

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

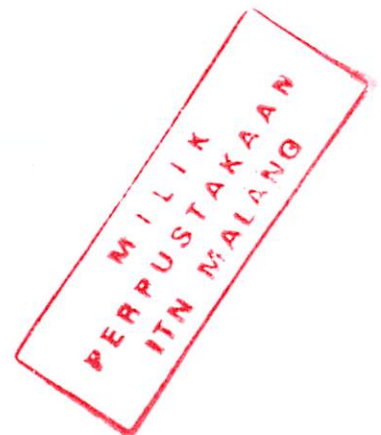


**ANALISA *ECONOMIC LOAD DISPATCH* DENGAN
METODE JARINGAN SYARAF *ADAPTIVE HOPFIELD***

SKRIPSI

**Disusun Oleh :
JAINAL MUHAJIRIN
NIM: 99.12.065**

APRIL 2005



LEMBAR PERSETUJUAN

**ANALISA *ECONOMIC LOAD DISPATCH* DENGAN
METODE JARINGAN SYARAF *ADAPTIVE HOPFIELD***


SKRIPSI

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

Disusun oleh

**JAINAL MUHAJIRIN
NIM: 99.12.065**

**Mengetahui,
Ketua Jurusan Teknik Elektro**



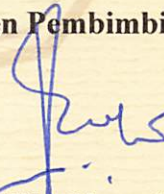
Ir. F. Yudi Limpraptono, MT
NIP. Y. 103 950 0274

**Diperiksa dan Disetujui,
Dosen Pembimbing I**



Ir. Choirul Saleh, MT
NIP. P. 101 808 0190

Dosen Pembimbing II



Ir. Eko Nurcahyo
NIP. 102 870 0172

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

Lembar Persembahan

Alhamdulillahirobbilalamin

Puji syukur kehadiran Allah Yang Maha Segalanya, dengan berkah dan rahmat-Mu akhirnya aku menyelesaikan skripsi ini dengan sebaik-baik petunjuk-Mu, Yaa Allah meskipun dalam penyelesaian skripsiku berliku-liku penuh ini itu tapi aku tau semua itu adalah jalanku menuju petunjuk-Mu dimana cahaya kebaikan menantiku, Yaa Allah limpahkan rahmat dan keselamatan yang sempurna kepada junjungan kami Nabi Muhammad S.A.W karena beliau-lah teladan kehidupanku. Terimakasih Yaa Allaaaahhhhhhh.....



Buat Ibu dan Bapak

Aku tidak tau harus berkata apa lagi, mungkin..... aku tidak akan dapat selesai jika tidak ada Ibu'-Bapak, terimakasih Bu'-terimakasih Pa' atas do'a dan segalanya, dengan apa aku harus membalas kebaikan Ibu'-Bapak.....???? Ibu'-Bapak adalah segalanya bagiku, mungkin untuk seumur hidupku aku tidak akan sanggup membalas kebaikan ibu'-Bapak, sekali lagi terima kasih Bu'-terima kasih Pa', sembah sujud (Yaa Allah) kudoakan untuk Ibu'-Bapakku.....

Buat Samberata

Untuk **Mbak Supik, Mbak Yuliani, Mbak Ratus, dan Mas arif,**

Mbak - Mas makasih yo do'ane, sekarang adekmu sudah lulus, sudah jadi sarjana, semua ini tak lepas dari bantuan Mbak - Mas kabeh, poko'e aku suwon akeh, juga tak lupa saya ucapkan terimakasih buat **Mbak Musihah** atas dukungannya dan do'anya

Buat Someone.....????

Makasih telah membantuku dan menyayangiku, karena dirimulah aku termotivasi menyelesaikan skripsiku, tapi... maafkan jika ta' kusebut namamu, tapi kuyakinkan aku akan selalu untukmu, selamanya.....

Untuk Adekku semua, **Imam, Dayat, Heni, dina** dan **Fairus** Makasih dukungane yo, tapi ko' ae yo sangune kalau aku sudah kerja, oke.....

Sahabat baekku, **Teo, Ipung, Sholeh, Kitem, hada', Toni, Evi, Khusnul, Nil, Ita, Indah, Yuyun** dan **Niken**

Buat temen-temen semua, **Joko, Rahmad, Farid (Saimo), Aris (Chontonk), Yudi, Ayu', Arif, Fariadi, Ubetz, Sholeh, Bowo, Gandhi, Vicky, Mehdy, Oky, Irul, Vidya, Rully, Edi (Mbah)** dan semuanya yang tak dapat ku sebutkan satu-persatu, makasih rek yo dukungane ambek sembarange, suwon-suwon akeh.....

..... **Pak Choirul** dan **Pak Eko**

Makasih pak ya bimbingannya, bapak memang terbaik, disaat saya mengalami kesusahan dan keadaan tertekan bapaklah bapakku di kampus yang dalam keadaan sesibuk apapun bapak masih menyempatkan waktu membimbing dan membantu saya, Makasih banyak pak ya.....

..... **Bu Ida**

Makasih bu selama ini sudah menjadi dosen waliku yang banyak membantuku untuk menyelesaikan segala hal di kampus, ibu adalah ibuku di kampus

..... **Mas Jayeng**

Mas makasih banyak yo... wis ngurusno sembarange di jurusan

Dan untuk semuanya yang telah membantu saya, saya ucapkan banyak terimakasih, semoga Allah membalas kebaikan semuanya, amin.....

ABSTRAKSI

ANALISA *ECONOMIC LOAD DISPATCH* DENGAN METODE JARINGAN SYARAF *ADAPTIVE HOPFIELD*

(Jainal Muhajirin, NIM. 9912065, T. Elektro E.L S-1, 58 Hal, 2005)

(Dosen Pembimbing I: Ir. Choirul Saleh, MT, Dosen pembimbing II: Ir. Eko Nurcahyo)

Kata Kunci: *Economic Load Dispatch, Jaringan Syaraf Adaptive hopfield*

Dalam skripsi ini membahas masalah optimasi untuk biaya bahan bakar pada suatu sistem tenaga listrik yang hasil dari analisis tersebut nantinya dapat digunakan sebagai salah satu acuan dalam mengefisiensikan biaya bahan bakar yang bisa digunakan oleh PT. Pembangkitan Jawa-Bali sehingga nilai kerugian yang sementara ini dimiliki oleh PT. PJB bisa berkurang, yang akhirnya nanti akan bisa menambah keuntungan bagi PT. PJB sebagai perusahaan penyedia energi listrik di Indonesia. Untuk penyelesaian masalah *Economic Dispatch* adalah menggunakan metode Jaringan Syaraf *Adaptive Hopfield*, program ini dipilih untuk menyelesaikan sejumlah kendala pertidaksamaan dan persamaan serta unit dengan beragam fungsi biaya bahan bakar. Program ini pada saat bersamaan, menghasilkan suatu rancangan operasi yang memenuhi persyaratan pengoperasian sistem tenaga listrik. Persyaratan pengoperasian sistem tenaga listrik yang terutama adalah daya yang dibangkitkan cukup memasok beban konsumen. Dengan menggunakan metode Jaringan Syaraf *Adaptive hopfield* ternyata dapat melakukan optimasi yang lebih murah dibandingkan dengan optimasi PT.PJB dengan metode ini biaya operasional Pada tanggal 10 Maret 2004 selisih biaya total PT. PLN PJB dengan metode Jaringan Syaraf *Adaptive Hopfield* sebesar Rp. 1.478.931.750 dioptimalkan 8.50%. Pada tanggal 13 Maret 2004 selisih biaya total PT. PJB dengan Jaringan Syaraf *Adaptive Hopfield* sebesar Rp. 1.139.345.889 dapat dioptimalkan 7.25%. Pada tanggal 14 Maret 2004 metode Jaringan Syaraf *Adaptive Hopfield* sebesar Rp. 714.070.092 dapat dioptimalkan sebesar 4.86%.

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT. Atas Karunia dan Hidayah-Nya skripsi ini dapat terselesaikan dengan baik, guna memenuhi persyaratan untuk mendapat gelar Sarjana Teknik pada Jurusan Teknik Elektro Konsentrasi Energi Listrik, Fakultas Teknologi Nasional Malang.

Penyusun menyampaikan terima kasih kepada pihak yang telah membantu, membimbing dan mengarahkan mulai dari awal hingga terselesainya skripsi ini. Ucapan terima kasih saya ucapkan kepada yang terhormat:

1. Bapak Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Muchtar Asroni, MSME, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro, Institut Teknologi Nasional Malang.
4. Bapak Ir. Choirul Saleh, MT, selaku dosen pembimbing I, dan bapak Ir. Eko Nurcahyo, selaku dosen pembimbing II yang memberikan masukan, dorongan semangat dan bantuannya.
5. Rekan-rekan Elektro khususnya angkatan 1999 atas segala kebersamaanya selama ini.
6. Serta semua pihak yang turut membantu dalam penyelesaian skripsi ini.

Kami menyadari bahwa skripsi ini masih jauh dari sempurna, oleh sebab itu kritik dan saran akan kami terima dengan senang hati.

Akhir kata penulis berharap semoga skripsi ini dapat bermanfaat bagi kita semua dalam memperkaya ilmu pengetahuan khususnya bagi mahasiswa pada Jurusan Teknik Elektro Energi Listrik.

Malang, April 2005

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR GRAFIK	xii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	4
1.7 Kontribusi.....	5
BAB II TEORI DASAR	
2.1 Sistem Tenaga Listrik.....	6
2.2 Operasi Sistem Tenaga Listrik	7
2.3 Karakteristik Pembangkit.....	10

2.3.1	Karakteristik <i>Input-Output</i>	10
2.3.2	Karakteristik <i>Hate Rate</i>	13
2.3.3	Karakteristik <i>Incremental Hate Rate</i> dan <i>Incremental Fuel Cost</i>	14
2.4	<i>Economic Dispatch</i>	15
2.4.1	Penyelesaian <i>Economic Dispatch</i> dengan Metode <i>Pengali La Grange</i>	16
2.4.2	Penyelesaian <i>Economic Dispatch</i> dengan Metode <i>Iterasi Lamda</i>	19
2.5	Fungsi Biaya Bahan Bakar	20
2.6	<i>Economic Dispatch</i> Dengan Mengabaikan Rugi-Rugi Transmisi	20

BAB III APLIKASI METODE JARINGAN SYARAF *ADAPTIVE HOPFIELD* UNTUK *ECONOMIC DISPATCH*

3.1	Penyelesaian Dengan Jaringan <i>Adaptive Hopfield</i>	24
3.1.1.	Jaringan <i>Adaptive Hopfield</i>	24
3.1.2.	Metode Pengaturan Slope	26
3.1.3.	Momentum	28
3.2	Algoritma <i>Adaptive Hopfield</i>	29
3.3	Flow Chart <i>Adaptive Hopfield</i>	30

BAB IV	ANALISA DATA MENGGUNAKAN METODE	
	JARINGAN SYARAF <i>ADAPTIVE HOPFIELD</i>	
4.1	Pendahuluan	31
4.2	Program Komputer Economic Dispatch Menggunakan Metode Jaringan Syaraf <i>Adaptive Hopfield</i>	31
4.3	Data Pembangkit Thermal.....	32
4.4	Aplikasi <i>Adaptive Hopfield</i> di PT. Pembangkitan Jawa-Bali	34
4.5	Beban Sistem.....	35
4.6	Hasil Perhitungan Dan Analisa Data.....	39
4.6.1	Hasil Perhitungan PT. PLN PJB	39
4.6.2	Tampilan Program Komputer Dan Hasil Perhitungan Dengan Metode Jaringan syaraf <i>Adaptive hopfield</i>	43
4.6.2.1	Hasil Perhitungan Menggunakan Metode Jaringan syaraf <i>Adaptive hopfield</i>	48
4.6.3	Perbandingan Hasil Perhitungan PT. PLN PJB Dengan Metode Jaringan Syaraf <i>Adaptive Hopfield</i> .	51
BAB V	KESIMPULAN	
5.1.	Kesimpulan.....	56
5.2.	Saran.....	57
	DAFTAR PUSTAKA	58
	LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1.	Unit <i>Boiler-Turbin-Generator</i>	11
Gambar 2.2.	Kurva Karakteristik <i>Input-Output</i> Pembangkit Thermal.....	13
Gambar 2.3.	Kurva Karakteristi <i>Hate Rate</i> Unit Pembangkit.....	14
Gambar 2.4.	Kurva Karakteristik <i>Incremental Fuel Cost Rate</i>	15
Gambar 2.5.	N Unit Melayani Beban P_R	16
Gambar 2.6.	Grafik Penyelesaian Dengan Metode <i>Iterai Lamda</i>	19
Gambar 2.7.	N Unit Pembangkit Thermal Melayani Beban P_{R2}	21
Gambar3.1.	Fungsi Perubahan Sigmoidal dengan Perbedaan Nilai dari Pendekatan Parameter	26
Gambar 4.1.	Tampilan Program Utama	43
Gambar 4.2.	Menampilkan <i>Input</i> Data pembangkit.....	44
Gambar 4.3.	Tampilan Parameter <i>Hopfield</i>	44
Gambar 4.4.	Tampilan Data Pembangkit.....	45
Gambar 4.5.	Tampilan data Pembebanan.....	45
Gambar 4.6.	Tampilan Data PT. PLN PJB	46
Gambar 4.6.	Hasil Optimasi Menggunakan Metode jaringan syaraf <i>Adaptive Hopfield</i>	47

DAFTAR TABEL

Tabel 4.1.	Data Unit Thermal pada PT. PLN Pembangkita Jawa-Bali	33
Tabel 4.2.	Unit Pembangkit Thermal yang Siap Operasi	34
Tabel 4.3.	Data Beban Unit Thermal pada PT. PLN PJB 10 Maret 2004....	36
Tabel 4.4.	Data Beban Unit Thermal pada PT. PLN PJB 13 Maret 2004....	37
Tabel 4.5.	Data Beban Unit Thermal pada PT. PLN PJB 14 Maret 2004....	38
Tabel 4.6.	Hasil Perhitungan Biaya Operasional Perjam PT. PLN PJB 10 Maret 2004	40
Tabel 4.7.	Hasil Perhitungan Biaya Operasional Perjam PT. PLN PJB 13 Maret 2004	41
Tabel 4.8.	Hasil Perhitungan Biaya Operasional Perjam PT. PLN PJB 14 Maret 2004	42
Tabel 4.9.	Hasil Perhitungan Menggunakan Metode jaringan Syaraf <i>Adaptive Hopfield</i> 10 Maret 2004	48
Tabel 4.10.	Hasil Perhitungan Menggunakan Metode jaringan Syaraf <i>Adaptive Hopfield</i> 13 Maret 2004	49
Tabel 4.11.	Hasil Perhitungan Menggunakan Metode jaringan Syaraf <i>Adaptive Hopfield</i> 14 Maret 2004	50
Tabel 4.12.	Perbandingan Biaya Operasional Perjam PT. PLN PJB Dan Metode Jaringan Syaraf <i>Adaptive Hopfield</i> . Rabu 10 Maret 2004	51

Tabel 4.13. Perbandingan Biaya Operasional Perjam PT. PLN PJB Dan Metode Jaringan Syaraf <i>Adaptive Hopfield</i> . Sabtu 13 Maret 2004	52
Tabel 4.14. Perbandingan Biaya Operasional Perjam PT. PLN PJB Dan Metode Jaringan Syaraf <i>Adaptive Hopfield</i> . Minggu 14 Maret 2004	53
Tabel 4.15. Perbandingan Total Biaya PT. PLN PJB Dengan Metode Jaringan Syaraf <i>Adaptive Hopfield</i>	53

DAFTAR GRAFIK

Grafik 4.1. Perbandingan Biaya Operasional Perjam PT. PLN PJB dan Metode Jaringan Syaraf <i>Adaptive Hopfield</i> Rabu 10 maret 2004	54
Grafik 4.2. Perbandingan Biaya Operasional Perjam PT. PLN PJB dan Metode Jaringan Syaraf <i>Adaptive Hopfield</i> Sabtu 13 maret 2004	54
Grafik 4.3. Perbandingan Biaya Operasional PT. PLN PJB dan Metode Jaringan Syaraf <i>Adaptive Hopfield</i> Minggu 14 maret 2004	55
Grafik 4.4. Perbandingan Total Biaya Operasional PT. PLN PJB Dengan Metode Jaringan Syaraf <i>Adaptive Hopfield</i>	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

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 dan rugi-rugi daya pada saluran transmisi. 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.

Koordinasi antara unit-unit pembangkit yang ada pada sistem tenaga listrik sangat diperlukan untuk mencapai biaya operasi yang seoptimum mungkin, dalam hal ini yang dimaksud adalah optimum secara ekonomis dengan tetap memperhatikan besar beban yang ada, sehingga dengan demikian nantinya akan

didapatkan nilai keuntungan pada PT. PLN (Persero) sebagai perusahaan penyedia energi listrik di Indonesia.

1.2. Permasalahan

Berdasarkan latar belakang diatas maka permasalahan yang ada adalah bagaimana mengoptimalkan biaya pembangkitan menggunakan *Economic Dispatch* yang dihasilkan pembangkit tenaga listrik thermal. Sehubungan dengan itu, maka judul skripsi ini adalah :

**“ ANALISA *ECONOMIC LOAD DISPATCH* DENGAN METODE
JARINGAN SYARAF *ADAPTIVE HOPFIELD* ”.**

1.3. Tujuan

Tujuan dari penulisan skripsi ini adalah memberikan analisa penerapan pembebanan ekonomis (*Economic Dispatch*) dengan Metode Jaringan Syaraf *Adaptive Hopfield* untuk optimasi biaya operasi pada suatu sistem tenaga listrik, yang mana dalam skripsi ini objek penelitian pada sistem tenaga listrik PT. PLN Pembangkit Jawa-Bali.

1.4. Batasan Masalah

Apabila ditinjau lebih jauh, permasalahan yang ada cukup luas dan kompleks. Oleh karena itu agar pembahasan lebih terarah tujuannya, maka perlu dilakukan pembatasan masalah :

1. Analisa perhitungan menggunakan metode Metode Jaringan Syaraf *Adaptive Hopfield* dengan mengabaikan Metode Pengaturan Bias

2. Hanya membahas unit pembangkit Thermal dan dibatasi pada pembangkit Thermal tenaga uap dan gas yang ada pada lingkungan kerja PT. PLN PJB.
3. Tidak membahas rugi-rugi saluran Transmisi.
4. Daya listrik yang dihasilkan setiap unit pembangkit harus disertai beberapa batasan (kapasitas minimum dan kapasitas maksimum).
5. Tidak membahas masalah Spinning Reserve.
6. Pembahasan di titik beratkan pada segi ekonomis daripada teknis.

1.5. Metodologi Penelitian

Metodologi yang dibahas dalam pembahasan skripsi ini dilakukan dengan langkah-langkah sebagai berikut :

1. Studi kepustakaan mengenai hal-hal yang berhubungan dengan pembahasan masalah.
2. Studi lapangan untuk mendapatkan data parameter unit thermal yang dibutuhkan dari obyek penelitian yaitu di PT. Pembangkit Jawa Bali yang diperlukan berpedoman pada teori yang diperoleh dari studi kepustakaan.
3. Perhitungan *Economic Dispatch* dengan metode Metode Jaringan Syaraf *Adaptive Hopfield*.
4. Membuat evaluasi, sehingga dapat disimpulkan dari perhitungan antara sebelum dan sesudah optimasi.

1.6. Sistematika Pembahasan

Penyusunan skripsi ini terbagi dalam beberapa sistematika bab pembahasan yang terdiri dari :

Bab I : Pendahuluan.

Dalam bab ini diuraikan latar belakang, rumusan masalah, tujuan masalah, batasan masalah, langkah pemecahan masalah dan sistematika pembahasan.

Bab II : Landasan Teori.

Dalam bab ini diuraikan tentang teori dasar Economic Dispatch, fungsi biaya bahan bakar, dan metode Metode Jaringan Syaraf *Adaptive Hopfield*.

Bab III : Aplikasi Metode *Jaringan Syaraf Adaptive Hopfield* pada Economic dispatch di PT. PLN Pembangkitan Jawa Bali.

Dalam bab ini diuraikan tentang data sistem PT. PLN PJB dan penggunaan Metode Jaringan Syaraf *Adaptive Hopfield* untuk menyelesaikan Economic Dispatch PT. PLN Pembangkitan Jawa Bali.

Bab IV : Hasil Simulasi Program.

Dalam bab ini diuraikan tentang seluruh hasil simulasi program untuk menyelesaikan masalah *Economic Dispatch* serta hasil perhitungan, analisa dan evaluasinya.

Bab V : Kesimpulan dan Saran.

1.7. Kontribusi

Adapun kontribusi dari tugas akhir ini adalah diharapkan langkah efisiensi bisa diambil oleh PT. PLN (Persero) sehingga nilai kerugian yang sementara ini diderita oleh PLN bisa berkurang, yang pada akhirnya nanti akan bisa menambah keuntungan bagi PLN sebagai perusahaan penyedia energi listrik di Indonesia.

BAB II

TEORI DASAR

2.1. Sistem Tenaga Listrik^[1]

Untuk keperluan penyediaan tenaga listrik bagi para pelanggan diperlukan berbagai peralatan listrik. Peralatan-peralatan listrik tersebut dihubungkan satu sama lain yang saling berhubungan dan secara keseluruhan membentuk suatu sistem tenaga listrik. Yang dimaksud dengan sistem tenaga listrik adalah satu kesatuan yang terintegrasi antara pembangkit tenaga listrik, Gardu induk (pusat beban) yang satu sama lain dihubungkan.

Pengelolaan sistem tenaga listrik merupakan persoalan yang cukup rumit, sehingga diperlukan suatu manajemen operasi yang baik. Manajemen operasi sistem tenaga listrik harus memikirkan bagaimana menyediakan tenaga listrik yang seekonomis mungkin dengan tetap memperhatikan mutu dan keandalan. Mutu dan keandalan diukur dengan frekuensi, tegangan dan jumlah gangguan. Masalah mutu tenaga listrik tidak semata-mata merupakan masalah operasi tenaga listrik tetapi erat kaitannya dengan pemeliharaan instalasi tenaga listrik dan juga pengembangan sistem tenaga listrik karena mengingat konsumsi tenaga listrik oleh pelanggan selalu bertambah dari waktu ke waktu oleh karena itu hasil-hasil operasi sistem tenaga listrik perlu di analisa dan dievaluasi untuk menjadi masukan bagi pemeliharaan instalasi serta pengembangan sistem tenaga listrik. Mutu tenaga listrik yang baik merupakan kendala/*counstraint* terhadap biaya

pengadaan tenaga listrik yang serendah mungkin, maka kompromi antar kedua hal yang merupakan masalah optimasi yang cukup kompleks.

Daya yang tersedia dalam sistem tenaga listrik tergantung kepada daya yang terpasang unit-unit pembangkit dan juga pada kesiapan operasi unit-unit tersebut. Berbagai faktor seperti gangguan kerusakan dan pemeliharaan rutin, menyebabkan unit pembangkit menjadi tidak siap operasi.

2.2. Sistem Operasi Pada Sistem Tenaga Listrik^[1]

Seperti telah diketahui bahwa dalam masalah pengaturan beban pada suatu operasi sistem tenaga listrik harus selalu dicapai suatu keadaan operasi yang bisa diandalkan dan cukup ekonomis.

Ada beberapa kinerja yang harus dilaksanakan untuk menjamin keandalan sistem operasi antara lain, pengaturan frekuensi dan tegangan sistem untuk berada pada harga normalnya karena adanya perubahan beban sistem. Dan seperti yang diketahui dan berulang kali disebutkan bahwa tenaga listrik tidak dapat disimpan sehingga dalam operasinya harus selalu dicapai keseimbangan antara penyediaan dengan pemenuhan kebutuhan daya serta perlu juga diingat bahwa sistem selalu berubah setiap saat. Maka sudah tentu jauh-jauh sebelumnya sudah harus diketahui atau diramalkan keadaan tersebut dengan tetap yaitu keadaan beban pada hari itu dari waktu ke waktu sampai selama 24 jam. Keadaan beban ini digambarkan sebagai kebutuhan daya sebagai fungsi dari waktu yang biasa disebut dengan lengkung beban harian. Lengkung beban harian ini adalah merupakan sesuatu yang sangat penting disamping karakteristik-karakteristik

lainya sehingga dalam operasi hariannya harus berdasarkan lengkung beban harian yang telah dibuat karena dengan lengkung baban harian ini dapat ditentukan perencanaan operasi pembangkit-pembangkit yang ada, baik itu unit pembangkit termal ataupun hidro. Tentu saja kebutuhan beban dalam suatu harinya tidak merata akan tetapi dari jam ke jam berbeda sesuai dengan kebutuhan konsumen. Berdasarkan lengkung beban yang telah ada maka dapat ditentukan berapa unit pembangkit yang harus bekerja dan siap bekerja pada hari itu.

Sebagai dasar pertimbangan yang sifatnya umum, untuk menentukan biaya produksi tenaga listrik yang dibutuhkan adalah dengan memperhatikan bahwa dalam keadaan beban minimum maka tenaga listrik yang dibutuhkan diberikan oleh unit pembangkit yang bekerja paling efisien pada keadaan tersebut. Pembangkit ini akan terus beroperasi atau dibebani sampai pada batas efisiensi maksimumnya. Dan apabila ternyata beban masih terus bertambah sedangkan unit pembangkit ini telah mencapai maksimumnya maka selanjutnya belum ditanggung oleh unit pembangkit yang lain yang belum mencapai efisiensi maksimumnya. Dengan dasar operasi yang demikian maka dapat dicapai keadaan operasi yang cukup ekonomis.

Akan tetapi dengan semakin berkembangnya sistem itu sendiri maka diperlukan suatu perencanaan pembangkitan yang optimum dengan biaya operasi yang ekonomis. Mengingat bahwa beban sistem adalah selalu berubah-ubah dari waktu ke waktu maka perlu untuk membuat secara grafis perubahan beban terhadap waktu.

Oleh karena biaya operasi untuk memproduksi daya listrik sangat besar, suatu pembangkit khususnya pembangkit thermal, maka untuk biaya operasi pembangkitan harus ditekan seekonomis mungkin untuk mendapatkan biaya operasi yang rendah, karena pada unit pembangkit thermal ini akan membutuhkan biaya operasi yang cukup tinggi sehingga usaha penghematan biaya bahan bakar akan sangat berarti. Dengan kata lain dengan mengkoordinasikan operasi pembangkit-pembangkit yang tersedia dengan tepat dan sesuai dengan beban maka didapat suatu keadaan operasi yang ekonomis.

Pembahasan mengenai operasi ekonomis adalah merupakan salah satu cara bagaimana menekan biaya produksi dari sistem tenaga listrik. Dalam hal ini maka metode yang dipakai adalah dengan memanfaatkan karakteristik dan menganalisa operasi dari sistem tersebut. Disamping karakteristik dari unit-unit pembangkit perlu juga diketahui karakteristik beban, karena karakteristik bebanlah maka dapat dianalisa pengaturan yang paling ekonomis dari setiap unit pembangkit. Adapun karakteristik yang perlu diketahui dari setiap unit pembangkit adalah :

1. Karakteristik input bahan bakar sebagai fungsi dari output daya.
2. Nilai panas sebagai fungsi output daya.
3. Kenaikan jumlah bahan bakar yang dibutuhkan jika terdapat perubahan beban.

Ketiga karakteristik tersebut merupakan pedoman menganalisa penjadwalan selanjutnya. Kemudian yang juga perlu diperhitungkan adalah variabel-variabel yang terdapat pada saluran transmisi, karena variabel-variabel ini

juga sangat menentukan ekonomis tidaknya penjadwalan pembangkit yang kita tentukan.

Maka untuk mencapai suatu operasi yang ekonomis pada suatu sistem tenaga listrik adalah dengan melakukan penjadwalan pada sistem pembangkit yang ada pada suatu sistem tenaga listrik yang ditinjau tersebut dengan memanfaatkan karakteristik dari setiap masing-masing unit pembangkit yang ada pada dasarnya bertujuan untuk menekan biaya pembangkitan agar diperoleh biaya yang sangat rendah sehingga dapat memuaskan pemakai energi listrik.

2.3. Karakteristik Pembangkit^[1]

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.

2.3.1. Karakteristik Input-Output^[1]

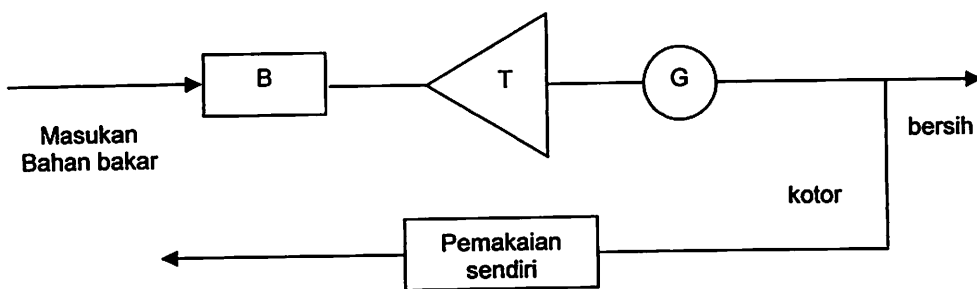
Hal yang paling mendasar dalam mengoptimalkan pembangkit secara ekonomis adalah membuat karakteristik *Input-Output* dari unit pembangkit

thermal. Karakteristik ini diperoleh dari desain perencanaan atau melalui test pembangkit. Adapun definisi dari karakteristik *Input-Output* pembangkit adalah formula yang menyatakan hubungan antara input pembangkit sebagai fungsi dari output pembangkit unit boiler-turbin-generator dapat digambarkan dalam gambar 2.1. dimana unit ini membuat sebuah boiler yang menghasilkan uap untuk turbin yang dikopel dengan rotor dari generator.

Pada pembangkit thermal Input diberikan dalam satuan panas Btu/jam atau Kal/jam dari bahan bakar yang diberikan pada boiler untuk menghasilkan output pembangkit. Sedangkan notasi yang digunakan adalah H(Mbtu/h) atau dalam satuan yang lain H (Mkal/h). selain itu input dari pembangkit dapat pula dinyatakan dalam nilai yang menyatakan besarnya biaya yang diperlukan untuk bahan bakar. Notasi yang digunakan adalah F(R/h). Hubungan antara H dan F dapat dinyatakan dalam rumus berikut ini :

$$F = H \times \frac{\text{Rupiah}}{\text{Mbtu}} \dots\dots\dots (2.1)$$

Dimana rupiah/Mbtu adalah nilai uang yang diperlukan persatuan panas dari bahan bakar.



Keterangan: B = Boiler, T = Turbin Uap, G = Generator

Gambar 2.1.
Unit Boiler-Turbin-Generator^[4]

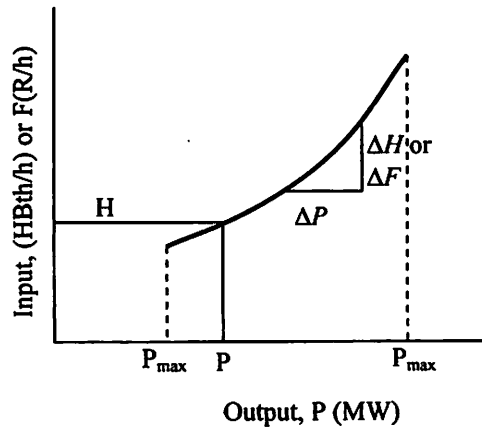
Seperti digambarkan dalam gambar 2.1. maka output pembangkit tidak hanya dihubungkan dengan beban tetapi juga untuk peralatan bantu dalam pembangkit. Disini output pembangkit didefinisikan sebagai daya yang dikeluarkan oleh generator untuk beban sistem diluar untuk keperluan pembangkit itu sendiri. Jadi untuk karakteristik *input-output*, daya output adalah berupa daya netto dari pembangkit, notasi yang digunakan adalah P (MW).

Generator akan mengeluarkan daya sesuai dengan beban yang ada. Semakin besar beban, semakin besar daya yang dikeluarkan oleh generator. Daya yang dikeluarkan generator dapat membesar sesuai peningkatan beban sampai dengan daya maksimum yang dapat dikeluarkan oleh generator. Semakin besar daya yang dikeluarkan oleh generator, semakin besar pula bahan bakar yang dimasukkan. Dengan kata lain jumlah bahan bakar yang dibakar merupakan fungsi dari daya keluaran generator. Namun hubungan antara bahan bakar dengan daya keluaran generator tidak linier, sebab bahan bakar melewati proses pembakaran yang memerlukan waktu.

Dari keterangan diatas, dapat dibentuk persamaan karakteristik *Input-Output* pembangkit yang dapat dilihat pada persamaan 2.2 dan persamaan 2.3 dibawah ini sedangkan kurva dari karakteristik Input-Output dapat dilihat pada gambar 2.2.

$$H = f(P), \text{ atau } \dots\dots\dots (2.2)$$

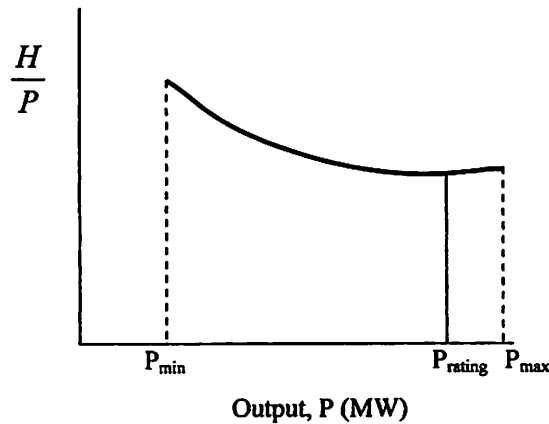
$$I = f(P) \dots\dots\dots (2.3)$$



Gambar 2.2
Kurva karakteristik *Input-Output* Pembangkit Thermal^[1]

2.3.2. Karakteristik *Hate Rate*^[1]

Karakteristik lain yang cukup penting bagi pembangkit thermal adalah karakteristik tingkat panas atau *Hate Rate Characteristic*. Fungsi ini menyatakan hubungan antara tingkat panas terhadap tingkat beban pusat listrik. Karakteristik ini umumnya memiliki korelasi dengan efisiensi mesin kalor yang digunakan. Gambar 2.3 diplot berdasarkan nilai H/P terhadap P . Pembangkit listrik thermal konvensional memiliki efisiensi kalor antara 30% sampai 35%, sehingga tingkat panas yang dimiliki berkisar antara 11400 BTU/KWH sampai 9800 BTU/KWH (1 KWH kira-kira setara dengan 3412 BTU). Karakteristik tingkat panas pada dasarnya antara lain menyatakan keadaan uap tingkat temperatur, tekanan kondensor dan siklus fluida kerja yang terjadi selama pembangkit listrik tenaga thermal tersebut beroperasi.



Gambar 2.3
Kurva Karakteristik *Hate–Rate* Unit Pembangkit^[1]

2.3.3. Karakteristik Incremental Hate Rate dan Incremental Fuel Cost^[1]

Perwujudan yang lain dari karakteristik pembangkit adalah karakteristik *Incremental Hate Rate* atau perubahan tingkat laju panas dan karateristik *Incremental Fuel Cost* atau tingkat perubahan tingkat laju biaya bahan bakar. Karakteristik Incremental Hate Rate menyatakan hubungan daya output sebagai fungsi Incremental Hate Rate. Sedangkan karateristik Incremental Fuel Cost menyatakan daya output sebagai fungsi Incremental Fuel Cost. Karakteristik Incremental Hate Rate ini menunjukkan besarnya perubahan Input energi bila ada perubahan output pembangkit pada megawatt output unit pembangkit.

Kurva karakteristik Incremental Hate Rate atau Fuel Cost dapat dilihat pada gambar 2.4. sedangkan persamaan Incremental Hate Rate dan persamaan Incremental Fuel Cost dapat dilihat pada persamaan 2.4. sampai persamaan 2.7.

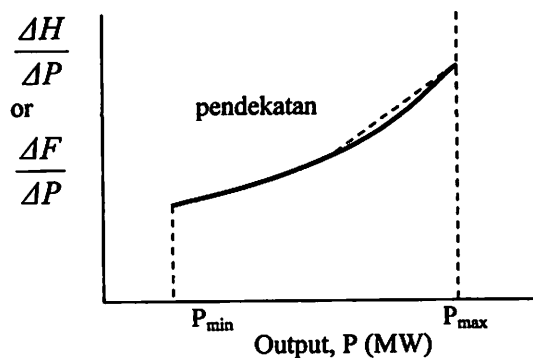
$$\text{Incremental Hate Rate} = \frac{\Delta H}{\Delta P} \left[\frac{\text{MBTU}}{\text{kwh}} \right] \dots\dots\dots (2.4)$$

$$\text{Incremental Fuel Cost} = \frac{\Delta F}{\Delta P} \left[\frac{\text{Rupiah}}{\text{kwh}} \right] \dots\dots\dots (2.5)$$

Dengan mengambil harga ΔP mendekati nol maka dapat dinyatakan dengan persamaan berikut ini

$$\text{Incremental Hate Rate} = \frac{dH}{dP} \left[\frac{\text{MBTU}}{\text{kwh}} \right] \dots\dots\dots (2.6)$$

$$\text{Incremental Fuel Cost} = \frac{dF}{dP} \left[\frac{\text{Rupiah}}{\text{kwh}} \right] \dots\dots\dots (2.7)$$



Gambar 2.4.
Kurva karakteristik Incremental Fuel Cost Rate^[1]

2.4. Economic Dispatch^[1]

Yang dimaksud *Economic Dispatch* adalah pembagian pembebanan pada pembangkit-pembangkit yang ada dalam sistem, secara optimal ekonomi pada harga beban tertentu. Tujuan dari *economic dispatch* adalah untuk mendapatkan biaya harga bahan bakar semurah mungkin dalam suatu sistem pembangkit pada beban tertentu. Dengan dilakukan *Economic Dispatch* maka didapat harga bahan bakar daya yang paling murah dalam suatu sistem pembangkit. Oleh karena itu

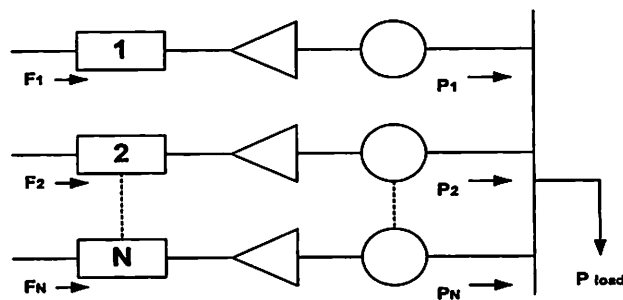
beban yang harus ditanggung oleh sistem pembangkit selalu berubah setiap periode waktu tertentu, maka perhitungan Economic Dispatch ini dilakukan untuk setiap harga beban tertentu. *Economic Dispatch* dapat dilakukan dengan beberapa cara yang akan dibahas pada sub bab dibawah ini.

2.4.1. Penyelesaian Economic Dispatch dengan metode pengali Lagrange ^[1]

Sistem dengan mengabaikan rugi-rugi transmisi dapat dilihat pada gambar

2.5. Sistem ini terdiri dari N unit generator thermal yang dihubungkan pada single busbar yang melayani beban P_R . Input dari masing-masing unit ditunjukkan oleh F_i yang mewakili biaya dari satu unit generator dan output dari masing-masing unit P_i adalah daya yang dihasilkan oleh satu unit generator.

Total biaya rata-rata yang ditanggung sistem adalah jumlah biaya dari masing-masing unit generator. Dan pembatas yang paling penting adalah jumlah dari output masing-masing unit generator sama dengan beban konsumen.



Gambar 2.5.
N unit melayani beban P_R ^[1]

Yang menjadi permasalahan adalah meminimumkan total biaya F_T dengan memperhatikan pembatas ϕ bahwa daya yang dihasilkan generator sama dengan

yang diterima beban. Secara matematika pernyataan tersebut diatas dapat dinyatakan dengan persamaan berikut :

$$F_T = F_1 + F_2 + F_3 + \dots + F_N$$

$$= \sum_{i=1}^N F_i(P_i) \dots\dots\dots (2.8)$$

$$\phi = 0 = P_R - \sum_{i=1}^N P_i \dots\dots\dots (2.9)$$

Persamaan ini adalah pembatas yang merupakan masalah dari optimasi dan ini dapat dipecahkan dengan metode kalkulus tingkat lanjut yang melibatkan fungsi Lagrange. Dimana fungsi Lagrange didapat dengan cara menambah pembatas ϕ yang telah dikalikan dengan factor pengali Lagrange λ pada fungsi F_T . Fungsi Lagrange dapat ditunjukkan dengan persamaan dibawah ini :

$$L = F_T + \lambda \phi \dots\dots\dots (2.10)$$

Dimana :

F_T = fungsi tujuan

λ = factor pengali

ϕ = fungsi pembatas (*constrain*)

Persamaan Lagrange diatas merupakan fungsi dari output pembangkit P_i dan factor pengali Lagrange λ . Keadaan dari optimasi dari fungsi tujuan F_T dapat diperoleh dengan operasi gradient dari persamaan Lagrange sama dengan nol.

$$\nabla L = 0 \dots\dots\dots (2.11)$$

$$\nabla F_T + \lambda \phi = 0 \dots\dots\dots (2.12)$$

$$\frac{\delta L}{\delta P} = \frac{\delta F_T}{\delta P_i} + \lambda \left[\frac{\delta P_R}{\delta P_i} - \frac{\delta P_i}{\delta P_i} \right] = 0 \text{ atau} \dots\dots\dots (2.13)$$

$$\frac{\delta F_i}{\delta P_i} + \lambda (0 - 1) = 0$$

$$\frac{\delta F_i}{\delta P_i} = \lambda \dots\dots\dots (2.14)$$

Persamaan terakhir ini menunjukkan bahwa bila digunakan biaya bahan bakar, F_T yang paling minimum maka Incremental Cost setiap unit generator pembangkit harus sama yaitu sebesar λ . Kondisi optimal ini tentunya dengan tetap memperhatikan pembatas yang ada yaitu bahwa daya dari setiap unit generator pembangkit harus lebih besar atau sama dengan daya output minimum atau sama dengan daya output maksimum yang diijinkan.

Dari N buah unit generator pembangkit dalam sistem tenaga yang telah dibahas dan beban sistem sebesar P_R , maka dapat diambil kesimpulan berikut :

$$\frac{\delta F_i}{\delta P_i} = \lambda \text{ ada N buah Persamaan}$$

$$P_{i_{\min}} \leq P_i \leq P_{i_{\max}} \text{ ada 2 N buah pertidaksamaan} \dots\dots\dots (2.15)$$

$$\sum_{i=1}^N P_i = P_R \text{ ada 1 buah pembatas}$$

Dari batasan pertidaksamaan pembatasan diatas dapat diperluas menjadi :

$$\frac{\delta F_i}{\delta P_i} = \lambda \text{ untuk } P_{i_{\min}} \leq P_i \leq P_{i_{\max}}$$

$$\frac{\delta F_i}{\delta P_i} \leq \lambda \text{ untuk } P_i = P_{i_{\max}} \dots\dots\dots (2.16)$$

$$\frac{\delta F_i}{\delta P_i} \geq \lambda \text{ untuk } P_i = P_{i_{\min}} \dots\dots\dots (2.17)$$

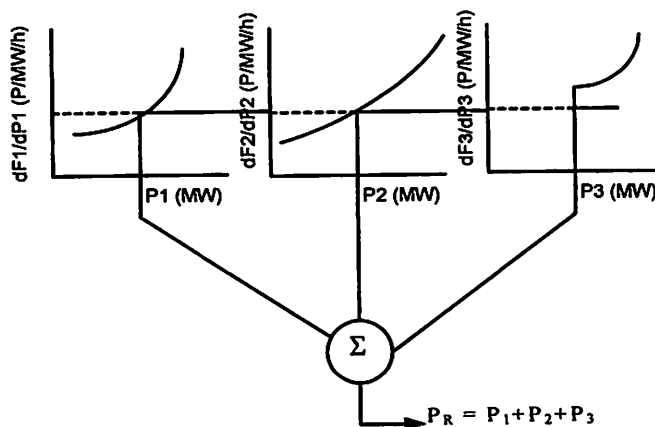
Karena F_i hanya sebagai fungsi P_i maka $\frac{\delta F_i}{\delta P_i}$ dapat diganti dengan $\frac{dF_i}{dP_i}$

2.4.2. Penyelesaian *Economic Dispatch* dengan metode iterasi Lamda^[1]

Dalam metode iterasi lamda, kita menentukan sembarang λ . Dari λ yang telah ditentukan, kita menghitung harga output masing-masing pembangkit dengan menggunakan syarat optimum.

Dengan menggunakan constrain diperiksa apakah jumlah total dari output sama dengan beban sistem. Bila jumlah dari P_1 , P_2 , dan P_3 , lebih kecil dari P_R (beban sistem). Maka ditentukan kembali harga λ kedua yang lebih besar dari λ pertama. Bila sebaliknya maka ditentukan harga λ kedua yang lebih kecil dari λ pertama.

Dengan telah diperoleh dua hasil perhitungan diatas maka secara ekstrapolasi dapat ditentukan harga λ selanjutnya sampai dicapai harga yang dikehendaki dimana $P_1 + P_2 + P_3 = P_R$



Gambar 2.6.
Grafik penyelesaian dengan metode iterasi lamda^[4]

2.5. Fungsi Biaya Bahan Bakar^[1]

Biaya bahan bakar merupakan unsur biaya yang penting dalam operasi sistem pembangkit thermal. Fungsi biaya bahan bakar $F_i(P_i)$ untuk tiap unit pembangkit terhadap daya keluaran diekspresikan dalam bentuk fungsi kuadrat, yang dapat dinyatakan sebagai berikut :

$$F_i(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i \dots\dots\dots (2.18)$$

Dimana : a_i, b_i, c_i = konstanta persamaan dari unit ke-i

P_{it} = daya keluaran dari unit ke-I pada jam t

Dalam pengoperasian secara ekonomis adalah penting untuk mengetahui biaya bahan bakar yang digunakan untuk membangkitkan daya yang diperlukan :

- Jenis bahan bakar
- Nilai kalori
- Harga bahan bakar

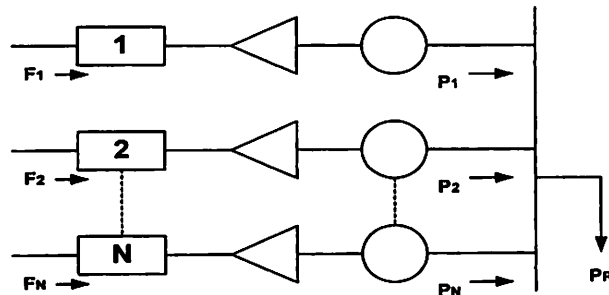
2.6. Economic Dispatch Dengan Mengabaikan Rugi-rugi Transmisi ^[1]

Dalam sistem tenaga listrik, kerugian transmisi merupakan kehilangan daya yang harus ditanggung oleh sisi pembangkit. Jadi dengan adanya kerugian daya tersebut merupakan tambahan beban bagi sistem tenaga listrik.

Sistem dengan mengabaikan rugi-rugi transmisi dapat dilihat pada gambar

2.9. Sistem ini terdiri dari N buah pembangkit thermal yang dihubungkan pada single bus bar yang melayani beban P_r . Input dari masing-masing pembangkit ditunjukkan oleh F_i yang mewakili biaya dari satu unit pembangkit dan output

dari masing-masing unit P_i adalah daya yang dihasilkan oleh satu unit pembangkit.



Gambar 2.7.
N Unit Pembangkit Thermal Melayani Beban P_R ^[1]

Total biaya rata-rata yang harus ditanggung oleh sistem adalah jumlah biaya dari masing-masing unit pembangkit. Dan pembatas yang paling penting adalah jumlah output dari dari masing-masing unit pembangkit sama dengan di konsumen. Yang menjadi permasalahan adalah meminimumkan total biaya F_T dengan memperhatikan pembatas ϕ bahwa daya yang dihasilkan oleh pembangkit sama dengan beban yang diterima. Secara matematika pernyataan yang tersebut diatas dapat dinyatakan dengan persamaan sebagai berikut :

$$F_T = F_1 + F_2 + F_3 + \dots + F_N \dots\dots\dots(2.19)$$

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

$$\phi = 0 = P_R - \sum_{i=1}^N P_i$$

Persamaan di atas adalah pembatas yang merupakan problem dari optimasi dan ini dapat dipecahkan dengan menggunakan kalkulus tingkat lanjut yang melibatkan fungsi Lagrange. Dimana fungsi ini didapat dengan cara

menambahkan pembatas ϕ yang telah dikalikan dengan factor pengali Lagrange λ pada fungsi tujuan F_T fungsi Lagrange dapat ditunjukkan dengan persamaan dibawah ini :

$$L = F_T + \lambda \cdot \phi$$

Persamaan Lagrange di atas fungsi output pembangkit P_i dan factor pengali Lagrange λ keadaan dari optimal dari fungsi tujuan F_T dapat diperoleh dengan operasi gradient persamaan Lagrange sama dengan 0.

$$\nabla L = 0 \dots\dots\dots (2.20)$$

$$\nabla F_T + \lambda \cdot \phi = 0 \dots\dots\dots (2.21)$$

$$\frac{\partial L}{\partial P} = \frac{\partial F_i}{\partial P_i} + \lambda \left(\frac{\partial P_R}{\partial P_i} - \frac{\partial P_i}{\partial P_i} \right) = 0 \dots\dots\dots (2.22)$$

$$\text{atau } \frac{\partial F_i}{\partial P_i} + \lambda \cdot (0 - 1) = 0 \dots\dots\dots (2.23)$$

$$\frac{\partial F_i}{\partial P_i} = \lambda \dots\dots\dots (2.24)$$

Persamaan terakhir ini menunjukkan bahwa bila digunakan biaya bahan bakar F_T yang paling minimum maka incremental cost setiap unit generator pembangkit harus sama yaitu sebesar λ kondisi optimal ini tentunya dengan tetap memp[erhatikan pembatas yang ada yaitu bahwa daya dari setiap unit generator pembangkit harus lebih besar atau sama dengan daya output minimum dan lebih kecil atau sama dengan daya output maksimum yang diizinkan.

Dari N buah pembangkit yang ada dalam sistem tenaga yang telah dibahas dan beban sistem sebesar P_R , maka dapat diambil kesimpulan sebagai berikut :

$$\frac{\partial F_i}{\partial P_i} = \lambda \dots\dots\dots (2.25)$$

$$P_{i \min} \leq P_i \leq P_{i \max} \text{ ada 2 N buah pertidaksamaan } \dots\dots\dots (2.26)$$

$$\sum_{i=1}^N P_i = P_R \dots\dots\dots (2.27)$$

Dari batasan pertidaksamaan pembatas di atas dapat di perluas menjadi :

$$\frac{\partial F_i}{\partial P_i} = \lambda \text{ untuk } P_{i \min} \leq P_i \leq P_{i \max} \dots\dots\dots (2.28)$$

$$\frac{\partial F_i}{\partial P_i} \leq \lambda \text{ untuk } P_i = P_{i \max} \dots\dots\dots (2.29)$$

$$\frac{\partial F_i}{\partial P_i} \geq \lambda \text{ untuk } P_i = P_{i \min} \dots\dots\dots (2.30)$$

Karena F_i hanya sebagai fungsi P_i maka $\frac{\partial F_i}{\partial P_i}$ dapat diganti dengan $\frac{\partial F_i}{\partial P_i}$.

BAB III

APLIKASI METODE JARINGAN SYARAF

ADAPTIVE HOPFIELD UNTUK ECONOMIC DISPATCH

3.1. Penyelesaian dengan Jaringan *Adaptive Hopfield*

Pendekatan konvensional untuk memecahkan masalah *Economic Load Dispatch* dengan menggunakan metode Jaringan *Adaptive Hopfield* membutuhkan jumlah pengulangan (iterasi) yang banyak dan seringkali berubah-ubah selama transien. Untuk mempercepat keseimbangan (*convergence*), maka metode pengaturan adaptif yang di bahas dalam skripsi ini adalah pengaturan slope.

3.1.1. Jaringan *Adaptive Hopfield*

Model neuron kontinyu adalah generalisasi dari jaringan *Hopfield* dimana energi perhitungan turun secara kontinyu seiring waktu. Untuk parameter keuagan sangat tinggi (λ) dari neuron, jaringan kontinyu melakukan dengan cara serupa seperti model diskrit. Ketika vector parameter nilai adalah simetris, maka fungsi energi dari Jaringan Syaraf *Hopfield* didefinisikan sebagai berikut ;

$$E = -1/2 \sum_i \sum_j T_{ij} V_i V_j - \sum_i I_i V_i + \sum_i \theta_i V_i \dots\dots\dots(3.1)$$

Dimana V_i adalah harga output dari neuron i , I_i adalah input keluaran pada neuron i , dan θ_i adalah ambang batas bias.

Pergerakan (*dynamics*) dari neuron didefinisikan :

$$\frac{dU_i}{dt} = \sum_j T_{ij} V_j + I_i \dots\dots\dots(3.2)$$

dimana U_i adalah total keluaran pada neuron i dan fungsi sigmoid didefinisikan :

$$V_i = g_i(\lambda U_i) = g_i\left(\frac{U_i}{U_0}\right) = \frac{1}{1 + \exp\left(-\frac{U_i + \theta_i}{U_0}\right)} \dots\dots\dots(3.3)$$

Stabilitas diketahui untuk menjamin dimana fungsi energi keluar batas dan dimana kenaikan nya ditemukan menjadi tidak positif

$$\frac{dE}{dt} = - \sum_i g_i(U_i) \left(\frac{dU_i}{dt}\right)^2 \dots\dots\dots(3.4)$$

Ketika $g_i(U_i)$ adalah fungsi kenaikan non monotone, maka masing-masing bentuk dari jumlah ini adalah tidak negatif. Oleh karena itu persamaan (3.4) akan lebih kecil dari nol. Evolusi waktu dari sistem adalah sebuah gerakan pada ruang yang mencari E minimum dan sampai berhenti pada angka tersebut.

Untuk memecahkan masalah ELD, fungsi energi dapat didefinisikan dengan persamaan sebagai berikut:

$$E = \frac{1}{2}A(D + L - \sum_i P_i)^2 + \frac{1}{2}B \sum_i (a_i + b_i P_i + c_i P_i^2) \dots\dots\dots(3.5)$$

dimana a_{ik}, b_{ik}, c_{ik} adalah biaya koefisien sebagai fungsi *discrete* P_i .

Dengan membandingkan (3.5) dengan (3.1) dimana mendekati nol, parameter bobot dan input eksternal pada neuron I dalam jaringan diberikan sebagai :

$$\begin{aligned} T_{ii} &= -A - Bc_i \\ T_{ij} &= -A \dots\dots\dots(3.6) \\ I_i &= A(D + L) - \frac{Bb_i}{2} \end{aligned}$$

dimana bobot diagonal adalah tidak nol. Hasil perubahan ini (3.2) kedalam bentuk update berikut :

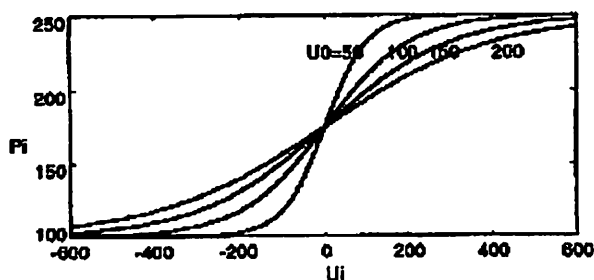
$$U_i(k) - U_i(k-1) = \sum_j T_{ij} V_j(k) + I_i \dots\dots\dots(3.7)$$

Fungsi sigmoid (3.3) dapat diubah menjadi bentuk batas daya sebagai berikut :

$$V_i(k+1) = (P_i^{max} - P_i^{min}) \frac{1}{1 + \exp\left(-\frac{U_i(k) + \theta_i}{U_0}\right)} + P_i^{min} \dots\dots\dots(3.8)$$

3.1.2. Metode Pengaturan Slope

Dalam keadaan transien, input neuron berubah-ubah mendekati antara nol ditunjukkan dalam gambar 1.



Gambar 1.
fungsi perubahan sigmoidal dengan perbedaan nilai dari pendekatan parameter

Beberapa input neuron berubah-ubah menjauhi dari nilai yang ditentukan. Jika parameter yang diperoleh terlalu tinggi, maka bentuk gelombang (osilasi) akan terjadi pada daerah saturasi. Jika parameter pada slope yang berada pada

daerah tersebut terlalu tinggi maka neuron-neuron itu tidak akan dalam keadaan stabil dan akan menyebabkan ketidakstabilan.

Ketika energi diperkecil dan keseimbangannya tergantung pada parameter yang diperoleh U_o , maka metode gradien-descent dapat dipakai untuk menyesuaikan parameter tersebut sebagai berikut :

$$U_o(k+1) = U_o(k) - \eta_s \frac{\partial E}{\partial U_o} \dots\dots\dots(3.9)$$

dimana η_s adalah tingkat pembelajaran.

Dari (3.5) dan (3.8), gradien energi dengan mengacu kepada parameter yang diperoleh dapat dihitung sebagai berikut :

$$\frac{\partial E}{\partial U_o} = \sum_{i=1}^n \frac{\partial E \partial P_i}{\partial P_i \partial U_o} \dots\dots\dots(3.10)$$

Rumus terbaru (3.9) membutuhkan pilihan yang sesuai dengan tingkat pembelajaran η_s . Untuk sebuah nilai kecil η_s konvergensi dapat dipenuhi tetapi kecepatannya rendah, di lain sisi jika tingkat pembelajaran terlalu tinggi, algoritmanya menjadi tidak stabil. Untuk terpenuhinya konvergensi dan dengan cepat, sebuah metode untuk menghitung tingkat pembelajaran adaptif maka dikembangkan prosedur berikut. Ini dapat ditunjukkan bahwa konvergensi dapat terpenuhi jika tingkat pembelajaran η_s , dipilih sebagai berikut :

$$0 < \eta_s < \frac{2}{g_{s, \max}^2} \dots\dots\dots(3.11)$$

Dimana $g_{s, \max} := \max \|g_s(k)\|, g_s(k) = \partial E(k) / \partial U_o$

Lebih jauh lagi, konvergensi optimal akan sesuai dengan

$$\eta_s^* = \frac{1}{g_{s,\max}^2} \dots\dots\dots(3.12)$$

Ini menunjukkan sebuah hasil yang menarik bahwa pada tingkat pembelajaran lain lebih besar dari pada η_s^* tidak memenuhi sebuah konvergensi yang cepat.

3.1.3. Momentum

Kecepatan konvergensi dapat diatur dengan menambahkan momentum dalam proses berikutnya. Momentum dapat dipakai ketika inputnya nanti dalam persamaan (3.7), parameter yang diperoleh berikutnya dalam persamaan (3.9):

$$U_i(k) - U_i(k-1) = \sum_j T_{ij} V_j(k) + \alpha_u \Delta U_i(k-1) \dots\dots\dots(3.13)$$

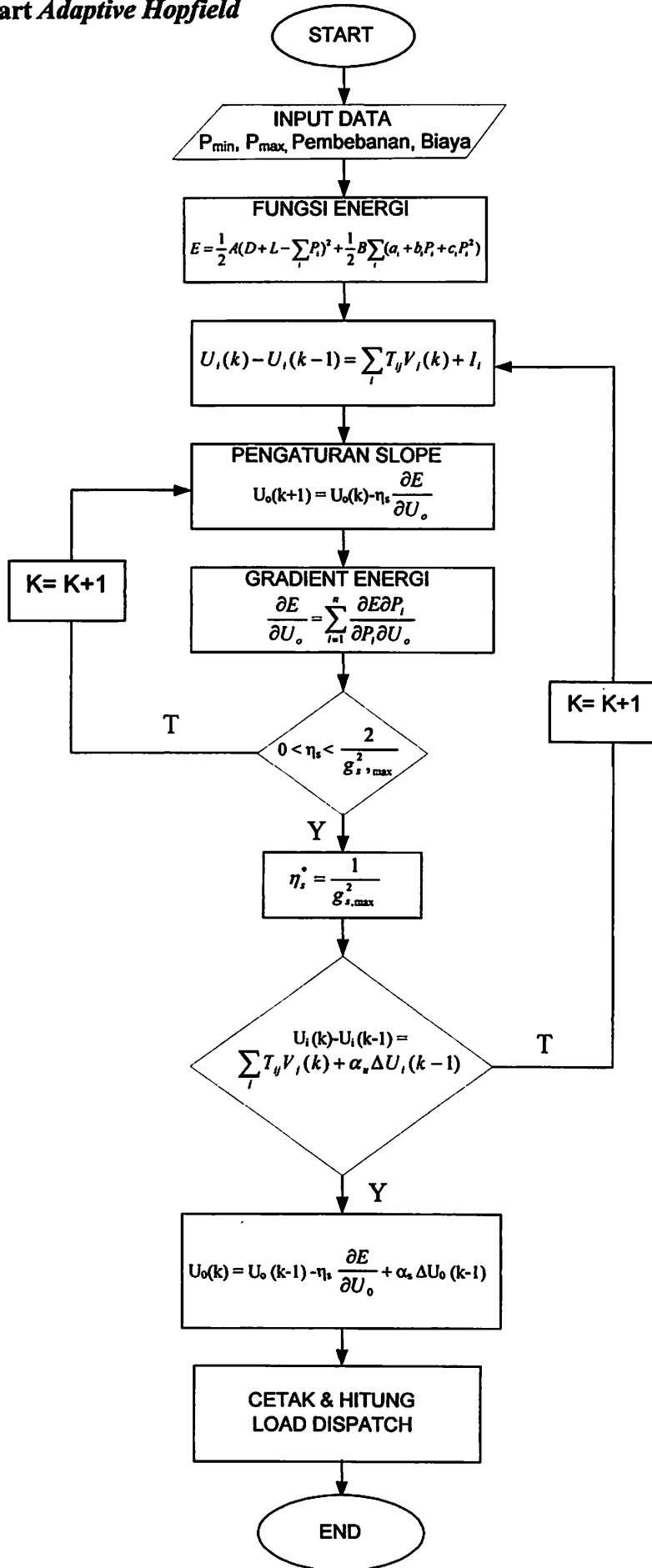
$$U_0(k) = U_0(k-1) - \eta_s \frac{\partial E}{\partial U_0} + \alpha_s \Delta U_0(k-1) \dots\dots\dots(3.14)$$

dimana α 's adalah faktor momentum.

3.3. Algoritma Adaptif Hopfield

- Langkah 1 : Set Input Data Pmin, Pmax, Pembebanan, Koefisien Biaya
- Langkah 2 : Set Persamaan Fungsi Energi
- Langkah 3 : Set Update fungsi $\Sigma T_{ij}V_j(k) + I_i$.
- Langkah 4 : Set Pengaturan Slope
- Langkah 5 : Set Persamaan Gradient Energi
- Langkah 6 : Set η_s sampai mendekati nilai yang paling konvergen
- Langkah 7 : Jika tidak konvergen maka akan kembali ke langkah ke-4 di pengaturan slope
- Langkah 8 : Jika nilai η_s sudah konvergen maka diteruskan ke langkah selanjutnya.
- Langkah 9 : Set Persamaan Momentum yaitu $U_i(k) - U_i(k-1)$.
- Langkah 10 : Jika langkah ke sembilan belum sesuai maka kembali ke langkah ke-3 yaitu set update fungsi.
- Langkah 11 : Jika sudah konvergen maka kita teruskan ke persamaan slope yaitu $U_0(k)$
- Langkah 12 : Setelah $U_0(k)$ diketahui maka kita dapat masukkan persamaan load economic dispatch yaitu $F = a_{ik} + b_{ik}P_i + c_{ik}P_i^2$

3.3. Flow Chart Adaptive Hopfield



BAB IV

ANALISA DATA MENGGUNAKAN

METODE JARINGAN SYARAF *ADAPTIVE HOPFIELD*

4.1. Pendahuluan

PT. PLN Pembangkit Jawa – Bali 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 Pembangkit Jawa-Bali harus menjaga efisiensi dan keadilan operasional penyediaan tenaga listrik dari pembangkit-pembangkit yang dimilikinya.

Dengan demikian merupakan suatu keharusan bagi seluruh jajaran PT.PLN Pembangkit Jawa-Bali agar selalu berupaya untuk meningkatkan kondisi penyediaan tenaga listrik dari pembangkit agar lebih ekonomis, bermutu dan didukung keandalan yang tinggi.

4.2. Program Komputer *Economic Dispatch* Dengan Metode Jaringan Syaraf *Adaptive Hopfield*

Dalam penyelesaian masalah ini digunakan bantuan program komputer. Program computer ini sangat berguna untuk mempercepat proses perhitungan membutuhkan ketelitian tinggi dan sering melibatkan iterasi yang membutuhkan waktu yang lama bila dikerjakan secara manual.

Program computer ini menggunakan bahasa pemrograman Borland Delphi versi 7.0, merupakan bahasa pemrograman terstruktur yang relatif mudah untuk dipelajari dan mudah penggunaannya.

4.3. Data Pembangkit Thermal

Pembangkit Thermal yang berada dalam pengawasan PT.PLN Pembangkit Jawa-Bali berjumlah 38 unit yang terdiri dari 5 blok Pembangkit Listrik Tenaga Gas dan Uap, 11 Pembangkit Listrik Tenaga Uap dan 5 Pembangkit Listrik Tenaga Gas. Adapun Data-data lebih lengkapnya dapat dilihat pada tabel 4.1, untuk harga bahan bakar berdasarkan statistic PLN 2003 dimana dipakai nilai tukar Rp. 9000 per satuan dolar Amerika.

Tabel 4.1
Data Unit Thermal Pada PT.PLN PJB

No	NamaPembangkit	Bahan Bakar	Kapasitas (MW)		Koefisien BiayaBahan Bakar		
			Min	Max	A	B	C
1	PLTU Paiton 1	Coal	225	370	3244978	111712,15	10,2971
2	PLTU Paiton 2	Coal	225	370	3244978	111712,15	10,2971
3	PLTGU Gresik GT 1.1	Gas	53	102	5467532,4	217963,548	34,155
4	PLTGU Gresik GT 1.2	Gas	53	102	5467532,4	217963,548	34,155
5	PLTGU Gresik GT 1.3	Gas	53	102	5467532,4	217963,548	34,155
6	PLTGU Gresik ST 1.0	Gas	250	480	10936203,3	72527,004	368,874
7	PLTGU Gresik GT 2.1	Gas	53	102	5467532,4	217963,548	34,155
8	PLTGU Gresik GT 2.2	Gas	53	102	5467532,4	217963,548	34,155
9	PLTGU Gresik GT 2.3	Gas	53	102	5467532,4	217963,548	34,155
10	PLTGU Gresik ST 2.0	Gas	250	480	10936203,3	72527,004	368,874
11	PLTGU Gresik GT 3.1	Gas	53	102	5467532,4	217963,548	34,155
12	PLTGU Gresik GT 3.2	Gas	53	102	5467532,4	217963,548	34,155
13	PLTGU Gresik GT 3.3	Gas	53	102	5467532,4	217963,548	34,155
14	PLTGU Gresik ST 3.0	Gas	250	480	10936203,3	72527,004	368,874
15	PLTU Gresik 1	Gas	43	85	1327126,68	217378,359	132,066
16	PLTU Gresik 2	Gas	43	85	1327126,68	217378,359	132,066
17	PLTU Gresik 3	Gas	90	175	5017369,5	169242,579	193,545
18	PLTU Gresik 4	Gas	90	175	5017369,5	169242,579	193,545
19	PLTG Gresik 1	Gas	5	16	352707,3	350680,77	903,969
20	PLTG Gresik 2	Gas	5	16	352707,3	350680,77	903,969
21	PLTG Gresik 3	Gas	5	16	352707,3	350680,77	903,969
22	PLTG Gilitimur 2	HSD	5	16	687181,85	683240,965	1762,3893
23	PLTG Gilitimur 2	HSD	5	16	687181,85	683240,965	1762,3893
24	PLTGU M. Karang GT 1.1	Gas	50	95	5730795	202052,97	108,045
25	PLTGU M. Karang GT 1.2	Gas	50	95	5730795	202052,97	108,045
26	PLTGU M. Karang GT 1.3	Gas	50	95	5730795	202052,97	108,045
27	PLTGU M. Karang ST 1.0	Gas	300	465	11560815	53685,135	460,845
28	PLTGU M. Tawar GT 1.1	HSD	72	138	14706521,25	433337,8	49,4605
29	PLTGU M. Tawar GT 1.2	HSD	72	138	14706521,25	433337,8	49,4605
30	PLTGU M. Tawar GT 1.3	HSD	72	138	14706521,25	433337,8	49,4605
31	PLTGU M. Tawar GT 2.1	HSD	72	138	14706521,25	433337,8	49,4605
32	PLTGU M. Tawar GT 2.2	HSD	72	138	14706521,25	433337,8	49,4605
33	PLTGU M. Tawar ST 1.0	HSD	162	202	672630	144191,717	519,1757
34	PLTU M. Karang 1	MFO	44	85	2417820,7	473895,41	120,77935
35	PLTU M. Karang 2	MFO	44	85	2417820,7	473895,41	120,77935
36	PLTU M. Karang 3	MFO	44	85	2417820,7	473895,41	120,77935
37	PLTU M. Karang 4	Gas	90	165	2949187,5	205217,145	83,79
38	PLTU M. Karang 5	Gas	90	165	2949187,5	205217,145	83,79

Sumber : Data Penawaran PT PLN PJB, JL. Ketingtang Baru NO. 11, Surabaya 60231

Catatan : Harga Batubara 253 Rp/Kg
 Harga MFO 1595 Rp/liter
 Harga HSD 1595 Rp/liter
 Harga Gas UP. Gresik 2,53 US\$/MMBTU
 Harga Gas UP. Gresik 2,45 US\$/MMBTU
 Nilai Tukar 9000 Rp/US\$

4.4. Aplikasi *Adaptive Hopfield* di PT.PLN Pembangkit Jawa-Bali

Perhitungan dan analisa ini dilakukan pada kebutuhan daya yang ditanggung PT.PLN Pembangkit Jawa Bali tanggal 10, 13 dan 14 Maret 2004. Analisa data dilakukan hanya untuk unit yang beroperasi, karena program komputer ini hanya untuk menghitung unit pembangkit yang siap beroperasi menjadi 22 unit yang dapat dilihat pada tabel 4.2.

Tabel 4.2
Unit Pembangkit Thermal Yang Siap Operasi

No	Nama Pembangkit
1	PLTU Paiton 1
2	PLTU Paiton 2
3	PLTGU Gresik GT 1.1
4	PLTGU Gresik GT 1.2
5	PLTGU Gresik GT 1.3
6	PLTGU Gresik ST 1.0
7	PLTGU Gresik GT 2.1
8	PLTGU Gresik GT 2.3
9	PLTGU Gresik GT 3.1
10	PLTGU Gresik GT 3.2
11	PLTGU Gresik GT 3.3
12	PLTGU Gresik ST 3.0
13	PLTU Gresik 1
14	PLTU Gresik 2
15	PLTU Gresik 3
16	PLTU Gresik 4
17	PLTGU M. Karang GT.1.1
18	PLTGU M. Karang GT.1.2

4.5. Beban Sistem

Dalam wilayah Jawa-Bali, pembangkit-pembangkit listrik yang ada dikoordinasi oleh PT. PLN Pembangkit Jawa-Bali. Proses *Economic Dispatch* dengan metode Jaringan Syaraf *Adaptive Hopfield* bertujuan untuk membuat rencana operasi yang optimum dalam sistem tenaga listrik yang dapat memenuhi kebutuhan beban dengan biaya operasi yang seekonomis mungkin.

Untuk mengetahui seberapa besar efisiensi dari metode ini, maka dilakukan evaluasi dengan mengambil data unit pembangkit thermal dan beban yang ditanggung oleh PT. PLN Pembangkit Jawa-Bali sebagai bahan perbandingan. Sedangkan kombinasi jadwal dan daya output pembangkit tenaga listrik dalam sistem PT. PLN Pembangkit Jawa-Bali tanggal 10, 13 dan 14 Maret 2004 terdapat pada lampiran. Untuk beban system terdapat pada tabel 4.3, 4.4 dan 4.5 (beban sistem yang ditanggung oleh pembangkit thermal saja).

Tabel 4.3
 Data Beban Unit Thermal pada PT.PLN PJB
 10 Maret 2004

JAM	Beban Sistem (MW)
01:00	3108
02:00	3024
03:00	2993
04:00	2974
05:00	2976
06:00	2916
07:00	2868
08:00	3202
09:00	3265
10:00	3281
11:00	3297
12:00	3220
13:00	3225
14:00	3226
15:00	3297
16:00	3372
17:00	3499
18:00	3600
19:00	3657
20:00	3642
21:00	3403
22:00	3388
23:00	3335
24:00	3316

Tabel 4.4
Data Beban Unit Thermal pada PT.PLN PJB
13 Maret 2004

JAM	Beban Sistem (MW)
01:00	2896
02:00	2864
03:00	2845
04:00	2866
05:00	2921
06:00	2806
07:00	2710
08:00	2856
09:00	3002
10:00	3020
11:00	3026
12:00	3030
13:00	3016
14:00	2901
15:00	2717
16:00	2796
17:00	2869
18:00	3374
19:00	3382
20:00	3373
21:00	3205
22:00	3015
23:00	2929
24:00	2869

Tabel 4.5
 Data Beban Unit Thermal pada PT.PLN PJB
 14 Maret 2004

JAM	Beban Sistem (MW)
01:00	2816
02:00	2678
03:00	2675
04:00	2694
05:00	2804
06:00	2611
07:00	2588
08:00	2746
09:00	2802
10:00	2816
11:00	2853
12:00	2789
13:00	2749
14:00	2657
15:00	2613
16:00	2709
17:00	2714
18:00	3255
19:00	3268
20:00	3269
21:00	2982
22:00	2876
23:00	2864
24:00	2822

4.6. Hasil Perhitungan dan Analisa Data

4.6.1. Hasil Perhitungan PT. PLN PJB

Dari data pembebanan harian pada lampiran dapat dihitung biaya operasional tiap jamnya dengan memasukkan ke persamaan fungsi biaya bahan bakar,. Sedangkan untuk hasil perhitungan beban dan biaya operasional perjamnya dapat dilihat pada tabel 4.6, 4.7 dan 4.8 dibawah ini.

Tabel 4.6
 Hasil Perhitungan Biaya Operasional Perjam PT. PLN PJB
 10 Maret 2004

Jam	Beban Sistem (MW)	PT. PLN PJB (Rp)
01:00	3108	680.306.108
02:00	3024	651.394.473
03:00	2993	646.667.419
04:00	2974	644.171.861
05:00	2976	640.361.923
06:00	2916	646.587.755
07:00	2868	626.319.642
08:00	3202	713.721.328
09:00	3265	725.817.463
10:00	3281	728.989.485
11:00	3297	732.577.863
12:00	3220	720.949.264
13:00	3225	723.177.188
14:00	3226	720.370.936
15:00	3297	733.154.110
16:00	3372	747.933.301
17:00	3499	805.181.358
18:00	3600	832.153.449
19:00	3657	844.634.828
20:00	3642	842.417.383
21:00	3403	757.459.805
22:00	3388	754.703.180
23:00	3335	744.635.040
24:00	3316	740.803.704

Tabel 4.7
 Hasil Perhitungan Biaya Operasional Perjam PT. PLN PJB
 13 Maret 2004

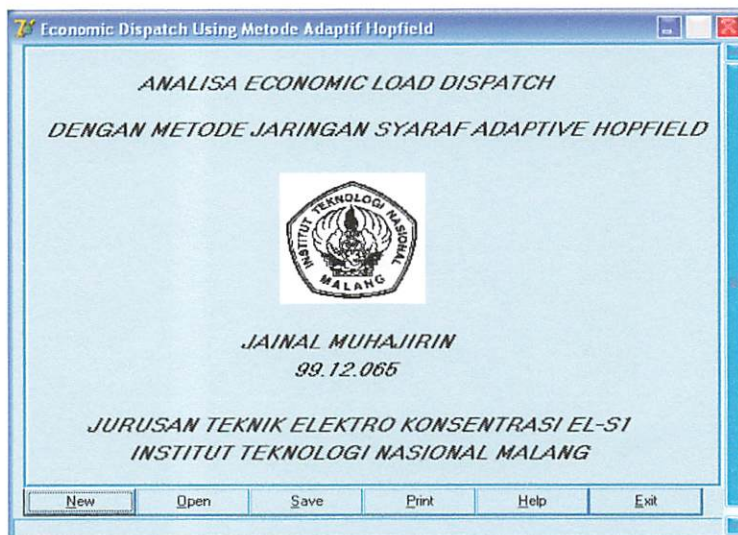
Jam	Beban Sistem (MW)	PT. PLN PJB (Rp)
01:00	2896	642.976.722
02:00	2864	627.840.757
03:00	2845	626.899.359
04:00	2866	629.878.888
05:00	2921	638.467.808
06:00	2806	622.986.844
07:00	2710	609.418.876
08:00	2856	629.093.132
09:00	3002	651.358.752
10:00	3020	655.538.620
11:00	3026	657.042.259
12:00	3030	658.662.629
13:00	3016	657.445.941
14:00	2901	637.045.376
15:00	2717	609.113.162
16:00	2796	622.239.104
17:00	2869	633.104.555
18:00	3374	755.125.576
19:00	3382	756.832.935
20:00	3373	755.769.182
21:00	3205	700.429.220
22:00	3015	657.757.376
23:00	2929	642.690.720
24:00	2869	632.703.376

Tabel 4.8
 Hasil Perhitungan Biaya Operasional Perjam PT. PLN PJB
 14 Maret 2004

JAM	Beban Sistem (MW)	PT. PLN PJB (Rp)
01:00	2816	618.251.323
02:00	2678	580.714.514
03:00	2675	580.742.086
04:00	2694	583.369.050
05:00	2804	601.290.761
06:00	2611	574.415.775
07:00	2588	570.967.006
08:00	2746	591.959.107
09:00	2802	601.814.600
10:00	2816	604.252.146
11:00	2853	611.074.946
12:00	2789	600.616.653
13:00	2749	595.203.725
14:00	2657	577.919.351
15:00	2613	570.955.908
16:00	2709	587.564.000
17:00	2714	588.687.073
18:00	3255	728.245.361
19:00	3268	730.147.988
20:00	3269	729.197.563
21:00	2982	639.723.642
22:00	2876	617.848.361
23:00	2864	615.562.714
24:00	2822	606.757.754

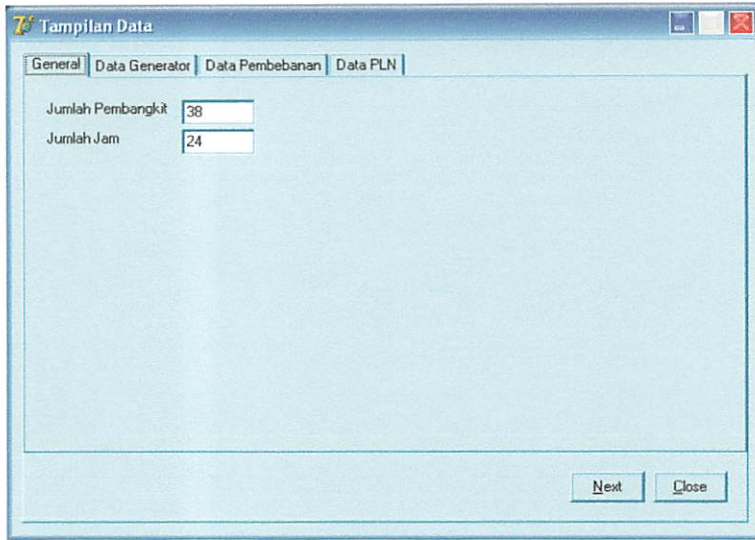
4.6.2. Tampilan Program Komputer dan Hasil Perhitungan dengan Metode Jaringan Syaraf *Adaptive Hopfield*

Tampilan program komputer yang telah dibuat ini adalah hasil perhitungan beban dan biaya operasional yang optimal, sedangkan tampilan utama dari program dapat dilihat pada gambar 4.1. dibawah ini.



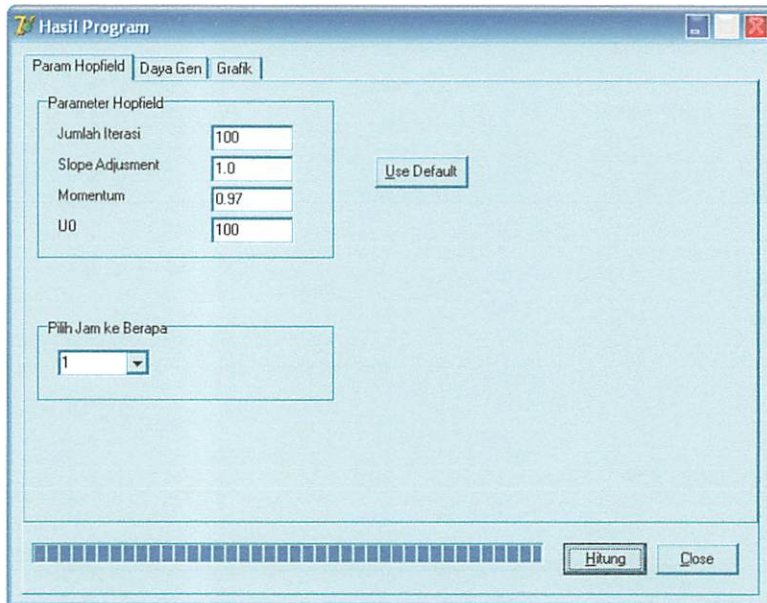
Gambar 4.1.
Tampilan Program Utama

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



Gambar 4.2.
Tampilan Input Data Pembangkit

Setelah data yang ada diinputkan kedalam komputer maka parameter-parameter yang digunakan adalah sebagai berikut :



Gambar 4.3.
Tampilan Parameter *Hopfield*

Gen	Nama	Pmax	Pmin	a0	a1	a2	Tup
1	PLTU Paton 1	370	225	3244978	111712.2	10.2971	72
2	PLTU Paton 2	370	225	3244978	111712.2	10.2971	72
3	PLTGU Gresik GT 1.1	102	53	5467532.4	217963.5	34.155	36
4	PLTGU Gresik GT 1.2	102	53	5467532.4	217963.5	34.155	36
5	PLTGU Gresik GT 1.3	102	53	5467532.4	217963.5	34.155	36
6	PLTGU Gresik ST 1.0	480	250	17177460.3	145165.581	4.554	36
7	PLTGU Gresik GT 2.1	102	53	5467532.4	217963.5	34.155	36
8	PLTGU Gresik GT 2.2	102	53	5467532.4	217963.5	34.155	36
9	PLTGU Gresik GT 2.3	102	53	5467532.4	217963.5	34.155	36
10	PLTGU Gresik ST 1.0	480	250	17177460.3	145165.581	4.554	36
11	PLTGU Gresik GT 3.1	102	53	5467532.4	217963.5	34.155	36
12	PLTGU Gresik GT 3.2	102	53	5467532.4	217963.5	34.155	36
13	PLTGU Gresik GT 3.3	102	53	5467532.4	217963.5	34.155	36
14	PLTGU Gresik ST 1.0	480	250	17177460.3	145165.581	4.554	36

Gambar 4.4.
Tampilan Data Pembangkit

	Load	Res
1	3108	400
2	3024	400
3	2993	400
4	2974	400
5	2976	400
6	2916	400
7	2868	400
8	3202	400
9	3265	400
10	3281	400
11	3297	400
12	3220	400
13	3225	400
14	3226	400
15	3207	400

Gambar 4.5.
Tampilan Data Pembebanan

	Jam 1	Jam 2	Jam 3	Jam 4	Jam 5	Jam 6	Jam 7	Jumlah
Gen 1	370	370	370	370	370	325	325	370
Gen 2	370	370	370	370	370	325	325	370
Gen 3	0	0	0	0	0	0	0	0
Gen 4	0	0	0	0	0	0	0	0
Gen 5	87	85	78	76	85	103	101	84
Gen 6	285	276	268	258	250	250	250	300
Gen 7	0	0	0	0	0	0	0	0
Gen 8	0	0	0	0	0	0	0	0
Gen 9	87	85	91	92	89	84	85	88
Gen 10	0	0	0	0	0	0	0	0
Gen 11	0	0	0	0	0	0	0	0
Gen 12	0	0	0	0	0	0	0	0
Gen 13	89	87	86	91	95	94	97	92
Gen 14	285	286	265	252	250	250	250	270

Gambar 4.6.
Tampilan Data PT. PLN PJB

Setelah menetapkan parameter tersebut maka komputasi dapat dilakukan, untuk menentukan kebutuhan beban yang disuplai dengan biaya yang minimum. Untuk proses komputasi digunakan software Borland Delphi versi 7.0 kemudian dieksekusi dengan menggunakan komputer berspesifikasi prosesor Intel 120 GHz Ram 256 Mb. Perhitungan dimulai pada jam 01:00 -24:00 selama 3 hari.

No	P Hopfd (kW)	P PLN (kW)	Cost Hopfd (Rp)	Cost PLN (Rp)	Selisih (Rp)
1	369	370	45.895.971	45.988.165	92.194
2	369	370	45.836.386	45.988.165	151.779
3	53	87	17.122.121	24.688.876	7.566.755
4	438	285	81.654.062	58.919.550	-22.734.513
5	56	87	17.738.918	24.688.876	6.949.958
6	65	89	19.741.452	25.136.826	5.395.374
7	478	285	87.647.110	58.919.550	-28.727.560
8	51	75	12.665.605	18.373.378	5.707.773
9	75	75	18.400.839	18.373.378	-27.461
10	101	90	24.154.099	21.816.918	-2.337.181
11	94	90	22.687.986	21.816.918	-871.068
12	453	435	82.625.417	80.069.945	-2.555.472
13	176	200	27.568.959	31.489.390	3.920.431
14	44	80	24.274.342	43.652.177	19.377.835
15	44	80	23.566.347	41.102.441	17.536.093

Biaya Hopfield: 623.080.162 Biaya PLN: 680.306.100 Selisih: 57.225.946

Hitung Close

Gambar 4.7.
 Hasil Optimasi Menggunakan Metode Jaringan Syaraf *Adaptive Hopfield*

4.6.2.1. Hasil Perhitungan Menggunakan Metode Jaringan Syaraf *Adaptive*

Hopfield

Berikut ini adalah hasil perhitungan biaya menggunakan metode Jaringan Syaraf *Adaptive Hopfield*

Tabel 4.9
Hasil Perhitungan Menggunakan
Metode Jaringan Syaraf *Adaptive Hopfield*
Rabu 10 Maret 2004

Jam	Beban Sistem (MW)	<i>Adaptive Hopfield</i> (Rp)
01:00	3108	623.080.162
02:00	3024	611.005.195
03:00	2993	613.458.967
04:00	2974	610.760.204
05:00	2976	606.592.740
06:00	2916	596.507.474
07:00	2868	592.219.847
08:00	3202	640.004.929
09:00	3265	661.065.167
10:00	3281	660.967.404
11:00	3297	661.472.766
12:00	3220	650.393.393
13:00	3225	647.386.367.
14:00	3226	647.047.958
15:00	3297	661.511.680
16:00	3372	679.184.696
17:00	3499	739.783.586
18:00	3600	761.965.592
19:00	3657	781.969.365
20:00	3642	775.757.757
21:00	3403	683.830.683
22:00	3388	682.280.314
23:00	3335	669.393.320
24:00	3316	667.917.550

Tabel 4.10
 Hasil Perhitungan Menggunakan
 Metode Jaringan Syaraf *Adaptive Hopfield*
 Sabtu 13 Maret 2004

Jam	Beban Sistem (MW)	<i>Adaptive Hopfield</i> (Rp)
01:00	2896	591.006.826
02:00	2864	593.119.569
03:00	2845	586.599.777
04:00	2866	592.280.510
05:00	2921	601.293.250
06:00	2806	579.930.349
07:00	2710	565.212.347
08:00	2856	588.257.008
09:00	3002	608.090.830
10:00	3020	616.350.396
11:00	3026	614.401.540
12:00	3030	614.321.981
13:00	3016	613.090.569
14:00	2901	593.238.859
15:00	2717	564.645.033
16:00	2796	579.726.274
17:00	2869	589.329.191
18:00	3374	676.642.296
19:00	3382	680.177.468
20:00	3373	677.778.127
21:00	3205	645.195.175
22:00	3015	609.888.611
23:00	2929	599.453.070
24:00	2869	591.046.224

Tabel 4.11
 Hasil Perhitungan Menggunakan
 Metode Jaringan Syaraf *Adaptive Hopfield*
 Minggu 14 Maret 2004

Jam	Beban Sistem (MW)	<i>Adaptive Hopfield</i> (Rp)
01:00	2816	579.348.294
02:00	2678	563.086.488
03:00	2675	559.725.375
04:00	2694	563.721.564
05:00	2804	576.714.951
06:00	2611	551.466.325
07:00	2588	548.911.274
08:00	2746	569.735.569
09:00	2802	575.610.386
10:00	2816	579.743.619
11:00	2853	589.028.432
12:00	2789	581.547.145
13:00	2749	571.769.067
14:00	2657	560.361.984
15:00	2613	553.896.933
16:00	2709	565.913.594
17:00	2714	565.243.620
18:00	3255	651.415.275
19:00	3268	655.095.378
20:00	3269	656.553.773
21:00	2982	606.155.654
22:00	2876	589.343.959
23:00	2864	595.354.791
24:00	2822	583.103.892

4.6.3. Perbandingan Hasil Perhitungan PT. PLN PJB dengan Metode Jaringan Syaraf Adaptive Hopfield.

Berikut ini adalah yang berisi perbandingan biaya pada PT. PLN PJB dan Jaringan Syaraf Adaptive Hopfield.

Tabel 4.12
Perbandingan Biaya Operasional Perjam
PT. PLN PJB dan Metode Jaringan Syaraf Adaptive Hopfield.
Rabu 10 Maret 2004

Jam	Beban Sistem (MW)	PT. PLN PJB (Rp)	Adaptive Hopfield (Rp)
01:00	3108	680.306.108	623.080.162
02:00	3024	651.394.473	611.005.195
03:00	2993	646.667.419	613.458.967
04:00	2974	644.171.861	610.760.204
05:00	2976	640.361.923	606.592.740
06:00	2916	646.587.755	596.507.474
07:00	2868	626.319.642	592.219.847
08:00	3202	713.721.328	640.004.929
09:00	3265	725.817.463	661.065.167
10:00	3281	728.989.485	660.967.404
11:00	3297	732.577.863	661.472.766
12:00	3220	720.949.264	650.393.393
13:00	3225	723.177.188	647.386.367.
14:00	3226	720.370.936	647.047.958
15:00	3297	733.154.110	661.511.680
16:00	3372	747.933.301	679.184.696
17:00	3499	805.181.358	739.783.586
18:00	3600	832.153.449	761.965.592
19:00	3657	844.634.828	781.969.365
20:00	3642	842.417.383	775.757.757
21:00	3403	757.459.805	683.830.683
22:00	3388	754.703.180	682.280.314
23:00	3335	744.635.040	669.393.320
24:00	3316	740.803.704	667.917.550

Tabel 4.13
 Perbandingan Biaya Operasional Perjam
 PT. PLN PJB dan Metode Jaringan Syaraf *Adaptive Hopfield*.
 Sabtu 13 Maret 2004

Jam	Beban Sistem (MW)	PT. PLN PJB (Rp)	<i>Adaptive Hopfield</i> (Rp)
01:00	2896	642.976.722	591.006.826
02:00	2864	627.840.757	593.119.569
03:00	2845	626.899.359	586.599.777
04:00	2866	629.878.888	592.280.510
05:00	2921	638.467.808	601.293.250
06:00	2806	622.986.844	579.930.349
07:00	2710	609.418.876	565.212.347
08:00	2856	629.093.132	588.257.008
09:00	3002	651.358.752	608.090.830
10:00	3020	655.538.620	616.350.396
11:00	3026	657.042.259	614.401.540
12:00	3030	658.662.629	614.321.981
13:00	3016	657.445.941	613.090.569
14:00	2901	637.045.376	593.238.859
15:00	2717	609.113.162	564.645.033
16:00	2796	622.239.104	579.726.274
17:00	2869	633.104.555	589.329.191
18:00	3374	755.125.576	676.642.296
19:00	3382	756.832.935	680.177.468
20:00	3373	755.769.182	677.778.127
21:00	3205	700.429.220	645.195.175
22:00	3015	657.757.376	609.888.611
23:00	2929	642.690.720	599.453.070
24:00	2869	632.703.376	591.046.224

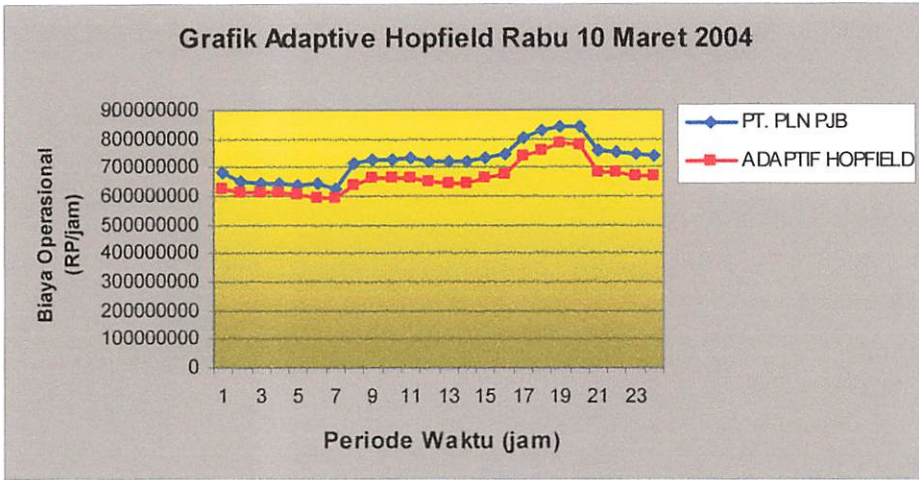
Tabel 4.14
Perbandingan Biaya Operasional Perjam
PT. PLN PJB dan Metode Jaringan Syaraf *Adaptive Hopfield*.
Minggu 14 Maret 2004

Jam	Beban Sistem (MW)	PT. PLN PJB (Rp)	<i>Adaptive Hopfield</i> (Rp)
01:00	2816	618.251.323	579.348.294
02:00	2678	580.714.514	563.086.488
03:00	2675	580.742.086	559.725.375
04:00	2694	583.369.050	563.721.564
05:00	2804	601.290.761	576.714.951
06:00	2611	574.415.775	551.466.325
07:00	2588	570.967.006	548.911.274
08:00	2746	591.959.107	569.735.569
09:00	2802	601.814.600	575.610.386
10:00	2816	604.252.146	579.743.619
11:00	2853	611.074.946	589.028.432
12:00	2789	600.616.653	581.547.145
13:00	2749	595.203.725	571.769.067
14:00	2657	577.919.351	560.361.984
15:00	2613	570.955.908	553.896.933
16:00	2709	587.564.000	565.913.594
17:00	2714	588.687.073	565.243.620
18:00	3255	728.245.361	651.415.275
19:00	3268	730.147.988	655.095.378
20:00	3269	729.197.563	656.553.773
21:00	2982	639.723.642	606.155.654
22:00	2876	617.848.361	589.343.959
23:00	2864	615.562.714	595.354.791
24:00	2822	606.757.754	583.103.892

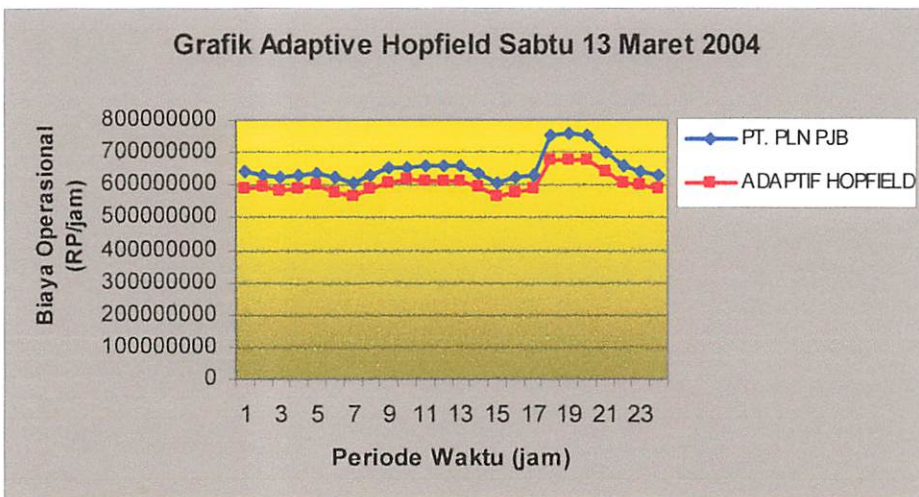
Tabel 4.15
Perbandingan Total Biaya Operasional
PT. PLN PJB Dengan Metode Jaringan Syaraf *Adaptive Hopfield*.

Periode Waktu (24 jam)	PT.PLN PJB (Rp)	<i>Adaptive Hopfield</i> (Rp)	Penghematan (Rp)
10 Maret 2004	17.404.488.866	15.925.557.116	1.478.931.750
13 Maret 2004	15.710.421.169	14.571.075.280	1.139.345.889
14 Maret 2004	14.707.281.407	13.992.847.342	714.434.065

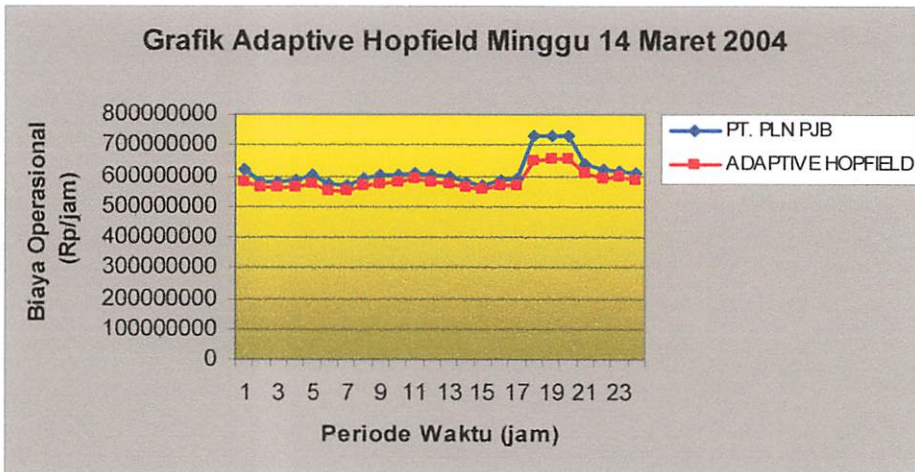
Grafik 4.1
 Perbandingan Biaya Operasional Perjam
 PT. PLN PJB dan Metode Jaringan Syaraf *Adaptive Hopfield*.
 Rabu 10 Maret 2004



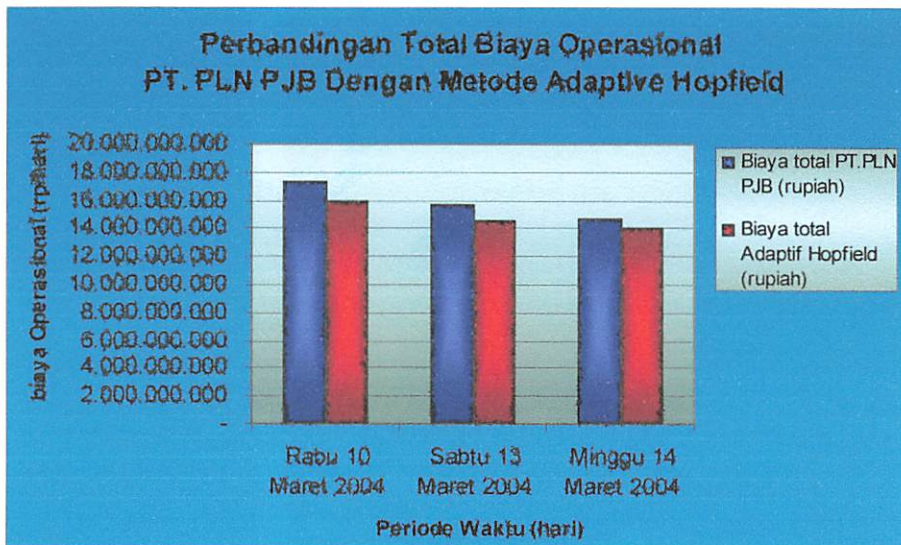
Grafik 4.2
 Perbandingan Biaya Operasional Perjam
 PT. PLN PJB dan Metode Jaringan Syaraf *Adaptive Hopfield*.
 Sabtu 13 Maret 2004



Grafik 4.3
 Perbandingan Biaya Operasional Perjam
 PT. PLN PJB dan Metode Jaringan Syaraf *Adaptive Hopfield*.
 Minggu 14 Maret 2004



Grafik 4.4
 Perbandingan Total Biaya Operasional
 PT. PLN PJB Dengan Metode Jaringan Syaraf *Adaptive Hopfield*.



BAB V

KESIMPULAN

5.1. Kesimpulan

Dari analisa program dan hasil perhitungan terhadap penggunaan Metode Jaringan Syaraf *Adaptive Hopfield* pada *economic dispatch* beban yang ditanggung PT. PLN Pembangkitan Jawa-Bali maka dapat diambil suatu kesimpulan:

1. Proses metode Jaringan Syaraf *Adaptive Hopfield* memberikan sebuah analisis penyelesaian yang cukup efektif dalam mengoptimalkan pembebanan dan sekaligus penghematan biaya total operasional PT. PLN PJB. Pada tanggal 10 Maret 2004 selisih biaya total PT. PLN PJB dengan metode Jaringan Syaraf *Adaptive Hopfield* sebesar Rp. 1.478.931.750 dioptimalkan 8.50%. Pada tanggal 13 Maret 2004 selisih biaya total PT. PJB dengan Jaringan Syaraf *Adaptive Hopfield* sebesar Rp. 1.139.345.889 dapat dioptimalkan 7.25%. Pada tanggal 14 Maret 2004 metode Jaringan Syaraf *Adaptive Hopfield* sebesar Rp. 714.434.065 dapat dioptimalkan sebesar 4.86%.

5.2. Saran

Berdasarkan kesimpulan diatas, ada saran yang mungkin bisa diajukan, antara lain:

1. Melakukan studi lebih lanjut untuk melakukan optimasi pada jumlah unit yang lebih besar atau jangka waktu yang dipelajari lebih lama untuk mengetahui sejauh mana kemampuan analisa dari metode ini apakah masih relevan untuk digunakan dalam sistem yang lebih besar.
2. Memungkinkan untuk dapat diaplikasikan pada PT. PLN Pembangkit Tenaga Listrik Jawa-Bali karena menghasilkan biaya operasional yang lebih ekonomis.

DAFTAR PUSTAKA

- [1]. A.J Wood dan B. F. Wollenberg, “ *Power Generation Operation, and Control* “ Second Edition, John Wiley & Sons. 1996.
- [2]. Djiteng Marsudi . Ir., “ *Operasi Sistem Tenaga Listrik* “, ISTN 1990
- [3]. William D. Stevenson, Jr. “ *Analisa Sistem Tenaga Listrik* “ , Edisi Keempat, Erlangga, Jakarta, 1993.
- [4]. Kwang Y. Lee, Arthit Sode-Yome and June Ho Park, “Adaptive Hopfield Neural Network for Economic Load Dispatch”, IEEE Transactions on Power System, Vol. 13, no.2, May 1998.
- [5]. Kusumadewi, Sri., “*Artificial intelegence*”, Graha ilmu, Jakarta, 2003.

- Berita Acara Ujian Skripsi
- Lembar Bimbingan Skripsi
- Formir bimbingan Skripsi
- Surat permohonan penelitian
- Data Perkembangan Jurnal P.I. PPN PJB
- Sisting Program

Lampiran



BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

1. Nama : **JAINAL MUHAJIRIN**
2. N.I.M. : **99.12.065**
3. Jurusan : **Teknik Elektro S-1**
4. Konsentrasi : **Teknik Energi Listrik**
5. Judul Skripsi : **ANALISA *ECONOMIC LOAD DISPATCH* DENGAN
METODE JARINGAN SYARAF *ADAPTIVE HOPFIELD***

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

6. Hari : **Jum'at**
7. Tanggal : **1 April 2005**
8. Dengan nilai : **72,8 (B)** *hm*



(Ir. Mochtar Asroni, MSME)

Ketua

Panitia Ujian Skripsi

(Ir. F. Yudi Limpraptono, MT)

Sekretaris

Anggota Penguji

(Ir. Yusuf Ismail Nakhoda, MT)

Penguji Pertama

(Ir. I Made Wartana, MT)

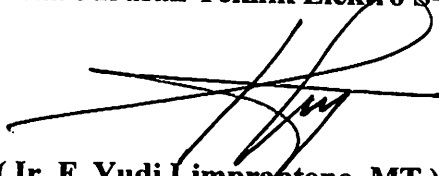
Penguji Kedua



LEMBAR BIMBINGAN SKRIPSI

Nama : JAINAL MUHAJIRIN
N.I.M. : 99.12.065
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : ANALISIA *ECONOMIC LOAD*
DISPATCH DENGAN METODE
JARINGAN SYARAF *ADAPTIVE*
HOPFIELD
Tanggal Mengajukan Skripsi : 15 November 2004
Tanggal Menyelesaikan Skripsi : 30 Maret 2005
Dosen Pembimbing : Ir. Choirul Saleh, MT.
Telah dievaluasi dengan nilai : 85 (Delapan puluh Lima) *85*

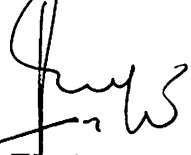
Mengetahui,
Ketua Jurusan Teknik Elektro S-1


(Ir. F. Yudi Limpraptono, MT)
NIP. Y. 103 950 0274

Diperiksa dan disetujui,
Dosen Pembimbing I


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

Dosen Pembimbing II


(Ir. Eko Nurcahyo)
NIP. 102 870 0172



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

FORMULIR BIMBINGAN SKRIPSI

Nama : JAINAL MUHAJIRIN
Nim : 99.12.065
Masa Bimbingan : 7 Februari 2005 s/d 7 Agustus 2005
Judul Skripsi : ANALISIA *ECONOMIC LOAD DISPATCH*
DENGAN METODE JARINGAN SYARAF
ADAPTIVE HOPFIELD

No.	Tanggal	Uraian	Parap Pembimbing
1.	12-03-2005	ACC Bab I	<i>CS</i>
2.	12-03-2005	Refisi Bab II. Teori Dasar Sistem Tenaga Listrik	<i>CS</i>
3.	14-03-2005	ACC Bab II	<i>CS</i>
4.	14-03-2005	Refisi Bab III. Cantumkan Flow Chart	<i>CS</i>
5.	19-03-2005	ACC Bab III	<i>CS</i>
6.	19-03-2005	ACC Bab IV	<i>CS</i>
7.	19-03-2005	ACC Bab V + Buat Abstraksi	<i>CS</i>
8.	19-03-2005	ACC Makalah Seminar Hasil	<i>CS</i>
9.	19-03-2005	ACC Ujian Skripsi	<i>CS</i>
10.			

Malang, 19 Maret 2005
Dosen Pembimbing,


Ir. Choirul Saleh, MT

Form. S-4b



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

FORMULIR BIMBINGAN SKRIPSI

Nama : JAINAL MUHAJIRIN
Nim : 99.12.065
Masa Bimbingan : 7 Februari 2005 s/d 7 Agustus 2005
Judul Skripsi : ANALISIA *ECONOMIC LOAD DISPATCH*
DENGAN METODE JARINGAN SYARAF
ADAPTIVE HOPFIELD

No.	Tanggal	Uraian	Parap Pembimbing
1.	12-03-2005	ACC Bab I	
2.	12-03-2005	Refisi Bab II. Teori Dasar Sistem Tenaga Listrik	
3.	14-03-2005	ACC Bab II	
4.	14-03-2005	Refisi Bab III. Cantumkan Flow Chart	
5.	19-03-2005	ACC Bab III	
6.	19-03-2005	ACC Bab IV	
7.	19-03-2005	ACC Makalah Seminar Hasil	
8.	19-03-2005	ACC Ujian Skripsi	
9.			
10.			

Malang, 19 Maret 2005
Dosen Pembimbing,

Ir. Eko Nurcahyo

Form. S-4b



Nomor : E2042335
Sifat : Biasa
Lampiran :-

Surabaya, 5 April 2005

KEPADA
Yth. DEKAN FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Bendungan Sigura-gura No. 2
MALANG - 65145

Perihal : Ijin Penelitian

Menunjuk surat Saudara, No.ITN.064/III.TA/2/2004 tgl. 21 September 2004;No.ITN-639/III.TA/2/2004 tgl. 9 Desember 2004;No.ITN.672/III.TA/2/2004 tgl. 14 Desember 2004;No.ITN.859/III.TA/2/2004 tgl. 1 Pebruari 2005;No.ITN-860/III.TA/2/2005 tgl. 1 Pebruari 2005;No.ITN.937/III.TA/2/2005 tgl. 24 Pebruari 2005;No.ITN.900/III.TA/2/2005 tgl. 19 Pebruari 2005;No.ITN.844/III.TA/2/2005 tgl. 29 Januari 2005, perihal : Survey / Permintaan Data untuk Tugas Akhir, maka perlu kami sampaikan hal-hal sebagai berikut :

1. Pada prinsipnya kami memberikan ijin kepada mahasiswa Saudara yang bernama :
 - o MEDYAYANTO JATI K - Nim 99 12 194
 - o JAINAL MUHAJIRIN - Nim 99 12 065
 - o CHOIRUL ANWAR - Nim 99 12 130
 - o VICKY DEWANTO - Nim 99 12 192
 - o HENDRA AGUS WAHYUDI - Nim 99 12 200
 - o VIDYA IS HINDRATNO - Nim 99 12 165
 - o ROBERT PARULIAN - Nim 97 12 070
 - o SAMUEL JULIAG T.A - Nim 97 12 068Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Nasional Malang, untuk melakukan Survey/Permintaan Data di PT Pembangkitan Jawa Bali - Kantor Pusat.
2. Kegiatan dimaksud akan dilaksanakan di PT Pembangkitan Jawa Bali - Kantor Pusat - Sub Direktorat Managemen Energi (EMG), Jl. Ketintang Baru No. 11 Surabaya 60231, selama kurun waktu 1 (satu) bulan, terhitung mulai tanggal 15 Maret 2005 sampai dengan 15 April 2005.
3. Kepada yang bersangkutan diberlakukan ketentuan bahwa :
 - o Tidak diperkenankan memanfaatkan data dan informasi untuk kepentingan di luar PT PJB sebelum mendapat persetujuan dari Pejabat yang berwenang.
 - o Seusai kegiatan harus membuat laporan kegiatan yang telah disahkan oleh pejabat PT PJB yang berwenang.
 - o Selama melaksanakan kegiatan wajib mengikuti ketentuan dan aturan yang berlaku bagi Karyawan / Tamu di lingkungan PT PJB.

Atas perhatian Saudara, kami sampaikan terima kasih.

Hormat kami,
Manajer Pelatihan dan Pengembangan

Gede Adnya Karsana

PT PEMBANGKITAN JAWA - BALI

Jl. Ketintang Baru 11 Surabaya 60231 - Indonesia, ☎ : 62-31-8283180 (hunting) Fax : 62-31-8291280 E-mail : pjb@pjb2.com

Tembusan

1. Manajer Pusat Manajemen Energi (EMC)



2. Slamet Mudani

3. Medyayanto Jati K

4. Jainal Muhajirin

5. Choirul Anwar

6. Vicky Dewanto

7. Hendra Agus Wahyudi

8. Vidya Is Hindratno

9. Robert Parulian

10. Samuel Juliag T.A



DATA PENAWARAN
PT PLN PEMBANGKITAN JAWA BALI
AGUSTUS 2002

No.	NAMA PEMBANGKIT	KAPASITAS			LAMA WAKTU (JAM)				BIAYA START UP (JUTA Rp)		KOEFSIEN BIAYA BAHAN BAKAR		
		Daya Terpasang	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
P	UP. PAITON PLTU #1/2 (COAL)	2 x 400	225	370	72	48	17	4	682.98	149.68	3244978	111712.15	10.2971
2	UP. GRESIK												
ST-6	GT 1-9 OC (GAS)	9 x 112	53	102	36	10	1	0	7.82	0	5467532.4	217963.548	34.155
	CC - 1.1.1 (GAS)		115	143	36	10	3	1	57.68	31.46	10936203.3	72527.004	368.874
	CC - 2.2.1 (GAS)		164	314	36	10	3	2	65.5	39.28	11795770.8	152515.737	6.831
	CC - 3.3.1 (GAS)	3 x 526	250	480	36	10	3	2	73.32	47.1	17177460.3	145165.581	4.554
	PLTU # 1/2 (GAS)	100	43	85	48	10	9	1	143.74	40.59	1327126.68	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
3	UP. MUARA KARANG												
ST-10	GT 1/2/3 - OC	3 x 107	50	95	36	10	1	0	7.35	0	5730795	202052.97	108.045
	CC - 1.1.1 (GAS)	153	110	150	36	10	3	1	54.22	29.67	11560815	53685.135	460.845
	CC - 2.2.1 (GAS)	317	200	300	36	10	3	2	61.57	36.92	16010064	127208.655	35.28
	CC - 3.3.1 (GAS)	508	300	465	36	10	3	2	68.92	44.27	31017735	87825.15	57.33
	MTW GT 1/2 - OC (HSD)	2 x 140	72	138	36	10	0	0	0	0	14706521.25	433337.8	49.4605
	MTW CC - 1.1.1 (HSD)	200	162	202	36	10	3	1	118.08	64.4	672630	144191.717	519.1757
	MTW CC - 2.2.1 (HSD)	420	210	403	36	10	3	2	134.1	80.42	30123040	303208.82	11.64715
	MTW CC - 3.3.1 (HSD)	640	315	605	36	10	3	2	160.1	96.42	43043399	288609.995	7.6584
ST-10	PLTU # 1/2/3 (MFO)	3 x 100	44	85	48	10	6	1	122.58	31.08	2417820.7	473895.41	120.77935
	PLTU # 4/5 (Gas)	2 x 200	90	165	48	10	11	2	215.34	89.29	2949187.5	205217.145	83.79

RENCANA : HARI/TANGGAL: RABU, 10 MARET 2004

SUB SISTEM REGION_1

PT. PLN PEMBANGKITAN TENAGA LISTRIK JAWA-RAJU

PLTU	Kapasitas (MW)	Jan																								
		00.30	01.00	01.30	02.00	02.30	03.00	03.30	04.00	04.30	05.00	05.30	06.00	06.30	07.00	07.30	08.00	08.30	09.00	09.30	10.00	10.30	11.00	11.30	12.00	12.30
PLTU	MKRNG10C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG0C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG10G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTU	MKRNG20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG10G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG20G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG30G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTU	MKRNG30C	435	435	410	410	410	410	410	410	410	410	410	410	410	425	425	410	410	410	410	410	410	410	410	410	410
	MKRNG	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165
	#4	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165
	#5	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165	165
	MKRNG	80	80	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85
	#1	80	80	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85
	#2	80	80	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85
	#3	80	80	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85
	MWVAR110C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MWVAR10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MWVAR20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MWVAR120C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MWVAR130C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MWVAR110G	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	
MWVAR110G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MWVAR120G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MWVAR130G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MWVAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MWVAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
PEMBELAN DARI LUAR PLN																										
PLTU	CSLAK #4	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182	182
	CSLAK #5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	CSLAK #6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	CSLAK #7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	CSLAK #8	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
PLTU	Keretapi Steel	4655	4553	4508	4453	4453	4453	4453	4453	4453	4453	4453	4453	4453	4453	4443	4393	4343	4125	4125	4328	4483	4591	4591	4591	4591
	Green Area-1	4482	4438	4379	4323	4224	4180	4143	4105	4309	4228	4093	4093	4151	4207	4445	4718	4819	4780	4859	4888	4830	4885	4881	4708	4541
	Sekah (*) (*)	173	117	129	130	229	273	310	348	241	322	350	250	-26	-62	-117	-225	-278	-189	-388	-297	-239	-284	-288	-175	-80
	Cadangan Sekeloa	54	108	104	121	121	121	121	121	70	70	130	125	172	172	178	123	50	50	50	50	50	50	50	50	75
	Cadangan Pater	189	272	318	372	372	372	372	372	275	275	282	482	878	875	472	342	244	244	244	244	244	244	244	244	284
	Green Area-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Green Area-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Green Area-4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Green Area-5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Green Area-6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*) Pembangkitan Area -1

RENCANA : HARI/TANGGAL: SABTU, 13 MARET 2004

SUB SISTEM REGION_1

PT. PLN PERBAHANGKIZAN TENAGA LISTRIK JAWA, GALLI

		Jan	13.00	13.30	14.00	14.30	15.00	15.30	16.00	16.30	17.00	17.30	18.00	18.30	19.00	19.30	20.00	20.30	21.00	21.30	22.00	22.30	23.00	23.30	24.00	Form-2	
PLTU	MKRNG10C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MKRNG10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MKRNG20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MKRNG20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MKRNG30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MKRNG30C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MKRNG10C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MKRNG20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG30C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTU	MKRNG #4	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	107	
	MKRNG #5	167	165	165	165	150	150	150	150	150	150	150	150	150	170	170	170	170	170	170	170	170	170	170	155	155	
	MKRNG #1	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	
	MKRNG #2	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	
	MKRNG #3	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	
PLTU	MWVAR110C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MWVAR110C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MWVAR20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MWVAR120C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MWVAR130C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MWVAR110C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MWVAR110C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	MWVAR20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MWVAR130C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MWVAR130C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTU	MWVAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	01 L9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
PEMBELIAN DARI LUAR PLN																											
PLTP	GSLAK #4	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	175	
	#5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	#6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	#7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	#8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	#9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
PLTU	Kerabatu Steel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	4425	4432	4398	4392	4345	4345	4353	4350	4403	4078	5078	5078	5078	5078	5078	5078	5078	4853	4833	4023	4823	4013	4753	4700	4853	4562	
	4637	4605	4656	4656	4604	4591	4562	4533	4702	5013	5100	5081	5104	5123	5118	5118	5119	4918	4809	4653	4639	4498	4407	4265	4609	4609	
	212	-253	-258	-284	-259	-246	-209	-175	-219	-335	172	-53	-75	-85	-90	-106	-85	14	135	174	235	233	333	333	333	-27	
	73	74	74	72	72	69	69	70	76	87	44	44	44	44	44	44	44	63	66	88	88	88	140	140	140	86	
	404	384	384	429	429	599	599	591	544	329	114	114	114	114	114	114	114	189	309	319	319	319	382	382	382	346	

Jam	00.30	01.00	01.30	02.00	02.30	03.00	03.30	04.00	04.30	05.00	05.30	06.00	06.30	07.00	07.30	08.00	08.30	09.00	09.30	10.00	10.30	11.00	11.30	12.00	12.30	
PLTA Area 4	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19
SUTAMI	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
BRANTAS	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66
PLTU #1	370	370	350	350	350	350	350	370	370	370	370	370	370	316	316	350	360	360	360	360	360	360	360	360	360	360
PLTU #2	370	370	350	350	350	350	350	370	370	370	370	370	370	316	316	350	360	360	360	360	360	360	360	360	360	360
GRSIR110C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR110C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR210C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR120C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR130C	85	85	78	78	78	77	77	87	87	87	102	102	102	100	100	85	85	90	90	103	103	105	105	110	110	112
GRSIR110C	155	155	155	131	131	148	148	155	155	155	160	160	160	161	161	177	177	177	177	175	175	181	181	182	182	182
GRSIR120C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR130C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR210C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR10C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR220C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR230C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR310C	87	87	91	90	90	92	92	97	97	87	83	83	83	85	85	85	87	87	87	93	93	95	95	102	102	103
GRSIR310C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR320C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR330C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR310C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR320C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR330C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR310C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR320C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR330C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR310C	134	134	147	147	142	146	146	148	158	158	158	158	158	136	136	132	132	155	155	154	147	147	147	157	157	155
GRSIR320C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR330C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR310C	120	120	122	122	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	120	120	121	121	121	121	121
GRSIR320C	121	121	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120
GRSIR330C	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
GRSIR310C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR320C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR330C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSIR310C	3004	3300	3256	3269	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287	3287
GRSIR320C	2068	2049	2022	1999	2093	2096	2077	2073	2075	2182	1963	1757	1717	1892	1862	2082	2089	2082	2089	2030	2092	1998	2038	2073	2027	2027
GRSIR330C	1238	1251	1236	1260	1194	1191	1210	1214	1212	1105	1319	1217	1127	1433	1460	1440	1453	1460	1440	1453	1492	1430	1528	1484	1409	1260
GRSIR310C	65	65	100	108	108	106	108	108	72	72	67	93	92	96	96	59	61	61	61	61	61	61	61	61	67	63
GRSIR320C	604	604	644	695	695	695	695	695	674	674	624	624	624	624	624	624	624	624	624	624	624	624	624	624	624	
GRSIR330C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

(*) Production Area -1
 (*) Production Area -1
 (*) Production Area -1
 (*) Production Area -1
 (*) Production Area -1

PT. PLN PEMBANGKITAN TENAGA LISTRIK JAWA-DALI

Jam	13.00	13.30	14.00	14.30	15.00	15.30	16.00	16.30	17.00	17.30	18.00	18.30	19.00	19.30	20.00	20.30	21.00	21.30	22.00	22.30	23.00	23.30	24.00	Ref-2	
PLTA Area 4																									
PLTA SUJAM	21	21	21	21	21	21	21	21	23	23	23	23	23	23	23	23	23	23	21	21	21	21	21	21	20
PLTA BRANTAS	90	90	90	90	90	90	90	90	100	100	100	100	100	100	100	100	100	100	90	90	90	90	90	90	92
	53	53	53	53	53	53	53	53	116	116	116	116	116	116	116	116	116	116	66	66	66	66	66	66	34
PLTU																									
PITON #1	360	360	360	360	360	360	360	360	360	360	370	370	370	370	370	370	370	370	370	370	370	370	370	370	360
PITON #2	360	360	360	360	360	360	360	360	360	360	350	370	370	370	370	370	370	370	370	370	370	370	370	370	359
GRSK110C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK120C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK130C	112	90	90	102	99	99	99	101	115	115	115	115	115	115	116	116	105	102	102	99	99	99	99	99	87
GRSK110C	182	100	100	153	153	175	175	175	185	185	186	186	186	186	186	186	175	174	174	168	168	168	150	175	
GRSK120C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK130C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK10C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK20C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK220C	100	93	93	83	83	97	97	98	98	115	115	115	115	115	116	116	115	115	97	97	92	92	87	87	87
GRSK20C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK210C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK220C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK230C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK310C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK10C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK20C3	96	87	87	89	89	96	96	98	98	105	105	106	106	105	105	105	97	97	87	87	85	85	85	85	87
GRSK30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK310C	155	156	156	126	126	157	157	157	162	162	162	163	163	162	162	162	156	156	156	146	146	135	135	155	
GRSK320C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSK330C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTU																									
GRSJK #3	121	121	121	121	120	119	119	119	172	172	172	175	175	175	175	174	174	174	114	114	115	114	114	121	
GRSJK #4	121	121	121	122	122	121	121	125	179	179	180	180	180	180	181	181	180	180	122	122	120	120	120	121	
GRSJK #1	73	73	73	73	73	73	73	73	88	88	88	88	88	89	89	89	89	89	72	72	73	73	73	73	
GRSJK #2	73	73	73	73	73	73	73	72	88	88	88	88	88	88	88	88	87	87	72	72	73	73	73	73	
PLTG																									
GLTMR #1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GLTMR #2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSJK #1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSJK #2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRSJK #3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pembangkitan Area-1	3517	3510	3477	3426	3426	3476	3426	3426	3426	3426	4353	4353	4353	4353	4353	4253	4098	3642	3727	3702	3562	3482	3342	3543	
Pembangkitan Area-1	2050	2214	2210	2230	1858	1870	1894	1939	2190	2170	2887	2897	2891	2841	2747	2660	2525	2418	2246	2138	2162	2182	2088	2157	
Saluran Tenaga (MVA)	1467	1308	1287	1196	1568	1546	1532	1487	1227	606	1466	1446	1456	1472	1497	1488	1430	1417	1311	1458	1424	1320	1254	1386	
Saluran Tenaga	65	63	62	62	106	108	110	112	115	88	37	37	37	37	44	62	56	75	69	68	66	70	68	69	
Saluran Tenaga	604	879	655	855	855	815	815	965	1035	552	227	227	227	227	242	347	482	587	719	744	784	784	804	804	

SUB SISTEM REGION_1

RENCANA : HARI/TANGGAL: MINGGU, 14 MARET 2004

PT. PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

Jenis	Jam																										
	00.30	01.00	01.30	02.00	02.30	03.00	03.30	04.00	04.30	05.00	05.30	06.00	06.30	07.00	07.30	08.00	08.30	09.00	09.30	10.00	10.30	11.00	11.30	12.00	12.30		
PLTGU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG10C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG30C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG10C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MKRNG30C	410	410	400	400	400	400	400	400	400	400	400	350	350	350	400	400	425	425	425	425	425	425	400	400	360		
PLTU	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90		
#4	150	150	150	150	150	150	150	150	150	150	150	165	165	155	155	165	165	165	165	185	185	165	150	150	150		
#5	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50		
#1	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50		
#2	65	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50		
#3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
PLTGU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI10C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI30C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI10C	350	350	325	325	325	325	325	325	325	300	300	300	300	300	350	350	350	350	350	350	350	350	350	350	350		
MITWARI20C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWARI30C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWAR GT 2.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
MITWAR GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
PEMBELIAN DARI LUAR PLN																											
PLTIP	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	150	
#4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
#5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
#6	100	100	100	100	100	100	100	100	100	100	100	100	100	100	75	75	110	110	110	110	110	110	110	100	100	100	
PLTU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Kanalisasi Steel	4520	4520	4587	4453	4453	4453	4453	4453	4550	4443	4425	4125	4125	4328	4457	4591	4591	4591	4591	4591	4591	4591	4573	4573	4453	4453	
(*) Pembangkitan Area -1	4482	4436	4379	4323	4224	4180	4143	4105	4309	4432	4093	4015	4120	4207	4445	4718	4819	4780	4859	4868	4850	4865	4861	4708	4541	4541	
(**) Bahan Area-1	38	64	208	130	229	273	310	346	241	118	350	410	5	-82	-117	-281	-228	-189	-288	-237	-238	-294	-268	-175	-68	-68	
Solasi (*) - (*)	54	108	104	121	121	121	121	121	70	70	130	125	172	172	128	123	50	50	50	50	50	50	50	70	75	75	
Cadangan Solasi*	169	272	316	372	372	372	372	372	275	275	382	482	675	675	472	342	244	244	244	244	244	244	244	284	284	364	
Udangan Putir																											

unit uAbout;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
Dialogs;

type

```
TfrmAbout = class(TForm)
private
    { Private declarations }
public
    { Public declarations }
end;
```

var

frmAbout: TfrmAbout;

implementation

{ \$R *.dfm }

end.

unit uComplex;

interface

uses uUtils;

type

```
TComplex=class
private
    FReal, FImag:double;
public
    constructor Create; overload;
    constructor Create(const aReal:double); overload;
    constructor Create(const aReal, aImag:double); overload;
    constructor Create(const aComplex:TComplex); overload;
    function GetAbs:double;
    function GetAngleRad:double;
    function GetAngleDeg:double;
    function Add(const aReal:double):TComplex; overload;
    function Add(const aComplex:TComplex):TComplex; overload;
    function Subtract(const aReal:double):TComplex; overload;
    function Subtract(const aComplex:TComplex):TComplex; overload;
    function Multiply(const aReal:double):TComplex; overload;
    function Multiply(const aComplex:TComplex):TComplex; overload;
    function Divide(const aReal:double):TComplex; overload;
    function Divide(const aComplex:TComplex):TComplex; overload;
    function Conj:TComplex;
    function Negative:TComplex;
    function toStringI(const rLen:integer):string;
    function toStringJ(const rLen:integer):string;
```

```
end;  
end;
```

```
function MatrixTranspose(const mat1:dArr2):dArr2;  
var i,j:integer;  
begin  
  SetLength(result,high(mat1)+1,high(mat1[0])+1);  
  for i:=0 to high(mat1) do  
    begin  
      for j:=0 to high(mat1[0]) do  
        begin  
          result[j,i]:=mat1[i,j];  
        end;  
      end;  
    end;  
end;
```

```
function MatrixNegative(const mat1:dArr2):dArr2;  
var i,j:integer;  
begin  
  SetLength(result,high(mat1)+1,high(mat1[0])+1);  
  for i:=0 to high(mat1) do  
    begin  
      for j:=0 to high(mat1[0]) do  
        begin  
          result[i,j]:=mat1[i,j]*-1;  
        end;  
      end;  
    end;  
end;
```

```
function EllGauss(const mat1:dArr2;const mat2:dArr1):dArr1;  
var i,j,k,Nmat:integer;  
    konst,value,DE,AE,sum:double;  
    tmp:dArr2;  
begin  
  if high(mat1)<>high(mat1[0]) then  
    begin  
      raise Exception.Create('Matrik 1 bukan matrik bujur  
sangkar!');  
    end;  
  if high(mat1[0])<>high(mat2) then  
    begin  
      raise Exception.Create('Jumlah kolom matrik 1 tidak sama  
dengan jumlah baris matrik 2!');  
    end;  
  Nmat:=high(mat1)+1;  
  SetLength(tmp,Nmat,Nmat+1);  
  for i:=0 to Nmat-1 do  
    begin  
      for j:=0 to Nmat-1 do  
        begin  
          tmp[i,j]:=mat1[i,j];  
        end;  
      end;  
    end;  
  for i:=0 to Nmat-1 do  
    begin  
      tmp[i,Nmat]:=mat2[i];  
    end;  
end;
```

```

    property Real:double read FReal write FReal;
    property Imag:double read FImag write FImag;
end;

CArr1=array of TComplex;
CArr2=array of array of TComplex;

implementation

{ TComplex }
//constructor
constructor TComplex.Create;
begin
    inherited Create;
    FReal:=0.0;
    FImag:=0.0;
end;

constructor TComplex.Create(const aReal:double);
begin
    inherited Create;
    FReal:=aReal;
    FImag:=0.0;
end;

constructor TComplex.Create(const aReal,aImag:double);
begin
    inherited Create;
    FReal:=aReal;
    FImag:=aImag;
end;

constructor TComplex.Create(const aComplex:TComplex);
begin
    inherited Create;
    FReal:=aComplex.FReal;
    FImag:=aComplex.FImag;
end;

//data operation

function TComplex.Add(const aReal:double):TComplex;
begin
    result:=TComplex.Create((FReal+aReal),FImag);
end;

function TComplex.Add(const aComplex:TComplex):TComplex;
begin
    result:=TComplex.Create((FReal+aComplex.FReal),(FImag+aComplex.FImag));
end;

function TComplex.Subtract(const aReal:double):TComplex;
begin
    result:=TComplex.Create((FReal-aReal),FImag);
end;

```



```

end;
function TComplex.Subtract(const aComplex:TComplex):TComplex;
begin
    result:=TComplex.Create((FReal-aComplex.FReal), (FImag-
aComplex.FImag));
end;

function TComplex.Multiply(const aReal:double):TComplex;
begin
    result:=TComplex.Create((aReal*FReal), (aReal*FImag));
end;

function TComplex.Multiply(const aComplex:TComplex):TComplex;
begin
    result:=TComplex.Create((FReal*aComplex.FReal-
FImag*aComplex.FImag),
(FReal*aComplex.FImag+FImag*aComplex.FReal));
end;

function TComplex.Divide(const aReal:double):TComplex;
begin
    result:=TComplex.Create((FReal/aReal), (FImag/aReal));
end;

function TComplex.Divide(const aComplex:TComplex):TComplex;
var denote:double;
begin
    denote:=sqr(aComplex.FReal)+sqr(aComplex.FImag);

result:=TComplex.Create(((FReal*aComplex.FReal+FImag*aComplex.FIma
g)/denote),
((FImag*aComplex.FReal-FReal*aComplex.FImag)/denote));
end;

function TComplex.GetAbs:double;
begin
    result:=sqrt(sqr(FReal)+sqr(FImag));
end;

function TComplex.GetAngleRad:double;
begin
    result:=arctan(FImag/FReal);
end;

function TComplex.GetAngleDeg:double;
var phi:double;
begin
    phi:=4*arctan(1);
    result:=GetAngleRad*180/phi;
end;

function TComplex.Conj:TComplex;
begin
    result:=TComplex.Create(FReal, -FImag);
end;

```

```

function TComplex.Negative:TComplex;
begin
  result:=TComplex.Create((FReal*-1),(FImag*-1));
end;

function TComplex.toStringI(const rLen:integer):string;
begin
  if FImag<0 then
    begin
      result:=RealToStr(FReal,rLen)+'-'+RealToStr(FImag,rLen)+'i';
    end
  else
    begin
      result:=RealToStr(FReal,rLen)+'+'+RealToStr(FImag,rLen)+'i';
    end;
end;

function TComplex.toStringJ(const rLen:integer):string;
begin
  if FImag<0 then
    begin
      result:=RealToStr(FReal,rLen)+'- j'+RealToStr(FImag,rLen);
    end
  else
    begin
      result:=RealToStr(FReal,rLen)+'+ j'+RealToStr(FImag,rLen);
    end;
end;

end.

unit uMatrix;

interface

uses uUtils, SysUtils;

function MatrixAdd(const mat1:dArr2;const
aValue:double):dArr2;overload;
function MatrixAdd(const mat1,mat2:dArr2):dArr2;overload;
function MatrixSub(const mat1:dArr2;const
aValue:double):dArr2;overload;
function MatrixSub(const mat1,mat2:dArr2):dArr2;overload;
function MatrixMul(const mat1:dArr2;const
aValue:double):dArr2;overload;
function MatrixMul(const mat1:dArr2;const
mat2:dArr1):dArr1;overload;
function MatrixMul(const mat1,mat2:dArr2):dArr2;overload;
function MatrixInvers(const mat1:dArr2):dArr2;
function MatrixTranspose(const mat1:dArr2):dArr2;
function MatrixNegative(const mat1:dArr2):dArr2;
function EllGauss(const mat1:dArr2;const mat2:dArr1):dArr1;

implementation

function MatrixAdd(const mat1:dArr2;const aValue:double):dArr2;

```

```

var i,j:integer;
begin
  SetLength(result,high(mat1)+1,high(mat1[0])+1);
  for i:=0 to high(mat1) do
    begin
      for j:=0 to high(mat1[0]) do
        begin
          result[i,j]:=mat1[i,j]+aValue;
        end;
      end;
    end;
end;

```

```

function MatrixAdd(const mat1,mat2:dArr2):dArr2;
var i,j:integer;
begin
  if (high(mat1)<>high(mat2)) or (high(mat1[0])<>high(mat2[0]))
  then
    begin
      raise Exception.Create('row dan col kedua matrik tidak
sama!');
    end;
    SetLength(result,high(mat1)+1,high(mat1[0])+1);
    for i:=0 to high(mat1) do
      begin
        for j:=0 to high(mat1[0]) do
          begin
            result[i,j]:=mat1[i,j]+mat2[i,j];
          end;
        end;
      end;
    end;
end;

```

```

function MatrixSub(const mat1:dArr2;const aValue:double):dArr2;
var i,j:integer;
begin
  SetLength(result,high(mat1)+1,high(mat1[0])+1);
  for i:=0 to high(mat1) do
    begin
      for j:=0 to high(mat1[0]) do
        begin
          result[i,j]:=mat1[i,j]-aValue;
        end;
      end;
    end;
end;

```

```

function MatrixSub(const mat1,mat2:dArr2):dArr2;
var i,j:integer;
begin
  if (high(mat1)<>high(mat2)) or (high(mat1[0])<>high(mat2[0]))
  then
    begin
      raise Exception.Create('Row dan Col kedua matrik tidak
sama!');
    end;
    SetLength(result,high(mat1)+1,high(mat1[0])+1);
    for i:=0 to high(mat1) do
      begin

```

```

    for j:=0 to high(mat1[0]) do
    begin
        result[i,j]:=mat1[i,j]-mat2[i,j];
    end;
end;
end;

```

```

function MatrixMul(const mat1:dArr2;const aValue:double):dArr2;
var i,j:integer;
begin
    SetLength(result,high(mat1)+1,high(mat1[0])+1);
    for i:=0 to high(mat1) do
    begin
        for j:=0 to high(mat1[0]) do
        begin
            result[i,j]:=mat1[i,j]*aValue;
        end;
    end;
end;

```

```

function MatrixMul(const mat1:dArr2;const mat2:dArr1):dArr1;
var i,j,k:integer;
    sum:double;
begin
    if high(mat1[0])<>high(mat2) then
    begin
        raise Exception.Create('Jumlah kolom matrik1 dan jumlah baris
matrik2 tidak sama');
    end;
    SetLength(result,high(mat2)+1);
    for i:=0 to high(mat1) do
    begin
        for j:=0 to high(mat2) do
        begin
            sum:=0.0;
            for k:=0 to high(mat1[0]) do
            begin
                sum:=sum+mat1[i,k]*mat2[k];
            end;
            result[i]:=sum;
        end;
    end;
end;
end;

```

```

function MatrixMul(const mat1,mat2:dArr2):dArr2;
var i,j,k:integer;
    sum:double;
begin
    if high(mat1[0])<>high(mat2) then
    begin
        raise Exception.Create('Jumlah kolom matrik1 dan jumlah baris
matrik2 tidak sama');
    end;
    SetLength(result,high(mat1)+1,high(mat2[0])+1);
    for i:=0 to high(mat1) do
    begin

```

```

for j:=0 to high(mat2[0]) do
begin
sum:=0;
for k:=0 to high(mat1[0]) do
begin
sum:=sum+mat1[i,k]*mat2[k,j];
end;
result[i,j]:=sum;
end;
end;
end;

```

```

function MatrixInvers(const mat1:dArr2):dArr2;
var i,j,k:integer;
    A,D:double;
begin
if high(mat1)<>high(mat1[0]) then
begin
raise Exception.Create('Bukan matrik bujur sangkat!');
end;
SetLength(result,high(mat1)+1,high(mat1[0])+1);
for i:=0 to high(mat1) do
begin
for j:=0 to high(mat1[C]) do
begin
result[i,j]:=mat1[i,j];
end;
end;
try
for i:=0 to high(result) do
begin
D:=result[i,i];
result[i,i]:=1;
for j:=0 to high(result[0]) do
begin
if D=0 then
begin
D:=0.00001;
end;
result[i,j]:=result[i,j]/D;
end;
for k:=0 to high(result) do
begin
if k<>i then
begin
A:=result[k,i];
result[k,i]:=0;
for j:=0 to high(result[0]) do
begin
result[k,j]:=result[k,j]-A*result[i,j];
end;
end;
end;
end;
except
raise exception.Create('matrik tidak bisa diinvers!');

```

```

    end;
end;

function MatrixTranspose(const mat1:dArr2):dArr2;
var i,j:integer;
begin
    SetLength(result,high(mat1)+1,high(mat1[0])+1);
    for i:=0 to high(mat1) do
        begin
            for j:=0 to high(mat1[0]) do
                begin
                    result[j,i]:=mat1[i,j];
                end;
            end;
        end;
    end;

function MatrixNegative(const mat1:dArr2):dArr2;
var i,j:integer;
begin
    SetLength(result,high(mat1)+1,high(mat1[0])+1);
    for i:=0 to high(mat1) do
        begin
            for j:=0 to high(mat1[0]) do
                begin
                    result[i,j]:=mat1[i,j]*-1;
                end;
            end;
        end;
    end;

function EllGauss(const mat1:dArr2;const mat2:dArr1):dArr1;
var i,j,k,Nmat:integer;
    konst,value,DE,AE,sum:double;
    tmp:dArr2;
begin
    if high(mat1)<>high(mat1[0]) then
        begin
            raise Exception.Create('Matrik 1 bukan matrik bujur
sangkar!');
        end;
    if high(mat1[0])<>high(mat2) then
        begin
            raise Exception.Create('Jumlah kolom matrik 1 tidak sama
dengan jumlah baris matrik 2!');
        end;
    Nmat:=high(mat1)+1;
    SetLength(tmp,Nmat,Nmat+1);
    for i:=0 to Nmat-1 do
        begin
            for j:=0 to Nmat-1 do
                begin
                    tmp[i,j]:=mat1[i,j];
                end;
            end;
        end;
    for i:=0 to Nmat-1 do
        begin
            tmp[i,Nmat]:=mat2[i];
        end;
    end;
end;

```

```

end;
for i:=0 to Nmat-1 do
begin
  if tmp[i,i]=0 then
  begin
    for k:=i+1 to Nmat-1 do
    begin
      if tmp[k,i]<>0 then
      begin
        for j:=0 to Nmat do
        begin
          Konst:=tmp[i,j];
          value:=tmp[k,j];
          tmp[i,j]:=value;
          tmp[k,j]:=Konst;
        end;
      end
    end;
  end;
end;
end;
for i:=0 to Nmat-1 do
begin
  DE:=tmp[i,i];
  for j:=0 to Nmat do
  begin
    tmp[i,j]:=tmp[i,j]/DE;
  end;
  for k:=i to Nmat-1 do
  begin
    if k<>i then
    begin
      AE:=tmp[k,i];
      for j:=0 to Nmat do
      begin
        tmp[k,j]:=tmp[k,j]-AE*tmp[i,j];
      end;
    end;
  end;
end;
end;
SetLength(result,Nmat);
for i:=Nmat-1 downto 0 do
begin
  if i<>Nmat-1 then
  begin
    Sum:=0;
    for j:=i+1 to Nmat-1 do
    begin
      Sum:=Sum+tmp[i,j]*result[j];
    end;
    result[i]:=tmp[i,Nmat]-Sum;
  end
  else
  begin
    result[i]:=tmp[i,Nmat];
  end;
end;
end;

```

```
end;  
end.
```

```
unit uMenu;
```

```
interface
```

```
uses
```

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

```
type
```

```
TfrmMenu = class(TForm)  
  Panel1: TPanel;  
  btnNew: TButton;  
  btnOpen: TButton;  
  btnExit: TButton;  
  StatusBar1: TStatusBar;  
  Panel2: TPanel;  
  OpenDialog1: TOpenDialog;  
  Label1: TLabel;  
  Label2: TLabel;  
  Label3: TLabel;  
  Label4: TLabel;  
  Button1: TButton;  
  Button2: TButton;  
  Button3: TButton;  
  Label5: TLabel;  
  Label6: TLabel;  
  Image2: TImage;  
  procedure btnExitClick(Sender: TObject);  
  procedure btnNewClick(Sender: TObject);  
  procedure btnOpenClick(Sender: TObject);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

```
var
```

```
  frmMenu: TfrmMenu;
```

```
implementation
```

```
uses uComplex, uUtils, uInputGen, uObjFunc, uGenerator;
```

```
{ $R *.dfm }
```

```
procedure TfrmMenu.btnExitClick(Sender: TObject);  
begin  
  gObjFunc.Free;  
  Application.Terminate;  
end;
```

```
procedure TfrmMenu.btnNewClick(Sender: TObject);
```



```

        aRes[i]:=Res;
        frmInput.fgLoad.Cells[1,i]:=FloatToStr(Load);
        frmInput.fgLoad.Cells[2,i]:=FloatToStr(Res);
    end;
    frmInput.fgPLN.RowCount:=Ngen+1;
    frmInput.fgPLN.ColCount:=Njam+1;
    SetLength(aPLN,Ngen+1,Njam+1);
    for i:=1 to Ngen do
    begin
        for j:=1 to Njam do
        begin
            Read(output,Load);
            aPLN[i,j]:=Load;
            frmInput.fgPLN.Cells[j,i]:=FloatToStr(Load);
        end;
        Readln(output);
    end;
    CloseFile(output);
    gObjFunc:=TObjFunc.Create(aLoad,aRes,aPLN,aGen);
    for i:=1 to Ngen do
    begin
        aGen[i].Free;
    end;
    frmInput.Caption:='Tampilan Data';
    frmInput.btnNext.Caption:='&Next';
    frmInput.Show;
end;
except
    MessageDlg('File Corrupt atau Error
Program!',mtWarning,[mbOK],0);
end;
end;

end.

```

```

unit uUtils;

```

```

interface

```

```

uses SysUtils;

```

```

type

```

```

    TSort=(asc,dec);

```

```

    TBatas=record

```

```

        min,max:double;

```

```

    end;

```

```

    TBatasArr1=array of TBatas;

```

```

    TBatasArr2=array of array of TBatas;

```

```

    dArr1=array of double;

```

```

    dArr2=array of array of double;

```

```

    iArr1=array of integer;

```

```

    iArr2=array of array of integer;

```

```

    bArr1=array of boolean;

```

```
bArr2=array of array of boolean;
sArr1=array of String;
```

```
TAlleleTCSC=record
  Lokasi,TypeAlat,Setting:double;
end;
```

```
TChromTCSC1=array of TAlleleTCSC;
```

```
TAlleleUpfc=record
  Status:boolean;
  TypeAlat, Tap, Sudut:double;
end;
```

```
TChromUpfc1=array of TAlleleUpfc;
```

```
function RealToStr(Num:double;Pecahan:byte):String;
function StrToReal(Huruf:string):double;
function Pangkat(Val,pangkat:double):double;
```

```
procedure Swap(var X,Y:byte);overload;
procedure Swap(var X,Y:integer);overload;
procedure Swap(var X,Y:word);overload;
procedure Swap(var X,Y:double);overload;
procedure Swap(var X,Y:extended);overload;
procedure Swap(var X,Y:string);overload;
procedure Swap(var X,Y:boolean);overload;
```

```
procedure BubleSort(var aData:dArr1;const aType:TSort);overload;
procedure BubleSort(var aData:iArr1;const aType:TSort);overload;
procedure BubleSort(var aData:sArr1;const aType:TSort);overload;
```

```
function DecodeBinToFloat1(const aData:bArr1):double;
function DecodeBinToFloat2(const aData:bArr2):dArr1;
```

```
function GetBatas(const aValue,aMin,aMax:double):double;
function GetFlip(const aFlip:double):boolean;
```

```
function GetBatasToReal(const aValue,aMin,aMax:double):double;
function GetRealToBatas(const aValue,aMin,aMax:double):double;
```

implementation

```
function RealToStr(Num:double;Pecahan:byte):String;
var Hasil:String;
    le:byte;
begin
  le:=sizeof(Num);
  Str(Num:le:Pecahan, Hasil);
  Result:=Hasil;
end;
```

```
function Pangkat(Val,pangkat:double):double;
begin
  Result:=exp(Pangkat*ln(Val));
end;
```

```

function StrToReal(Huruf:string):double;
var Temp:double;
    Code:integer;
begin
    val(Huruf,Temp,Code);
    Result:=Temp;
end;

procedure Swap(var X,Y:byte);
var tmp:byte;
begin
    tmp:=X;
    X:=Y;
    Y:=tmp;
end;

procedure Swap(var X,Y:integer);
var tmp:integer;
begin
    tmp:=X;
    X:=Y;
    Y:=tmp;
end;

procedure Swap(var X,Y:word);
var tmp:word;
begin
    tmp:=X;
    X:=Y;
    Y:=tmp;
end;

procedure Swap(var X,Y:double);
var tmp:double;
begin
    tmp:=X;
    X:=Y;
    Y:=tmp;
end;

procedure Swap(var X,Y:extended);
var tmp:extended;
begin
    tmp:=X;
    X:=Y;
    Y:=tmp;
end;

procedure Swap(var X,Y:string);
var tmp:string;
begin
    tmp:=X;
    X:=Y;
    Y:=tmp;
end;

```

```

procedure Swap(var X,Y:boolean);
var tmp:boolean;
begin
  tmp:=X;
  X:=Y;
  Y:=tmp;
end;

```

```

procedure BubleSort(var aData:dArr1;const aType:TSort);
var i,j:integer;
begin
  for i:=1 to (high(aData)-1) do
  begin
    for j:=i to high(aData) do
    begin
      if aType=asc then
      begin
        if aData[i]>aData[j] then
        begin
          Swap(aData[i],aData[j]);
        end;
      end
      else if aType=dec then
      begin
        if aData[i]<aData[j] then
        begin
          Swap(aData[i],aData[j]);
        end;
      end;
    end;
  end;
end;

```

```

procedure BubleSort(var aData:iArr1;const aType:TSort);
var i,j:integer;
begin
  for i:=1 to (high(aData)-1) do
  begin
    for j:=i to high(aData) do
    begin
      if aType=asc then
      begin
        if aData[i]>aData[j] then
        begin
          Swap(aData[i],aData[j]);
        end;
      end
      else if aType=dec then
      begin
        if aData[i]<aData[j] then
        begin
          Swap(aData[i],aData[j]);
        end;
      end;
    end;
  end;
end;

```

```
end;  
end;
```

```
procedure BubleSort(var aData:sArr1;const aType:TSort);  
var i,j:integer;  
begin  
  for i:=1 to (high(aData)-1) do  
    begin  
      for j:=i to high(aData) do  
        begin  
          if aType=asc then  
            begin  
              if aData[i]>aData[j] then  
                begin  
                  Swap(aData[i],aData[j]);  
                end;  
            end  
          else if aType=dec then  
            begin  
              if aData[i]<aData[j] then  
                begin  
                  Swap(aData[i],aData[j]);  
                end;  
            end;  
          end;  
        end;  
      end;  
    end;  
  end;  
end;
```

```
function DecodeBinToFloat1(const aData:bArr1):double;  
var i:integer;  
    powerof2,sa:double;  
begin  
  result:=0;  
  powerof2:=1;  
  sa:=pangkat(2,high(aData))-1;  
  for i:=high(aData) downto 1 do  
    begin  
      if aData[i]=true then  
        begin  
          result:=result+powerof2;  
        end;  
      powerof2:=powerof2*2;  
    end;  
  result:=result/sa;  
end;
```

```
function DecodeBinToFloat2(const aData:bArr2):dArr1;  
var i,j:integer;  
    Data:bArr1;  
begin  
  SetLength(Data,high(aData[0])+1);  
  SetLength(result,high(aData)+1);  
  for i:=1 to high(aData) do  
    begin  
      for j:=1 to high(aData[0]) do  
        begin
```

```

        Data[j]:=aData[i,j];
    end;
    result[i]:=DecodeBinToFloat1(Data);
end;
end;

function GetBatas(const aValue,aMin,aMax:double):double;
begin
    if aValue>1.0 then raise Exception.Create('Value tidak boleh
lebih dari 1');
    if aValue<0.0 then raise Exception.Create('Value tidak boleh
kurang dari 0');
    result:=aMin+aValue*(aMax-aMin);
end;

function GetFlip(const aFlip:double):boolean;
begin
    result:=false;
    if random<=aFlip then result:=true;
end;

function GetBatasToReal(const aValue,aMin,aMax:double):double;
begin
    result:=aMin+aValue*(aMax-aMin);
end;

function GetRealToBatas(const aValue,aMin,aMax:double):double;
begin
    result:=(aValue-aMin)/(aMax-aMin);
end;

end.

```