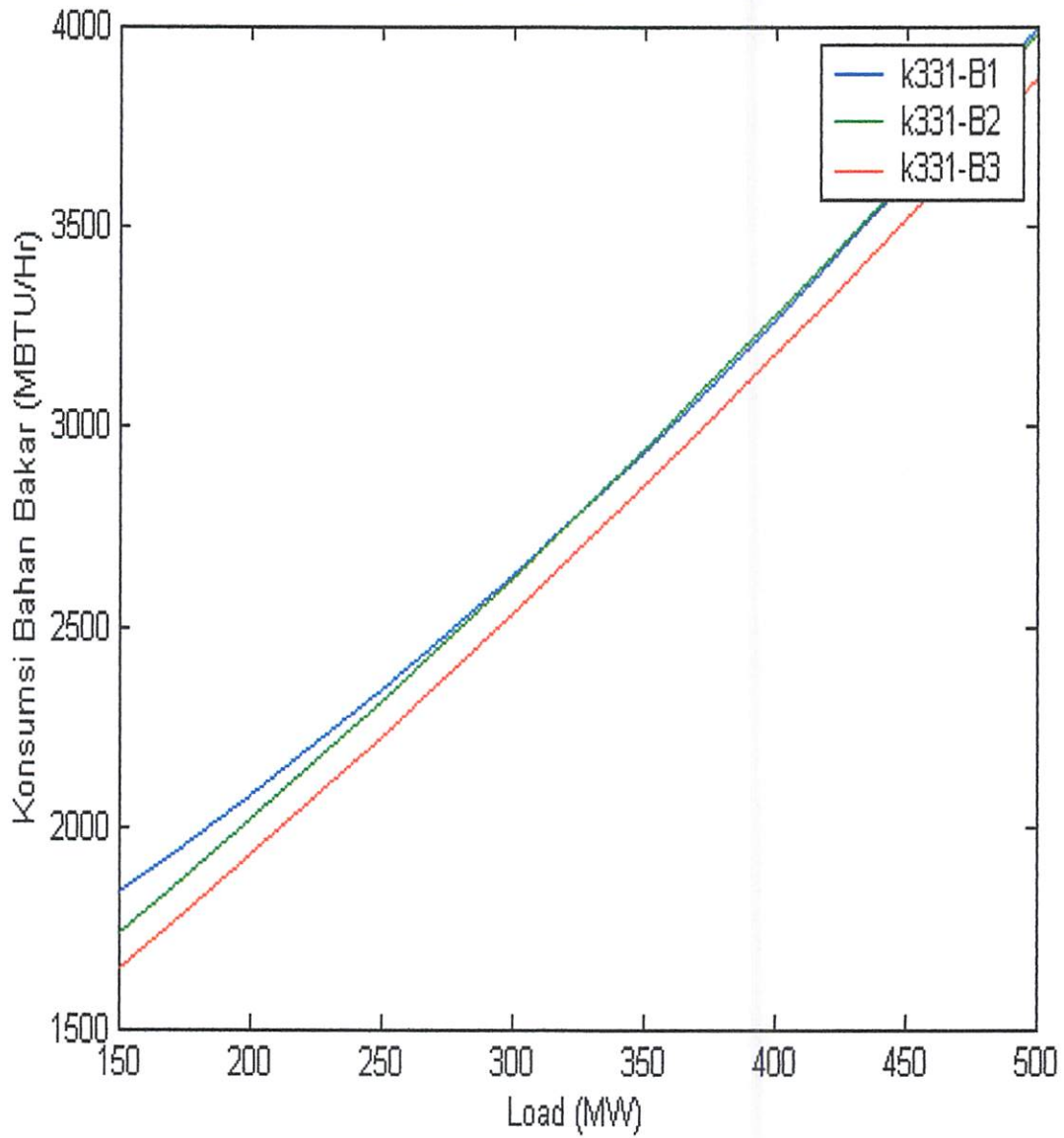
Grafik 4.3 :
Total aliran BBG per tingkat beban
Pada Blok 3



Grafik 4.4 :

Total aliran BBG per tingkat beban
Pada Kombinasi 3.3.1

Grafik Konsumsi Bahan Bakar



4.3 Perhitungan Persamaan Karakteristik Masukan-Keluaran(Bahan Bakar) masing-masing Blok Unit Pembangkitan PLTGU Gresik.

Melihat kondisi seperti yang telah diuraikan pada bab sebelumnya, maka langkah pertama yang perlu dilakukan adalah menentukan persamaan karakteristik masukan-keluaran ($H = f(P)$) dalam (MBTU/H) dan $F = f(P)$ dalam (\$ US/H).

Dari kurva aliran BBG sebagai fungsi beban (grafik 4.1, grafik 4.2, grafik 4.3) terlihat bahwa karakteristik masukan-keluaran sebuah Blok Pembangkit Listrik Termal sudah cukup teliti. Sebagai catatan, bahwa harga persatuan energi bahan bakar (natural gas) yang dikonsumsi PLTGU Gresik sebesar 2,53 (\$ US/MBTU).

Hasil dari perhitungan sebagaimana terlampir adalah sebagai berikut :

- **Persamaan Karakteristik Masukan-Keluaran (Bahan Bakar) PLTGU Gresik dengan konfigurasi 3.3.1 pada Blok 1.**

$$F_1(P_1) = 0,0112590 P_1^2 + 8,2427 P_1 + 3,166 \cdot 10^3 \left[\frac{\$US}{Hr} \right],$$

Maka persamaan laju tambahan biaya bahan bakar diperoleh dengan menurunkan persamaan biaya bahan bakar terhadap beban

$$\left(\frac{dF_1(P_1)}{dP_1} \right) \text{ sehingga menghasilkan :}$$

$$\left(\frac{dF_1(P_1)}{dP_1} \right) = 0,022518 P_1 + 8,2427 \left[\frac{\$US}{MWH} \right]$$

- **Persamaan Karakteristik Masukan-Keluaran (Bahan bakar) PLTGU Gresik dengan konfigurasi 3.3.1 pada Blok 2.**

$$F_2(P_2) = 0,0062063 P_2^2 + 12,1771 P_2 + 2,4282 \cdot 10^3 \left[\frac{\$US}{Hr} \right],$$

Maka persamaan laju tambahan biaya bahan bakar diperoleh dengan menurunkan persamaan biaya bahan bakar terhadap beban

$\left(\frac{dF_1(P_1)}{dP_1} \right)$ sehingga menghasilkan :

$$\left(\frac{dF_1(P_1)}{dP_1} \right) = 0,0214126 P_2 + 12,1771 \left[\frac{\$US}{MWH} \right]$$

- **Persamaan Karakteristik Masukan-Keluaran (Bahan bakar) PLTGU Gresik dengan konfigurasi 3.3.1 pada Blok 3.**

$$F_3(P_3) = 0,0053446 P_3^2 + 12,5668 P_3 + 2,1676 \cdot 10^3 \left[\frac{\$US}{Hr} \right],$$

Maka persamaan laju tambahan biaya bahan bakar diperoleh dengan menurunkan persamaan biaya bahan bakar terhadap beban

$\left(\frac{dF_1(P_1)}{dP_1}\right)$ sehingga menghasilkan :

$$\left(\frac{dF_1(P_1)}{dP_1}\right) = 0,0106892 P_1 + 12,5668 \left[\frac{\$US}{MWH} \right]$$

Contoh 1

Untuk beban 150 MW pada blok 1 maka biaya bahan bakarnya akan dapat diketahui dengan cara:

$$F_1(P_1) = 0,0112590 P_1^2 + 8,2427 P_1 + 3,166 \cdot 10^3 \left[\frac{\$US}{Hr} \right],$$

$$F_1(150) = 0,0112590 150^2 + 8,2427 150 + 3,166 \cdot 10^3 \left[\frac{\$US}{Hr} \right],$$

$$= 4.655,73 (\$ US/Hr)$$

Begitu juga sama cara perhitungannya pada blok 2, dan 3, disesuaikan dengan persamaan masing-masing blok.

Untuk mencari laju pertambahan bahan bakarnya maka:

$$\left(\frac{dF_1(P_1)}{dP_1}\right) = 0,022518 P_1 + 8,2427 \left[\frac{\$US}{MWH} \right]$$

$$\left(\frac{dF_1(P_1)}{dP_1}\right) = 0,022518 150 + 8,2427 \left[\frac{\$US}{MWH} \right]$$

$$\left(\frac{dF_1(P_1)}{dP_1}\right) = 11,26 (\$ US/Hr)$$

Untuk blok 2, dan 3 dicari dengan cara yang sama sesuai dengan persamaannya.

Dari persamaan biaya bahan bakar dan laju tambahan biaya bahan bakar didapatkan nilai masing-masing terhadap tingkat beban, kemudian ditabulasikan dan diplot dalam diagram pancarnya.

Tabel 4.4

**Tabel Fungsi Biaya Bahan Bakar Masing-masing Blok
Terhadap Tingkat Beban**

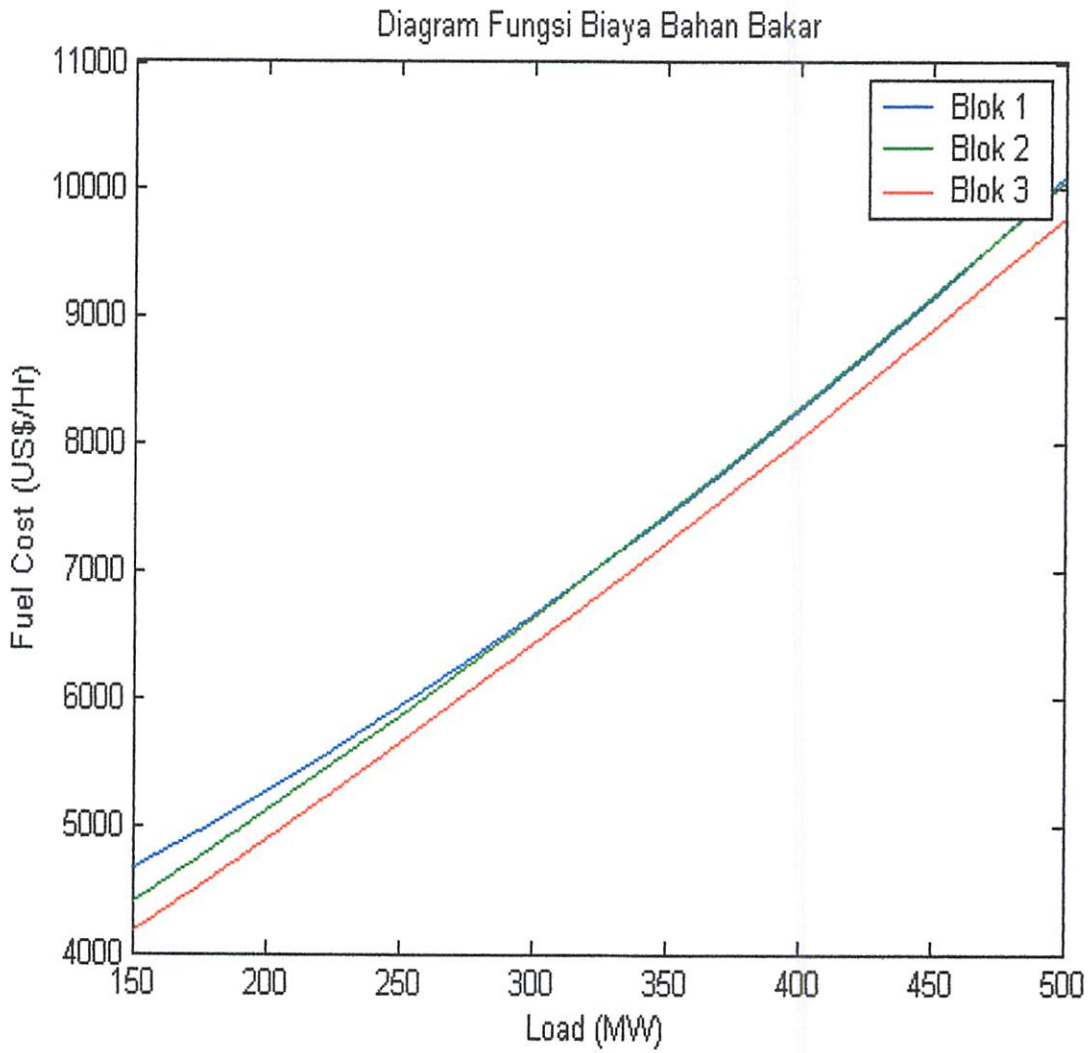
No	Beban Blok (MW)	Biaya Bahan Bakar (\$ US/Hr)		
		Blok 1	Blok 2	Blok 3
1	150	4.655,73	4.394,41	4.172,86
2	175	4.953,28	4.749,26	4.530,45
3	200	5.264,90	5.111,88	4.894,73
4	225	5.590,59	5.482,25	5.265,68
5	250	5.930,36	5.860,37	5.643,32
6	275	6.284,19	6.246,26	6.207,64
7	300	6.652,11	6.639,90	6.418,64
8	325	7.034,09	7.041,30	6.816,32
9	350	7.430,15	7.450,46	7.220,68
10	375	7.840,29	7.867,37	7.631,72
11	400	8.264,49	8.292,05	8.049,44
12	425	8.702,77	8.724,48	8.473,84
13	450	9.155,13	9.164,67	8.904,92
14	475	9.621,55	9.612,62	9.342,69
15	500	10.102,06	10.068,32	9.787,13

Tabel 4.5

**Tabel Fungsi Laju Tambahan Bahan Bakar Masing-masing Blok
Terhadap Tingkat Beban**

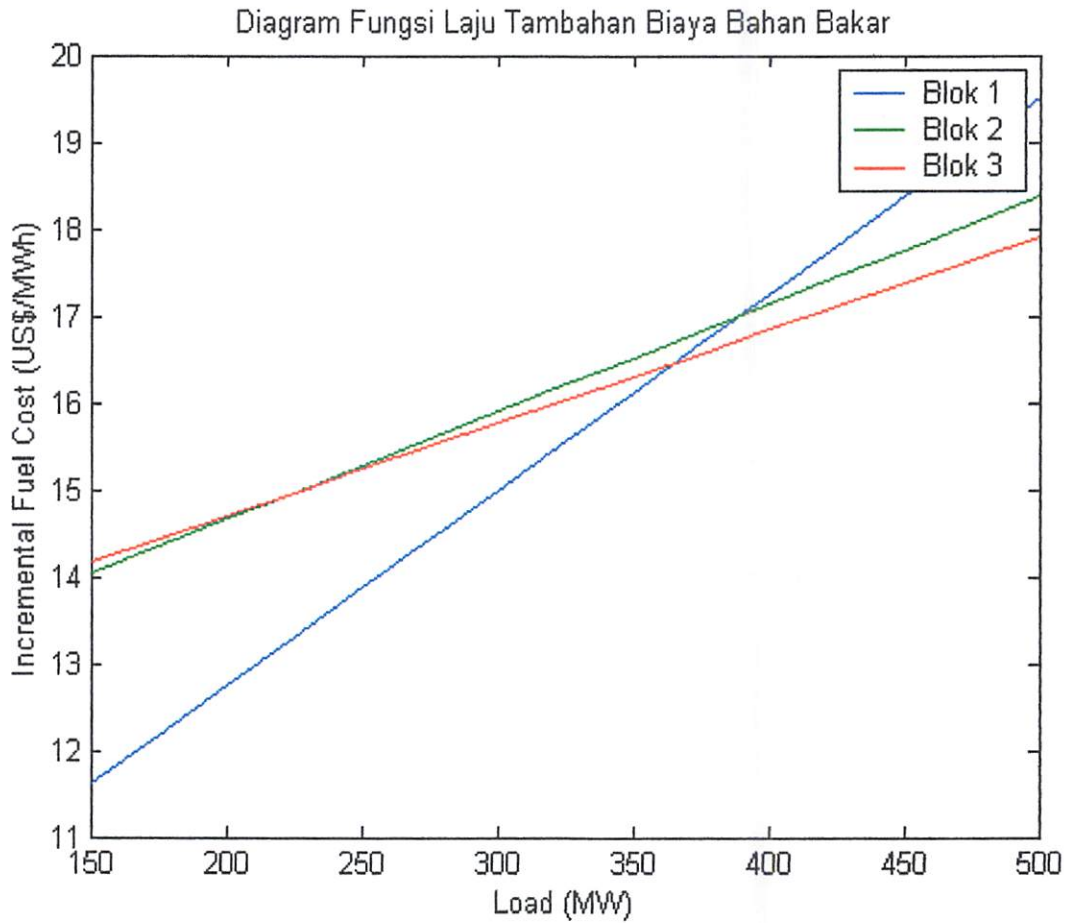
No	Beban Blok (MW)	Laju Tambahan B.Bakar (\$ US/Hr)		
		Blok 1	Blok 2	Blok 3
1	150	11,26	14,03	14,17
2	175	12,18	14,34	14,43
3	200	12,74	14,66	14,70
4	225	13,30	14,97	14,97
5	250	13,87	15,28	15,23
6	275	14,43	15,59	15,50
7	300	14,99	15,90	15,77
8	325	15,56	16,21	16,04
9	350	16,12	16,52	16,30
10	375	16,68	16,83	16,57
11	400	17,25	17,14	16,84
12	425	18,37	17,45	17,11
13	450	18,93	17,76	17,37
14	475	18,93	18,03	17,64
15	500	19,50	18,38	17,91

Grafik 4.5 :
Diagram Fungsi Biaya Bahan Bakar
Pada PLTGU Unit Pembangkitan Gresik
Konfigurasi 3.3.1



Grafik 4.6

Diagram Fungsi Laju Tambahan Biaya Bahan Bakar
 PLTGU Unit Pembangkitan Gresik Konfigurasi 3.3.1



Dari grafik dapat diamati bahwa turunan pertama dari fungsi biaya bahan bakar adalah fungsi laju tambahan bahan bakar, untuk blok 1 memiliki kemiringan yang jauh lebih curam dibandingkan dengan blok 2, dan blok 3. Hal ini menunjukkan bahwa untuk kenaikan setiap satu satuan tingkat beban yang sama masing-masing blok, akan menyebabkan biaya untuk memproduksi

Tabel 4.6

Tabel Fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing

Blok Pada PLTGU Gresik

375(MW) - 725(MW)

No	Beban (MW)	Tingkat Beban masing-masing Blok (MW)			Biaya Bahan Bakar (\$ US/Hr)			
		Blok 1	Blok 2	Blok 3	Blok 1	Blok 2	Blok 3	Total (Σ)
1	375	125	125	125	4372	4047	3822	12242
2	400	125	125	150	4372	4047	4173	12592
3	425	125	125	175	4,372	4047	4530	12950
4	450	125	125	200	4372	4047	4895	13314
5	475	125	125	225	4372	4047	5265	13689
6	500	125	125	250	4372	4047	5643	14063
7	525	125	125	275	4372	4047	6027	14447
8	550	125	125	300	4372	4047	6418	14838
9	575	125	125	325	4372	4047	6816	15235
10	600	125	125	350	4372	4047	7220	15,640
11	625	125	225	375	4372	5482	7631	17486
12	650	125	225	300	4372	5482	6418	16273
13	675	125	250	300	4372	5861	6418	16651
14	700	125	275	300	4372	6247	6418	17037
15	725	125	300	300	4372	6640	6418	17431

Tabel 4.7

Tabel Fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing Blok Pada PLTGU Gresik

750 (MW) – 1125(MW)

No	Beban (MW)	Tingkat Beban masing-masing Blok (MW)			Biaya Bahan Bakar (\$ US/Hr)			
		Blok 1	Blok 2	Blok 3	Blok 1	Blok 2	Blok 3	Total (Σ)
16	750	125	300	325	4372	6640	6816	17828
17	775	125	300	350	4372	6640	7220	18233
18	800	125	300	375	4372	6640	7631	18644
19	825	225	300	300	5591	6640	6418	18649
20	850	250	300	300	5930	6640	6418	18989
21	875	275	300	300	6284	6640	6418	19343
22	900	300	300	300	6652	6640	6418	19711
23	925	300	300	325	6652	6640	6816	20108
24	950	300	300	350	6652	6640	7220	20513
25	975	300	300	375	6652	6640	7631	20924
26	1000	300	350	350	6652	7451	7220	21323
27	1025	300	350	375	6652	7451	7631	21734
28	1050	300	375	375	6652	7868	7631	22151
29	1075	350	350	375	7430	7451	7631	22512
30	1100	375	375	350	7840	7868	7220	22928
31	1125	375	375	375	7840	7868	7631	23339

Tabel 4.8
Tabel Fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing
Blok Pada PLTGU Gresik

1150(MW) – 1500(MW)

No	Beban (MW)	Tingkat Beban masing- masing Blok (MW)			Biaya Bahan Bakar (\$ US/Hr)			
		Blok 1	Blok 2	Blok 3	Blok 1	Blok 2	Blok 3	Total (Σ)
32	1150	375	375	400	7840	7868	8049	23757
33	1175	300	375	500	6652	7868	9786	24306
34	1200	350	375	475	7430	7868	9342	24640
35	1225	375	375	475	7840	7868	9342	25050
36	1250	375	375	500	7840	7868	9786	25494
37	1275	375	400	500	7840	8293	9786	25919
38	1300	350	475	475	7430	9613	9342	26385
39	1325	375	475	475	7840	9613	9342	26796
40	1350	375	475	500	7840	9613	9786	27240
41	1375	375	500	500	7840	10069	9786	27696
42	1400	400	500	500	8265	10069	9786	28120
43	1425	475	475	475	9622	9613	9342	28577
44	1450	475	475	500	9622	9613	9786	29021
45	1475	475	500	500	9622	10069	9786	29477
46	1500	500	500	500	10102	10069	9786	29958

4.4 Perhitungan Distribusi Beban Blok.

Sebelum Metode Evolutionary Programming dapat diterapkan untuk menghitung distribusi beban pada masing-masing blok, maka langkah pertama yang harus dilakukan adalah menentukan fungsi karakteristik masukan-keluaran ($H = f(P)$ (MBTU/H) dan ($F = f(P)$ (\$US/H)) untuk masing-masing blok, dan hal tersebut telah dilakukan pada sub bab (4.3).

Hal lain yang harus diperhatikan adalah nilai batasan yang telah ditentukan. Nilai ini terkait dengan batas maksimum dan minimum kapasitas maksimum dan minimum masing-masing blok. Batasan tersebut adalah :

- a) Untuk $P_i \geq P_i(\max)$, maka $P_i = P_i(\max)$
- b) Untuk $P_i \leq P_i(\min)$, maka $P_i = P_i(\min)$
- c) Untuk $P_i(\min) \leq P_i \leq P_i(\max)$, maka $P_i = P_i$ (hasil optimasi).

4.4.1. Penggunaan Program Komputer Dengan Metode Evolutionary

Programming

Untuk menentukan Optimasi Economic Dispatch pada PLTGU Gresik maka digunakan bantuan komputer. Dengan menggunakan program komputer ini, maka waktu komputasi yang dibutuhkan pada saat menganalisis data akan jauh lebih cepat karena data yang dianalisis banyak dan dibutuhkan ketelitian yang tinggi.

Program komputer ini menggunakan bahasa pemrograman Borland Delphi versi 7.0, merupakan bahasa pemrograman terstruktur yang dapat digunakan untuk berbagai macam aplikasi.

4.4.2. Tampilan program

1. Tampilan utama dari program

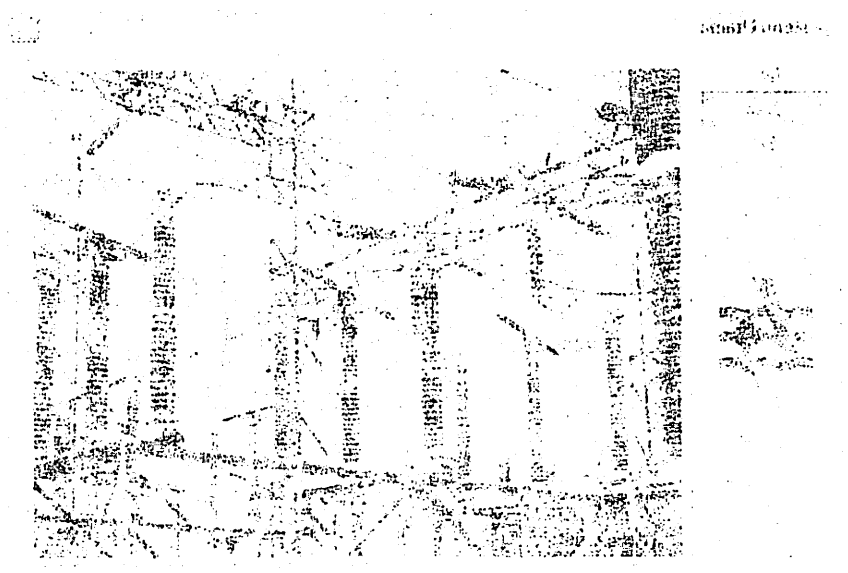


Gambar 4.1. Tampilan Utama Program

Program komputer ini menggunakan bahasa pemrograman Pascal
Dipik: versi 7.0, merupakan bahasa pemrograman terapan yang dapat
digunakan untuk berbagai macam aplikasi.

4.4.3. Contoh program

1. Contoh hasil output program



Gambar 4.1. Contoh hasil output program

2. Kemudian setelah itu tekan tombol buka data untuk membuka file yang tersimpan.

The screenshot shows a software window titled "Tampilan Data" with a tabbed interface. The "General" tab is selected. It contains two input fields: "Jumlah Pembangkit" with the value "3" and "Jumlah Jam" with the value "46". At the bottom right, there are two buttons labeled "Next" and "Close".

Gambar 4.2. Input Data General

3. Setelah itu masukkan data CCPP Unit Pembangkitan Gresik yang beroperasi pada pola konfigurasi 3.3.1 – 3.3.1 – 3.3.1.

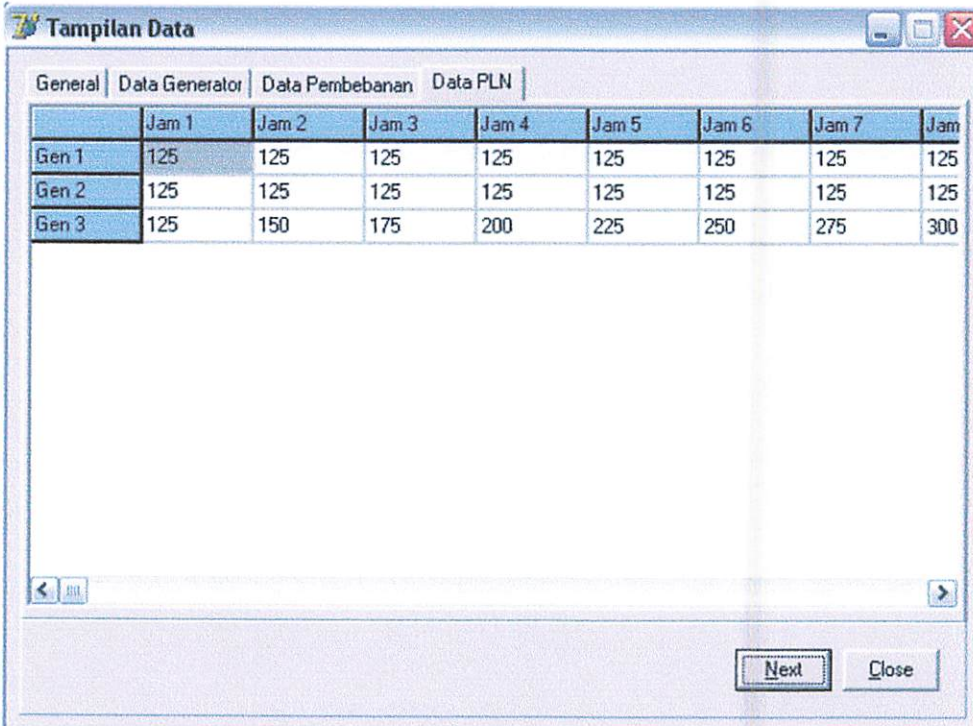
The screenshot shows the same "Tampilan Data" window, but now the "Data Generator" tab is selected. It displays a table with the following data:

Gen	Nama	Pmax	Pmin	a0	a1	a2	Tup
1	B1	500	125	3166	8.2427	0.01126	0
2	B2	500	125	2429.2	12.1771	0.00621	0
3	B3	500	125	2167.6	12.5668	0.00534	0

At the bottom right, there are "Next" and "Close" buttons.

Gambar 4.3. Input Data Generator

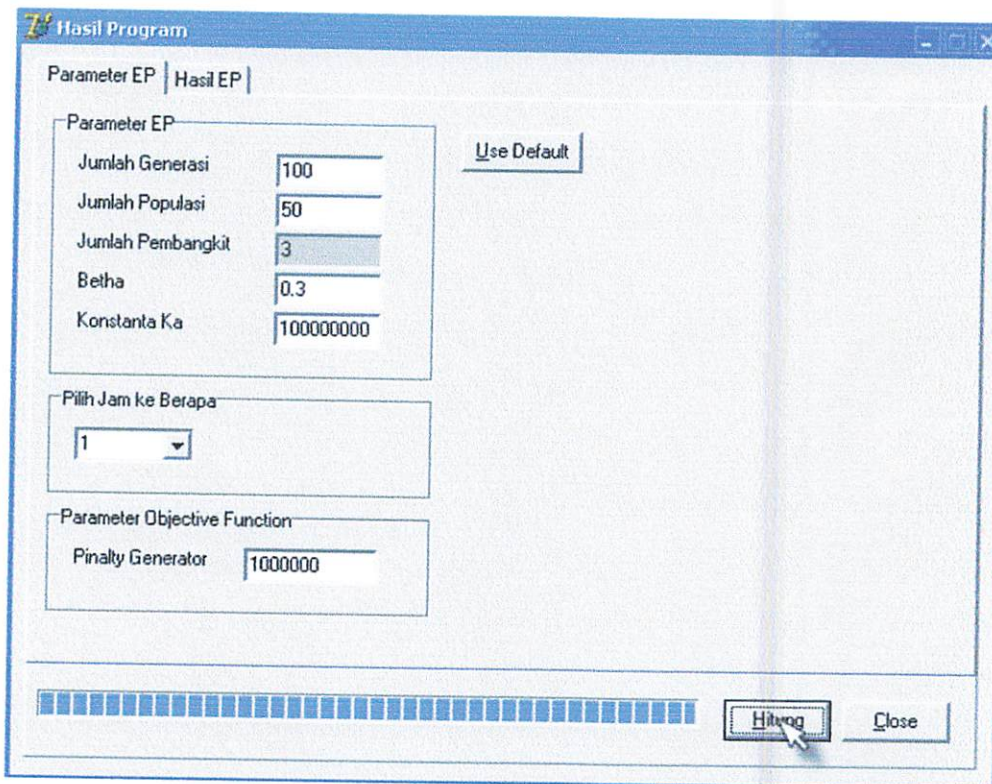
4. Masukkan data tingkatan pembebanan tiap-tiap Blok CCPP Unit Pembangkitan Gresik yang beroperasi pada pola konfigurasi 3.3.1 – 3.3.1 – 3.3.1. mulai kapasitas minimum tiap-tiap blok (125 MW) sampai kapasitas maksimum tiap-tiap blok (500 MW)



	Jam 1	Jam 2	Jam 3	Jam 4	Jam 5	Jam 6	Jam 7	Jam
Gen 1	125	125	125	125	125	125	125	125
Gen 2	125	125	125	125	125	125	125	125
Gen 3	125	150	175	200	225	250	275	300

Gambar 4.4. Input Data Tingkatan Pembebanan Pola Gresik

5. Setelah menetapkan parameter tersebut maka komputasi dapat dilakukan, untuk menentukan kebutuhan beban yang disuplai dengan biaya yang minimum perjam.



The image shows a software window titled "Hasil Program" with two tabs: "Parameter EP" and "Hasil EP". The "Parameter EP" tab is active. It contains several input fields and a "Use Default" button. The parameters are:

Parameter	Value
Jumlah Generasi	100
Jumlah Populasi	50
Jumlah Pembangkit	3
Betha	0.3
Konstanta Ka	100000000

Below these fields is a section titled "Pilih Jam ke Berapa" with a dropdown menu set to "1".

At the bottom of the window, there is a "Parameter Objective Function" section with a "Pinalty Generator" field set to "1000000".

At the bottom right of the window, there are two buttons: "Hitung" and "Close". A mouse cursor is pointing at the "Hitung" button.

Gambar 4.5. Parameter *Evolutionary Programming*

6. Tekan tombol Hitung untuk menampilkan hasil komputasi, untuk menentukan hasil optimasi pembebanan ekonomis dengan *Evolutionary Programming*, yaitu *Cost per Jam* untuk melihat hasil perbandingan biaya pola Evolutionary Programming dan pola gresik pada tingkatan beban tertentu serta melihat selisih biayanya dengan kebutuhan beban yang disuplai dengan biaya yang minimum.

The screenshot shows a software window titled "Hasil Program" with a tab labeled "Hasil EP". It contains a table with the following data:

No	PEP (MW)	P PLN (MW)	Cost EP (\$)	Cost PLN (\$)	Selisih (\$)
1	150	125	4,656	4,372	-283
2	125	125	4,047	4,047	0
3	125	150	3,822	4,173	351

Below the table, there are input fields for summary statistics: Biaya EP (12,525), Biaya PLN (12,592), and Selisih (67). At the bottom right, there are buttons for "Hitung" and "Close".

Gambar 4.6.

Hasil Optimasi Pembebanan Ekonomis dengan
Evolutionary Programming

Tabel 4.9
Tabel Fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing
Blok Dengan Metode Evolutionary Programming

375(MW) - 725(MW)

No	Beban (MW)	Tingkat Beban masing- masing Blok (MW)			Biaya Bahan Bakar (\$ US/Hr)			
		Blok 1	Blok 2	Blok 3	Blok 1	Blok 2	Blok 3	Total (Σ)
1	375	125	125	125	4372	4047	3822	12242
2	400	150	125	125	4656	4047	3822	12525
3	425	175	125	125	4953	4047	3822	12823
4	450	200	125	125	5265	4047	3822	13134
5	475	225	125	125	5591	4047	3822	13460
6	500	246	129	125	5874	4104	3822	13799
7	525	253	142	129	5976	4293	3878	14146
8	550	258	152	140	6047	4417	4033	14496
9	575	263	161	151	6114	4550	4185	14849
10	600	269	170	161	6191	4680	4334	15205
11	625	274	180	172	6264	4815	4485	15564
12	650	279	188	183	6339	4941	4645	15925
13	675	284	198	194	6411	5079	4800	16290
14	700	289	207	204	6486	5216	4955	16657
15	725	294	216	215	6565	5350	5113	17027

Tabel 4.10

Tabel Fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing Blok Dengan Metode Evolutionary Programming

750(MW) – 1125(MW)

No	Beban (MW)	Tingkat Beban masing-masing Blok (MW)			Biaya Bahan Bakar (\$ US/Hr)			
		Blok 1	Blok 2	Blok 3	Blok 1	Blok 2	Blok 3	Total (Σ)
16	750	299	226	225	6637	5492	5272	17400
17	775	304	235	236	6713	5628	5435	17776
18	800	309	244	247	6785	5769	5601	18155
19	825	314	253	258	6867	5911	5758	18536
20	850	319	262	269	6948	6045	5927	18921
21	875	324	272	279	7022	6196	6090	19308
22	900	329	281	290	7103	6333	6262	19698
23	925	334	290	300	7182	6485	6425	20091
24	950	340	299	311	7264	6628	6595	20487
25	975	345	308	322	7344	6769	6773	20886
26	1000	350	318	333	7424	6922	6941	21287
27	1025	356	326	343	7520	7059	7113	21692
28	1050	360	336	354	7592	7215	7291	22099
29	1075	365	345	365	7672	7372	7465	22509
30	1100	370	354	376	7757	7519	7645	22922
31	1125	375	363	387	7840	7675	7823	23338

Tabel 4.11

Tabel Fungsi Biaya Bahan Bakar dan Distribusi Beban Masing-masing Blok Dengan Metode Evolutionary Programming

1150(MW) – 1500(MW)

No	Beban (MW)	Tingkat Beban masing-masing Blok (MW)			Biaya Bahan Bakar (\$ US/Hr)			
		Blok 1	Blok 2	Blok 3	Blok 1	Blok 2	Blok 3	Total (Σ)
32	1150	380	373	397	7928	7832	7997	23757
33	1175	385	382	408	8012	7986	8180	24178
34	1200	390	391	418	8102	8140	8361	24603
35	1225	395	400	429	8186	8298	8546	25030
36	1250	401	410	440	8277	8458	8726	25460
37	1275	406	419	451	8362	8614	8918	25893
38	1300	411	428	462	8451	8773	9105	26329
39	1325	416	437	472	8542	8937	9288	26768
40	1350	421	446	483	8627	9100	9483	27209
41	1375	426	456	493	8722	9266	9666	27654
42	1400	433	467	500	8837	9478	9786	28101
43	1425	441	484	500	8998	9770	9786	28554
44	1450	450	500	500	9161	10069	9786	29011
45	1475	474	500	500	9622	10069	9786	29477
46	1500	500	500	500	10102	10069	9786	29958

Tabel 4.12

**Tabel Hasil Penghematan Biaya Bahan Bakar Dengan Metode
Evolutionary Programming dibanding PLTGU Gresik**

Beban 375 (MW) – 725 (MW)

No	Total Beban	Biaya Bahan Bakar		Penghematan Biaya Bahan Bakar
	[MW]	Evolutionary Programming	PLTGU Gresik	[SUS/Hr]
		[SUS/Hr]		
1	375	12242	12242	0
2	400	12525	12592	67
3	425	12823	12950	127
4	450	13134	13314	180
5	475	13460	13685	225
6	500	13799	14063	263
7	525	14146	14447	361
8	550	14496	14838	341
9	575	14849	15235	386
10	600	15205	15640	435
11	625	15564	16051	487
12	650	15925	16273	348
13	675	16290	16651	361
14	700	16657	17037	380
15	725	17027	17431	404

Tabel 4.13

**Tabel Hasil Penghematan Biaya Bahan Bakar Dengan Metode
Evolutionary Programming dibanding PLTGU Gresik**

Beban 750 (MW) – 1125 (MW)

No	Total Beban	Biaya Bahan Bakar		Penghematan Biaya Bahan Bakar
	[MW]	Evolutionary Programming	PLTGU Gresik	[\$US/Hr]
		[\$US/Hr]		
16	750	17400	17828	428
17	775	17776	18233	457
18	800	18155	18644	489
19	825	18536	18649	113
20	850	18921	18987	68
21	875	19308	19343	35
22	900	19698	19711	13
23	925	20091	20108	17
24	950	20487	20513	26
25	975	20886	20924	38
26	1000	21287	21323	36
27	1025	21692	21734	43
28	1050	22099	22151	52
29	1075	22509	22512	3
30	1100	22922	22928	6
31	1125	23398	23339	1

Tabel 4.14.

**Tabel Hasil Penghematan Biaya Bahan Bakar Dengan Metode
Evolutionary Programming dibanding PLTGU Gresik**

Beban 1150 (MW) – 1500 (MW)

No	Total Beban	Biaya Bahan Bakar		Penghematan Biaya Bahan Bakar
	[MW]	Evolutionary Programming	PLTGU Gresik	[\$US/Hr]
		[\$US/Hr]		
32	1150	23757	23757	0
33	1175	24178	24306	128
34	1200	24603	24640	37
35	1225	25030	25050	20
36	1250	25460	25494	34
37	1275	25843	25919	26
38	1300	26329	26385	56
39	1325	26768	26796	28
40	1350	27210	27240	30
41	1375	27654	27696	42
42	1400	28102	28120	18
43	1425	28554	28577	23
44	1450	29011	29021	11
45	1475	29477	29477	0
46	1500	29958	29958	0

Catatan :

Dari table-tabel diatas tampak bahwa pola pembebanan yang selama ini diterapkan oleh Unit Pembangkitan Gresik masih memiliki selisih biaya bahan bakar yang dapat dihemat bila menggunakan dengan Metode Evolutionary Programming.

Apabila mengacu pada hasil penghematan biaya bahan bakar pada table (4.6), (4.7) dan (4.8) jumlahnya sangat bervariasi, seperti contoh jika kita melihat bahwa biaya bahan bakar yang dapat dihemat dapat mencapai 457 [\$US/Hr] pada tingkat beban 775 [MW] dan 489 [\$US/Hr] pada tingkat beban 800 [MW].

BAB V

KESIMPULAN

5.1 Kesimpulan

1. Pola Pembebanan dengan metode Evolutionary Programming dalam skripsi ini menggunakan asumsi karakteristik masukan keluaran masing-masing Blok tidak sama, hal ini berbeda dengan asumsi yang digunakan PLTGU Gresik, dan dapat diambil kesimpulan dengan metode Evolutionary Programming lebih ekonomis.
2. Dengan membandingkan hasil yang diperoleh maka pola pembebanan yang diterapkan di PLTGU Gresik dengan Metode Evolutionary Programming berbeda. Sebagai contoh pada permintaan beban 800 (MW), biaya bahan bakar apabila menggunakan pola pembebanan Gresik adalah 18644 [\$US/Hr]. Metode Evolutionary Programming memerlukan biaya bahan bakar 18155 [\$US/Hr], sehingga biaya yang bisa dihemat 489 [\$US/Hr].

5.2 Saran

Dalam skripsi ini hanya dilakukan pola pembagian beban pada kombinasi 3.3.1, yaitu pola konfigurasi pada kapasitas maksimum. Hal ini dilakukan akibat data test yang tersedia terbatas pada kombinasi 3.3.1 saja. Perlu dipertimbangkan untuk membandingkan pemakaian bahan bakar pada tingkat beban yang sama dengan kombinasi yang berbeda. Untuk itu perlu dilakukan pengetesan terhadap pola kombinasi 2.2.1 dan 1.1.1, agar diketahui karakteristik masukan keluaran masing-masing Blok dengan kombinasi tersebut.

DAFTAR PUSTAKA

- [1]. A.J Wood dan B. F. Wollenberg, “ *Power Generation Operation, and Control* “
Second Edition, John Wiley & Sons. 1996
- [2]. R. Ghandass. Dr P Venkatesh, Prof T G Palanivelu, Prof K Manivannan, ”
*Evolutionary Programming Solution of Economic Load Dispatch with Combine
Cycle Power Plant* “ Written discussion on this paper will be received until
November 30,2000.
- [3]. Djiteng Marsudi . Ir., “ *Operasi Sistem Tenaga Listrik* “, ISTN 1990
- [4]. William D. Stevenson, Jr. “ *Analisa Sistem Tenaga Listrik* “ , Edisi Keempat,
Erlangga, Jakarta, 1993

LAMPIRAN





INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ENERGI LISTRIK

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : AGUS YULIANTO
N.I.M. : 99.12.068
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : ANALISIS OPTIMASI *ECONOMIC DISPATCH*
COMBINED CYCLE POWER PLANT DENGAN
METODE EVOLUTIONARY PROGRAMMING

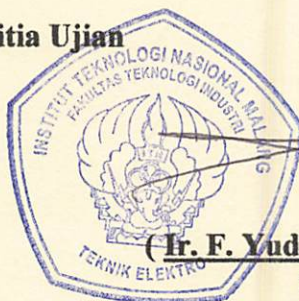
Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)

Hari : Jum'at
Tanggal : 30 September 2005
Dengan Nilai : 74 (B+) *hm*



(Ir. Mochtar Asroni, MSME)
Ketua

Panitia Ujian



(Ir. F. Yudi Limprapto, MT)
Sekretaris

Anggota Penguji

Andayati

(Ir. Taufik Hidayat, MT)
Penguji Pertama

Djojo Priatmono

(Ir. Djojo Priatmono, MT)
Penguji Kedua



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ENERGI LISTRIK

LEMBAR BIMBINGAN SKRIPSI

1. Nama : AGUS YULIANTO
2. N.I.M. : 99.12.068
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Elektro Energi Listrik
5. Judul Skripsi : ANALISIS OPTIMASI *ECONOMIC DISPATCH COMBINED CYCLE POWER PLANT* DENGAN METODE EVOLUTIONARY PROGRAMMING
6. Tanggal Mengajukan Skripsi : 27 Mei 2005
7. Tanggal Menyelesaikan Skripsi : 15 September 2005
8. Dosen Pembimbing : Ir. Choirul Saleh, MT.
9. Telah dievaluasi dengan nilai : 80,00(Delapan Puluh Koma Nol) *Sm*

Mengetahui,

Ketua Jurusan Teknik Elektro S-1



(Ir. F. Yudi Limprapto, MT)
NIP. Y. 1039500274

Diperiksa dan disetujui,

Dosen Pembimbing

(Ir. Choirul Saleh, MT)
NIP. 101 880 0190



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ENERGI LISTRIK

FORMULIR PERBAIKAN SKRIPSI

Nama : AGUS YULIANTO
NIM : 99.12.068
Masa Bimbingan : 27 Mei 2005 - 15 September 2005
Judul Skripsi : ANALISI OPTIMASI *ECONOMIC DISPATCH COMBINE CYCLE* POWER PLANT DENGAN METODE EVOLUTIONARY PROGRAMMING

No.	Tanggal	Uraian	Paraf Pembimbing
1.	30 - 09 -2005	Perbaiki tujuan dengan kesimpulan.	
2	30 - 09 -2005	Buat Tabel sebelum dan sesudah Optimasi	

Disetujui

(Ir. Taufik Hidayat, MT.)
Penguji Pertama

(Ir. Djojo Priatmono, MT.)
Penguji Kedua

Mengetahui,
Dosen Pembimbing

(Ir. Choirul Saleh, MT.)



**LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1**

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika *)

1	Nama Mahasiswa : AGUS Y		Nim : 99.12.068	
2	Waktu pengajuan	Tanggal :	Bulan :	Tahun :
3	Spesifikasi judul (berilah tanda silang)			
	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen		
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer		
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi		
	d. Sistem Kendali Industri	h. lainnya		
4	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) :		Mengetahui, Ketua Jurusan	
	Ir. Choirul Saleh Choirul Saleh, MT		 Ir. F. Yudi Limpraptono, MT Nip. Y. 1039500274	
5	Judul yang diajukan mahasiswa :	ANALISIS OPTIMASI ECONOMIC DISPATCH COMBINED CYCLE POWER PLANT DENGAN METODE EVOLUTIONARY PROGRAMMING		
6	Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu		
7	Catatan :		
	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu		Disetujui, , 200..... Dosen Choirul Saleh	

Perhatian :

1. Formulir Pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan : *) coret yang tidak perlu
*) dilingkari a, b, c, atau g. sesuai bidang keahlian



INSTITUT TEKNOLOGI NASIONAL
JL. Bendungan Sigura – gura No. 2
M A L A N G

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : Agus Yulianto
Nim : 99. 12. 068
Semester : XIV
Jurusan : Teknik Elektro S – 1
Konsentrasi : Energi Listrik

Dengan ini menyatakan bersedia / tidak bersedia *) membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**ANALISIS OPTIMASI ECONOMIC DISPATCH COMBINED CYCLE POWER
PLANT DENGAN METODE EVOLUTIONARY PROGRAMMING**

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Maret 2005
Kami yang membuat pernyataan,

Catatan :
Setelah disetujui agar formulir ini
Diserahkan mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut
*) coret yang tidak perlu


Ir. Choirul Saleh, MT
Nip : 1018800190



INSTITUT TEKNOLOGI NASIONAL
JL. Bendungan Sigura – gura No. 2
MALANG

Lampiran : 1 (Satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak / Ibu Ir. Choirul Saleh, MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama : **AGUS YULIANTO**
Nim : **99. 12. 068**
Jurusan : Teknik Elektro S-1
Konsentrasi : Energi Listrik

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama / Pendamping *), untuk penyusunan Skripsi dengan judul (proposal terlampir) :

ANALISIS OPTIMASI ECONOMIC DISPATCH COMBINED CYCLE
POWER PLANT DENGAN METODE EVOLUTIONARY
PROGRAMMING

Adapun Tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.


Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.

Malang, Maret 2005

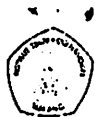
Mengetahui
Ketua Jurusan Teknik Elektro


Ir. F. Yudi Limpraptono, MT
NIP. Y . 1039500274

Hormat kami,

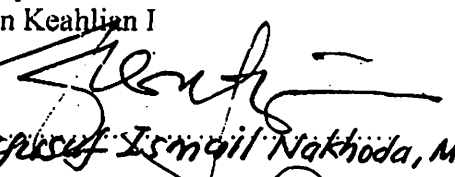
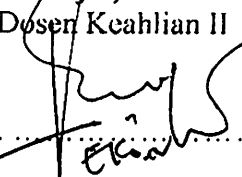
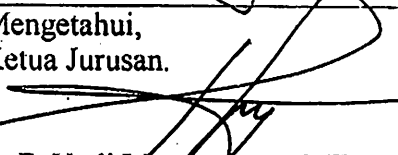
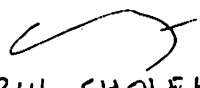

Agus Yulianto
Nim : 99/12 068

*) coret yang tidak perlu



BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika *)

1	Nama Mahasiswa : <u>AGUS YULIANTO</u>		Nim : <u>99.12.068</u>	
2	Keterangan	Tanggal <u>27</u>	Waktu	Tempat
	Pelaksanaan	<u>27-Mei-2005</u>	<u>13.00</u>	Ruang : <u>LAB ASTE</u>
Spesifikasi judul **):				
3	a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri		e. Elektronika & Komponen f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainnya	
4	Judul Proposal yang diseminarkan Mahasiswa	<u>ANALISIS OPTIMASI ECONOMIC DISPATCH COMBINED CYCLE POWER PLANT DENGAN METODE EVOLUTIONARY PROGRAMMING</u>		
5	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian		
6	Catatan :			
Persetujuan Judul Skripsi :				
7	Disetujui, Dosen Keahlian I  <u>Ir. Yusuf Ismail Nakhoda, MT</u>		Disetujui, Dosen Keahlian II  <u>Ir. CHOIRUL SHOLEH, MT</u>	
	Mengetahui, Ketua Jurusan.  <u>Ir. F. Yudi Limpraptono, MT</u> Nip. Y. 1039500274		Disetujui, Calon Dosen Pembimbing ybs.  <u>Ir. F. YUDI LIMPRAPTONO, MT</u>	

Perhatian :

- *) coret yang tidak perlu
- ***) dilingkari a, b, c, atau g. sesuai bidang keahlian.



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

3NI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 7 Juni 2005

Nomor : ITN-509/I.SKP /2/'05
Lampiran : satu lembar
Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. **Ir. CHOIRUL SALEH, MT**
Dosen Institut Teknologi Nasional
di -
Malang

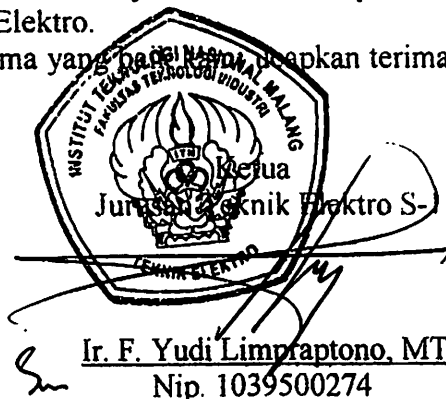
Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi melalui seminar proposal yang telah dilakukan untuk mahasiswa :

Nama : AGUS YULIANTO
Nim : 9912068
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro
Konsentrasi : T. Energi Listrik (S-1)

Dengan ini pembimbingan skripsi tersebut kami serahkan sepenuhnya kepada saudara/I selama masa waktu **6 (enam) bulan** terhitung mulai tanggal:

27 Mei 2005 s/d 27 Nov. 2005

Adapun tugas tersebut merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik, Jurusan Teknik Elektro.
Demikian atas perhatian serta kerjasama yang dapat kami dapatkan terima kasih


Ir. F. Yudi Limpraptono, MT
Nip. 1039500274

Tindakan :

1. Mahasiswa yang bersangkutan
2. Arsip.

Form. S-4a



FORMULIR BIMBINGAN SKRIPSI

Nama : AGUS YULIANTO
Nim : 99.12.068
Masa Bimbingan : 27 Mei 2005 s/d 27 November 2005

Judul Skripsi : ANALISIS OPTIMASI *ECONOMIC DISPATCH* COMBINED
CYCLE POWER PLANT DENGAN METODE
EVOLUTIONARY PROGRAMMING

No	Tanggal	Uraian	Parap Pembimbing
1.	3-6-2005	Perbaiki pada cara penulisan Referensi	
2.	5-6-2005	Perbaiki Rumusan Masalah	
3.	14-6-2005	ACC Bab I dan Bab II	
4.	17-6-2005	Beri keterangan pada gambar	
5.	20-6-2005	Sempurnakan Analisa Pembahasan	
6.	25-6-2005	ACC Bab III	
7.	27-6-2005	ACC Bab IV	
8.	30-6-2005	Uji Validasi Program	
9.	28-7-2005	Acc buat Makalah Seminar	
10.	15-9-2005	Acc Ujian Skripsi	

Malang,
Dosen Pembimbing

Ir. Choirul Saleh, MT

Form. S-4b

7-11-05
ITAJ

Evolutionary Programming Solution of Economic Load Dispatch with Combined Cycle Power Plant

R Gnanadass, Associate Member

Dr P Venkatesh, Associate Member

Prof T G Palanivelu, Non-member

Prof K Manivannan, Fellow

This paper was received on February 25, 2001. Written discussion on this paper will be received until November 30, 2000.

A maiden attempt is made to explore the feasibility of combined cycle Power plant (CCPP) in economic load dispatch (ELD) problem. In this paper evolutionary programming (EP) method has been proposed to solve the ELD problem for IEEE systems having quadratic and CCPP fuel cost functions. This method results in better optimal solution due to its global search characteristic. Results of evolutionary programming based economic load dispatch (EPELD) for IEEE 14 bus system having quadratic cost function are compared with those obtained by quadratic programming, Hessian method and linear programming method. The parameters used in EP method for the IEEE system are also compared with previously published work. The performance of EP on convergence for IEEE test systems is also presented.

Keywords: Economic load dispatch; Evolutionary programming; Combined cycle Power plant

NOTATION

- a_i, b_i, c_i : fuel cost coefficient of i th generator, \$/MW² h, \$/MW h, \$/h
- b_k, c_k : cost coefficients of CCPP in k th region, \$/MW h, \$/h
- $F_i(P_i), F_t$: fuel cost of generator, overall fuel cost, \$/h
- n : number generators
- P_D : system power demand, MW
- P_i : real power output of generator, MW
- P_L : total system transmission loss, MW
- P_s : slack bus power, MW
- P_t, P_s : parent trial vector, Slack bus power Vector
- $Q_{i,max}$: maximum reactive power limit of generator, MVAR
- $Q_{i,min}$: minimum reactive power limit of generator, MVAR

INTRODUCTION

The main objective of ELD is to minimise the fuel cost while satisfying the load demand. The classical Lambda iteration method has been used to solve ELD problem. This method uses equal increment cost criterion for systems without transmission losses and penalty factors using B_{mn} matrix for considering the losses. Other methods such as Gradient, Newton, Linear programming, Interior point and Quadratic programming method have also been applied to solve ELD problem. In all these methods, the fuel cost function is chosen to be of quadratic form. The fuel cost characteristic becomes more non-linear when valve point loading effects are included.

Recently combined cycle power plants (CCPP) have shown their importance in both developing and developed countries in order to improve the efficiency of generation. From the survey, the fuel cost characteristic of the CCPP is of non-smooth and non-differentiable form. Thus, the nonlinearity in the ELD problem is further increased.

Numerical optimization techniques and evolutionary computation (EC) methods can play a key role in future in solving the ELD problem.

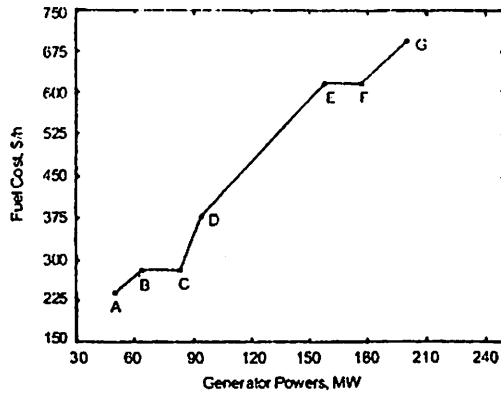
EC technique consists of three main areas such as genetic algorithms (GA), evolution strategies (ES) and evolutionary programming (EP). Each of these methods represented their

data structures in binary (or) real coded (or) in both forms. Operators such as crossover, mutation are also used in EC techniques.

Researchers have developed the solution of ELD problem using genetic algorithm for the same three units system by assuming one unit as a combined cycle co-generation plant". But the algorithm required large number of iterations/ generations when the system has large number of units. In order to minimize the number of generations and avoid the loss of useful chromosome for further generation micro genetic algorithm (MGA) was developed¹². In the process of generation of the MGA, the best strings are retained using Elitism. Crossover and mutation operations in binary are performed in case of GA and MGA. Among the EC methods, EP is better in obtaining global optimum, which relies on mutation rather than crossover". Due to inherent floabilities in fitness function and ease in coding, the EP method produces the better optimum solution with less number of iterations.

COMBINED CYCLE POWER PLANT (CCPP)

In CCPP, gas and steam turbines are working in combination to generate electric power. As an example the modelling of the plant in IEEE 14 bus system is carried out by following descriptions. The first generator of this system is operated between 50 MW and 200 MW. This generator is assumed to be combined cycle type with two 75 MW gas turbines and one 50 MW steam turbine. Five points of 50 bIW, 82.75 MW, 93.75 MW, 176.625 MW and 200 MW from original quadratic curve values are fixed and used for creating CCPP cost ehatacusinia. The first gas turbine is operating in ail open cycle till it reaches 85'16 of its capacity (63.75 MW). During 50-63.75 MW power generation this gas turbine is operated in the open cycle. Thus linear region AB is formulated. The exhaust heat produced by the gas turbine is enough to operate the steam turbine. Hence, during 63.75 MW-82,875 MW of power generation steam turbine is allowed to operate without any additional fuel. It results in a constant region BC in the fuel cost characteristic. During 82.875 N4W-93.75 blixr of generation both the gas and steam turbines are operated in a dosed cycle (CD region). Similarly, the operation of steam turbine and the second gas turbine result in regions DE, EF and FG. The fuel costs of CCPP are computed by substituting generator powers in its quadratic fuel cost functions. The cost characteristic of CCPP is modeled as shown in Figure 1 and Table 1. Likewise in all the test systems the cost chancteristic of a generator having quadratic function is modified as CCPP.



ECONOMIC LOAD DISPATCH PROBLEM FORMULATION

The objective of the ELD problem is to minimize the total cost of generating units.

Objective function

$$F_t = \sum_{i=1}^n F_i(P_i) \$ / h \quad (1)$$

subjected to the constraint balance and in-equality of real power balance

$$\sum_{i=1}^n P_i - P_L - P_D = 0 \quad (2)$$

And in- equality of real limits on the generator outputs

$$P_{i_{\min}} \leq P_{i_{\max}}, i = 1, \dots, n \quad (3)$$

EVOLUTIONARY PROGRAMMING BASED ECONOMIC LOAD DISPATCH (EPELD)

Evolutionary programming is a probabilistic search technique, which generates the initial parent vectors distributed uniformly in intervals within the limits and obtains global optimum solution over number of iterations. The main stages of this techniques are initialization, creation of offspring vectors by mutation and competition and selection of best vectors to evaluate best fitness solution. The implementation of EP algorithm is given below.

(i) Initialisation

The initial population (k number of parent vectors) is generated after satisfying the equations (2) and (3). The elements of parent vectors (P) are the real power outputs of generating units disvibuced uniformly between their minimum and maximum limits. Slack bus generator vector is calculated using Newton-Raphson method for the above generations.

(ii) Mutation

An off-spring vector p,' is created from each parent vector by adding gaussian random variable with zero mean and standard deviation σ_i , denoted as

$$P_i = P_i + N(0, \sigma_i^2) \quad \text{for } i = 1, 2, \dots, n-1 \quad (4)$$

where

$$\sigma_i = \beta \times \sum_{i=1}^{n-1} F_i(P_i) / f_{\min}(P_{i_{\max}} - P_{i_{\min}}) \quad (5)$$

The created offspring vector must satisfy the minimum and maximum generation limits of the units. The slack bus generator vector for the above off-spring vector generations is also calculated using Newton-Raphson method.

The constrained optimization ELD problem is converted into unconstrained optimization problem using penalty factor (PF) μ given in the equation (6). This becomes the fitness function in EP method.

$$\text{Fitness function} = F_i + \text{PF} \times \Phi \quad (6)$$

The second term in equation (6) becomes zero during initialization and its gets a non-zero value after mutation only if generator vectors violate its limits. Therefore, only F_i becomes the fitness function and is computed for the offspring vectors similar to the parent vectors.

Because the real power generations at all intervals are control variables which are self constrained. On the other hand, the operating ramp rate limitations are constrained by adding them as the exact penalty.

(iii) Competition and Selection

The parent vial vectors P_i and the corresponding offspring P_i ; (2k number of vectors) compete with each other for survival within the competing pool. The competition is based on the cost parent vectors cost of parent vectors $F_i(P_i)$ with the corresponding cost of off-spring vectors $F_i(P_i')$ in this population. The best vector having minimum cost, whether parent vector P_i ; (or) offspring vector P_i' is selected for the new parent for the next generation. Initialisation and mutation are repeated until there is no appreciable improvement in the fitness value.

RESULTS AND DISCUSSION

EP was applied to solve the ELD problem and was demonstrated with IEEE 14 and 30 bus systems. The system load P_D was 259 MW and 283 MW, respectively. The minimum solution was obtained by assuming the fuel cost characteristics of generators in quadratic (case A) and any one of the generator as CCPP (case B). Actual transmission losses of IEEE test systems were computed by Newton-Raphson method. A MATLAB program was developed for EPELD algorithm in P-111 700 MHz system.

IEEE System

The fuel cost of characteristic of first generator was assumed to be of CCPP and other generators as quadratic nature. The coefficients and generation emits for case A are taken from 'Contaxis, et al' and case B are given in Table. The minimum solution obtained using EPELD for two cases after several runs is given. The execution time for each case was approximately 5 s.

The comparison of ELD results between EP and other optimization methods such as quadratic programming, Hessian method and linear programming. It reveals that the fuel cost obtained by EP method is in close with those obtained by other methods. The convergence performance of the EPELD method is tested for 50 iterations to obtain minimum solution is shown. The optimal solution was obtained within 20 iterations. From it is seen that there is no vast variation in the three generator powers after several runs. Thus, the consistency of EPELD algorithm is verified.

Fig. 1 Data Unit MW Generation by The EP Method For The-3 Unit System

Unit	a_i	b_i	c_i	P_{\max}	P_{\min}
1	0.001562	7.92	561	600	100
2	0.00194	7.85	310	400	100
3	0.00482	7.97	78	200	50

	Unit 1	Unit 2	Unit 3	TOTAL
EP	250	400	200	850

Table 1. Summary of result

Method	Cost (\$)
Evolutionary Programming	8178.431

CONCLUSION

The performance of the proposed method. New England Test System.

Fig. 1 unit data parameter for the -3 unit system and computing time from 3 time simulation with scaling factor 0.3, maximum generation 100 while varied the population size 50.

It can be seen from Fig. 1 that when the population size (the number of candidate solution in each generation) increase, total production cost decrease while the computing time increase.

In table 1, we compared the result obtained from evolutionary programming, and it can be seen that the proposed method gave a better solution than that of evolutionary programming.

REFERENCE

- A. J. Wood dan B. F. Wollenberg, " *Power Generation Operation, and Control* " Second Edition, John Wiley & Sons. 1996
- H Saadat. 'power System Analysis.' WCB/McGraw-Hill, 1999
- A Momoh, M E El-Hawary and A Rambabu. ' A Review of Selected Optimal Power Literatur to 1993, Part-I: Newton, linier Programming and Interior Point Method. ' *IEEE Transaction on Power System*, vol 14, no 1,1999, pp 105.
- S B Shebel and K Briting. ' *Refined Genetic Algoritma Economic Dispatch Example.* ' *IEEE Transaction on Power System*, vol 10, no 1 1995,pp 117-124.
- H T Yang, P C Yang and C L Huang. 'Evolutionary Programming Based Economic Dispatch for Unit with Non-smooth Fuel Cost Function. ' *IEEE Transaction on Power System*, vol 11, no 1, 1996,pp 112-118.
- Nanda, D P Kothari and S C L Huang. 'New Optimal Power-dispatch Algoritma using Fletcher's Quadratic Method. *IEE Proceeding*,vol 136, 1989,pp 153-161.
- A Achyuthakan. *Master Thesis Report in Asian Institute of Technologi*, Thalland,1997.
- Turevich and K P wong.'Evolutionary Programing Based Optimal Power Flow Algoritma. *IEEE Transaction on Power System* ,vol 1, 1986,pp 1245-1250



**PT PLN (PERSERO)
PENYALURAN DAN PUSAT PENGATUR BEBAN JAWA BALI
REGION JAWA TIMUR & BALI**

Jl. Suningrat No. 45 Taman Sidoarjo 61257

Telepon : (031) 7882113, 7882114
Facsimile : (031) 7882578, 7881024

Kotak Pos : 4119 SBS
Bank : Bank Mandiri

Nomor : 059 / 330 / RJTB / 2005.
Surat Sdr. No. : ITN-593/III.TA/2/2004.
ITN-811/III.TA/2/2005.
Lampiran : 1 (satu) lampiran.
Perihal : Ijin Survey/ Pengambilan Data.

09 MAR 2005

Kepada

Yth. Dekan Fakultas Teknik.
Institut Teknologi Nasional Malang
Di
MALANG

Menunjuk surat Saudara nomor : ITN-593/III.TA/2/2004 tanggal 2 Desember 2004 dan ITN-811/III.TA/2/2005 tanggal 24 Januari 2005 perihal : Survey/ Permintaan Data, dengan ini diberitahukan bahwa kami tidak keberatan untuk memberikan ijin kepada Mahasiswa Saudara, bernama :

- AGUS YULIANTO NIM : 99.12.068.
- ROY AGUSTIAWAN RAMDHANY NIM : 98.12.121.

Untuk melakukan Praktek Kerja pada PT. PLN (Persero) P3B Region Jawa Timur dan Bali Bidang OPHAR, dengan persyaratan sebagai berikut :

1. Mahasiswa tersebut diatas supaya mengisi dan menanda tangani Surat Pernyataan 1 (satu) lembar bermeterai Rp. 6.000,-
2. Mahasiswa yang bersangkutan agar mematuhi peraturan/ketentuan yang berlaku di PT. PLN (PERSERO) sehingga faktor-faktor kerahasiaan harus benar-benar diutamakan.
3. Semua biaya perjalanan, penginapan, makan dan lain sebagainya tidak menjadi tanggungan PT. PLN (Persero) P3B Region Jawa Timur dan Bali.
4. Buku Laporan Kerja Praktek Mahasiswa tersebut agar dikirimkan kepada PT. PLN (Persero) P3B Region Jawa Timur dan Bali 1 (satu) buah.
5. Untuk informasi lebih lanjut dapat menghubungi PT. PLN (Persero) P3B Region Jawa Timur dan Bali Cq. Bidang Enjiniring.

Demikian harap maklum dan terima kasih atas perhatian saudara.

a.n. MANAGER
PH MANAGER BIDANG ENJINIRING,



Tembusan Yth. :

1. M.SDM PLN P3B.
2. Sdr. Agus Yulianto
3. Sdr. Roy Agustiawan Ramdhany



Nomor : P126 354
Sifat : Biasa
Lampiran :

Tanggal, 12 Agustus 2005

Kepada YTH :
Dekan Fakultas Teknologi Industri
Institut Teknologi Nasional Malang (ITN)
Kampus I : Jl. Bendungan Sigura gura No.2
MALANG - 65145

Perihal : Survey Data

Dengan Hormat,
Menunjuk surat Saudara :

Nomor : ITN-1504/III.TA/2/2005
Tanggal : 02 Agustus 2005
Perihal : Survey / Permintaan Data

Dengan ini kami beritahukan, bahwa permohonan mahasiswa Saudara dari Institut Teknologi Nasional Malang Jurusan Teknik Elektro untuk melaksanakan survey data atas nama :

=>AGUS YULIANTO NIM: 9912068
=>RHOMA DWI Y NIM : 9912 174

Kami bisa memenuhi permohonan mahasiswa saudara, tentang kegiatan Survey dan Permintaan Data di PT Pembangkitan Jawa Bali Unit Pembangkitan Gresik pada tahun2005

Demikian atas perhatiannya kami ucapkan terima kasih

MANAJER,

Deputi Manajer SDM



Wicnoe Satrijono

PT PJB UNIT PEMBANGKITAN GRESIK

Jln. Harun Tohir, GRESIK - 61112 - Indonesia, »: 62-31-3984540 Fax : 62-31-3981568 E-mail: upgrk@pjb2.com

No.	NAMA PEMBANGKIT	KAPASITAS			LAMA WAKTU (JAM)				BIAYA START UP (JUTA Rp)		KOEFSIEN BIAYA BAHAN BAKAR		
		Daya Terpasang (MW)	MIN (MW)	MAX (MW)	MIN UP TIME	MIN DOWN TIME	COLD START UP	HOT START UP	COLD START UP	HOT START UP	a0	a1	a2
1	UP. GRESIK CC- 3.3.1												
	BLOK 1 (GAS)	526	125	500	36	10	3	2	24,4	47,1	3,166	82,427	0,0112590
	BLOK 2 (GAS)	526	125	500	36	10	3	2	24,4	47,1	24,282	121,771	0,0062063
	BLOK 3 (GAS)	526	125	500	36	10	3	2	24,4	47,1	21,676	125,668	0,0053446
	PLTU # 1/2 (GAS)	100	43	85	48	10	9	1	143.74	40.59	1327126.7	217378.359	132.066
	PLTU # 3/4 (GAS)	200	90	175	48	10	9	2	229.5	92.52	5017369.5	169242.579	193.545
	PLTG GRESIK 1-3 (GAS)	3 x 20	5	16	3	1	1	0	6.13	0	352707.3	350680.77	903.969
	PLTG GILITIMUR 1-2 (HSD)	2 x 20	5	16	3	1	1	0	6.33	0	687181.85	683240.965	1762.3893

Catatan :

Harga gas UP Gresik	2.53	US\$/MMBTU
Nilai Tukar	9000	Rp/US\$
Harga Batubara	235	Rp/kg
Nilai Kalor Batubara	5100	kCal/kg
Harga MFO	1595.5	Rp/liter
Harga HSD	1595.5	Rp/liter
Nilai Kalor MFO	9777.46	kCal/liter
Nilai Kalor HSD	9063.04	kCal/liter



DATA PEMBEBANAN DISTRIBUSI BEBAN BLOK
KONFIGURASI 3.3.1-3.3.1-3.3.1
PT. PLN PJB II UP. GRESIK

NO	Beban Total	Beban Blok 1	Beban Blok 2	Beban Blok 3	BBG Blok 1	BBG Blok 2	BBG Blok 3	Total Atrian BBG
	MW	MW	MW	MW	KNM ³ / H	KNM ³ / H	KNM ³ / H	KNM ³ / H
1	375	125	125	125	39.21	39.21	39.21	117.63
2	400	125	125	150	39.21	39.21	44.78	123.20
3	425	125	125	175	39.21	39.21	47.98	126.40
4	450	125	125	200	39.21	39.21	52.31	130.73
5	475	125	125	225	39.21	39.21	55.62	134.04
6	500	125	125	250	39.21	39.21	58.65	137.07
7	525	125	125	275	39.21	39.21	62.98	141.40
8	550	125	125	300	39.21	39.21	65.54	143.96
9	575	125	125	325	39.21	39.21	70.12	148.54
10	600	125	125	350	39.21	39.21	74.03	152.45
11	625	125	125	375	39.21	39.21	77.96	156.38
12	650	125	225	300	39.21	55.62	65.54	160.37
13	675	125	250	300	39.21	58.65	65.54	163.40
14	700	125	275	300	39.21	62.98	65.54	167.73
15	725	125	300	300	39.21	65.54	65.54	170.29
16	750	125	300	325	39.21	65.54	70.12	174.87
17	775	125	300	350	39.21	65.54	74.03	178.78
18	800	125	300	375	39.21	65.54	77.96	182.71
19	825	225	300	300	55.62	65.54	65.54	186.70
20	850	250	300	300	58.65	65.54	65.54	189.73
21	875	275	300	300	62.98	65.54	65.54	194.06
22	900	300	300	300	65.54	65.54	65.54	196.62
23	925	300	300	325	65.54	65.54	70.12	201.20



**DATA PEMBEBANAN DISTRIBUSI BEBAN BLOK
KONFIGURASI 3.3.1-3.3.1-3.3.1
PT. PLN PJB II UP. GRESIK**

NO	Behan	Behan	Behan	Behan	BBG	BBG	BBG	Total
	Total	Blok 1	Blok 2	Blok 3	Blok 1	Blok 2	Blok 3	Ahiran BBG
	MW	MW	MW	MW	KNM ³ / H	KNM ³ / H	KNM ³ / H	KNM ³ / H
24	950	300	350	350	65.54	65.54	74.03	205.11
25	975	300	375	375	65.54	65.54	77.96	209.04
26	1000	300	350	350	65.54	74.03	74.03	213.60
27	1025	300	350	375	65.54	74.03	77.96	217.53
28	1050	300	375	375	65.54	77.96	77.96	221.46
29	1075	350	350	375	74.03	74.03	77.96	226.02
30	1100	375	375	350	77.96	77.96	74.03	229.95
31	1125	375	375	375	77.96	77.96	77.96	233.88
32	1150	375	375	400	77.96	77.96	83.04	238.96
33	1175	300	375	500	65.54	77.96	83.04	243.56
34	1200	350	375	475	74.03	77.96	95.81	247.80
35	1225	375	375	475	77.96	77.96	95.81	251.73
36	1250	375	375	500	77.96	77.96	100.06	255.98
37	1275	375	400	500	77.96	83.04	100.06	261.06
38	1300	350	475	475	74.03	95.81	95.81	265.65
39	1325	375	475	475	77.96	95.81	95.81	269.58
40	1350	375	475	500	77.96	100.06	100.06	273.83
41	1375	375	500	500	77.96	100.06	100.06	278.08
42	1400	400	500	500	83.04	100.06	100.06	283.16
43	1425	475	475	475	95.81	95.81	95.81	287.43
44	1450	475	475	500	95.81	95.81	100.06	291.68
45	1475	475	500	500	95.81	100.06	100.06	295.93
46	1500	500	500	500	100.06	100.06	100.06	300.18



DATA TOTAL ALIRAN BAHAN BAKAR GAS PER TINGKAT BEBAN PADA TIAP BLOK

No	Tingkat Beban	Total Aliran BBG	
		KNM ³ /Hr	MBTU/Hr
1	150	46,55	1838,33
2	175	49,37	1939,69
3	200	53,11	2097,39
4	225	56,83	2244,30
5	250	59,67	2356,45
6	275	63,31	2460,71
7	300	65,25	2576,82
8	325	69,70	2752,55
9	350	73,77	2913,28
10	375	78,82	3112,72
11	400	83,59	3301,09
12	425	88,06	3477,62
13	450	92,69	3660,46
14	475	96,05	3793,15
15	500	99,98	3948,35

Blok 1

No	Tingkat Beban	Total Aliran BBG	
		KNM ³ /Hr	MBTU/Hr
1	150	43,56	1720,44
2	175	47,42	1872,92
3	200	51,28	2025,44
4	225	55,15	2177,91
5	250	59,01	2330,43
6	275	62,87	2482,91
7	300	66,73	2635,42
8	325	70,59	2787,94
9	350	73,92	2919,21
10	375	78,23	3089,53
11	400	82,73	3267,44
12	425	87,28	3447,13
13	450	91,83	3626,82
14	475	96,38	3806,46
15	500	100,93	3986,15

Blok 2

No	Tingkat Beban	Total Aliran BBG	
		KNM ³ /Hr	MBTU/Hr
1	150	41,44	1636,84
2	175	45,27	1787,78
3	200	49,09	1938,67
4	225	52,91	2089,61
5	250	56,73	2240,55
6	275	60,55	2391,48
7	300	64,37	2542,38
8	325	68,20	2693,32
9	350	72,31	2855,98
10	375	75,88	2996,73
11	400	80,22	3168,08
12	425	84,70	3344,96
13	450	89,18	3521,85
14	475	93,66	3698,77
15	500	98,13	3875,65

Blok 3

*1 KNM³ = 37,3258 MSCF

*HHV Bahan Bakar Gas = 1058,02 BTU/SCF

1	MMSCF	=	10^3 MSCF	Volume = Berat / BD	
1	MSCF	=	10^3 SCF	Berat = Vol x BD	
1	BSCF	=	10^9 SCF	BTU	= British Thermal Unit
1	BSCF	=	10^6 MSCF	MMBTU	= Millions BTU
1	BSCF	=	10^3 MMSCF	MMBTU	= 10^6 BTU
1	MMBTU	=	1 MSCF	1 BTU	= 3,968 kalori = 293×10^{-6} kWh
1	MMBTU	=	10^6 BTU	1 kalori	= 0,252 BTU
1	MMSCF	=	10^6 SCF	Nilai kalor	= btu/lb atau btu/SCF (ARCO → 900 s.d 1300)
1	BAREL	=	158,98 Liter	Nilai kalor HSD	= 19570 btu/lb
	SFC	=	0,0077 MSCF / kWh	Nilai kalor MFO	= 18635 btu/lb
		=	0,0077 MMBTU / kWh	Nilai kalor IDO	= 19140 btu/lb
		=	7700 BTU / kWh	1 lb	= 0,555 kg
	SCF	=	Standard Cubic Feed	Heat rate	= kilo kalori / kWh
	MSCF	=	Meter Standard Cubic Feed		
				Heat rate BBM	= <u>Pem.BBM(Liter) x (Nilai Kalor (BTU/lb) x 0,555) x BD</u> Produksi (kWh)
				Heat rate Batu bara	= <u>Pem.B.Bara (kg) x Nilai Kalor (kcal / kg)</u> Produksi (kWh)
				Heat rate Gas	= <u>Pem.Gas (MMBTU) x 1000000 x 0,252</u> Produksi (kWh)

LAMPIRAN:

1. Metode Kuadratis Minimum
2. Data Teknis CCPP Unit Pembangkitan Gresik
3. Flowchart Penyelesaian masalah
4. Persamaan biaya bahan bakar (Mathlab Ver. 5.3.1)
5. Distribusi beban Metode BP dan FP (Mathlab Ver. 5.3.1)
6. SFC, Fuel Cost, Incremental Fuel Cost, Heat Rate (Mathlab Ver. 5.3.1)
7. Perbandingan Fuel Cost BP dan FP dengan Pola Gresik, dan Penghematan yang diperoleh (Mathlab Ver. 5.3.1)
8. Listing Program (Mathlab Ver. 5.3.1)

METODE KUADRATIS MINIMUM

Dari himpunan titik-titik $(x_1, y_1), \dots, (x_n, y_n)$, pendekatan persamaan derajat dua yang menyatakan hubungan antara x_i dan y_i secara matematis dapat ditulis sebagai berikut:

$$f(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + E$$

dimana c_0, c_1, c_2 koefisien dan E adalah kesalahan atau tepatnya perbedaan nilai data dengan nilai yang diperoleh dari model, sehingga dapat ditulis;

$$E = y - c_0 - c_1 \cdot x - c_2 \cdot x^2$$

Jumlah kuadrat dari kesalahan adalah:

$$S_r = \sum_{i=1}^n E_i^2 = \sum_{i=1}^n (y_i - c_0 - c_1 \cdot x_i - c_2 \cdot x_i^2)^2$$

dimana n adalah banyaknya titik.

S_r diturunkan masing-masing terhadap c_0, c_1, c_2 , maka diperoleh;

$$\frac{\partial S_r}{\partial c_0} = -2 \cdot \sum_{i=1}^n (y_i - c_0 - c_1 \cdot x_i - c_2 \cdot x_i^2)$$

$$\frac{\partial S_r}{\partial c_1} = -2 \cdot \sum_{i=1}^n [x_i \cdot (y_i - c_0 - c_1 \cdot x_i - c_2 \cdot x_i^2)]$$

$$\frac{\partial S_r}{\partial c_2} = -2 \cdot \sum_{i=1}^n [x_i^2 \cdot (y_i - c_0 - c_1 \cdot x_i - c_2 \cdot x_i^2)]$$

Agar S_f minimum, maka masing-masing harga turunan diatas harus =0.

$$0 = \sum_{i=1}^n Y_i - \sum_{i=1}^n C_0 - \sum_{i=1}^n C_1 \cdot x_i - \sum_{i=1}^n C_2 \cdot x_i^2$$

$$0 = \sum_{i=1}^n Y_i \cdot x_i - \sum_{i=1}^n C_0 \cdot x_i - \sum_{i=1}^n C_1 \cdot x_i^2 - \sum_{i=1}^n C_2 \cdot x_i^3$$

$$0 = \sum_{i=1}^n Y_i \cdot x_i^2 - \sum_{i=1}^n C_0 \cdot x_i^2 - \sum_{i=1}^n C_1 \cdot x_i^3 - \sum_{i=1}^n C_2 \cdot x_i^4$$

Bila $\sum_{i=1}^n C_0 = n \cdot C_0$, maka persamaan diatas dapat ditulis;

$$\sum_{i=1}^n C_2 \cdot x_i^2 + \sum_{i=1}^n C_1 \cdot x_i + n \cdot C_0 = \sum_{i=1}^n Y_i$$

$$\sum_{i=1}^n C_2 \cdot x_i^3 + \sum_{i=1}^n C_1 \cdot x_i^2 + \sum_{i=1}^n C_0 \cdot x_i = \sum_{i=1}^n Y_i \cdot x_i$$

$$\sum_{i=1}^n C_2 \cdot x_i^4 + \sum_{i=1}^n C_1 \cdot x_i^3 + \sum_{i=1}^n C_0 \cdot x_i^2 = \sum_{i=1}^n Y_i \cdot x_i^2$$

merupakan tiga persamaan dengan tiga variabel yang berbeda.

Bentuk matrik dari persamaan-persamaan diatas adalah.

$$A = \begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i & n \\ \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i^4 & \sum_{i=1}^n x_i^3 & \sum_{i=1}^n x_i^2 \end{bmatrix}, \quad B = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n y_i \cdot x_i \\ \sum_{i=1}^n y_i \cdot x_i^2 \end{bmatrix}, \quad C = \begin{bmatrix} C_2 \\ C_1 \\ C_0 \end{bmatrix}$$

setelah nilai C_0, C_1, C_2 diperoleh dari penyelesaian matrik diatas, maka didapatkan persamaan kuadrat yang

Lampiran

mewakili fungsi dari himpunan titik-titik tersebut, yaitu:

$$y = f(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2$$

Maka apabila terdapat data sebagai berikut:

Sebagai Masukan :

Y = adalah Konsumsi Bahan Bakar per Blok (dalam MBTU/hours)

X = Adalah Tingkat Beban (dalam MW)

N = Adalah jumlah data

Persamaan Kuadratis Minimum adalah:

$$Y = f(x) = C_0 + C_1 \cdot x + C_2 \cdot x^2$$

Penggunaannya dalam menentukan Persamaan biaya bahan bakar,

$$H_i(P_i) = a + b \cdot x + c \cdot x^2$$

Maka dilakukan usaha mencari nilai a, b, c. Dapat dilakukan dengan membentuk persamaan:

$$\sum Y_i = a \sum X_i^2 + b \sum X_i + c n$$

$$\sum X_i Y_i = a \sum X_i^3 + b \sum X_i^2 + c \sum X_i$$

$$\sum X_i^2 Y_i = a \sum X_i^4 + b \sum X_i^3 + c \sum X_i^2$$

Setelah persamaan tersebut terbentuk, dapat diselesaikan nilai konstanta a, b, c dengan menggunakan Determinan Matrik, sehingga dapat pula dibentuk persamaan biaya bahan bakar untuk masing-masing Blok

Determinan Matrik untuk mencari nilai konstanta a, b, c :

$$a = \frac{\begin{vmatrix} \sum Y_i & \sum X_i & n \\ \sum X_i Y_i & \sum X_i^2 & \sum X_i \\ \sum X_i^2 Y_i & \sum X_i^3 & \sum X_i^2 \end{vmatrix}}{\begin{vmatrix} \sum X_i^2 & \sum X_i & n \\ \sum X_i^3 & \sum X_i^2 & \sum X_i \\ \sum X_i^4 & \sum X_i^3 & \sum X_i^2 \end{vmatrix}}$$

$$b = \frac{\begin{vmatrix} \sum X_i^2 & \sum Y_i & n \\ \sum X_i^3 & \sum X_i Y_i & \sum X_i \\ \sum X_i^4 & \sum X_i^2 Y_i & \sum X_i^2 \end{vmatrix}}{\begin{vmatrix} \sum X_i^2 & \sum X_i & n \\ \sum X_i^3 & \sum X_i^2 & \sum X_i \\ \sum X_i^4 & \sum X_i^3 & \sum X_i^2 \end{vmatrix}}$$

i= Blok ke-i

$$c = \frac{\begin{vmatrix} \sum X_i^2 & \sum X_i & Y_i \\ \sum X_i^3 & \sum X_i^2 & \sum X_i Y_i \\ \sum X_i^4 & \sum X_i^3 & \sum X_i^2 Y_i \end{vmatrix}}{\begin{vmatrix} \sum X_i^2 & \sum X_i & n \\ \sum X_i^3 & \sum X_i^2 & \sum X_i \\ \sum X_i^4 & \sum X_i^3 & \sum X_i^2 \end{vmatrix}}$$

setelah diketahui nilai konstanta a, b, c, maka persamaan biaya bahan bakar sebagai fungsi beban:

$$H_1(P_1) = a P_1^2 + b P_1 + c$$

Validasi Program pada IEEE Test Sistem

Untuk mensimulasi program penyelesaian Economic Dispatch dengan metode Evolutionary Programming (EP) dalam skripsi ini menggunakan system IEEE test yaitu : data pada jurnal “*Evolutionary Programming Solution of Economic Load Dispatch with Combine Cycle Power Plants*”, untuk jumlah pembangkit sebanyak 3 unit dan periode waktu 1 jam, data-data selengkapnya seperti pada tabel 1 dan 2.

Tabel 1. Data Unit Pembangkit (Data IEEE)

No	Nama Pembangkit	Kapasitas (MW)		Koefisien Biaya Bahan Bakar		
		P maks	P min	a	b	c
1	Unit 1	600	100	0,001562	7,92	561
2	Unit 2	600	100	0,001940	7,85	310
3	Unit 3	200	50	0,004820	7,97	78

Tabel 2. Data Validasi (Data IEEE)

No	Nama Pembangkit	Beban (MW)
1	Unit 1	250
2	Unit 2	400
3	Unit 3	200
Total		850

Tampilan Software Metode EP data Jurnal dengan beban system 850 MW

Tampilan Data

General | Data Generator | Data Pembebanan | Data PLN

Jumlah Pembangkit

Jumlah Jam

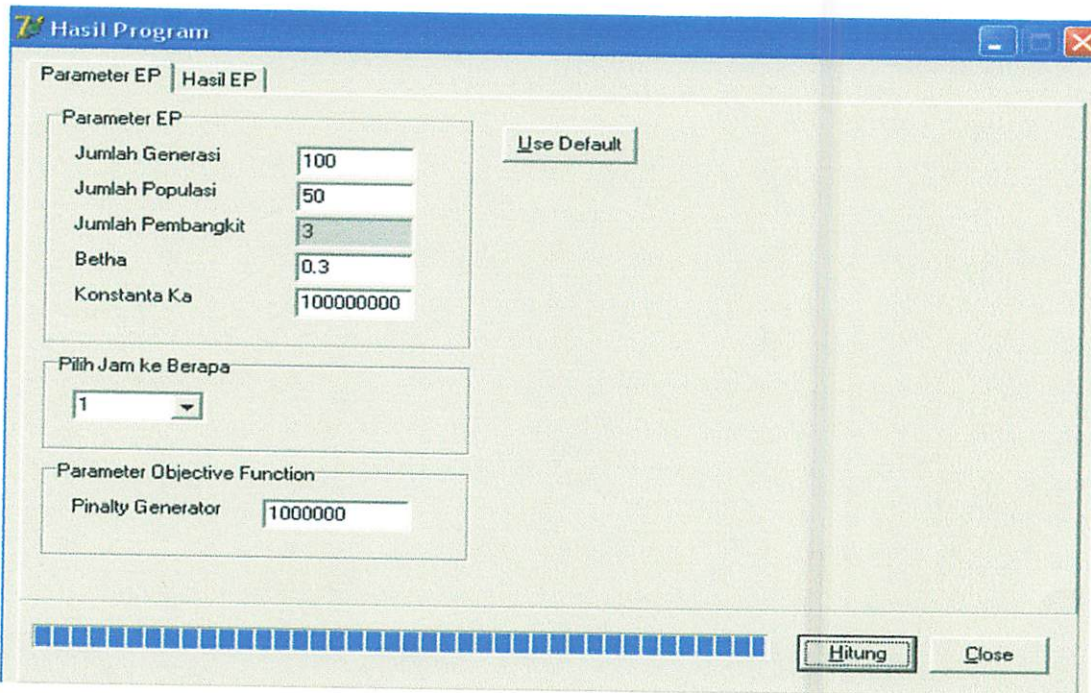
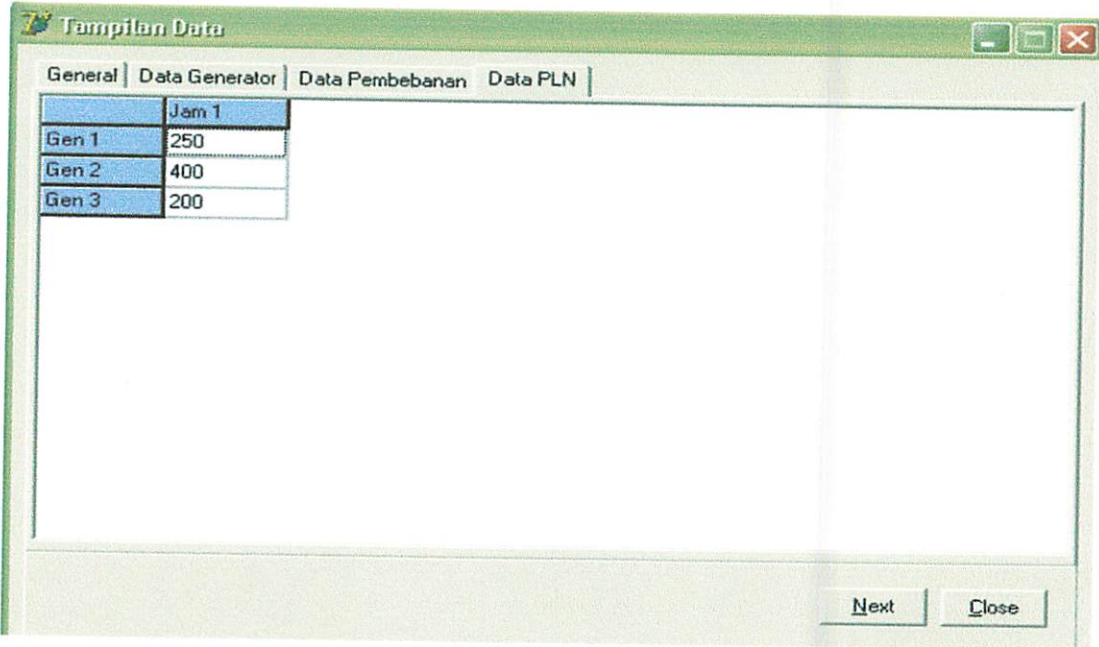
Next Close

Tampilan Data

General | Data Generator | Data Pembebanan | Data PLN

Gen	Nama	Pmax	Pmin	a0	a1	a2	Tup
1	B 1	600	100	561	7.92	0.001562	1
2	B 2	600	100	310	7.85	0.00194	1
3	B 3	200	50	78	7.97	0.00482	1

Next Close



7 Hasil Program

Parameter EP Hasil EP

No	P EP (MW)	P PLN (MW)	Cost EP (\$)	Cost PLN (\$)	Selisih (\$)
1	393	250	3,916	2,639	-1,277
2	335	400	3,153	3,760	607
3	122	200	1,126	1,865	739

Biaya EP 8,194 Biaya PLN 8,264 Selisih 69

Hitung Close

Dengan memakai data IEEE diatas maka diperoleh hasil sebagai berikut dengan parameter generasi = 100, populasi = 50, jumlah pembangkit = 3, komutasi dilakukan jam 1 dengan beban 850 MW.

Setelah dilakukan proses perbandingan data yang ada pada jurnal dan data pada hasil program, maka didapatkan biaya operasional pada data jurnal sebesar \$. 8178.43.- dan biaya operasional pada hasil program sebesar \$. 8194.-. Maka selisih antara biaya operasional pada jurnal dan biaya operasional hasil program adalah \$. 15.5686.-(error 0,19).

Listing Program DELPHI 7.0

unit InputGen;

interface

uses

Windows, Messages, SysUtils,
Variants, Classes, Graphics, Controls,
Forms,
Dialogs, StdCtrls, ComCtrls,
ExtCtrls, Grids;

type

TfrmInput = class(TForm)
 PageControl1: TPageControl;
 Panel1: TPanel;
 TabSheet1: TTabSheet;
 TabSheet2: TTabSheet;
 TabSheet3: TTabSheet;
 btnClose: TButton;
 btnNext: TButton;
 SaveDialog1: TSaveDialog;
 Label1: TLabel;
 Label2: TLabel;
 edtNGen: TEdit;
 edtNjam: TEdit;
 fgGen: TStringGrid;
 fgLoad: TStringGrid;
 TabSheet4: TTabSheet;
 fgPLN: TStringGrid;
 procedure btnCloseClick(Sender:
TObject);
 procedure FormCreate(Sender:
TObject);
 procedure edtNGenChange(Sender:
TObject);
 procedure edtNjamChange(Sender:
TObject);
 procedure btnNextClick(Sender:
TObject);
 private
 { Private declarations }
 public
 { Public declarations }
end;

var

frmInput: TfrmInput;

implementation

uses uObjFunc, uHasil;

{ \$R *.dfm }

procedure

TfrmInput.btnCloseClick(Sender:
TObject);
begin
 Close;
end;

procedure

TfrmInput.FormCreate(Sender:
TObject);
begin
 fgGen.Cells[0,0] := 'Gen';
 fgGen.Cells[1,0] := 'Nama';
 fgGen.Cells[2,0] := 'Pmax';
 fgGen.Cells[3,0] := 'Pmin';
 fgGen.Cells[4,0] := 'a0';
 fgGen.Cells[5,0] := 'a1';
 fgGen.Cells[6,0] := 'a2';
 fgGen.Cells[7,0] := 'Tup';
 fgGen.Cells[8,0] := 'Tdown';
 fgGen.Cells[9,0] := 'Sh';
 fgGen.Cells[10,0] := 'Sc';
 fgGen.Cells[11,0] := 'Tcold';
 fgGen.Cells[12,0] := 'InitSt';
 fgGen.Cells[13,0] := 'Ramp Rate';
 fgLoad.Cells[0,0] := '';
 fgLoad.Cells[1,0] := 'Load';
 fgLoad.Cells[2,0] := 'Res';
end;

procedure

TfrmInput.edtNGenChange(Sender:
TObject);
var i: integer;
begin
 if edtNgen.Text="" then
 begin
 fgGen.RowCount:=2;
 fgPLN.RowCount:=2;
 end
 else
 begin

```
fgGen.RowCount:=StrToInt(edtNgen.  
Text)+1;
```

```
fgPLN.RowCount:=StrToInt(edtNgen.  
Text)+1;
```

```
for i:=1 to StrToInt(edtNgen.Text)  
do
```

```
begin
```

```
fgGen.Cells[0,i]:=IntToStr(i);
```

```
fgPLN.Cells[0,i]:='Gen
```

```
'+IntToStr(i);
```

```
end;
```

```
end;
```

```
end;
```

```
procedure
```

```
TfrmInput.edtNjamChange(Sender:  
TObject);
```

```
var i:integer;
```

```
begin
```

```
if edtNjam.Text="" then
```

```
begin
```

```
fgLoad.RowCount:=2;
```

```
fgPLN.ColCount:=2;
```

```
end
```

```
else
```

```
begin
```

```
fgLoad.RowCount:=StrToInt(edtNjam.  
Text)+1;
```

```
fgPLN.ColCount:=StrToInt(edtNjam.T  
ext)+1;
```

```
for i:=1 to StrToInt(edtNjam.Text)  
do
```

```
begin
```

```
fgLoad.Cells[0,i]:=IntToStr(i);
```

```
fgPLN.Cells[i,0]:='Jam
```

```
'+IntToStr(i);
```

```
end;
```

```
end;
```

```
end;
```

```
procedure
```

```
TfrmInput.btnNextClick(Sender:  
TObject);
```

```
var input:TextFile;
```

```
NamaFile, Nama:string;
```

```
Pmin,Pmax,a2,a1,a0,Sh,Sc,Ramp,Load  
,Res:double;
```

```
i,j,Tup,Tdown,Tcold,InitSt,Ngen,Njam  
:integer;
```

```
begin
```

```
if btnNext.Caption='&Save' then
```

```
begin
```

```
if SaveDialog1.Execute then
```

```
begin
```

```
NamaFile:=SaveDialog1.FileName;
```

```
AssignFile(input, NamaFile+'.txt');
```

```
Rewrite(input);
```

```
Ngen:=StrToInt(edtNgen.Text);
```

```
Njam:=StrToInt(edtNjam.Text);
```

```
Writeln(input, Ngen);
```

```
Writeln(input, Njam);
```

```
for i:=1 to Ngen do
```

```
begin
```

```
Nama:=fgGen.Cells[1,i];
```

```
Pmax:=StrToFloat(fgGen.Cells[2,i]);
```

```
Pmin:=StrToFloat(fgGen.Cells[3,i]);
```

```
a0:=StrToFloat(fgGen.Cells[4,i]);
```

```
a1:=StrToFloat(fgGen.Cells[5,i]);
```

```
a2:=StrToFloat(fgGen.Cells[6,i]);
```

```
Tup:=StrToInt(fgGen.Cells[7,i]);
```

```
Tdown:=StrToInt(fgGen.Cells[8,i]);
```

```
Sh:=StrToFloat(fgGen.Cells[9,i]);
```

```
Sc:=StrToFloat(fgGen.Cells[10,i]);
```

```
Tcold:=StrToInt(fgGen.Cells[11,i]);
```

```
InitSt:=StrToInt(fgGen.Cells[12,i]);
```

```
Ramp:=StrToFloat(fgGen.Cells[13,i]);
```

```
Writeln(input, Pmax:7:0, '
```

```
', Pmin:7:0, ',
```

```
a0:9:4, ', a1:9:4, ', a2:9:5, ', Tup, '
```

```
', Tdown, ',
```

```
Sh:7:0, ', Sc:7:0, ', tcold, ', InitSt, '
```

```
', Ramp:7:0, ', Nama);
```

```
end;
```

```
for i:=1 to Njam do
```

```
begin
```

```

Load:=StrToFloat(fgLoad.Cells[1,i]);

Res:=StrToFloat(fgLoad.Cells[2,i]);
  WriteLn(input,Load,Res);
end;
fgPLN.RowCount:=Ngen+1;
fgPLN.ColCount:=Njam+1;
for i:=1 to Ngen do
begin
  for j:=1 to Njam do
    Load:=StrToFloat(fgPLN.Cells[j,i]);
    Write(input,Load:7:2,');
  end;
  WriteLn(input,");
end;
CloseFile(input);
end;
else if btnNext.Caption='&Next' then
begin
  frmHasil.edtNparam.Text:=IntToStr(g
ObjFunc.Ngen);
  frmHasil.cmbJam.Items.Clear;
  for i:=1 to gObjFunc.Njam do
    begin
      frmHasil.cmbJam.AddItem(IntToStr(i)
,nil);
    end;
  frmHasil.cmbJam.Text:='1';
  frmHasil.Show;
end;
end;
end.

```

```

unit uGenerator;

interface

type
  TPembangkit=class
  private
    FNama:string;

    FPmax,FPmin,Fa2,Fa1,Fa0,FSh,FSc,F
    Daya,FRamp:double;

    FTup,FTdown,FTcold,FInitSt:integer;
    function GetAFLC:double;
  public
    constructor Create;overload;
    constructor Create(const
    rNama:string;
      const
    rPmin,rPmax,ra2,ra1,ra0,rSh,rSc,rRam
    p:double;
      const
    rTup,rTdown,rTcold,rInitSt:integer);ov
    erload;
    constructor Create(const
    rPembangkit:TPembangkit);overload;
    procedure Assign(const
    rPembangkit:TPembangkit);
    function GetBiaya(const
    rDaya:double):double;
    function GetDaya(const
    rLamda:double):double;
    function GetLamda(const
    rDaya:double):double;
    property Nama:string read FNama
    write FNama;
    property Pmax:double read FPmax
    write FPmax;
    property Pmin:double read FPmin
    write FPmin;
    property a2:double read Fa2 write
    Fa2;
    property a1:double read Fa1 write
    Fa1;
    property a0:double read Fa0 write
    Fa0;
    property Sh:double read FSh write
    FSh;

```

```

property Sc:double read FSc write
FSc;
property Ramp:double read FRamp
write FRamp;
property Tup:integer read FTup
write FTup;
property Tdown:integer read
FTdown write FTdown;
property Tcold:integer read FTcold
write FTcold;
property InitSt:integer read FInitSt
write FInitSt;
property Daya:double read FDaya
write FDaya;
property AFLC:double read
GetAFLC;
end;

```

```
TGenArr=array of TPembangkit;
```

implementation

```

//constructor
constructor TPembangkit.Create;
begin
inherited Create;
end;

constructor TPembangkit.Create(const
rNama:string;
const
rPmin,rPmax,ra2,ra1,ra0,rSh,rSc,rRam
p:double;
const
rTup,rTdown,rTcold,rInitSt:integer);
begin
inherited Create;
FNama:=rNama;
FPmin:=rPmin;
FPmax:=rPmax;
Fa2:=ra2;
Fa1:=ra1;
Fa0:=ra0;
FSh:=rSh;
FSc:=rSc;
FRamp:=rRamp;
FTup:=rTup;
FTdown:=rTdown;
FTcold:=rTcold;
FInitSt:=rInitSt;

```

```
end;
```

```

constructor TPembangkit.Create(const
rPembangkit:TPembangkit);
begin
inherited Create;
FNama:=rPembangkit.Nama;
FPmin:=rPembangkit.Pmin;
FPmax:=rPembangkit.Pmax;
Fa2:=rPembangkit.a2;
Fa1:=rPembangkit.a1;
Fa0:=rPembangkit.a0;
FSh:=rPembangkit.Sh;
FSc:=rPembangkit.Sc;
FRamp:=rPembangkit.Ramp;
FTup:=rPembangkit.Tup;
FTdown:=rPembangkit.Tdown;
FTcold:=rPembangkit.Tcold;
FInitSt:=rPembangkit.InitSt;
end;

```

function

```

TPembangkit.GetAFLC:double;
begin
Result:=fa0/fPmax+fa1+fa2*fPmax;
end;

```

```

procedure TPembangkit.Assign(const
rPembangkit:TPembangkit);
begin
FNama:=rPembangkit.Nama;
FPmin:=rPembangkit.Pmin;
FPmax:=rPembangkit.Pmax;
Fa2:=rPembangkit.a2;
Fa1:=rPembangkit.a1;
Fa0:=rPembangkit.a0;
FSh:=rPembangkit.Sh;
FSc:=rPembangkit.Sc;
FRamp:=rPembangkit.Ramp;
FTup:=rPembangkit.Tup;
FTdown:=rPembangkit.Tdown;
FTcold:=rPembangkit.Tcold;
FInitSt:=rPembangkit.InitSt;
end;

```

//data operation

```

function TPembangkit.GetBiaya(const
rDaya:double):double;
begin

```



```

result:=0;
if rDaya<>0 then
begin
result:=Fa2*sqr(rDaya)+Fa1*rDaya+Fa0;
end;
end;

```

```

function TPembangkit.GetDaya(const
rLamda:double):double;
begin
result:=(rLamda-Fa1)/(2*Fa2);
if result>FPmax then result:=FPmax;
if result<FPmin then result:=FPmin;
end;

```

```

function
TPembangkit.GetLamda(const
rDaya:double):double;
begin
result:=2*Fa2*rDaya+Fa1;
end;

```

end.

unit uObjFunc;

interface

uses
uUtils,uGenerator,SysUtils,uMatrix;

```

type
TObjFunc=class
private
FNgen,FNjam:integer;
FBeban,FRes,FAFLC:dArr1;
FSortAFLC:iArr1;
FPLN:dArr2;
FGen:TGenArr;
function GetBeban:dArr1;
function GetRes:dArr1;
function GetPLN:dArr2;
function GetGen:TGenArr;
procedure SetGen(const
rGen:TGenArr);
procedure SetBeban(const
rBeban:dArr1);

```

```

procedure SetRes(const rRes:dArr1);
procedure SetPLN(const
rPLN:dArr2);
function isON(const
rFlip:double):boolean;
function isServe(const
rJam:integer;const
rChrom:bArr1):boolean;
function isRampRate(const
rJam:integer;const
rPL:dArr2):boolean;
function FindAFLC(const
ri:integer):integer;
procedure RepairAFLC(var
rChrom:bArr2);
function CreateChromBase:bArr2;
function
CreateChromONOFF:bArr2;
function HitungEcoDis(const
rJam:integer;
const rChrom1:bArr1):dArr1;
function GetSortAFLC:iArr1;
function GetSortChrom(const
rRank:integer):bArr1;
function HitungCostGen(const
rPL:dArr2):dArr2;
function HitungCostSUC(const
rPL:dArr2):dArr2;
function GantiChrom(const
rChrom:bArr2):bArr2;
function GetSwap(const
rChrom:bArr2):bArr2;
function doCariGreyZone(const
rChrom:bArr2):bArr2;
public
constructor Create;overload;
constructor Create(const
rBeban,rRes:dArr1;
const rPLN:dArr2;
const
rGen:TGenArr);overload;
function getRandomChrom(const
rFlip:double):bArr2;
function getConstructSolution(const
rFlip:double):bArr2;
procedure setLocalSearch(var
rChrom:bArr2);
procedure doHitungChrom(const
rChrom:bArr2);

```

```

    var
rCostTotal:double);overload;
    procedure doHitungChrom(const
rChrom:bArr2;
    var rPL:dArr2;
    var rCostPerJam:dArr1;
    var
rCostTotal:double);overload;
    procedure doHitungPLN(
    var rCostPerJam:dArr1;
    var rCostTotal:double);
    procedure doHitungPL(const
rPL:dArr2;
    var rCostPerJam:dArr1;
    var rCostTotal:double);
    procedure doExecute(var
rChrom:bArr2;
    var rPL:dArr2;
    var rCostPerJam:dArr1;
    var rCostTotal:double);
    function HitungEcoDisGrad(const
rIterasi,rJam:integer;
    const rAlpha:double;
    const rChrom1:bArr1):dArr1;
    destructor Destroy;override;
    property Ngen:integer read FNgen
write FNgen;
    property Njam:integer read FNjam
write FNjam;
    property Gen:TGenArr read GetGen
write SetGen;
    property Beban:dArr1 read
GetBeban write SetBeban;
    property PLN:dArr2 read GetPLN
write SetPLN;
    property Res:dArr1 read GetRes
write SetRes;
    property SortAFLC:iArr1 read
GetSortAFLC;
end;

```

```
var gObjFunc:TObjFunc;
```

```
implementation
```

```

//constructor
constructor TObjFunc.Create;
begin
    inherited Create;
    FNgen:=0;

```

```

    FNjam:=0;
end;

```

```

constructor TObjFunc.Create(const
rBeban,rRes:dArr1;
    const rPLN:dArr2;
    const rGen:TGenArr);
var i,j,Ncek:integer;
begin
    inherited Create;
    FNgen:=high(rGen);
    FNjam:=high(rBeban);
    Ncek:=high(rRes);
    if FNjam<>Ncek then raise
Exception.Create('Dimensi matrik
tidak sama!');
    SetLength(FGen,FNgen+1);
    SetLength(FBeban,FNjam+1);
    SetLength(FRes,FNjam+1);
    SetLength(FAFLC,FNgen+1);
    for i:=1 to FNgen do
begin
        FGen[i]:=TPembangkit.Create(rGen[i]
);
        FAFLC[i]:=FGen[i].AFLC;
    end;
    for i:=1 to FNjam do
begin
        FBeban[i]:=rBeban[i];
        FRes[i]:=rRes[i];
    end;

SetLength(FPLN,FNgen+1,FNjam+1);
    for i:=1 to FNgen do
begin
        for j:=1 to FNjam do
begin
            FPLN[i,j]:=rPLN[i,j];
        end;
    end;
    FSortAFLC:=GetSortAFLC;
end;

```

```
//data accessing
```

```

function TObjFunc.GetBeban:dArr1;
var i:integer;
begin
    SetLength(result,FNjam+1);
    for i:=1 to FNjam do

```

```

begin
  result[i]:=FBeban[i];
end;
end;

function TObjFunc.GetRes:dArr1;
var i:integer;
begin
  SetLength(result, FNjam+1);
  for i:=1 to FNjam do
  begin
    result[i]:=FRes[i];
  end;
end;

function TObjFunc.GetPLN:dArr2;
var i,j:integer;
begin
  SetLength(result, FNgen+1, FNjam+1);
  for i:=1 to FNgen do
  begin
    for j:=1 to FNjam do
    begin
      result[i,j]:=FPLN[i,j];
    end;
  end;
end;

function TObjFunc.GetGen:TGenArr;
var i:integer;
begin
  SetLength(result, FNgen+1);
  for i:=1 to FNgen do
  begin

result[i]:=TPembangkit.Create(FGen[i]
);
  end;
end;

procedure TObjFunc.SetGen(const
rGen:TGenArr);
var i:integer;
begin
  FNgen:=high(rGen);
  SetLength(FGen, FNgen+1);
  SetLength(FAFLC, FNgen+1);
  for i:=1 to FNgen do

```

```

begin
  FGen[i]:=TPembangkit.Create(rGen[i]
);
  FAFLC[i]:=FGen[i].AFLC;
  end;
  FSortAFLC:=GetSortAFLC;
end;

procedure TObjFunc.SetBeban(const
rBeban:dArr1);
var i,Ncek:integer;
begin
  if FNjam>0 then
  begin
    Ncek:=high(rBeban);
    if FNjam<>Ncek then raise
Exception.Create('Dimensi matrik
tidak sama!');
  end
  else
  begin
    FNjam:=high(rBeban);
  end;
  SetLength(FBeban, FNjam+1);
  for i:=1 to FNjam do
  begin
    FBeban[i]:=rBeban[i];
  end;
end;

procedure TObjFunc.SetRes(const
rRes:dArr1);
var i,Ncek:integer;
begin
  if FNjam>0 then
  begin
    Ncek:=high(rRes);
    if FNjam<>Ncek then raise
Exception.Create('Dimensi matrik
tidak sama!');
  end
  else
  begin
    FNjam:=high(rRes);
  end;
  SetLength(FRes, FNjam+1);
  for i:=1 to FNjam do
  begin
    FRes[i]:=rRes[i];
  end;
end;

```

```

end;

procedure TObjFunc.SetPLN(const
rPLN:dArr2);
var i,j,Ncek:integer;
begin
  if FNgen>0 then
    begin
      Ncek:=high(rPLN);
      if FNgen<>Ncek then raise
Exception.Create('Dimensi matrik
tidak sama!');
    end
  else
    begin
      FNgen:=high(rPLN);
    end;
    if FNjam>0 then
      begin
        Ncek:=high(rPLN[0]);
        if FNjam<>Ncek then raise
Exception.Create('Dimensi matrik
tidak sama!');
      end
    else
      begin
        FNjam:=high(rPLN[0]);
      end;
    end;

SetLength(FPLN,FNgen+1,FNjam+1);
for i:=1 to FNgen do
  begin
    for j:=1 to FNjam do
      begin
        FPLN[i,j]:=rPLN[i,j];
      end;
    end;
  end;

//data processing
function TObjFunc.isON(const
rFlip:double):boolean;
begin
  result:=false;
  if random<=rFlip then result:=true;
end;

function TObjFunc.isServe(const
rJam:integer;const
rChrom:bArr1):boolean;

var i:integer;
load,sBebanMin,sBebanMax:double;
begin
  result:=true;
  sBebanMin:=0;
  sBebanMax:=0;
  for i:=1 to FNgen do
    begin
      if rChrom[i]=true then
        begin
          sBebanMin:=sBebanMin+FGen[i].Pmi
n;
          sBebanMax:=sBebanMax+FGen[i].Pm
ax;
        end;
      end;
      load:=FBeban[rJam]+FRes[rJam];
      if load<sBebanMin then result:=false;
      if load>sBebanMax then
        result:=false;
      end;

function TObjFunc.isRampRate(const
rJam:integer;const
rPL:dArr2):boolean;
var i:integer;
delta:double;
begin
  result:=true;
  for i:=1 to FNgen do
    begin
      if rJam>1 then
        begin
          delta:=rPL[i,rJam]-rPL[i,rJam-1];
          if delta>0 then
            begin
              if delta>FGen[i].Ramp then
                begin
                  result:=false;
                  break;
                end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

function
TObjFunc.HitungEcoDis(const
rJam:integer;
    const rChrom1:bArr1):dArr1;
var i,j:integer;
    Status:bArr1;

LoadCek,Pa,Pb,Lmd,LoadSplit:double;
aBeban,diffa2,diffa1,Cek,tes:double;
begin
SetLength(Status,FNgen+1);
for i:=1 to FNgen do
begin
    Status[i]:=rChrom1[i];
    FGen[i].Daya:=0;
end;
aBeban:=FBeban[rJam];
LoadCek:=aBeban;
LoadSplit:=aBeban;
for i:=1 to 15 do
begin
    Pa:=0;
    Pb:=0;
    for j:=1 to FNgen do
begin
    if Status[j] then
begin
        diffa2:=FGen[j].a2*2;
        diffa1:=FGen[j].a1;
        Pa:=Pa+1/diffa2;
        Pb:=Pb+diffa1/diffa2;
    end;
end;
if Pa<>0 then
begin
    Lmd:=(LoadSplit+Pb)/Pa;
end
else
begin
    Lmd:=LoadSplit+Pb;
end;
Cek:=0;
for j:=1 to FNgen do
begin
    if Status[j] then
begin
        diffa2:=2*FGen[j].a2;
        diffa1:=FGen[j].a1;
        FGen[j].Daya:=(Lmd-
diffa1)/diffa2;

        if FGen[j].Daya<FGen[j].Pmin
then
begin
            FGen[j].Daya:=FGen[j].Pmin;
end;
        if FGen[j].Daya>FGen[j].Pmax
then
begin
            FGen[j].Daya:=FGen[j].Pmax;
end;
        end;
        Cek:=Cek+FGen[j].Daya;
        end;
        tes:=LoadCek-Cek;
        if (tes<0.0001) and (tes>-0.0001)
then
begin
            break;
end
else if tes>0 then
begin
            for j:=1 to fNgen do
begin
                if Status[j] then
begin
                    if FGen[j].Daya=FGen[j].PMax
then
begin
                        Status[j]:=false;
                        LoadSplit:=LoadSplit-
FGen[j].Daya;
                        if LoadSplit<0 then
begin
                            LoadSplit:=LoadSplit+fGen[j].Daya;
                            Status[j]:=true;
                        end;
                    end;
                end;
            end;
        end;
    end;
end
else if tes<0 then
begin
            for j:=1 to fNgen do
begin
                if Status[j] then
begin
                    if FGen[j].Daya=FGen[j].Pmin
then
begin

```

```

    Status[j]:=false;
    LoadSplit:=LoadSplit-
FGen[j].Daya;
    if LoadSplit<0 then
        begin
LoadSplit:=LoadSplit+FGen[j].Daya;
            Status[j]:=true;
        end;
    end;
end;
end;
end;
end;
end;
end;
SetLength(result,FNgen+1);
for i:=1 to FNgen do
begin
    result[i]:=0;
    if rChrom1[i] then
        begin
            result[i]:=FGen[i].Daya;
        end;
    end;
end;
end;

```

```

function
TObjFunc.HitungEcoDisGrad(const
rIterasi,rJam:integer;
    const rAlpha:double;
    const rChrom1:bArr1):dArr1;

```

```

function CariChrom1(const
rJam:integer):bArr1;
var i:integer;
begin
    SetLength(result,FNgen+1);
    for i:=1 to FNgen do
        begin
            if FPLN[i,rJam]<>0 then
                begin
                    result[i]:=true;
                end
            else
                begin
                    result[i]:=false;
                end;
            end;
        end;
    end;
end;

```

```

function InitDaya(const rJam:integer;

```

```

    const rChrom1:bArr1;
    var rNgenON:integer):dArr1;
var i,sa:integer;
    sBebanMin,dBeban,tmp:double;
    Pgen:dArr1;
begin
    SetLength(Pgen,FNgen+1);
    sBebanMin:=0;
    rNgenON:=0;
    for i:=1 to FNgen do
        begin
            sa:=FSortAFLC[i];
            Pgen[sa]:=0;
            if rChrom1[sa]=true then
                begin
                    inc(rNgenON);
                    Pgen[sa]:=FGen[sa].Pmin;
sBebanMin:=sBebanMin+Pgen[sa];
                end;
            end;
            dBeban:=FBeban[rJam]-sBebanMin;
            for i:=1 to FNgen do
                begin
                    sa:=FSortAFLC[i];
                    if rChrom1[sa]=true then
                        begin
                            if dBeban>0 then
                                begin
                                    tmp:=Pgen[sa];
                                    Pgen[sa]:=Pgen[sa]+dBeban;
                                    if Pgen[sa]>FGen[sa].Pmax then
                                        begin
                                            Pgen[sa]:=FGen[sa].Pmax;
                                        end;
                                    dBeban:=dBeban-(Pgen[sa]-tmp);
                                end;
                            end;
                        end;
                    SetLength(result,rNgenON+1);
                    sa:=0;
                    for i:=1 to FNgen do
                        begin
                            if rChrom1[i]=true then
                                begin
                                    inc(sa);
                                    result[sa]:=Pgen[i];
                                end;
                            end;
                        end;
                    end;
                end;
            end;
end;

```

```

function CariAveLamda(const
rPgen:dArr1;
    const rChrom1:bArr1):double;
var i,sa:integer;
    sLamda:double;
begin
    sLamda:=0;
    sa:=0;
    for i:=1 to FNgen do
    begin
        if rChrom1[i]=true then
        begin
            inc(sa);
        end;
    end;
    sLamda:=sLamda+FGen[i].GetLamda(
rPgen[sa]);
    end;
    result:=sLamda/sa;
end;

```

```

function CariPgen(const
rChrom:bArr1;
    const rLamda:double):dArr1;
var i:integer;
begin
    SetLength(result,FNgen+2);
    for i:=1 to FNgen do
    begin
        if rChrom[i]=true then
        begin
            result[i]:=FGen[i].GetDaya(rLamda);
        end;
    end;
end;

```

```

function CaridL(const rJam:integer;
    const rChrom1:bArr1;
    const rPgen:dArr1;
    const rLamda:double):dArr1;
var i,sa:integer;

```

```

minLamda,maxLamda,sumPgen:doubl
e;
begin
    SetLength(result,high(rPgen)+2);
    sumPgen:=0;
    sa:=0;
    for i:=1 to FNgen do

```

```

begin
    if rChrom1[i]=true then
    begin
        inc(sa);
    end;
    minLamda:=FGen[i].GetLamda(FGen[
i].Pmin);
    maxLamda:=FGen[i].GetLamda(FGen
[i].Pmax);
    sumPgen:=sumPgen+rPgen[sa];
    if (rLamda<minLamda) or
(rLamda>maxLamda) then
        //if (maxLamda<rLamda) then
        begin
            result[sa]:=0;
        end
    else
        begin
            result[sa]:=FGen[i].GetLamda(rPgen[s
a])-rLamda;
        end;
    end;
    result[sa+1]:=FBeban[rJam]-
sumPgen;
end;

```

```

function CariMaxdL(const
rdL:dArr1):double;
var i:integer;
begin
    result:=rdL[1];
    for i:=2 to high(rdL) do
    begin
        if result<abs(rdL[i]) then
        begin
            result:=abs(rdL[i]);
        end;
    end;
end;

```

```

function MatrixHessian(const
rChrom1:bArr1):dArr2;
var i,j,sa,sb,NgenON:integer;
begin
    NgenON:=0;
    for i:=1 to FNgen do
    begin

```

```

if rChrom1[i]=true then
begin
inc(NgenON);
end;
end;

SetLength(result,NgenON+2,NgenON
+2);
sa:=0;
for i:=1 to FNgen do
begin
if rChrom1[i]=true then
begin
inc(sa);
sb:=0;
for j:=1 to FNgen do
begin
if rChrom1[j]=true then
begin
inc(sb);
if sb=sa then
begin
result[sa,sb]:=2*FGen[i].a2;
end
else
begin
result[sa,sb]:=0;
end;
end;
end;
end;
end;
sa:=0;
for i:=1 to FNgen do
begin
if rChrom1[i]=true then
begin
inc(sa);
result[NgenON+1,sa]:=-1;
result[sa,NgenON+1]:=-1;
end;
end;
result[NgenON+1,NgenON+1]:=0;
end;

function BalikKeSemula(const
rPgen:dArr1;
const rChrom1:bArr1):dArr1;
var i,sa:integer;
begin

```

```

SetLength(result,FNgen+1);
sa:=0;
for i:=1 to FNgen do
begin
if rChrom1[i]=true then
begin
inc(sa);
result[i]:=rPgen[sa];
end;
end;
end;

procedure UpdateLamda(const
rdL:dArr1;
const rChrom1:bArr1;
const rHessian:dArr2;
var rPgen:dArr1;
var rLamda:double);
var i,sa:integer;
invHessian:dArr2;
dX:dArr1;
begin
invHessian:=MatrixInvers(rHessian);

invHessian:=MatrixNegative(invHessi
an);
dX:=MatrixMul(invHessian,rdL);
sa:=0;
for i:=1 to FNgen do
begin
if rChrom1[i]=true then
begin
inc(sa);
rPgen[sa]:=rPgen[sa]+dX[sa];
if rPgen[sa]>FGen[i].Pmax then
rPgen[sa]:=FGen[i].Pmax;
if rPgen[sa]<FGen[i].Pmin then
rPgen[sa]:=FGen[i].Pmin;
end;
end;
rLamda:=rLamda+dX[FNgen+1];
end;

var i,NgenON:integer;
status:bArr1;
Hessian:dArr2;
Pgen,dL:dArr1;

sPmax,sPmin,Lamda,Alpha,error,max:
double;

```



```

begin
  SetLength(status,FNgen+1);
  for i:=1 to FNgen do
  begin
    status[i]:=rChrom1[i];
  end;
  sPmax:=0;
  sPmin:=0;
  for i:=1 to FNgen do
  begin
    if status[i]=true then
    begin
      sPmin:=sPmin+FGen[i].Pmin;
      sPmax:=sPmax+FGen[i].Pmax;
    end;
  end;
  SetLength(result,FNgen+1);
  if sPmin>FBeban[rJam] then
  begin
    for i:=1 to FNgen do
    begin
      if status[i]=true then
      begin
        result[i]:=FGen[i].Pmin;
      end;
    end;
  end
  else if sPmax<FBeban[rJam] then
  begin
    for i:=1 to FNgen do
    begin
      if status[i]=true then
      begin
        result[i]:=FGen[i].Pmax;
      end;
    end;
  end
  else
  begin
    error:=rAlpha;
  end;
Pgen:=InitDaya(rJam,rChrom1,NgenO
N);
Lamda:=CariAveLamda(Pgen,rChrom
1);
  for i:=1 to rIterasi do
  begin
    dL:=CaridL(rJam,rChrom1,Pgen,Lamd
a);

```

```

    max:=cariMaxdL(dL);
    if max<error then break;

    Hessian:=MatrixHessian(rChrom1);

    UpdateLamda(dL,rChrom1,Hessian,Pg
en,Lamda);
    end;

    result:=BalikKeSemula(Pgen,rChrom1
);
    end;
  end;

  function
  TObjFunc.HitungCostGen(const
  rPL:dArr2):dArr2;
  var i,j:integer;
  begin

    SetLength(result,FNgen+1,FNjam+1);
    for i:=1 to FNgen do
    begin
      for j:=1 to FNjam do
      begin

        result[i,j]:=FGen[i].GetBiaya(rPL[i,j]);
      end;
    end;
  end;

  function
  TObjFunc.HitungCostSUC(const
  rPL:dArr2):dArr2;
  var i,j,init,tcold:integer;
  begin

    SetLength(result,FNgen+1,FNjam+1);
    for i:=1 to FNgen do
    begin
      init:=FGen[i].InitSt;

      tcold:=FGen[i].Tdown+FGen[i].Tcold;
      for j:=1 to FNjam do
      begin
        result[i,j]:=0;
        if rPL[i,j]<>0 then
        begin
          if init>0 then
          begin

```

```

    init:=init+1;
end
else if init<0 then
begin
    if abs(init)<=tcold then
    begin
        result[i,j]:=FGen[i].Sh;
    end
    else
    begin
        result[i,j]:=FGen[i].Sc;
    end;
    init:=1;
end;
end
else if rPL[i,j]=0 then
begin
    if init>0 then
    begin
        init:=-1;
    end
    else if init<0 then
    begin
        init:=init-1;
    end;
end;
end;
end;
end;

procedure
TObjFunc.doHitungChrom(const
rChrom:bArr2;
    var rCostTotal:double);
var i,j:integer;
    PLa:dArr1;
    PL,CostGen,CostSUC:dArr2;
    chrom1:bArr1;
begin
    SetLength(PL,FNgen+1,FNjam+1);
    SetLength(chrom1,FNgen+1);
    SetLength(PLa,FNgen+1);
    for i:=1 to FNjam do
    begin
        for j:=1 to FNgen do
        begin
            chrom1[j]:=rChrom[j,i];
        end;
        PLa:=HitungEcoDis(i,chrom1);
        for j:=1 to FNgen do

```

```

begin
    PL[j,i]:=PLa[j];
end;
end;
CostGen:=HitungCostGen(PL);
CostSUC:=HitungCostSUC(PL);
rCostTotal:=0;
for i:=1 to FNjam do
begin
    for j:=1 to FNgen do
    begin
        rCostTotal:=rCostTotal+CostGen[j,i]+
        CostSUC[j,i];
    end;
end;
end;

procedure
TObjFunc.doHitungChrom(const
rChrom:bArr2;
    var rPL:dArr2;
    var rCostPerJam:dArr1;
    var rCostTotal:double);
var i,j:integer;
    PLa:dArr1;
    CostGen,CostSUC:dArr2;
    chrom1:bArr1;
begin
    SetLength(rPL,FNgen+1,FNjam+1);
    SetLength(rCostPerJam,FNjam+1);
    SetLength(chrom1,FNgen+1);
    SetLength(PLa,FNgen+1);
    for i:=1 to FNjam do
    begin
        for j:=1 to FNgen do
        begin
            chrom1[j]:=rChrom[j,i];
        end;
        PLa:=HitungEcoDis(i,chrom1);
        for j:=1 to FNgen do
        begin
            rPL[j,i]:=PLa[j];
        end;
    end;
    CostGen:=HitungCostGen(rPL);
    CostSUC:=HitungCostSUC(rPL);
    SetLength(rCostPerJam,FNjam+1);
    rCostTotal:=0;
    for i:=1 to FNjam do

```

```

begin
  rCostPerJam[i]:=0;
  for j:=1 to FNgen do
  begin
    rCostPerJam[i]:=rCostPerJam[i]+Cost
    Gen[j,i]+CostSUC[j,i];
  end;

  rCostTotal:=rCostTotal+rCostPerJam[i
  ];
  end;
end;

```

```

procedure
TObjFunc.doHitungPL(const
rPL:dArr2;
  var rCostPerJam:dArr1;
  var rCostTotal:double);
var i,j:integer;
  CostGen,CostSUC:dArr2;
begin
  SetLength(rCostPerJam,FNjam+1);
  CostGen:=HitungCostGen(rPL);
  CostSUC:=HitungCostSUC(rPL);
  rCostTotal:=0;
  for i:=1 to FNjam do
  begin
    rCostPerJam[i]:=0;
    for j:=1 to FNgen do
    begin
      rCostPerJam[i]:=rCostPerJam[i]+Cost
      Gen[j,i]+CostSUC[j,i];
    end;

    rCostTotal:=rCostTotal+rCostPerJam[i
    ];
  end;
end;

```

```

procedure
TObjFunc.doHitungPLN(var
rCostPerJam:dArr1;
  var rCostTotal:double);
var i,j:integer;
  CostGen,CostSUC:dArr2;
begin
  SetLength(rCostPerJam,FNjam+1);
  CostGen:=HitungCostGen(FPLN);

```

```

  CostSUC:=HitungCostSUC(FPLN);
  rCostTotal:=0;
  for i:=1 to FNjam do
  begin
    rCostPerJam[i]:=0;
    for j:=1 to FNgen do
    begin
      rCostPerJam[i]:=rCostPerJam[i]+Cost
      Gen[j,i]+CostSUC[j,i];
    end;

    rCostTotal:=rCostTotal+rCostPerJam[i
    ];
  end;
end;

```

```

rCostTotal:=rCostTotal+rCostPerJam[i
];
end;
end;

```

```

//data output
function
TObjFunc.GetSortAFLC:iArr1;
var i,j,tmp:integer;
  tmpAFLC:double;
begin
  SetLength(result,FNgen+1);
  for i:=1 to FNgen do
  begin
    result[i]:=i;
  end;
  for i:=1 to fNgen-1 do
  begin
    for j:=i to fNgen do
    begin
      if FAFLC[i]>FAFLC[j] then
      begin
        tmpAFLC:=FAFLC[i];
        FAFLC[i]:=FAFLC[j];
        FAFLC[j]:=tmpAFLC;
        tmp:=result[i];
        result[i]:=result[j];
        result[j]:=tmp;
      end;
    end;
  end;
  for i:=1 to fNgen do
  begin
    fAFLC[i]:=fGen[i].AFLC;
  end;
end;
end;

```

```

function
TObjFunc.GetSortChrom(const
rRank:integer):bArr1;
var i:integer;
begin
  SetLength(result,FNgen+1);
  for i:=1 to FNgen do
  begin
    result[i]:=false;
  end;
  for i:=1 to rRank do
  begin
    result[FSortAFLC[i]]:=true;
  end;
end;

```

```

function TObjFunc.FindAFLC(const
ri:integer):integer;
var i:integer;
begin
  result:=1;
  for i:=1 to FNgen do
  begin
    if FSortAFLC[i]=ri then
    begin
      result:=i;
      break;
    end;
  end;
end;

```

```

procedure TObjFunc.RepairAFLC(var
rChrom:bArr2);
var i,j,pos,k:integer;
begin
  for i:=1 to FNjam do
  begin
    for j:=1 to Ngen do
    begin
      if rChrom[j,i]=true then
      begin
        pos:=FindAFLC(j);
        for k:=1 to pos do
        begin
          if
rChrom[FSortAFLC[k],i]=false then
            begin
              rChrom[FSortAFLC[k],i]:=true;
              rChrom[j,i]:=false;

```

```

            break;
          end;
        end;
      end;
    end;
  end;
end;

```

```

function
TObjFunc.CreateChromBase:bArr2;
var i,j,k:integer;
    chrom1:bArr1;
    serve:boolean;
begin

```

```

  SetLength(result,FNgen+1,FNjam+1);
  SetLength(chrom1,FNgen+1);
  for i:=1 to FNjam do
  begin
    for j:=1 to FNgen do
    begin
      chrom1:=GetSortChrom(j);
      serve:=isServe(i,chrom1);
      if serve=true then
      begin
        for k:=1 to FNgen do
        begin
          result[k,i]:=chrom1[k];
        end;
        break;
      end;
    end;
  end;
end;

```

```

function
TObjFunc.CreateChromONOFF:bArr2
;
var i,j,k:integer;
    chrom1:bArr1;
    serve:boolean;
begin

```

```

  SetLength(result,FNgen+1,FNjam+1);
  SetLength(chrom1,FNgen+1);
  for i:=1 to FNjam do
  begin
    for j:=FNgen downto 1 do
    begin
      chrom1:=GetSortChrom(j);

```

```

serve:=isServe(i,chrom1);
if serve=true then
begin
  for k:=1 to FNgen do
  begin
    result[k,i]:=chrom1[k];
  end;
  break;
end;
end;
end;
end;

function
TObjFunc.getRandomChrom(const
rFlip:double):bArr2;
var i,j:integer;
    chromBase,chromONOFF:bArr2;
begin
  chromBase:=CreateChromBase;

chromONOFF:=CreateChromONOFF;
SetLength(result,Ngen+1,Njam+1);
for i:=1 to Ngen do
begin
  for j:=1 to Njam do
  begin
    if chromBase[i,j]=true then
    begin
      result[i,j]:=true;
    end
    else
    begin
      if chromONOFF[i,j]=true then
      begin
        if isON(rFlip)=true then
        begin
          result[i,j]:=true;
        end
        else
        begin
          result[i,j]:=false;
        end;
      end;
    end;
  end;
end;
end;
end;
end;
RepairAFLC(result);
result:=GetSwap(result);
end;

```

```

function
TObjFunc.getConstructSolution(const
rFlip:double):bArr2;
var i,j:integer;
    chromBase,chromONOFF:bArr2;
begin
  chromBase:=CreateChromBase;

chromONOFF:=CreateChromONOFF;
SetLength(result,Ngen+1,Njam+1);
for i:=1 to Ngen do
begin
  for j:=1 to Njam do
  begin
    if chromBase[i,j]=true then
    begin
      result[i,j]:=true;
    end
    else
    begin
      if chromONOFF[i,j]=true then
      begin
        if isON(rFlip)=true then
        begin
          result[i,j]:=true;
        end
        else
        begin
          result[i,j]:=false;
        end;
      end;
    end;
  end;
end;
end;
end;
RepairAFLC(result);
result:=GetSwap(result);
end;

procedure
TObjFunc.setLocalSearch(var
rChrom:bArr2);
var i,j:integer;
    CostBest,CostCek:double;
    greyChrom:bArr2;
begin
  doHitungChrom(rChrom,CostBest);

greyChrom:=doCariGreyZone(rChrom
);
for i:=1 to FNgen do

```

```

begin
  for j:=1 to FNjam do
  begin
    if greyChrom[i,j]=true then
    begin
      rChrom[i,j-1]:=true;
doHitungChrom(rChrom, CostCek);
      if CostBest>CostCek then
      begin
        CostBest:=CostCek;
      end
      else
      begin
        rChrom[i,j-1]:=false;
      end;
    end;
  end;
end;
end;

```

```

function TObjFunc.GetSwap(const
rChrom:bArr2):bArr2;
var i,j,k,init,pos:integer;
begin

```

```

SetLength(result,FNgen+1,FNjam+1);
for i:=1 to FNgen do
begin
  init:=FGen[i].InitSt;
  for j:=1 to FNjam do
  begin
    result[i,j]:=rChrom[i,j];
    if result[i,j]=true then
    begin
      if init<0 then
      begin
        if abs(init)>=FGen[i].Tdown
then
        begin
          init:=1;
        end
        else
        begin
          pos:=j+init;
          if pos<1 then
          begin
            pos:=1;
          end;
          for k:=pos to j-1 do

```

```

begin
  result[i,k]:=true;
end;
end;
end;
else if init>0 then
begin
  init:=init+1;
end;
end
else if result[i,j]=false then
begin
  if init<0 then
  begin
    init:=init-1;
  end
  else if init>0 then
  begin
    if init>=FGen[i].Tup then
    begin
      init:=-1;
    end
    else
    begin
      result[i,j]:=true;
      init:=init+1;
    end;
  end;
end;
end;
end;
end;
end;
end;
end;
end;

```

```

//destructor
destructor TObjFunc.Destroy;
var i:integer;
begin
  try
    for i:=1 to FNgen do
    begin
      FGen[i].Free;
    end;
  finally
    inherited Destroy;
  end;
end;
end;

```

```

function TObjFunc.GantiChrom(const
rChrom:bArr2):bArr2;
var i,j:integer;

```

```

begin
SetLength(result,FNgen+1,FNjam+1);
for i:=1 to FNgen do
begin
for j:=1 to FNjam do
begin
result[i,j]:=rChrom[i,j];
end;
end;
end;
end;

```

```

function
TObjFunc.doCariGreyZone(const
rChrom:bArr2):bArr2;
var i,j,init,tcold:integer;
begin

```

```

SetLength(result,FNgen+1,FNjam+1);
for i:=1 to FNgen do
begin
for j:=1 to FNjam do
begin
result[i,j]:=false;
end;
end;
for i:=1 to FNgen do
begin
init:=FGen[i].InitSt;

tcold:=FGen[i].Tdown+FGen[i].Tcold;
for j:=1 to FNjam do
begin
if rChrom[i,j]=true then
begin
if init<0 then
begin
if abs(init)=(tcold+1) then
begin
result[i,j]:=true;
end;
init:=1;
end
else if init>0 then
begin
init:=init+1;
end;
end
else if rChrom[i,j]=false then
begin

```

```

if init<0 then
begin
init:=init-1;
end
else if init>0 then
begin
init:=-1;
end;
end;
end;
end;
end;
end;

```

```

procedure TObjFunc.doExecute(var
rChrom:bArr2;
var rPL:dArr2;
var rCostPerJam:dArr1;
var rCostTotal:double);
var i,j:integer;
CostBest,CostCek:double;
chromBase,greychrom:bArr2;
begin
chromBase:=CreateChromBase;
rChrom:=GetSwap(chromBase);
doHitungChrom(rChrom,CostBest);

greychrom:=doCariGreyZone(rChrom
);
for i:=1 to FNgen do
begin
for j:=1 to FNjam do
begin
if greychrom[i,j]=true then
begin
rChrom[i,j-1]:=true;

doHitungChrom(rChrom,CostCek);
if CostBest>CostCek then
begin
CostBest:=CostCek;
end
else
begin
rChrom[i,j-1]:=false;
end;
end;
end;
end;
end;
end;

```