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

SKRIPSI

PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA-BALI

Disusun oleh;
HENDRA ANDRIANI
99.12.129

MARET 2006

LEMBAR PERSETUJUAN

PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA-BALI

SKRIPSI

Disusun Guna Melengkapi dan Memenuhi Syarat-Syarat
Guna Mencapai Gelar Sarjana Teknik

Disusun Oleh :

HENDRA ANDRIANI
NIM. 99.12.129

Mengetahui, Ketua Jurusan Teknik Elektro	Menyetujui, Dosen Pembimbing
Ir. F. Yudi Limpraptono, MT	Ir. H. CHOIRI
NIP Y.103 950 0274	NIP. 130703042

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

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

SKRIPSI

PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA-BALI

Disusun oleh;
HENDRA ANDRIANI
99.12.129

MARET 2006

LEMBAR PERSETUJUAN

PENJADWALAN UNIT PEMBANGKIT TERMAL
DENGAN METODE ANT COLONY SEARCH ALGORITHM
PADA PT. PEMBANGKITAN JAWA-BALI

SKRIPSI

Disusun Guna Melengkapi dan Memenuhi Syarat-Syarat
Guna Mencapai Gelar Sarjana Teknik

Disusun Oleh :

HENDRA ANDRIANI
NIM. 99.12.129

Mengetahui,	**Menyetujui,**
Ketua Jurusan Teknik Elektro	**Dosen Pembimbing**
Ir. F. Yudi Limpraptono, MT	**Ir. H. CHOIRI**
NIP Y.103 950 0274	**NIP. 130703042**

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

ABSTRAKSI

PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA-BALI

(Hendra Andriani, Nim. 99.12.129, Teknik Elektro Energi Listrik S-1)
(Dosen Pembimbing : Ir.H Choiri)

Kata Kunci: *Ant Colony Search Algorithm*, Komitmen Unit, Optimasi Biaya Pembangkitan.

Dalam melayani kebutuhan daya listrik yang tidak tetap dari waktu ke waktu, sehingga menimbulkan suatu permasalahan yaitu bagaimana bagaimana mengoperasikan suatu sistem tenaga listrik yang selalu dapat memenuhi permintaan daya pada setiap saat, dengan keandalan yang tinggi dan harga yang murah. Oleh karena itu suatu operasi pada beban tertentu, perhitungan ekonomis harus tetap merupakan suatu prioritas atau nilai yang harus diperhitungkan disamping hal-hal 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 cadangan berputar (*spinning reserve*) pada sistem pembangkitan Jawa-Bali.

Skripsi ini menganalisis permasalahan komitmen unit atau penentuan penjadwalan operasi unit-unit pembangkit dalam melayani beban sistem selama periode waktu tertentu dengan menggunakan metode *ant colony Search Algorithm* (ACSA). Hasil dari analisa tersebut nantinya dapat digunakan sebagai salah satu acuan dalam operasi pembangkitan dan penyaluran daya yang ekonomis dan optimal, terutama mengenai biaya pembangkitan. *Input* dari program ini adalah koefisien biaya bahan bakar (*fuel cost*), daya maksimum dan minimum dan data pembebanan tiap jam tiap unit pembangkit. Sedangkan hasil akhir dari program ini yaitu hasil penjadwalan operasi unit-unit pembangkit, perhitungan pembebanan ekonomis, grafik biaya antara PT. Pembangkitan Jawa-Bali dan komitmen unit dengan metode *ant colony Search Algorithm* (ACSA) serta total biaya pembangkitan ekonomis.

KATA PENGANTAR

Dengan rahmat Allah SWT, dan mengucapkan syukur kehadirat-Nya atas karunia yang dilimpahkan kepada saya sehingga dapat menyelesaikan skripsi yang berjudul **"PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA-BALI"**.

Skripsi ini bertujuan untuk memenuhi kurikulum akademik yang harus ditempuh oleh setiap mahasiswa ITN Malang dalam menempuh sekaligus mengakhiri pendidikan pada jenjang S-1 pada Jurusan Teknik Elektro Konsentrasi Teknik Energi Listrik.

Sebelum dan selama penyusunan skripsi ini, penyusun telah banyak mendapatkan bantuan dan bimbingan dari berbagai pihak. Untuk itu pada kesempatan ini penyusun menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Ir.Abraham Lomi, MSEE, selaku Rektor ITN Malang.
2. Bapak Ir. F. Yudi Limpraptono, MT, selaku Kepala Jurusan Teknik Elektro ITN Malang.
3. Bapak Ir. H. Choiri, selaku Dosen Pembimbing.
4. Bapak dan Ibu dosen jurusan Teknik Elektro Energi Listrik
5. Bapak – bapak karyawan bagian personalia dan lapangan di PT. Pembangkitan Jawa-Bali.

6. Bapak dan ibuku, yang sangat berarti dalam kehidupan penyusun, dimana doa serta restu dan keridhaannya senantiasa penyusun harapkan.
7. Teman- teman di jurusan Teknik Elektro Energi Listrik Institut Teknologi Nasional Malang, terutama angkatan '99 yang telah membantu dalam penyelesaian skripsi ini.
8. Semua pihak yang tidak dapat saya sebutkan, yang telah membantu dalam penyelesaian skripsi ini.

Penyusun menyadari sepenuhnya akan segala kekurangan yang ada dalam penyusunan skripsi ini, maka dengan kerendahan hati penyusun mengharapkan kritik dan saran demi penyempurnaan skripsi ini.

Akhirnya penyusun berharap semoga dalam skripsi ini dapat membantu serta bermanfaat bagi rekan-rekan mahasiswa khususnya jurusan Teknik Elektro Energi Listrik Institut Teknologi Nasional Malang

Malang, Maret 2006

Penyusun

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR GRAFIK

BAB I

PENDAHULUAN

1.1 Latar Belakang

Harga dasar tenaga listrik dari waktu ke waktu terus mengalami kenaikan yang cukup tinggi. Satu hal yang menjadi penyebabnya adalah besarnya biaya operasional yang harus ditanggung PLN, yang selama ini masih disubsidi oleh pemerintah, sedangkan jumlah pembangkit yang dimiliki oleh PLN sangat banyak, antara lain: PLTA, PLTGU, PLTU, PLTD dan PLTG. Pembangkit-pembangkit yang berada dikawasan Jawa dan Bali telah terinterkoneksi, sehingga suatu lokasi yang pada saat tertentu kekurangan daya listrik dapat disuplai oleh pembangkit dari lokasi lain. Tiap-tiap pembangkit tenaga listrik mempunyai karakteristik yang berbeda, walaupun dari jenis pusat pembangkit yang sama. Hal ini terjadi karena perbedaan output pembangkit, pemakaian bahan bakar, usia unit pembangkit dan masih banyak faktor yang lainnya.

Pola beban yang optimum bagi subsistem termal harus diikuti oleh unit-unit pembangkit termal. Dalam mengikuti pola beban ini perlu dicari kombinasi unit-unit pembangkit termal yang beroperasi agar dicapai hasil operasi yang optimum, dengan biaya bahan bakar yang minimum. Konsekuensinya adalah bahwa ada unit pembangkit termal yang perlu distart dan distop kembali dalam periode optimasi. Untuk unit pembangkit termal, proses start-stop bukanlah hal yang sederhana, karena dalam proses tersebut terdapat sejumlah kalori yang hilang saat unit distop sehingga unit menjadi dingin dan perlu dipanaskan lagi

waktu start. Apabila dikehendaki waktu start yang pendek maka harus dilakukan pemanasan terus pada unit pembangkit termal, tentu saja hal ini memerlukan bahan bakar yang harus diperhitungkan.

Beban sistem tenaga listrik selalu berubah menurut waktu, maka dengan demikian beban unit pembangkit termal dan hidro juga berubah-ubah menurut waktu dalam partisipasinya untuk melayani beban sistem. Hal ini mengakibatkan biaya bahan bakarnya persatuan waktu dalam rupiah perjam juga ikut mengalami perubahan menurut waktu.

Tenaga listrik tidak dapat disimpan dalam skala besar, karenanya harus disediakan pada saat dibutuhkan, akibatnya timbul persoalan dalam menghadapi kebutuhan daya listrik yang tidak tetap dari waktu ke waktu. Bagaimana mengoperasikan suatu sistem tenaga listrik yang selalu dapat memenuhi kebutuhan daya pada setiap saat, dengan kualitas yang baik dan harga yang murah. Apabila daya yang dikirim dari bus-bus pembangkit jauh lebih besar dari pada permintaan daya pada bus-bus beban, maka akan timbul persoalan pemborosan energi pada perusahaan listrik terutama pada pembangkit termal. Sedangkan apabila daya yang dibangkitkan dan dikirimkan lebih rendah atau tidak memenuhi kebutuhan beban konsumen, maka akan terjadi pemadaman lokal pada bus-bus beban, yang akibatnya merugikan pihak konsumen. Oleh karena itu, diperlukan penyesuaian antara pembangkitan dengan permintaan daya.

Sistem tenaga listrik terdiri dari sejumlah unit pembangkit yang saling terinterkoneksi. Penentuan penjadwalan pembangkit adalah faktor penting karena ini berhubungan dengan biaya operasional pembangkit dan memegang peranan

penting dalam perencanaan operasional pembangkitan sehingga didapat operasi pembangkitan yang optimum dalam sistem tenaga listrik secara keseluruhan. Perencanaan operasional pembangkitan sistem tenaga listrik sangat erat kaitannya dengan masalah komitmen unit, yaitu masalah pemilihan unit-unit pembangkit mana yang akan dioperasikan untuk memenuhi kebutuhan beban listrik. Komitmen unit ini nantinya akan berpengaruh pada biaya bahan bakar. Oleh karena itu, perlu diupayakan pengoptimalan operasi unit pembangkit agar dicapai biaya operasional pembangkitan yang minimum.

Ada banyak metode yang digunakan untuk memecahkan masalah komitmen unit, diantaranya: metode *skala prioritas, branch and bound, dynamic programming, simulated annealing, lagrange relaxation, genetic algorithm, tabu search, fuzzy logic, bender decomposition, evolutionary programming* dan masih banyak lagi. Dalam skripsi ini akan digunakan evaluasi dari sebuah metode yang berdasarkan dari pergerakan dan peradaban suatu makhluk hidup (*swarm Inteligence),* yaitu *Ant Colony Search Algorithm (ACSA).* Dimana ACSA merupakan salah satu metode yang terinspirasikan dari cara hidup sekelompok semut dalam menentukan jalan terpendek antara sumber makanan dan sarang mereka. Menurut para peneliti *ethnologi dan kebiasan hewan* setiap semut dapat menentukan jalur terpendek antara sarang mereka dengan sumber makanan tanpa bantuan visual. Karakteristik inilah yang digunakan oleh metode ini dalam menyelesaikan berbagai permasalahan perhitungan, khususnya optimalisasi. Pada mulanya metode ini muncul sebagai sebuah metode guna menyelesaikan permasalahan optimasi yang terdapat dalam menentukan pola pergerakan yang

efisien pada penjual keliling (traveling salesman) yang kemudian berkembang dan dapat digunakan untuk berbagai masalah optimasi. Salah satu masalah optimasi yang dapat diselesaikan adalah masalah optimasi biaya dimana dalam menentukan biaya optimal digunakan fungsi kuadrat.

Keunggulan *Ant Colony Search Algorithm* ada pada penggunaan logika yang sederhana sehingga dapat menurunkan permintaan memori komputer atau dengan kata lain dengan penghematan dalam segi penggunaan memori akan dicapai waktu penghitungan yang lebih singkat.

1.2 Rumusan Masalah

Dalam mengoperasikan pusat-pusat pembangkit listrik, diharapkan adanya suatu optimalisasi dari biaya operasional pembangkit dengan menekannya serendah mungkin. Untuk itu diperlukan penentuan penjadwalan unit pembangkit yang ekonomis. Dengan menggunakan evaluasi dari metode *Ant Colony Search Algorithm* diharapkan dapat diperoleh alternatif solusi terbaik dalam memecahkan masalah komitmen unit pada pusat-pusat pembangkit listrik.

Berdasarkan pada diskripsi permasalahan dan latar belakang tersebut diatas maka skripsi ini diberi judul :

PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA BALI

1.3 Tujuan Pembahasan

Berdasarkan permasalahan yang telah dikemukakan diatas, maka skripsi ini bertujuan :

- Untuk menentukan penjadwalan pembangkitan daya dari unit-unit pembangkit termal yang akan melayani kebutuhan beban yang berubah tiap jam dengan menggunakan metode *Ant Colony Search Algorithm.*
- Untuk mengetahui besarnya biaya operasi pembangkitan dengan menggunakan metode *Ant Colony Search Algorithm.*
- Untuk menganalisa serta mengevaluasi kelayakan penerapan metode pembebanan pada *Ant Colony Search Algorithm* untuk pemecahan masalah komitmen unit di PT. Pembangkitan Jawa-Bali, guna mendapatkan suatu perencanaan dan penjadwalan pembangkitan daya termal yang optimal, ekonomis, efektif dan efisien.

1.4 Batasan Masalah

Permasalahan dalam sistem tenaga listrik sangat luas sekali, khususnya pada komitmen unit, sehingga dalam menganalisa permasalahan perlu diadakan pembatasan-pembatasan. Didalam penulisan skripsi ini pembatasan yang dilakukan adalah sebagai berikut :

- Penjadwalan dilakukan dalam tiga hari (3x24 jam dengan range tiap jam) pada periode studi tanggal 27, 30, 31 juli 2005
- Tidak membahas masalah rugi-rugi saluran transmisi.

- Pembahasan dititik beratkan pada segi ekonomis, schingga tidak terlalu membahas segi teknis.
- Tidak membahas *combined cycle* pada PLTGU.
- Untuk ST (*Steam Turbine*) pada *combined cycle*, diambil data parameter dari pola PLTGU CC-3.3.1 yang beroperasi.
- Tidak membahas masalah biaya cadangan berputar (*spinning reserve*), hanya memperhatikan kendala batasan cadangan berputar.
- Tidak ada biaya untuk *shutdown* unit.
- Data biaya dan parameter unit pembangkit menggunakan data penawaran PT. Pembangkitan Jawa-Bali Agustus 2002 (sebelum kenaikan biaya bahan bakar).

1.5 Metodologi Pembahasan

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

- Studi kepustakaan mengenai hal-hal yang berhubungan dengan pembahasan masalah.
- Studi lapangan untuk mendapatkan data-data parameter unit-unit termal yang dibutuhkan dari objek penelitian yaitu pada PT. PJB yang diperlukan dengan berpedoman pada teori yang diperoleh dan studi kepustakaan.
- Menentukan optimasi penjadwalan pembebanan pada komitmen unit dengan metode *Ant Colony Search Algorithm*.

- Perhitungan *Economic Dispatch* menggunakan iterasi lambda untuk pembangkit unit termal.
- Membuat evaluasi, sehingga dapat disimpulkan apakah metode ini layak ataukah tidak layak dilihat dari sisi nilai ekonomisnya.

1.6 Sistematika Pembahasan

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

BAB I PENDAHULUAN

Berisikan masalah umum mengenai latar belakang penulisan, rumusan masalah, tujuan penulisan skripsi, pembatasan masalah, metodologi pembahasan, sistematika penulisan dan kontribusi penelitian.

BAB II LANDASAN TEORI

Berisi antara lain tentang pengenalan sistem tenaga listrik, karakteristik masukan-keluaran, karakteristik laju tambahan biaya bahan bakar, pembebanan ekonomis pembangkit listrik dengan metode *Ant Colony Search Algorithm* dan teori tentang optimasi dengan *Ant Colony Search Algorithm.*

BAB III DATA SISTEM UNIT PEMBANGKIT TERMAL PADA PT. PJB

Berisi tentang Data unit termal, aplikasi metode *Ant Colony Search Algorithm* pada PT. PJB dan data beban sistem, dengan data yang diperoleh selama survey dilapangan sebagai variabelnya. Juga dicantumkan data program pembanding yang dipakai sebagai acuan untuk memvalidasi program komputer.

BAB IV ANALISA DATA DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PJB

Berisi tentang Program Komputer, Algoritma program dan hasil perhitungan serta analisa data dengan metode *Ant Colony Search Algorithm*.

BAB V PENUTUP

Berisi tentang kesimpulan akhir dan saran.

1.7 KONTRIBUSI

Dengan adanya analisa ini nantinya diharapkan dapat memberikan alternatif terbaik dalam pemecahan permasalahan komitmen unit yang lebih mudah karena tanpa penurunan permasalahan dengan waktu perhitungan relatif cepat, sehingga kemungkinan dapat diaplikasikan dilapangan pada PT. Pembangkitan Jawa-Bali.

BAB II

LANDASAN TEORI

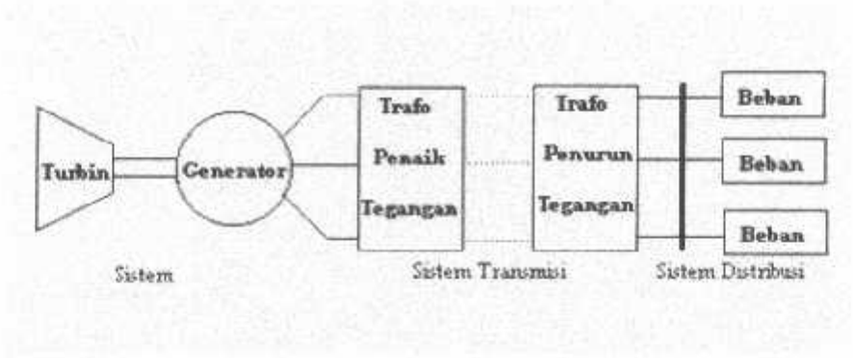
2.1 Sistem Tenaga Listrik

Karena berbagai persoalan teknis, tenaga listrik hanya dapat dibangkitkan pada lokasi tertentu. Mengingat pemakai tenaga listrik atau pelanggan tenaga listrik tersebar diberbagai tempat, maka penyaluran tenaga listrik terdiri dari berbagai penanganan teknis.

Tenaga listrik dibangkitkan di pusat-pusat listrik seperti PLTA, PLTD, PLTU, PLTG dan PLTGU, kemudian disalurkan melalui saluran transmisi setelah tegangannya dinaikkan terlebih dahulu oleh Transformator Penaik Tegangan yang terdapat di pusat listrik.

Setelah tenaga listrik disalurkan melalui transmisi, maka sampailah tenaga listrik tersebut di Gardu Induk (GI) yang untuk kemudian tegangannya diturunkan oleh Trafo Penurun Tegangan menjadi tegangan menengah atau tegangan distribusi primer.

Jaringan keluar dari gardu induk umumnya disebut jaringan distribusi, sedangkan antara pusat listrik dengan gardu induk disebut jaringan transmisi. Setelah disalurkan melalui jaringan distribusi primer maka tenaga listrik kemudian diturunkan tegangannya oleh gardu distribusi menjadi tegangan distribusi sekunder atau tegangan rendah, dan baru kemudian disalurkan ke konsumen, seperti pada gambar 2-1.



Gambar 2-1 Elemen Pokok Sistem Tenaga Listrik

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

Untuk memenuhi kebutuhan tenaga listrik bagi para konsumen, diperlukan berbagai peralatan listrik. Berbagai peralatan listrik ini dihubungkan satu sama yang lain secara kesceluruhan membentuk suatu sistem tenaga listrik. Sehingga yang disebut Sistem Tenaga Listrik disini adalah sekumpulan pusat listrik dan gardu induk yang satu dengan yang lain dihubungkan oleh jaringan transmisi sehingga merupakan sebuah kesatuan interkoneksi.

Biaya operasi dari sistem tenaga listrik pada umunya merupakan bagian biaya yang terbesar dari biaya operasi suatu sistem tenaga listrik, secara garis besar biaya dari sistem tenaga listrik terdiri atas :

- Biaya pembelian tenaga listrik
- Biaya pegawai
- Biaya bahan bakar dan materi operasi
- Biaya lain-lain

Dari keempat biaya tersebut, biaya bahan bakar pada umumnya adalah biaya yang terbesar. Untuk PLN biaya bahan adalah kira-kira 60% dari biaya operasi secara keseluruhan.

Karena daya listrik yang dibangkitkan harus sama dengan tenaga listrik yang dibutuhkan oleh konsumen, maka manajemen operasi sistem tenaga listrik harus memperhatikan ha-hal sebagai berikut :

- Prakiraan beban
- Syarat-syarat pemeliharaan peralatan
- Keandalan yang diinginkan
- Pengaturan dan penyaluran beban
- Proses tenaga listrik yang ekonomis

Dari kelima hal diatas masih harus seringkali dikaji ulang terhadap berbagai kendala seperti :

- Aliran beban dalam jaringan
- Daya hubung singkat
- Gangguan yang sering kali menimpa peralatan
- Stabilitas sistem
- Penyediaan suku cadang dan dana

Dengan memperhatikan kendala – kendala diatas maka seringkali harus dilakukan pengaturan kembali terhadap rencana pemeliharaan dan alokasi beban. Makin besar sistem, maka makin banyak hal yang harus diamati dan dikoordinasi, sehingga diperlukan perencanaan, pelaksanaan, pengendalian dan evaluasi sistem yang cermat.

Makin besar suatu sistem tenaga listrik, maka semakin banyak unsur yang harus dikoordinasikan dan harus diamati. Dalam mengoperasikan sistem tenaga listrik sering dijumpai beberapa persoalan. Hal ini antara lain disebabkan karena pemakaian tenaga listrik selalu berubah dari waktu ke waktu. Biaya bahan bakar yang relatif tinggi serta kondisi alam dan lingkungan yang sering mengganggu jalannya operasi pada sistem tenaga listrik. Berbagai persoalan pokok yang dihadapi dalam pengoperasian sistem tenaga listrik adalah :

a. **Pengaturan Frekuensi**

   Apabila daya yang dibangkitkan dalam sistem lebih kecil dari beban sistem, maka frekuensi akan turun dan sebaliknya apabila daya yang dibangkitkan lebih besar dari beban sistem maka frekuensi akan naik.

b. **Tegangan Dalam Sistem**

   Dalam penyediaan tenaga listrik bagi para pelanggan, tegangan yang konstan, merupakan salah satu faktor utama yang harus dipenuhi.

c. **Pemeliharaan peralatan**

   Peralatan yang dioperasikan dalam sistem tenaga harus dipelihara secara periodik agar tidak cepat rusak dan apabila ada kerusakan hendaknya langsung diperbaiki.

d. Biaya Operasi

Secara garis besar biaya operasi tenaga listrik terdiri dari biaya pembelian tenaga listrik, biaya pegawai, biaya bahan bakar dan material operasi. Dari beberapa biaya operasi diatas biaya bahan bakar merupakan biaya operasi yang terbesar suatu perusahaan listrik sehingga perlu teknik teknik optimasi untuk menekan biaya tersebut.

e. Perkembangan Sistem

Perkembangan sistem tenaga listrik seirama dengan perkembangan pemakaian tenaga listrik oleh konsumen.

f. Gangguan Sistem

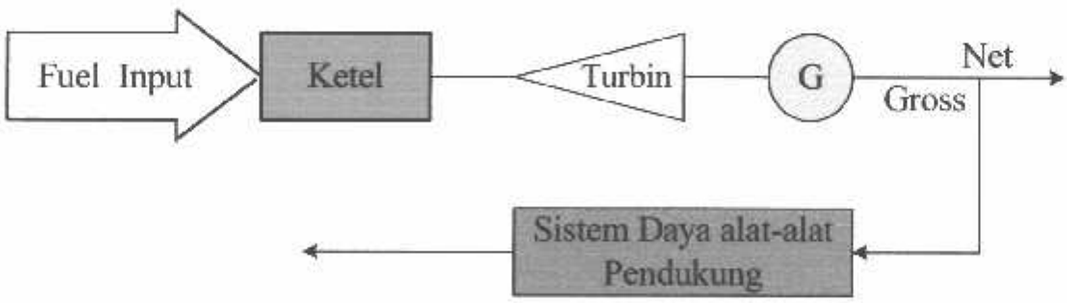
Gangguan dalam sistem tenaga listrik, adalah suatu keadaan yang sepenuhnya tidak dapat dihindari. Penyebab gangguan terbesar salah satunya adalah petir.

Dalam usaha pengadaan energi listrik dikenal berbagai jenis pembangkit yang di kelompokkan pada jenis awal energi (wujud sebelum energi tersebut dikonversikan). Dalam skripsi ini akan dibahas jenis pembangkit listrik tenaga termal yang akan diterangkan lebih lanjut pada sub bab berikut.

2.2 Prinsip Pembangkit Listrik Tenaga Termal

Jenis pembangkit ini pada dasarnya menciptakan suatu tekanan pada ruang tertutup dengan menggunakan pemanasan dengan tujuan menggerakan turbin. Pembangkit jenis ini biasanya dikelompokan lagi berdasarkan sumber panas atau bahan yang digunakan untuk memperoleh panas. Secara garis besar prinsip kerja

suatu pembangkit termal adalah memanaskan sejumlah air dalam ketel dengan konstruksi khusus, lalu air tersebut akan berubah menjadi uap. Sebagai mana kita ketahui uap merupakan bentuk pemuaian atau ekspansi dari zat cair akibat pemanasan dengan kata lain terjadi penambahan massa jenis zat tersebut. Jika ketel yang digunakan didesain hanya mempunyai satu jalan keluar untuk uap tersebut dan jalan tersebut ditutup maka akan terjadi tekanan yang terus bertambah didalam ketel akibat dari ekspansi yang terus menerus seiring dengan pemanasan yang dilakukan. Setelah tekanan didalam ketel sudah mencapai batas maksimal dari ketel tersebut mampu menahan, jalur keluar yang pada awalnya ditutup dibuka kembali. Uap air yang bertekanan tinggi ini pun akan segera keluar dengan deras melalui saluran tersebut, diujung saluran tersebut telah diletakkan turbin yang terhubung dengan generator, sehingga ketika uap bertekanan tinggi tersebut menabrak turbin akan menyebabkan turbin tersebut mampu berputar dan menyebabkan kumparan didalam generator ikut berputar memotong fluksi magnet yang ada disekeliling kumparan tersebut sehingga timbullah energi listrik. Proses ini dapat dilihat seperti pada gambar 2.2.



Gambar 2-2 Skema Kerja Unit Termal

2.3. Karakteristik Pembangkit Tenaga Listrik

2.3.1 Pembangkit Listrik Termal

Hal yang paling mendasar dalam optimasi ekonomi dari sebuah pembangkit termal adalah menentukan karakteristik masukan-keluaran (*input-output characteristics*) pusat listrik tersebut. Dalam mendefinisikan karakteristik masukan-keluaran, akan dibicarakan tentang *gross input* dan *net output* yang dihasilkan pusat listrik tersebut. *Gross input* pembangkit listrik termal menyatakan jumlah keseluruhan bahan bakar yang diperlukan, sedangkan *net output* adalah daya nyata (*real power*) yang dihasilkan generator.

Tipe sebuah pembangkit listrik tenaga termal tampak pada gambar 2-2. Bagan tersebut tediri atas sebuah ketel yang menghasilkan uap untuk menggerakkan turbin uap yang dikopel dengan sebuah generator. Daya listrik yang dihasilkan tidak seluruhnya disalurkan ke sistem tetapi sebagai kecil digunakan untuk mengoperasikan peralatan yang terdapat pada pusat listrik tersebut, seperti pompa, kompresor dan sebagainya, serta untuk mencatu peralatan kontrol, komunikasi, penerangan dan komputer.

2.3.2 Karakteristik Masukan – Keluaran (*input-output characteristics*)

Definisi dari karakteristik masukan-keluaran pembangkit tenaga listrik adalah formula yang menyatakan hubungan antara masukan pembangkit sebagai fungsi keluaran pembangkit. Karakteristik masukan-keluaran ini merupakan hal yang paling mendasar dalam optimasi ekonomi dari unit pembangkit termal. Masukan sebuah pembangkit listrik termal umumnya dinyatakan sebagai

banyaknya energi per satuan waktu dari bahan bakar yang diberikan ke ketel untuk menghasilkan daya listrik yang merupakan keluaran dari pusat listrik tersebut.

Terdapat dua notasi yang umum digunakan :

H dengan satuan [Mbtu/hour]

F dengan satuan [$US/hour]

Dimana $F = H \times \$US/Btu$ dan *$US/Btu* menyatakan harga bahan bakar per satuan energi yang dikandung oleh bahan bakar tersebut.

Sedangkan keluaran dari pembangkit listrik termal adalah daya nyata yang dihasilkan oleh generator dikurangi dengan daya nyata yang dipakai oleh pusat listrik tersebut. Notasi yang umum digunakan adalah :

P dengan satuan [MW]

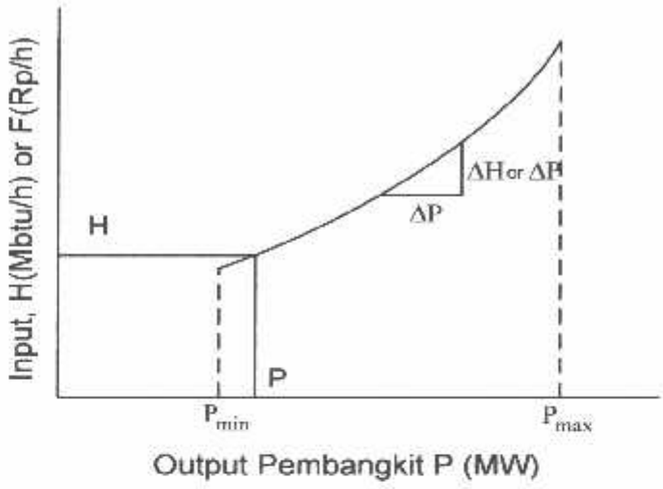
Jadi dapat disimpulkan masukan pusat listrik merupakan fungsi terhadap keluarannya, maka hubungan tersebut dapat ditulis sebagai berikut:

$H = f(P)$ *[Mbtu/h] atau* $F = f(P)$*[$US/h]*

Pembahasan selanjutnya akan berpedoman atas dasar fungsi biaya bahan bakar ($F = f(P)$ *[$US/h])* , sedangkan kurva dari karakteristik masukan-keluaran dari sebuah pembangkit tenaga termal yang telah diidealkan ditunjukan pada grafik 2-1. Masukan adalah sebagai ordinat, yang berupa banyaknya energi yang diperlukan persatuan waktu [Mbtu/h] atau dapat juga merupakan biaya bahan bakar yang dikonsumsi per satuan waktu [$US/h]. Sedangkan keluaran adalah daya listrik [MW] yang dihasilkan blok tersebut untuk melayani beban sistem.

GRAFIK 2 – 1

Kurva Karakteristik Masukan – Keluaran Pembangkit Termal



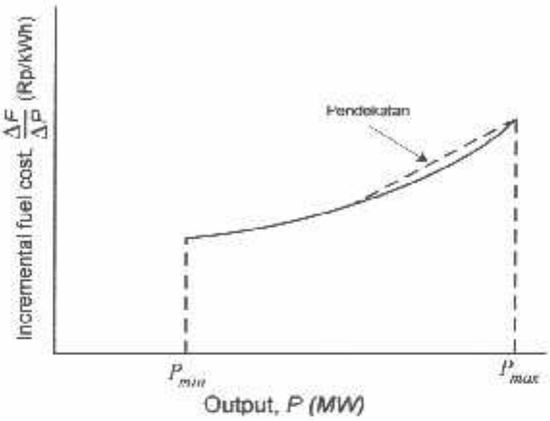
2.3.3 Karakteristik Laju Tambahan Biaya Bahan Bakar

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

Secara luas fungsi biaya bahan bakar akan digunakan untuk menentukan pembebanan ekonomis dari sebuah pembangkit listrik tenaga termal. Tampak pada grafik 2-2 kurva laju tambahan biaya bahan bakar yang telah diidealkan melalui pendekatan linier dari sebuah pembangkit listrik termal.

GRAFIK 2 – 2

Kurva Karakteristik Laju Tambahan Biaya Bahan Bakar



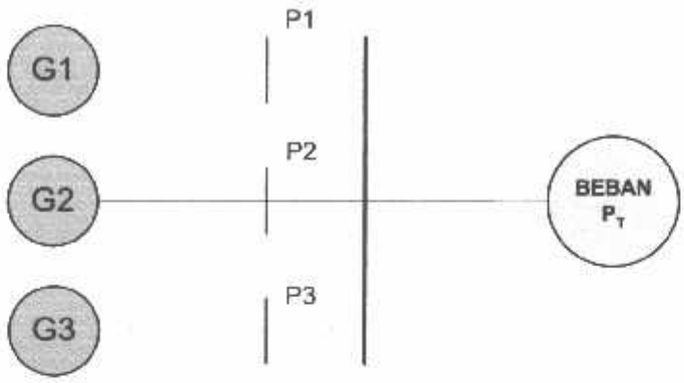
2.4 Komitmen Unit

Kebutuhan akan energi listrik sudah mengalami pergeseran yang cukup signifikan dari masa lalu hingga sekarang, dimana pada masa lalu listrik hanya digunakan sebagai penerangan saja. Sedangkan pada masa sekarang hampir sepenuhnya menjadi bagian penting dari hidup manusia seiring dengan perkembangan teknologi yang ada saat ini, dimana semua peralatan yang digunakan menggunakan listrik sebagai sumber energinya. Apalagi pada sektor industri listrik adalah bagian terpenting dalam kelangsungan proses produksinya. Misalnya dalam siklus harian yang terbagi dalam interval waktu 1 jam selama 24 jam, beban listrik dalam sistem tenaga selalu berubah. Oleh karena itu operasi pusat-pusat pembangkit dalam sistem tenaga harus selalu dikoordinasikan dalam pembagian pembebanan secara optimal atau

seekonomis mungkin pada setiap perubahan beban dalam interval waktu untuk siklus waktu tertentu yang dikenal dengan istilah *Unit Commitment* .

Guna memenuhi kebutuhan akan energi listrik ini, maka diperlukan suatu pengaturan operasi sistem tenaga listrik yang ekonomis dimana bukan hanya faktor pemenuhan kebutuhan beban yang harus dipenuhi tetapi juga faktor kualitas, keandalan dan nilai ekonomis menjadi syarat mutlak yang harus diperhatikan.

Untuk mengatasi masalah diatas, maka sistem tenaga listrik yang ada sekarang ini menggunakan sistem interkoneksi, dimana prinsip dari sistem ini adalah menggabungkan beberapa pusat pembangkit yang tersebar diberbagai lokasi, baik unit hdiro maupun termal secara paralel melalui suatu jaringan transmisi bertegangan tinggi untuk menyuplai beban gabungan (*infinite bus*). Hal ini berarti bahwa seluruh unit pembangkit yang berada dalam suatu wilayah menjadi suatu kesatuan yang terpadu,seperti pada gambar 2-3. Untuk sistem interkoneksi yang besar, yang terdiri dari banyak unit pembangkit dan banyak pusat beban (GI), sarana pengendalian operasi dengan menggunakan sarana komunikasi saja tidak cukup, tetapi harus ditambah dengan peralatan *telemetering* dan alat – alat pengolah data elektronis seperti komputer. Selain itu, pada pengoperasian sistem yang terinterkoneksi PLN berkewajiban menyediakan energi listrik dengan rating tegangan yang berada dalam batas – batas tertentu.



GAMBAR 2-3

Sistem Interkoneksi

Tetapi pada kenyataannya, pengoperasian secara interkoneksi menimbulkan masalah teknis yang cukup rumit dan komplek. Hal ini disebabkan ada dua hal yang saling bertentangan pada PLN (Persero) dalam menjalankan misinya. Misi utama PLN (Persero) adalah sebagai perusahaan jasa teknik kelistrikan adalah mencari laba, dan untuk itu suatu langkah yang ekonomis harus diambil untuk memperoleh keuntungan yang maksimal atas modal yang ditanamkan, yaitu dengan mengoptimalkan pengoperasiannya. Mengoptimalkan pengoperasian berarti harus dicapai biaya operasi yang seminimal mungkin, khususnya biaya bahan bakar karena bahan bakar merupakan unsur terbesar dalam total biaya operasi. Kemudian misi dari PLN (Persero) lainnya adalah mengolah dan menyediakan energi listrik bagi masyarakat dengan kualitas dan keandalan yang terbaik. Melihat kedua misi ini, maka diperlukan suatu perencanaan penyaluran

dan penyediaan energi listrik yang memenuhi faktor kualitas dan keandalan dengan biaya yang seekonomis mungkin.

Komitmen unit merupakan suatu metode solusi yang tepat untuk mengatasi permasalahan diatas guna mencari jadwal unit pembangkit yang harus beroperasi untuk periode waktu tertentu agar dicapai biaya operasi yang seekonomis mungkin. Pada masalah komitmen unit diasumsikan bahwa ada sejumlah N unit pembangkit yang tersedia dan harus dioperasikan untuk memenuhi permintaan beban.

2.4.1 Kendala Pada Komitmen Unit.

Dalam pengoperasian pembangkit untuk memenuhi permintaan beban terdapat berbagai kendala yang merupakan syarat pembatas. Kendala tersebut antara lain :

- **Cadangan berputar (spinning reserve)**

  Cadangan berputar adalah kata yang digunakan untuk menggambarkan jumlah total daya yang mungkin dibangkitkan dari semua unit yang tersinkronisasi pada sistem yang dikurangi dengan beban saat itu dan rugi- rugi penyaluran yang terjadi. Cadangan berputar harus ada untuk menghindari penurunan frekwensi sistem yang terlalu besar bila satu atau lebih unit harus keluar dari sistem dengan kata lain harus ada cukup cadangan pada unit-unit lain untuk menutupi kekurangan suplai daya yang hilang dalam periode waktu tertentu. Cadangan berputar harus dialokasikan untuk menaati aturan-aturan tertentu, biasanya aturan yang dipakai adalah bahwa cadangan tersebut berupa sebuah persentase yang

diberikan terhadap beban puncak yang diperkirakan atau cadangan tersebut harus mampu menutupi kehilangan daya dari unit yang paling besar yang dibebani penuh dalam suatu periode tertentu.

Besar cadangan berputar tersebut harus ditentukan secara hati – hati, sebab seringkali penentuan yang didasarkan untuk menjaga keandalan sistem berbenturan dengan biaya pengoperasian yang diusahakan seekonomis mungkin, misalnya jika cadangan berputar kecil dan unit pembangkit terbesar mengalami gangguan dan *trip* sehingga unit tersebut keluar secara mendadak dari sistem, maka cadangan berputar tersebut tidak cukup untuk mengatasi kekurangan pembangkitan yang terjadi dan untuk meghindari sistem *collapase* maka perlu dilakukan pelepasan beban dan ini mengakibatkan keandalan sistem menurun. Makin besar cadangan berputar dalam sistem, maka makin handal pula sistem tersebut dalam menghindari gangguan, tetapi makin besar pula biaya operasi terutama biaya bahan bakarnya karena adanya cadangan berputar tersebut. Oleh karena itu perlu adanya kesepakatan antara pemenuhan dan pengoptimalan biaya operasi.

- **Kendala Unit Termal**

Unit termal biasanya memerlukan operator untuk mengoperasikannya terutama ketika dinyalakan ataupun pada saat dimatikan. Sebuah unit termal hanya dapat dijalankan dibawah perubahan temperatur yang gradual, dan ini diterjemahkan kedalam sebuah periode

waktu dalam jam yang dibutuhkan untuk membawa unit tersebut *on-line*. Hal ini menyebabkan kendala – kendala antara lain :

a. Minimum Up Time

Minimum up time adalah interval waktu minimum dimana suatu unit yang dihidupkan (ON) tidak boleh dimatikan (OFF) kembali sebelum melewati batas waktunya (Up Time).

b. Minimum Down Time

Minimum Down Time adalah interval waktu minimum dimana suatu unit yang dimatikan (OFF) tidak boleh dihidupkan (ON) kembali sebelum melewati batas waktunya (Down Time).

2.4.2 Biaya start-up

Biaya start-up adalah biaya yang diperlukan oleh pembangkit untuk start dari keadaan tidak beroperasi sampai pembangkit beroperasi (terhubung ke sistem tenaga listrik). Ada dua macam biaya start-up yaitu :

a. Biaya Start-up pada kondisi dingin (cooling)

kondisi ini terjadi karena saat pembangkit dilepas dari sistem (tidak beroperasi), temperatur boiler dibiarkan turun dari temperatur kerjanya, sehingga pada saat beroperasi kembali perlu dilakukan pemanasan kembali.

b. Biaya Start-up pada kondisi panas (banking)

Kondisi ini terjadi karena saat pembangkit dilepas dari sistem (tidak beroperasi), temperatur boiler dijaga pada temperatur kerja.

2.4.3 Fungsi Biaya Bahan Bakar

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

$$F_i(P_i^t) = a_i + b_i P_i^t + c_i (P_i^t)^2 \quad \text{..... (2.1)}$$

Dimana :

a_i, b_i, c_i = konstanta persamaan unit ke i

P_i^t = keluaran daya unit pembangkit pada jam t

2.4.4 Fungsi Obyektif dan Formulasi Masalah Komitmen Unit

Sasaran dari masalah *unit commitment* adalah bagaimana cara menjadwalkan unit pembangkit pada periode waktu tertentu guna melayani kebutuhan sistem secara optimal, sehingga didapatkan biaya operasi minimum tanpa melanggar kendala – kendala yang dimasukkan. Oleh karena itu, fungsi obyektif dinyatakan sebagai jumlah dari fungsi biaya bahan bakar dan biaya start up dari unit pembangkit, dapat dinyatakan dengan :

$$\text{Min } F(P_i^t, u_i^t) = \sum_{t=1}^{T} \sum_{i=1}^{N} \left[F_i(P_i^t) + ST_i \left(1 - u_i^{t-1}\right) \right] u_i^t \quad \text{.....(2.2)}$$

dengan $F_i(Pf_{i,t}) = a_i + b_i P_{it} + c_i P_{it}^2$

Sehubungan dengan minimalisasi dari total biaya operasi, terdapat syarat batas, sebagai berikut :

a) Batasan kesimbangan daya (*power balance constraint*)

$$P_{load}^{t} - \sum_{i=1}^{N} P_i^t u_i^t = 0 \quad \text{.....(2.3)}$$

b) Batasan pembangkitan (*generation limits constraint*)

$$P_i^{\min} \le P_i^t \le P_i^{\max} \quad \text{.....(2.4)}$$

c) Batasan cadangan berputar (*spinning reserve constraint*)

$$P_{load}^{t} + R^t - \sum_{i=1}^{N} P_i^{\max} u_i^t \le 0 \quad \text{.....(2.5)}$$

d) Batasan Minimum up time (*Minimum up time constraint*)

$$u_i^t = 1 \text{ for } \sum_{h=t-T_i^{up}}^{t-1} u_i^t < T_i^{up} \quad \text{.....(2.6)}$$

e) Batasan minimum down time (*Minimum down time constraint*)

$$u_i^t = 0 \text{ for } \sum_{h=t-T_i^{down}}^{t-1} \left(1 - u_i^t\right) < T_i^{down} \quad \text{.....(2.7)}$$

Adapun parameter-parameter yang digunakan dalam formulasi masalah komitmen unit adalah :

$F_i(P_{it})$ – fungsi biaya bahan bakar unit termal ke-*i* dengan a_i, b_i dan c_i adalah konstanta persamaan dari unit ke-*i*

p_{it} = daya keluaran dari unit termal ke-*i* pada jam *t*

$F_i(p)$ = biaya untuk menghasilkan unit-unit daya *p* oleh unit ke-*i*

ST_i – biaya start up dari unit ke-*i*

P_{load}^{t} = beban pada waktu *t*

R^t = cadangan daya pada waktu *t*

U_i^t = status on atau off dari unit ke-i pada jam t, $U_{it} = 0$ ketika unit Off dan $U_{it} = 1$ ketika unit on

N = Jumlah dari pembangkit termal

T = jumlah jam

2.5 Pembebanan Ekonomis Unit Pembangkit Listrik

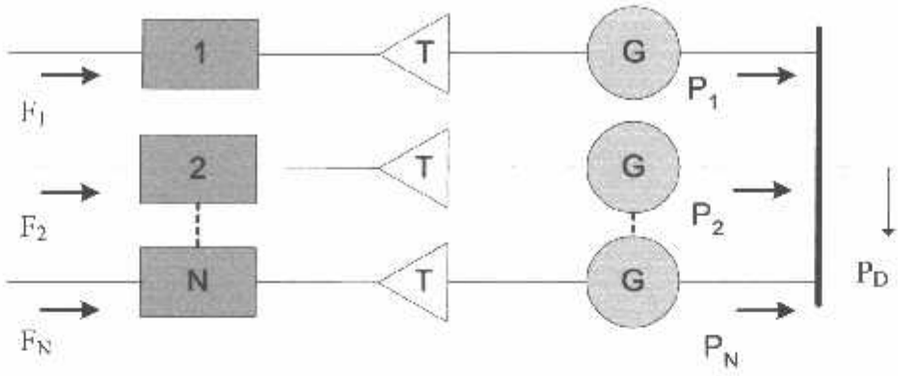
Seperti yang telah diketahui bahwa beban sistem selalu berfluktuasi setiap waktu. Perubahan beban itu harus diikuti oleh unit-unit pembangkit tenaga listrik, sehingga perlu difikirkan bagaimana membagi beban secara ekonomis diantara unit-unit pembangkit tenaga listrik yang beroperasi. Pembebanan ekonomis *(economic dispatch)* unit pembangkit listrik adalah pembagian atau pendistribusian beban sistem pada unit-unit pembangkit listrik yang beroperasi secara optimal sehingga mendapatkan biaya operasi pembangkit yang minimum. Penyelesaian *economic dispatch* dapat dilakukan dengan beberapa cara yang akan dibahas pada sub bab dibawah ini

2.5.1 Penyelesaian Economic Dispatch dengan Metode Pengali Lagrange

Sistem tenaga listrik dengan mengabaikan rugi-rugi transmisi dapat dilihat pada gambar 2-4. Sistem ini memperlihatkan pembangkit termal yang terdiri atas N buah unit yang dihubungkan pada sebuah bus bar untuk melayani total beban sebesar P_D. Masukan untuk setiap unit ke-i adalah F_i yang menyatakan tingkat biaya dari masing-masing unit, dan keluaran dari masing-masing unit P_i adalah daya listrik yang dibangkitkan oleh tiap-tiap unit.

Biaya total F_T yang ditanggung sistem adalah jumlah biaya dari tiap-tiap unit pembangkit. Dan batasan yang paling penting dari pengoperasian pembangkit

termal tersebut adalah daya listrik yang dihasilkan harus sama dengan besarnya beban konsumen.



Gambar 2-4 N Unit Pembangkit Termal Melayani Beban P_D

Yang menjadi permasalahan adalah meminimumkan total biaya F_T dengan memperhatikan pembatas φ bahwa daya dihasilkan unit pembangkit sama dengan yang diterima beban. Secara matematis pernyataan tersebut diatas dapat dinyatakan dengan persamaan berikut :

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

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

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

$$P_D = \sum_{i=1}^{N} P_i \quad \ldots\ldots (2.9)$$

$$P_{Li} - \sum_{i=1}^{N} P_i = \varphi = 0 \quad \ldots\ldots (2.10)$$

dimana P_{Li} = kebutuhan beban, dan

P_i = jumlah daya yang dihasilkan

Penyelesaian permasalahan optimasi seperti ini dapat diselesaikan dengan metode yang menyangkut fungsi *lagrange* :

$$£ = F_T + \lambda\varphi$$

atau

$$£ = \sum_{i=1}^{N} F_i(P_i) + \lambda \ (P_D - \sum_{i=1}^{N} P_i) \quad \text{..... (2.11)}$$

dimana λ = *Lagrange Multipliers*.

Untuk mencari harga optimal dari fungsi *lagrange* terhadap P_i, dapat diperoleh dengan operasi gradien dari persamaan *lagrange* sama dengan nol.

$$\nabla £ = 0$$

$$\frac{\partial L}{\partial P_i} = \frac{\partial F_T}{\partial P_i} + \lambda \cdot \left(\frac{\partial P_D}{\partial P_i} - \frac{\partial P_i}{\partial P_i} \right) = 0 \quad \textit{atau}$$

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

$$\frac{\partial F_i}{\partial P_i} = \lambda \quad \text{..... (2.12)}$$

Persamaan diatas menunjukkan bahwa distribusi beban yang optimal terjadi apabila semua unit pembangkit beroperasi pada tingkat laju tambahan biaya bahan bakar yang sama, yang ternyata sama dengan nilai λ. Kondisi optimal ini tentunya memerlukan persamaan-persamaan pembatas (*constraint*) agar keluaran dari setiap unit pembangkit harus lebih besar atau sama dengan keluaran minimum, dan lebih kecil atau sama dengan keluaran maksimum yang diijinkan.

Dari *N* buah unit pembangkit yang telah dibahas, maka dapat diambil kesimpulan sebagai berikut :

$$\frac{\partial F_i}{\partial P_i} = \lambda \qquad \text{ada } N \text{ buah persamaan}$$

$$P_{i\,min} \le P_i \le P_{i\,maks} \qquad \text{ada } 2N \text{ buah pertidaksamaan}$$

$$\sum_{i=1}^{N} P_i = P_{it} \qquad \text{ada 1 buah pembatas}$$

Batasan-batasan diatas dapat diperluas menjadi :

$$\frac{\partial F_i}{\partial P_i} = \lambda \text{ untuk } P_{i\,min} \le P_i \le P_{i\,maks}$$

$$\frac{\partial F_i}{\partial P_i} \le \lambda \text{ untuk } P_i = P_{i\,maks}$$

$$\frac{\partial F_i}{\partial P_i} \ge \lambda \text{ untuk } P_i = P_{i\,min}$$

2.5.2 Penyelesaian Economic Dispatch Dengan Metode Iterasi Lamda

Pada metode iterasi lamda (λ), lamda ditentukan dahulu kemudian dihitung keluaran untuk tiap-tiap unit pembangkit. Bila jumlah dari daya keluaran tiap-tiap unit pembangkit tidak sama dengan daya beban maka dilakukan perhitungan kembali harga lamda berikutnya. Bila lamda masukan untuk yang pertama ditentukan menghasilkan daya keluaran yang lebih rendah dari daya beban dan harga lamda yang kedua ditentukan menghasilkan daya keluaran lebih dari daya beban, maka dapat dilakukan interpolasi. Atau dapat pula dilakukan pada lamda pertama, penambahan untuk mendapatkan lamda berikutnya yang mendekati solusi sebenarnya, setelah beberapa kali iterasi.

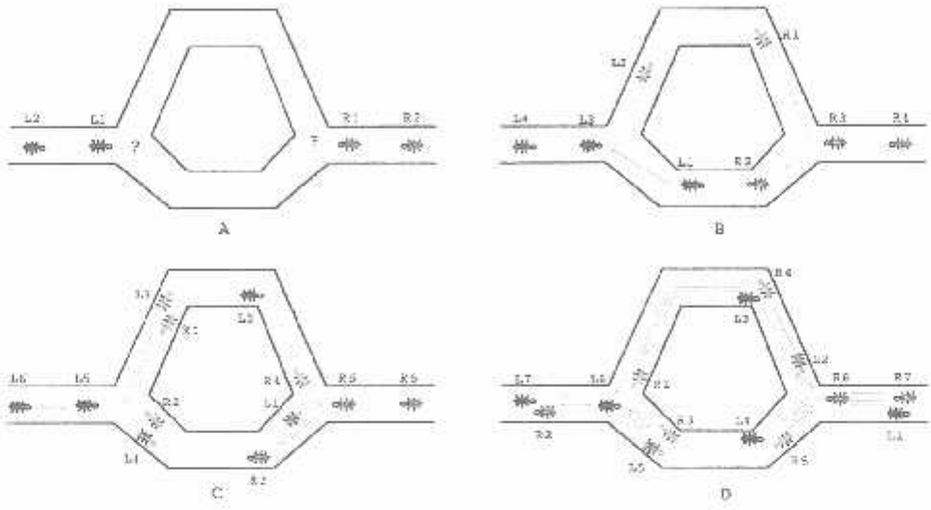
2.6 Teori Dasar Ant Colony Search Algorithm (ACSA)

Metode *Ant Colony Search Algorithm* (ACSA) terinspirasi dari perilaku koloni semut yang digunakan untuk memecahkan fungsi atau masalah– masalah optimasi kombinatorial. *Ant Colony Search Algorithm* adalah suatu metode yang didasarkan dari pola pergerakan semut dalam mencari dan menentukan jalur teroptimal dari sarangnya menuju sumber makanan. Sekelompok semut selalu dapat menemukan jalur makanan paling dekat dengan sarang mereka menggunakan semacam getah yang disebut peromone. Peromone berfungsi sebagai sarana komunikasi yang fital diantara semut. Jadi secara garis besar semut berkomunikasi melalui jumlah peromone pada jalur yang mereka tempuh semakin banyak jumlah peromone semakin banyak semut yang melaluinya. Pada pengaplikasiannya di optimasi pembangkit listrik termal metode ini digunakan untuk menentukan kombinasi fungsi obyektif yang paling optimal pada setiap jamnya untuk siklus waktu 24 jam dan akhirnya menentukan besarnya daya yang harus dipikul unit termal untuk suatu siklus waktu 24 jam.

Seperti diketahui, semut mampu mencari jalur terpendek dari sumber makanan menuju sarang tanpa menggunakan isyarat visual. Mereka juga mampu beradaptasi dengan perubahan dalam lingkungan; misalkan, mencari jalur terpendek jika jalur yang lama sudah tidak memungkinkan untuk dilalui karena adanya hambatan.

Proses ini diilustrasikan oleh gambar dibawah. Semut berjalan menuju sumber makanan, dalam hal ini seekor semut meletakkan peromon pada saat

berjalan dan secara probabilistik semut lebih suka jalur yang lebih kaya peromon daripada yang tidak.



L= semut bergerak dari kiri ke kanan
R= semut bergerak dari kanan ke kiri

GAMBAR 2-5
Perilaku Semut Ketika Mendapat Rintangan

Pada saat menemukan rintangan semut tidak dapat melanjutkan perjalanan sehingga harus harus memilih ke kanan atau ke kiri. Setengah semut memilih ke kanan dan yang lain ke kiri. Semut yang memilih jalur terpendek akan lebih cepat membentuk jalur feromone yang putus dibandingkan dengan semut yang memilih jalur yang lebih panjang. Sehingga, jalur yang lebih pendek akan mendapatkan feromone dalam jumlah yang lebih besar per unit waktu, dan selanjutnya, sejumlah besar semut akan memilih jalur yang lebih pendek ini. Karena ini, semua semut akan dengan cepat memilih jalur yang lebih pendek .

Semua semut berjalan dengan kecepatan yang hampir sama. Waktu yang dibutuhkan untuk melintasi sisi yang lebih panjang dari sebuah rintangan lebih

lama dari sisi yang lebih pendek. Semut lebih suka jalur dengan feromone yang lebih banyak menyebabkan akumulasi ini terjadi lebih cepat pada jalur yang lebih pendek. Perilaku semut ini bisa digunakan untuk masalah optimasi dalam mencari penjadwalan unit pembangkit.

2.6.1 Metode Ant Colony Search Algorithm Dalam Penyelesaian Optimasi

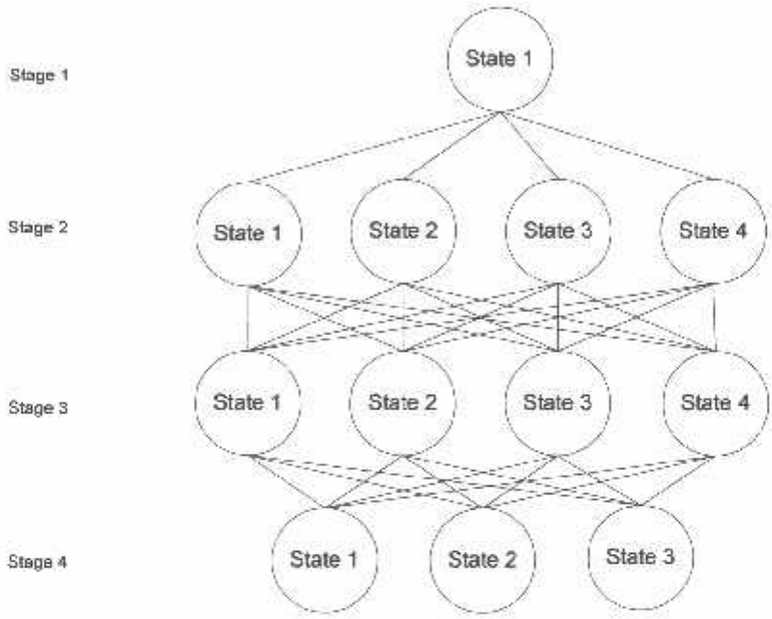
Pada penjadwalan unit-unit pembangkit yang akan beroperasi tiap periode waktu dalam jam, pembagian bebannya harus dalam keadaan optimal atau seekonomis mungkin. Dari jumlah unit pembangkit yang banyak, maka untuk menentukan unit mana yang harus beroperasi dan tidak beroperasi pada jam tertentu, dapat diperhitungkan dengan membuat kombinasi keadaan dari unit-unit yang ada dan berada pada keadaan operasi yang optimal, kemudian baru dipilih kombinasi mana yang termurah biaya operasinya.

Untuk menyelesaikan komitmen unit dengan ACSA, ruang pencarian dari masalah penjadwalan pembangkitan dibangun dengan menggunakan konsep pembuatan keputusan multi-proses. Gambar dibawah menggambarkan prosedur komputasi dari metode yang diajukan. Blok komputasi utama dibahas di bawah ini

- **Langkah 1 : Inisialisasi produksi semut**

  Langkah pertama, koloni semut pertama dibangkitkan. Semut di letakkan pada keadaan awal dan nilai feromon dari τ_0 juga diberikan pada langkah ini. Gambar dibawah memplot *multi-stage search space*. Yakni berdasarkan pada konsep proses multi tahap ini, ruang pencarian dari masalah penjadwalan

pembangkitan termal bisa dihasilkan. Semua permutasi yang mungkin merupakan ruang pencarian ini. Setiap tahap berisi beberapa keadaan, sedangkan orde keadaan dipilih pada setiap tahap digabungkan sebagai perjalanan yang bisa dicapai dianggap sebagai solusi yang mungkin untuk masalah ini



Gambar 2-6
Multi-stage search space

- **Langkah 2: Evaluasi fitness**

Dalam langkah ini, kesesuaian semua semut dinilai berdasarkan fungsi obyektif terkait, yang bisa dinyatakan sebagai berikut

$$f(\mu) = \sum_{i=1}^{n} tc(s_{\mu(i)}, s_{\mu(i+1)}) \quad \text{.....(2.13)}$$

Dengan kesesuaian evaluasi dari semut terkait, feromon bisa ditambahkan pada arah khusus dimana semut ini telah dipilih. dimana f fungsi biaya produksi tiap pembangkit, tc (s_i , s_j) adalah biaya perpindahan antara s_i dan s_j, dan μ (i) for i=1,...,n menyatakan perubahan.

Dalam aplikasi komitmen unit dilakukan pergerakan apakah sistem melanggar atau tidak kendala-kendala pembangkit? Bila tidak memenuhi maka jalur itu tidak akan dipilih oleh semut, dan bila memenuhi maka akan dilakukan perhitungan awal guna meminimalisir biaya operasi dengan persamaan: $minimize\ cost = \sum_{i}^{N_g} F_i(p_i)$, N_g: biaya operasi pembangkit tiap unit, $F_i(P_i)$: fungsi biaya dan P_i: besar pembangkit unit i. Selanjutnya dilakukan pendekatan fungsi kuadratik besar pembangkit P_i :

$F_i(P_i) = a_iP_i+b_iP_i+c_i$. bila hasil kecil maka biaya operasional murah dan bila sebaliknya maka biaya opersasi mahal.

Dengan fitness yang telah di evaluasi dari hubungan diantara semut, peromon dapat ditambahkan pada jalur tertentu yang telah dilalui oleh semut. Dengan metode ini jumlah feromon yang berhubungan dengan jalur yang berbedapun terbentuk. Dengan demikian jalur yang paling sering dikunjungi akan mengumpulkan feromon terbanyak.

Dengan kata lain kombinasi dengan biaya yang paling optimal akan mendapatkan nilai probabilitas terbesar sehingga kemungkinan untuk kombinasi paling optimal terpilih lebih besar dari yang tidak optimal.

- **Langkah 3 : Pelepasan Semut *(Ant Dispatch)***

Pada langkah ini, semut memilih suatu jalur berdasarkan kandungan pheromon dan jarak yang akan ditempuhnya pada suatu jalur. Cara pengambilan jalur ini dapat dilukiskan dengan persamaan berikut:

$$p_{ij}^{k} = \frac{[\tau_{ij}][\eta_{ij}]^{\varepsilon}}{\sum_{N=J_i^k}[\tau_{im}][\eta_{im}]^{\beta}} \quad \text{.....(2.14)}$$

Dimana:

$\tau_{ij}(t)$ = pheromone yang diletakkan antara simpul *i* dan *j* waktu *t*

η_{ij} = perbandingan terbalik dari panjang jalur

ε = parameter jarak

$P_{ij,k}$ = kemungkinan yang ada untuk perpindahan unit

Dengan mengacu pada persamaan diatas, setiap pergerakan semut menuju dari simpul satu ke simpul lainnya berdasarkan dari nilai τ_{ij} dan n_{ij}. Ketika nilai τ_{ij} menjadi lebih besar, ini berarti bahwa terdapat banyak kawanan yang melewati jalur tersebut, schingga jalur tersebut dapat dikatakan sebagai solusi optimal. Ketika nilai n_{ij} miningkat, nilai ini merepresentasikan bahwa simpul yang terdekat dari simpul sebelumnya mempunyai kemungkinan jarak terdekat. Pada studi penjadwalan pembangkit thermal, ini dapat dilihat sebagai perbedaan antara biaya total sesungguhnya dengan biaya total yang baru. Jika semut "k" diletakkan pada kcadaan i memilih kcadaan j selanjutnya untuk dituju, kemudian gerakan itu bisa dinyatakan dalam persamaan:

- Jika $q>q_0$ pilih keadaan berikutnya secara acak dengan kemungkinan seblumnya

$$p_{ij}^{k} = \frac{[\tau_{ij}][n_{ij}]^{\beta}}{\sum_{N \in i_k^*}[\tau_{iu}][n_{iu}]^{\beta}} \quad \text{.....(2.15)}$$

- Jika $q<q_0$ pilih keadaan berikutnya dengan jarak terbaik dari pheromone lokal Masksimum : $[\tau_{ij}][\eta_{ij}]^{\beta}$

q adalah nilai acak penyebaran atau distribusi sedangkan q_0 adalah parameter. Jika *m* merupakan notasi yang diberikan pada sejumlah ekor semut, lalu setiap *m* ekor semut melakukan pergerakan sebanyak *m* kali pada setiap interasinya dalam interval waktu (*t, t+1*). Ketika mencari sebuah pemecahan masalah, tingkat feromone dari jalur yang telah dikunjungi bertambah dengan dinamis, Persamaan diatas bertujuan untuk memperluas ruang lingkup pencarian yang disebut aturan update pheromone lokal $\left(\tau(r,s) \leftarrow (1-\rho)\tau(r,s) + \rho.\Delta\tau(r,s)\right)$

Dimana ρ : pengatur parameter, $\Delta \rho$ (r,s)= τ_0 dan τ_0 : nilai pheromone awal .

Setelah interasi ke *n*, seluruh semut telah menyelesaikan perjalanannya. Jalur paling pendek yang ditemukan oleh para semut diperbolehkan untuk memperbaharui jumlah pheromonnya. Pada langkah ini nilai probabilitas diperbaharui kembali dengan upadate global pheromone yaitu: $\tau(r,s) \leftarrow (1-\alpha).\tau(r,s) + \alpha.\Delta\tau(r,s)$

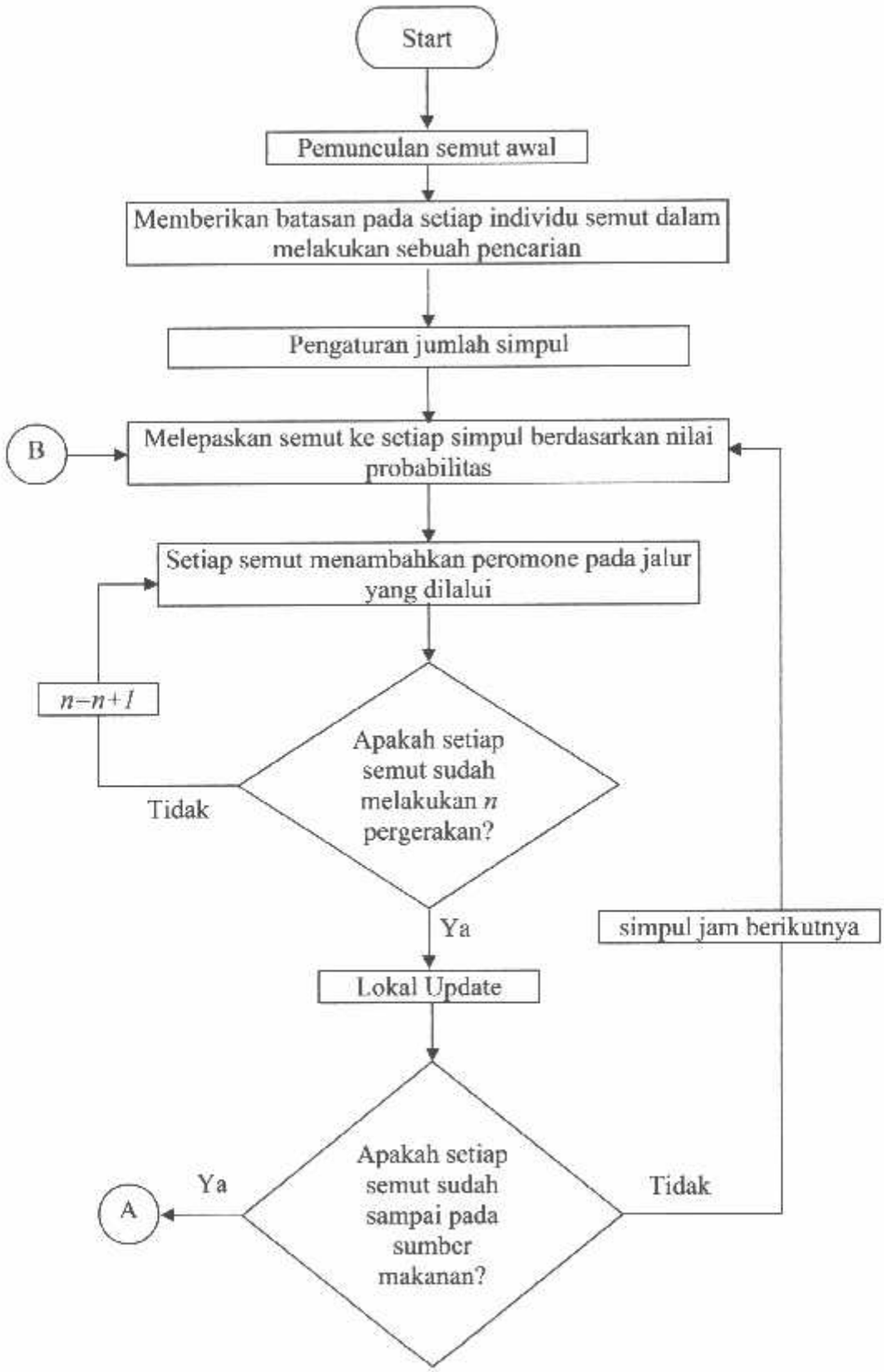
α=parameter pheromone

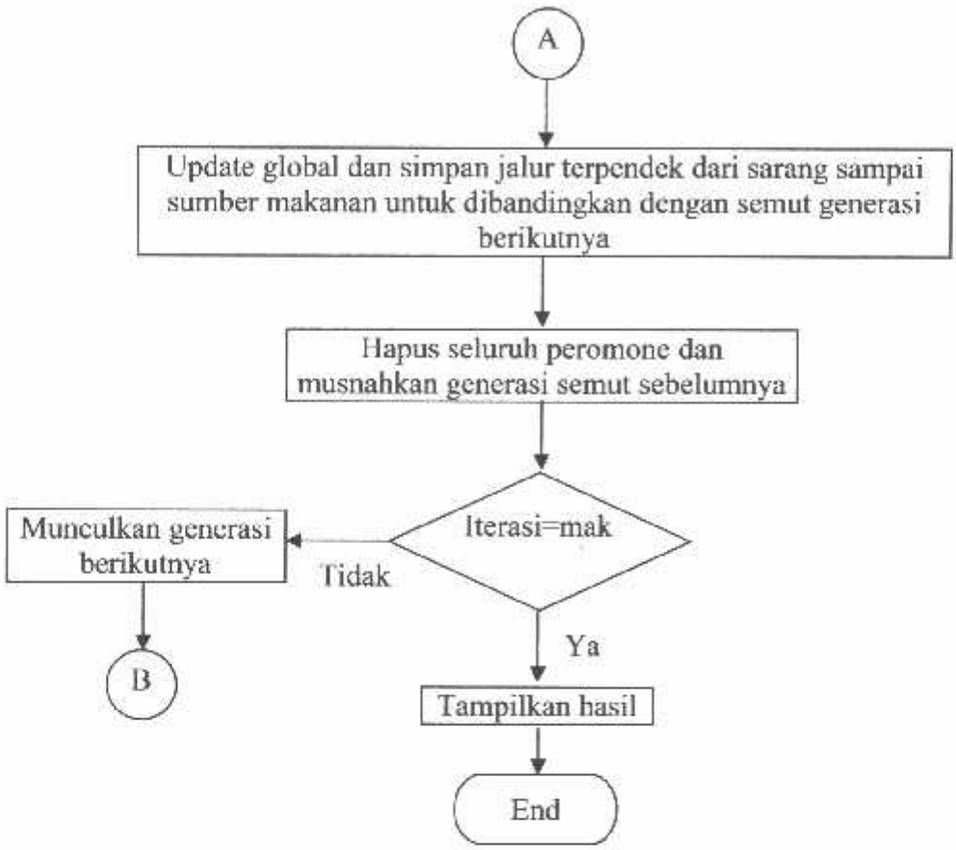
Jalur terpendek ini juga disimpan sebagai bahan perbandingan untuk iterasi yang akan datang., karena langkah ini telah mengevaluasi segala kemungkinan keadaan yang terjadi.

- **Langkah 4 : Kriteria pemberhentian *(stop criteria)***

Proses penghitungan berlangsung sampai interasi sebelum titik ambang yang telah ditetapkan tercapai, atau semua semut telah memilih jalur yang sama. Jumlah dari semut dan interasi ditetapkan secara coba-coba. Setiap jalur yang dipilih oleh semut pada setiap interasinya harus dikaji ulang. Jika sebuah jalur yang lebih baik ditemukan didalam proses maka jalur inipun akan disimpan sebagai referensi berikutnya. Jalur yang paling baik diantara seluruh interasi dapat dikatakan sebagai solusi optimal dari masalah penjadwalan. Dalam algoritma, nilai dari ρ dan σ dipilih antara 0 dan 1, dengan tujuan untuk meningkatkan pelaksanaan perhitungan. Sementara itu untuk nilai dari ε dipilih tidak terlalu besar agar kontribusi dari peromon dan jarak akan menemukan titik keseimbangan yang lebih baik.

Secara global metode *ant colony search algorithm* dapat dilukiskan sebagai berikut:


Start
Pemunculan semut awal
Memberikan batasan pada setiap individu semut dalam melakukan sebuah pencarian
Pengaturan jumlah simpul
B
Melepaskan semut ke setiap simpul berdasarkan nilai probabilitas
Setiap semut menambahkan peromone pada jalur yang dilalui
n=n+1
Apakah setiap semut sudah melakukan n pergerakan?
Tidak
Ya
simpul jam berikutnya
Lokal Update
Apakah setiap semut sudah sampai pada sumber makanan?
Ya
Tidak
A


A
Update global dan simpan jalur terpendek dari sarang sampai sumber makanan untuk dibandingkan dengan semut generasi berikutnya
Hapus seluruh peromone dan musnahkan generasi semut sebelumnya
Iterasi=mak
Tidak
Munculkan generasi berikutnya
B
Ya
Tampilkan hasil
End

BAB III

DATA SISTEM UNIT PEMBANGKIT TERMAL PADA PT. PEMBANGKITAN JAWA-BALI

3.1 Pendahuluan

PT. Pembangkitan Jawa-Bali dalam penyelenggaraan usaha ketenaga listrikan berdasarkan prinsip industri dan perniagaan yang sehat, dituntut mampu memanfaatkan sebesar-besarnya peluang pasar dalam bidang tenaga listrik. Dalam hal tersebut, PT. Pembangkitan Jawa-Bali harus menjaga efisiensi dan keandalan opersional penyediaan tenaga lsitrik dari pembangkit-pembangkit yang dimilikinya.

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

3.2 Data Pembangkit Termal

Pembangkit termal yang berada pada kawasan PT. Pembangkitan Jawa-Bali berjumlah 37 unit yang terdiri dari 5 (lima) blok pembangkit listrik tenaga gas dan uap, 11 (sebelas) pembangkit listrik tenaga uap dan 5 (lima) pembangkit listrik tenaga gas. Adapun data-data lebih lengkapnya dapat dilihat pada tabel 3-1 dan tabel 3-2, untuk harga bahan bakar berdasarkan statistik PLN tahun 2002 dipakai nilai tukar Rp. 9000,- per satuan dolar Amerika ($ US).

Tabel 3 - 1
Data Unit Termal Pada PT. Pembangkitan Jawa - Bali

No	Nama Pembangkit	Bahan Bakar	Kapasitas (MW)		Lama Waktu (Jam)			
			Min	Max	MUT	MDT	Cold Start	Hot Start
1	PLTU Paiton 1	Coal	225	370	72	48	17	4
2	PLTU Paiton 2	Coal	225	370	72	48	17	4
3	PLTGU Gresik GT 1.1	Gas	53	102	36	10	1	0
4	PLTGU Gresik GT 1.2	Gas	53	102	36	10	1	0
5	PLTGU Gresik GT 1.3	Gas	53	102	36	10	1	0
6	PLTGU Gresik CC 3.3.1	Gas	250	480	36	10	3	2
7	PLTGU Gresik GT 2.1	Gas	53	102	36	10	1	0
8	PLTGU Gresik GT 2.2	Gas	53	102	36	10	1	0
9	PLTGU Gresik GT 2.3	Gas	53	102	36	10	1	0
10	PLTGU Gresik CC 3.3.1	Gas	250	480	36	10	3	2
11	PLTGU Gresik GT 3.1	Gas	53	102	36	10	1	0
12	PLTGU Gresik GT 3.2	Gas	53	102	36	10	1	0
13	PLTGU Gresik GT 3.3	Gas	53	102	36	10	1	0
14	PLTGU Gresik CC 3.3.1	Gas	250	480	36	10	3	2
15	PLTU Gresik 1	Gas	43	85	48	10	9	1
16	PLTU Gresik 2	Gas	43	85	48	10	9	1
17	PLTU Gresik 3	Gas	90	175	48	10	9	2
18	PLTU Gresik 4	Gas	90	175	48	10	9	2
19	PLTG Gresik 1	Gas	5	16	3	1	1	0
20	PLTG Gresik 2	Gas	5	16	3	1	1	0
21	PLTG Gilitimur 1	HSD	5	16	3	1	1	0
22	PLTG Gilitimur 2	HSD	5	16	3	1	1	0
23	PLTGU M. Karang GT 1.1	Gas	50	95	36	10	1	0
24	PLTGU M. Karang GT 1.2	Gas	50	95	36	10	1	0
25	PLTGU M. Karang GT 1.3	Gas	50	95	36	10	1	0
26	PLTGU M. Karang CC 3.3.1	Gas	300	465	36	10	3	2
27	PLTGU M. Tawar GT 1.1	HSD	72	138	36	10	0	0
28	PLTGU M. Tawar GT 1.2	HSD	72	138	36	10	0	0
29	PLTGU M. Tawar 1.3	HSD	72	138	36	10	0	0
30	PLTGU M. Tawar GT 2.1	HSD	72	138	36	10	0	0
31	PLTGU M. Tawar GT 2.2	HSD	72	138	36	10	0	0
32	PLTGU M. Tawar CC 3.3.1	HSD	315	605	36	10	3	2
33	PLTU M. Karang 1	MFO	44	85	48	10	6	1
34	PLTU M. Karang 2	MFO	44	85	48	10	6	1
35	PLTU M. Karang 3	MFO	44	85	48	10	6	1
36	PLTU M. Karang 4	Gas	90	165	48	10	11	2
37	PLTU M. Karang 5	Gas	90	165	48	10	11	2

Sumber : Data penawaran PT. PJB, Jl. Ketintang Baru No. 11 Surabaya 60231

Keterangan : MUT = Minimum Up Time
MDT = Minimum Down Time

Tabel 3 – 2
Data Biaya dan Parameter Unit Termal Pada PT. Pembangkitan Jawa – Bali

No	Nama Pembangkit	Biaya Start - Up (Juta Rp)		Koefisien Biaya Bahan Bakar		
		Cold Start – Up	Hot Start - Up	a	b	c
1.	PLTU Paiton 1	682.98	149.68	3244978	111712.15	10.2971
2.	PLTU Paiton 2	682.98	149.68	3244978	111712.15	10.2971
3.	PLTGU Gresik GT 1.1	7.82	0	5467532.4	217963.548	34.155
4.	PLTGU Gresik GT 1.2	7.82	0	5467532.4	217963.548	34.155
5.	PLTGU Gresik GT 1.3	7.82	0	5467532.4	217963.548	34.155
6.	PLTGU Gresik CC 3.3.1	73.32	47.10	17177460.3	145165.581	4.554
7.	PLTGU Gresik GT 2.1	7.82	0	5467532.4	217963.548	34.155
8.	PLTGU Gresik GT 2.2	7.82	0	5467532.4	217963.548	34.155
9.	PLTGU Gresik GT 2.3	7.82	0	5467532.4	217963.548	34.155
10.	PLTGU Gresik CC 3.3.1	73.32	47.10	17177460.3	145165.581	4.554
11.	PLTGU Gresik GT 3.1	7.82	0	5467532.4	217963.548	34.155
12.	PLTGU Gresik GT 3.2	7.82	0	5467532.4	217963.548	34.155
13.	PLTGU Gresik GT 3.3	7.82	0	5467532.4	217963.548	34.155
14.	PLTGU Gresik CC 3.3.1	73.32	47.10	17177460.3	145165.581	4.554
15.	PLTU Gresik 1	143.74	40.59	1327126.68	217378.359	132.066
16.	PLTU Gresik 2	143.74	40.59	1327126.68	217378.359	132.066
17.	PLTU Gresik 3	229.50	92.52	5017369.5	169242.579	193.545
18.	PLTU Gresik 4	229.50	92.52	5017369.5	169242.579	193.545
19.	PLTG Gresik 1	6.13	0	352707.3	350680.77	903.969
20.	PLTG Gresik 2	6.13	0	352707.3	350680.77	903.969
21.	PLTG Gilitimur 1	6.33	0	687181.85	683240.965	1762.3893
22.	PLTG Gilitimur 2	6.33	0	687181.85	683240.965	1762.3893
23.	PLTGU M. Karang GT 1.1	7.35	0	5730795	202052.97	108.045
24.	PLTGU M. Karang GT 1.2	7.35	0	5730795	202052.97	108.045
25.	PLTGU M. Karang GT 1.3	7.35	0	5730796	202052.97	108.045
26.	PLTGU M. Karang CC 3.3.1	68.92	44.27	31017735	87825.150	57.3300
27.	PLTGU M. Tawar GT 1.1	0	0	14706521.25	433337.8	49.4605
28.	PLTGU M. Tawar GT 1.2	0	0	14706521.25	433337.8	49.4605
29.	PLTGU M. Tawar 1.3	0	0	14706521.25	433337.8	49.4605
30.	PLTGU M. Tawar GT 2.1	0	0	14706521.25	433337.8	49.4605
31.	PLTGU M. Tawar GT 2.2	0	0	14706521.25	433337.8	49.4605
32.	PLTGU M. Tawar CC 3.3.1	160.10	96.42	43043399	288609.995	7.65800
33.	PLTU M. Karang 1	122.58	31.08	2417820.7	473895.41	120.77935
34.	PLTU M. Karang 2	122.58	31.08	2417820.7	473895.41	120.77935
35.	PLTU M. Karang 3	122.58	31.08	2417820.7	473895.41	120.77935
36.	PLTU M. Karang 4	215.34	89.29	2949187.5	205217.145	83.73
37.	PLTU M. Karang 5	215.34	89.29	2949187.5	205217.145	83.73

Sumber: Data penawaran PT. PJB, Jl. Ketintang Baru No. 11 Surabaya 60231

Catatan:

Harga Batu bara	253 Rp/Kg
Harga MFO	1595,5 Rp/Liter
Harga HSD	15945,5 Rp/Liter
Harga Gas UP Gresik	2,53 US$/MMBTU
Harga Gas UP M. Karang	2,45 US$/ MMBTU
Nilai tukar	9000 Rp/US$

Data total biaya pembangkitan PT. Pembangkitan Jawa Bali yang digunakan sebagai data pembanding dari metode *Ant Colony Search Algorithm* diperoleh dengan menggunakan persamaan pada subbab 2.3.3, dimana a, b, dan c merupakan konstanta dari fungsi biaya bahan bakar yang telah didapat dari data PT. Pembangkitan Jawa Bali pada tabel 3.2. Untuk tiap unit dan diselesaikan mulai dari unit pertama sampai pada unit terakhir yang beroperasi tiap periode atau jam.

3.3 Penjadwalan Unit Pembangkit

Penjadwalan unit pembangkit tenaga listrik merupakan tugas harian yang penting untuk membuat rencana operasi harian. Penjadwalan unit pembangkit listrik ini berisi tentang status on-off unit pembangkit dan disertai dengan pembebanannya 24 jam sampai 168 jam dengan interval waktu 1 jam. Penentuan unit pembangkit yang harus di start dan di stop dapat dipilih dengan memperhatikan kendala (*constraint*) pembangkitan yang telah ditentukan, sehingga didapat pilihan yang optimum dalam arti dapat memenuhi kebutuhan sistem dengan operasi minimum.

Perhitungan dan analisa dari metode *Ant Colony Search Algorithm* dilakukan pada kebutuhan daya yang ditanggung PT. Pembangkitan Jawa Bali pada tanggal 27, 30, dan 31 Juli 2005. Analisa data dilakukan untuk waktu tiga hari tersebut, karena terdapat karateristik kurva yang berlainan dengan keterangan sebagai berikut:

- Tanggal 27 Juli 2005 adalah hari rabu, merupakan beban kerja atau beban penuh.
- Tanggal 30 Juli 2005 adalah hari sabtu, merupakan hari dengan beban setengah penuh
- Tanggal 31 Juli 2005 adalah hari minggu, merupakan beban pada hari libur

Berdasarkan data unit termal yang terdapat pada PT. Pembangkitan Jawa Bali pada sistem tenaga listrik pada tabel 3.1, ternyata pada saat dilakukan pengambilan data, semua unit pembangkit termal dalam kondisi siap beroperasi. Maka input data unit pembangkit termal dapat disusun yang siap beroperasi pada tanggal 27,30, dan 31 Juli 2005.

Data beban harian sistem yang diperoleh dari PT. Pembangkitan Jawa Bali, terdapat data hasil perhitungan mengenai jumlah total pembangkitan, beban total, dan cadangan berputar tiap jam dalam tiap-tiap area. Data-data ini tidak dipakai dalam skripsi ini, karena data tersebut menyangkut sistem secara keseluruhan dalam satu area. Dalam satu area terdapat lebih dari satu perusahaan penyedia energi listrik, seperti PT. Pembangkitan Jawa Bali dan PT Indonesia Power perusahaan milik swasta yang terdapat pada area IV.

Model perhitungan yang digunakan dalam melakukan perhitungan optimalisasi penjadwalan unit-unit pembangkit termal baik PLTU, PLTG, ataupun PLTGU dapat menggunakan karateristik tiap unit termal, meskipun karateristik tiap blok saling tergantung antara unit gas (gas turbin) atau GT dan unit uap (steam turbin) atau ST sehingga sering disebut *combined cycle*. Untuk

memudahkan perhitungan total biaya operasi, dilakukan menggunakan pendekatan per unit termal, dimana data parameter tiap unit GT dapat diambil parameter kombinasi CC 3.3.1.

PT. Pembangkitan Jawa Bali tidak memiliki dasar yang pasti dalam menentukan besarnya cadangan berputar tiap periode jam, tetapi PT. Pembangkitan Jawa Bali menggunakan asumsi bahwa nilai cadangan berputar diambil dari daya terpasang terbesar dari unit pembangkit yang mengalami gagal operasi. Dalam hal ini PT. Pembangkitan Jawa Bali daya terpasang terbesar adalah unit pembangkit PLTU Paiton yang memiliki daya terpasang sebesar 400 MW sebagai nialai cadangan berputar tiap periode jamnya.

3.4 Beban Sistem

Unit-unit Pembangkit yang ada dalam wilayah Jawa – Bali dikoordinasi oleh PT. Pembangkitan Jawa Bali. Proses penjadwalan unit-unit pembangkit dengan menggunakan metode kombinasi *Ant Colony Search Algorithm* bertujuan untuk membuat perencanaan penjadwalan unit pembangkit dalam sistem tenaga listrik untuk dapat memenuhi kebutuhan beban sistem dengan biaya operasi yang seekonomis mungkin.

Untuk dapat mengetahui besarnya pengaruh, dan efisiensi dari metode ini, maka dilakukan evaluasi dan analisa dengan mengambil data unit pembangkit termal dan data beban yang ditanggung PT. Pembangkitan Jawa Bali. Untuk kombinasi penjadwalan dan daya output pembangkit tenaga listrik dalam sistem PT. Pembangkitan Jawa Bali dapat diambil data pada tanggal 27, 30, 31 Juli 2005.

TABEL 3.3
DATA BEBAN UNIT-UNIT PEMABANGKIT TERMAL
PADA PT. PEMBANGKITAN JAWA BALI

JAM	Rabu 27 Juli 2005		Sabtu 30 Juli 2005		Minggu 31 Juli 2005	
	Beban Sistem	Cadangan Berputar	Beban Sistem	Cadangan Berputar	Beban Sistem	Cadangan Berputar
01.00	2300	400	2525	400	2275	400
02.00	2175	400	2300	400	1755	400
03.00	2090	400	2170	400	1755	400
04.00	2090	400	2170	400	1740	400
05.00	2240	400	2470	400	1895	400
06.00	2215	400	2250	400	1970	400
07.00	1990	400	1940	400	1642	400
08.00	2250	400	2065	400	1565	400
09.00	2540	400	2190	400	1615	400
10.00	2590	400	2190	400	1675	400
11.00	2590	400	2210	400	1625	400
12.00	2340	400	2165	400	1575	400
13.00	2575	400	2140	400	1575	400
14.00	2575	400	2190	400	1575	400
15.00	2575	400	2265	400	1575	400
16.00	2475	400	2130	400	1575	400
17.00	2457	400	2197	400	1689	400
18.00	2951	400	2849	400	2689	400
19.00	2981	400	2989	400	2929	400
20.00	2981	400	2934	400	2924	400
21.00	2951	400	2914	400	2904	400
22.00	2664	400	2582	400	2632	400
23.00	2430	400	2375	400	2330	400
24.00	2405	400	2300	400	2215	400

Sumber: Data Penawaran PT. PJB, JL. Ketintang Baru 11 Surabaya 60231

3.5 Data Validasi Program

Pengujian validasi program dilakukan dengan menggunakan jurnal T. Sum-im, W. Ongsakul, " **Ant Colony Search Algorithm for Unit Commitment**", IEEE 2003, yang digunakan sebagai data validasi program. Dalam pengujiannya digunakan 10 unit pembangkit selama 24 jam. Data unit pembangkit dan data beban akan diberikan pada tabel 3-4 dan tabel 3-5

TABEL 3-4
Data generator untuk validasi sistem
dengan 10 unit pembangkit

No unit	P^{max} (MW)	P^{min} (MW)	a	b	c	T^{up}	T_{down}	S_h	Sc	T_{cold} (hour)	Initial State
1	455	150	1000	16.19	0.00048	8	8	4500	9000	5	8
2	455	150	970	17.26	0.00031	8	8	5000	10000	5	8
3	130	20	700	16.60	0.00200	5	5	550	1100	4	-5
4	130	20	680	16.50	0.00211	5	5	560	1120	4	-5
5	162	25	450	19.70	0.00398	6	6	900	1800	4	-6
6	80	20	370	22.26	0.00712	3	3	170	340	2	-3
7	85	25	480	27.74	0.00079	3	3	260	520	2	-3
8	55	10	660	25.92	0.00413	1	1	30	60	0	-1
9	55	10	665	27.27	0.00222	1	1	30	60	0	-1
10	55	10	670	27.79	0.00173	1	1	30	60	0	-1

Sumber: T. Sum-im, W. Ongsakul 2003

TABEL 3-5
Data beban untuk validasi dengan 10 unit pembangkit

Jam	Beban (MW)	Reserve (MW)
1	700	70
2	750	75
3	850	85
4	950	95
5	1000	100
6	1100	110
7	1150	115
8	1200	120
9	1300	130
10	1400	140
11	1450	145
12	1500	150
13	1400	140
14	1300	130
15	1200	120
16	1050	105
17	1000	100
18	1100	110
19	1200	120
20	1400	140
21	1300	130
22	1100	110
23	900	90
24	800	80

Sumber: T. Sum-im, W. Ongsakul 2003

BAB IV

ANALISA DATA DENGAN METODE *ANT COLONY SEARCH ALGORITHM* PADA PT. PEMBANGKITAN JAWA-BALI

4.1 Program Komputer Metode Ant Colony Search Algorithm

Dalam pemecahan masalah komitmen unit digunakan bantuan program komputer. Program ini sangat membantu untuk mempercepat proses perhitungan yang membutuhkan ketelitiean yang tinggi dan sering melibatkan iterasi yang memerlukan waktu lama bila dilakukan perhitungan secara manual.

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

4.2 Algoritma Program

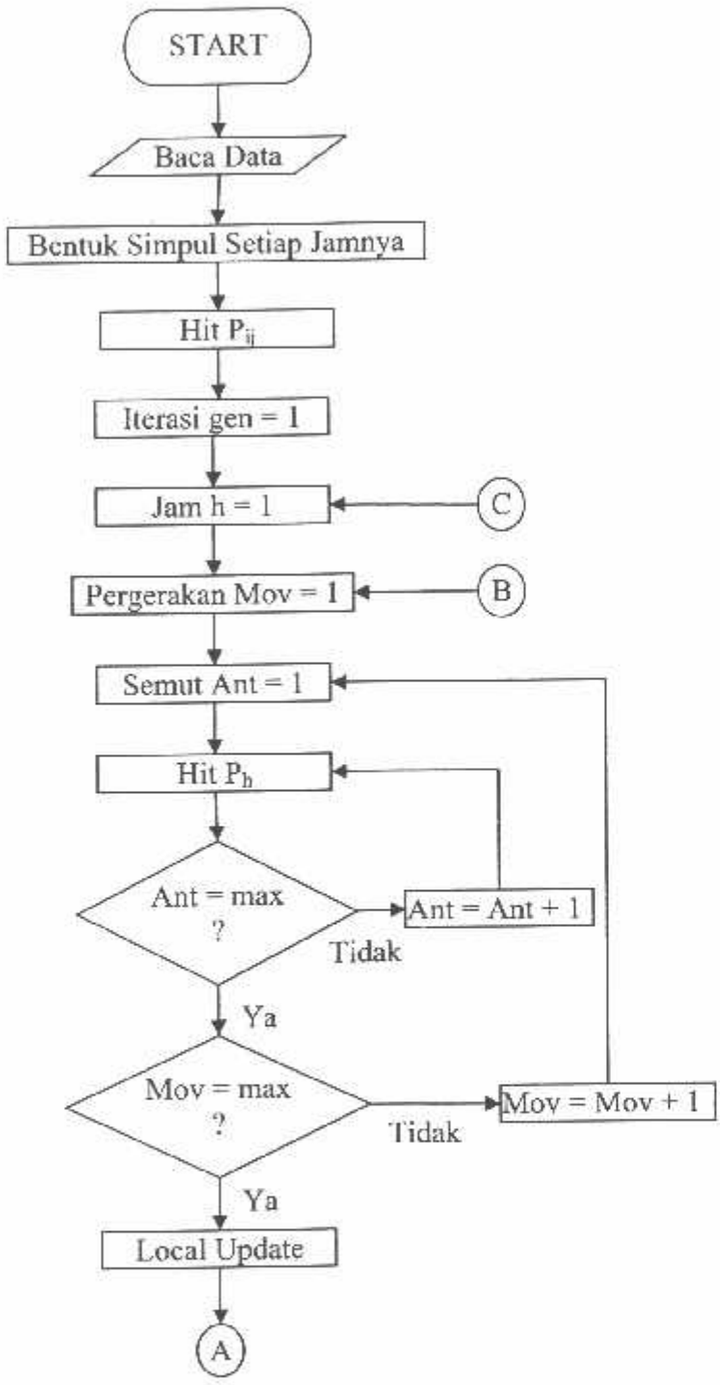
Urutan langkah-langkah dalam program komputer yang digunakan dapat dilihat pada algoritma program berikut :

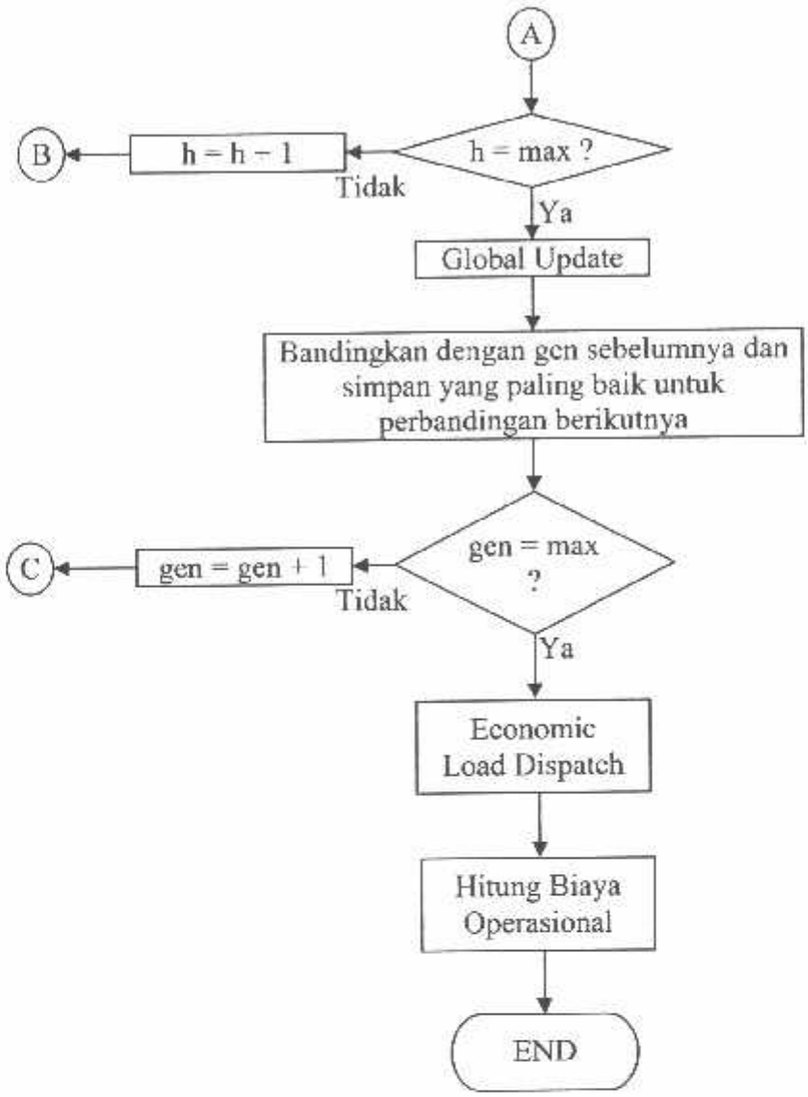
1. Memasukkan data parameter unit pembangkit termal dan data pembebanan harian untuk periode waktu 24 jam.

   Data tiap-tiap unit pembangkit termal yang diperlukan adalah jumlah unit pembangkit, daya maksimum dan daya minimum, konstanta persamaan biaya bahan bakar, harga bahan bakar, biaya start up, *minimum up time* dan *minimum down time*

2. Masukkan parameter *Ant Colony Search Algorithm* yang meliputi jumlah iterasi, jumlah semut, jumlah pergerakan, koefisien jarak (d), konstanta beta, konstanta alpha dan nilai awal pheromone.
3. Memunculkan semut generasi pertama.
4. Menghitung nilai probabilitas pada setiap jalur.
5. Melepaskan semut disetiap simpul.
6. Laksanakan kunjungan kesetiap simpul sampai jumlah pergerakan yang diinputkan terpenuhi kemudian perbaharui jumlah pheromone secara lokal pada jalur yang optimal.
7. Setelah seluruh semut menyelesaikan perjalanannya (telah menemukan sumber makanan) maka dilakukan pembaharuan secara keseluruhan (Global Update) jumlah pheromone pada jalur teroptimal dari sarang semut sampai pada sumber makanan.
8. Simpan segala informasi tentang jalur optimal yang telah didapat dari generasi sebelumnya untuk dilakukan perbandingan terhadap generasi berikutnya.
9. Bersihkan pheromone dari semua jalur.
10. Munculkan generasi semut berikutnya guna mencari kemungkinan jalur yang lebih optimal dari jalur sebelumnya.
11. Hentikan proses jika generasi yang diinputkan telah terpenuhi atau sebagian generasi semut terakhir memilih jalur yang sama denagn jalur yang teroptimal yang dihasilkan dari generasi terdahulu.

4.3 Flowchart *Ant Colony Search Algorithm*

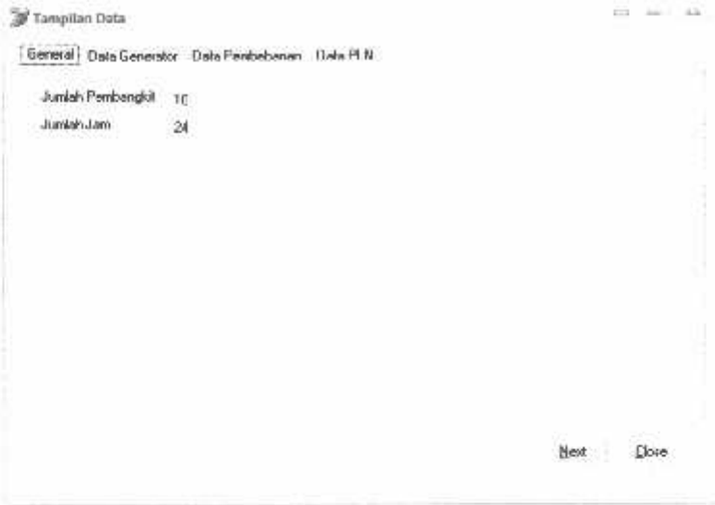



A
B
h = h - 1
Tidak
h = max ?
Ya
Global Update
Bandingkan dengan gen sebelumnya dan simpan yang paling baik untuk perbandingan berikutnya
C
gen = gen + 1
Tidak
gen = max ?
Ya
Economic Load Dispatch
Hitung Biaya Operasional
END

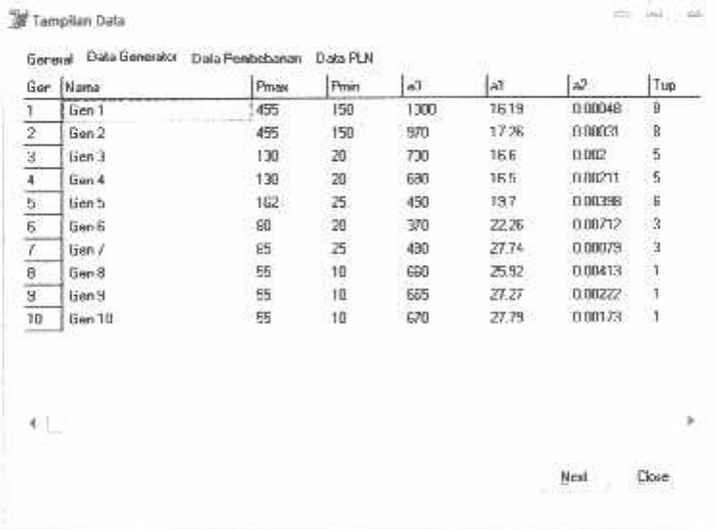
4.4 Hasil Uji Validasi

Sebelum melakukan komputasi program optimalisasi pembebanan unit pembangkit termal pada sistem PT. Pembangkitan Jawa – Bali dengan metode *Ant Colony Search Algorithm,* terlebih dahulu dilakukan pengujian validasi dengan menggunakan Jurnal *Ant Colony Search Algorithm for Unit Commitment.* T-Sum In and W. Ongsakul, IEEE, 2003. dan data yang terlampir pada bab III diatas. Untuk proses komputasi digunakan prosesor AMD Athlon XP™ 2600+, 1,53 GHz dengan RAM 512 Mb. Hasil perhitungan Komputasi pada *Ant Colony Search Algorithm* adalah sebagai berikut :

GAMBAR 4.1
Tampilan Metode Ant Colony Search Algorithm

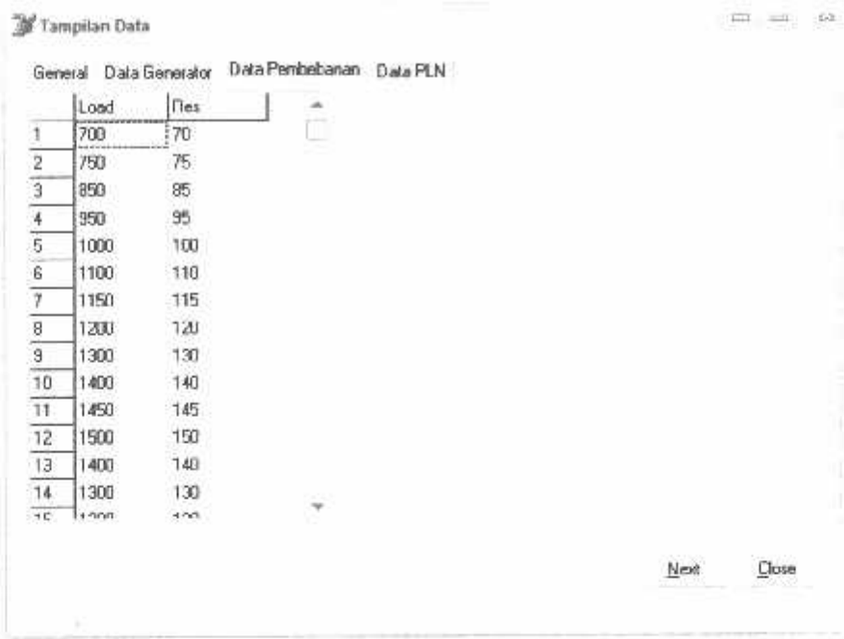


GAMBAR 4.2.
Tampilan Data Untuk Validasi

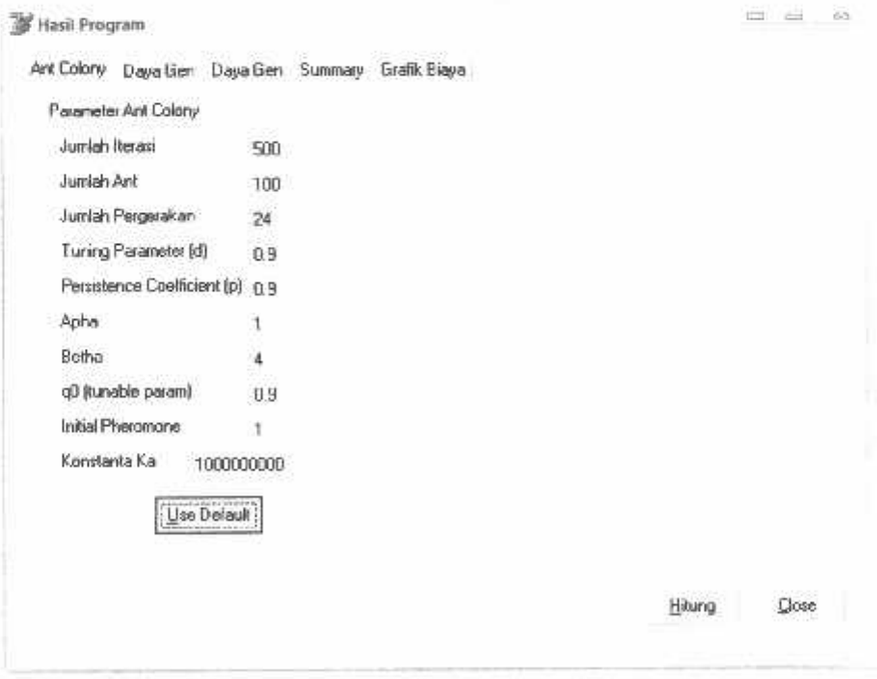


Gen	Nama	Pmax	Pmin	a0	a1	a2	Tup
1	Gen 1	455	150	1000	16.19	0.00048	9
2	Gen 2	455	150	970	17.26	0.00031	8
3	Gen 3	130	20	700	16.6	0.002	5
4	Gen 4	130	20	680	16.5	0.00211	5
5	Gen 5	162	25	450	19.7	0.00398	6
6	Gen 6	80	20	370	22.26	0.00712	3
7	Gen 7	85	25	480	27.74	0.00079	3
8	Gen 8	55	10	660	25.92	0.00413	1
9	Gen 9	55	10	665	27.27	0.00222	1
10	Gen 10	55	10	670	27.79	0.00173	1

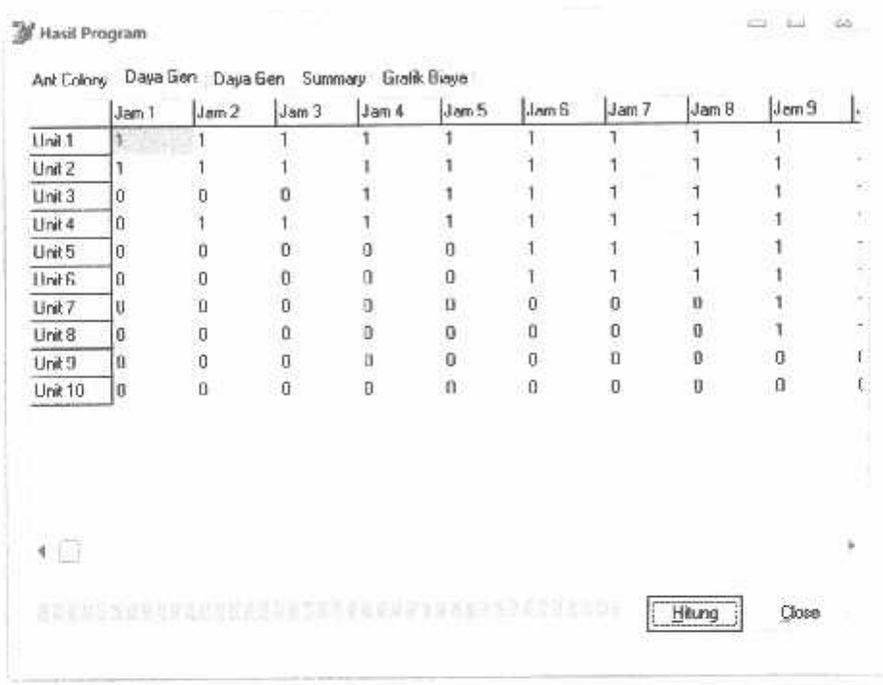
GAMBAR 4.3.
Data Generator Untuk Validasi



GAMBAR 4.4.
Data Pembebanan Untuk Validasi



GAMBAR 4.5
Tampilan Parameter Validasi



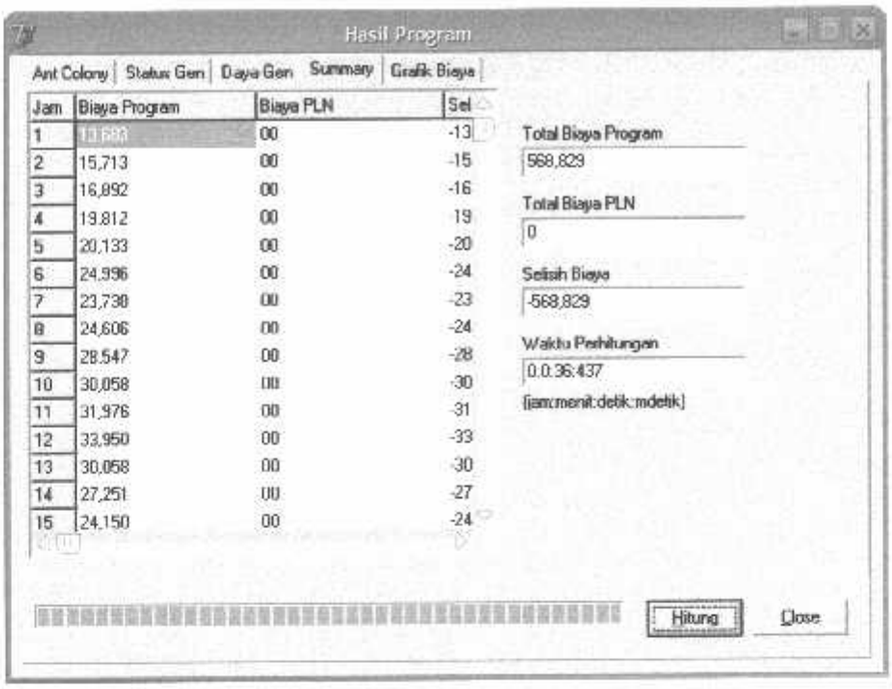
GAMBAR 4.6.
Hasil Perhitungan Penjadwalan Unit Pembangkit Menurut Program

Tabel 4.1
Hasil Perhitungan Penjadwalan Unit Pembangkit Menurut Program

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

Tabel 4.2
Hasil Perhitungan Penjadwalan Unit Pembangkit menurut jurnal

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



GAMBAR 4.7
Hasil Validasi Program Terhadap Jurnal

Dari hasil pengujian disini dapat dilihat bahwa program tersebut layak digunakan, karena dari hasil perhitungan program tersebut mendekati hasil yang ada di jurnal ***Ant Colony Search Algorithm for Unit Commitment***, T-Sum In and W. Ongsakul, IEEE, 2003. dengan menghasilkan biaya total USD 568,829. Pada tampilan tersebut dapat dilihat bahwa hasil perhitungan dari metode *Ant Colony Search Algorithm* mendekati yang ada di jurnal *Ant Colony Search Algorithm for Unit Commitment*, T-Sum In and W. Ongsakul, IEEE, 2003 yaitu sebesar USD 564,049, dengan selisih biaya sebesar USD 4,240 (0,84%). Sehingga program tersebut layak digunakan.

4.5 Hasil Perhitungan dan Analisa Data

Program optimasi pembebanan unit pembangkit termal pada PT. PJB dengan metode kombinasi *Ant Colony Search Algorithm*, terdiri dari beberapa tahap yang secara keseluruhan dijabarkan sebagai berikut:

1. Tahap input data dengan inisialisasi data karateristik tiap unit pembangkit dan data beban sistem tiap jamnya.
2. Mencari perhitungan dan pencarian nilai yang paling minimum untuk biaya, beban sistem, dan cadangan berputar.
3. Pencarian kombinasi penjadwalan unit-unit pembangkit yang paling murah untuk melayani beban sistem pada tiap jamnya.

Seluruh unit termal yang siap beroperasi dalam PT. Pembangkitan Jawa Bali terdiri dari 37 unit pembangkit. Pola kombinasi dari 37 unit pembangkit pada periode waktu 24 jam yaitu pada tanggal 27 juli, 30 juli, dan 31 juli 2005 dapat dilihat pada tabel 4.3, 4.4, dan 4.5 yang digunakan sebagai bahan perbandingan dengan hasil kombinasi metode kombinasi *Ant Colony Search Algorithm*

Table 4.3
Kombinasi Unit pembangkit Termal
Pada Rabu, 27 Juli 2005

Jam	Status ON dan OFF Unit Pembangkit	Beban (MW)	Status ON Unit Pembangkit
01.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2300	14 unit
02.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2175	14 unit
03.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2090	13 unit
04.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2090	13 unit
05.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2240	13 unit
06.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2215	13 unit
07.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	1990	13 unit
08.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2250	13 unit
09.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2540	13 unit
10.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2590	13 unit
11.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2590	13 unit
12.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2340	13 unit
13.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2575	13 unit
14.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2575	13 unit
15.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2575	13 unit
16.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1	2475	13 unit
17.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2457	18 unit
18.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2951	18 unit
19.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2981	18 unit
20.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2981	18 unit
21.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2951	18 unit
22.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2664	18 unit
23.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2430	14 unit
24.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 1	2405	13 unit

Keterangan : 1 = *ON* dan 0 = *OFF*

Tabel 4.4
Kombinasi Unit pembangkit Termal
Pada Sabtu, 30 Juli 2005

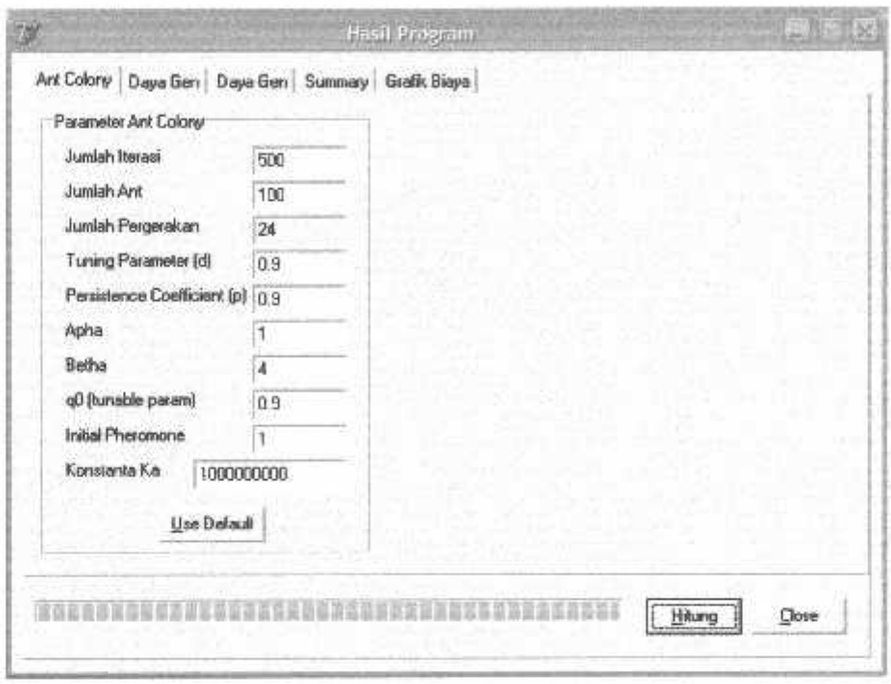
Jam	Status ON dan OFF Unit Pembangkit	Beban (MW)	Status ON Unit Pembangkit
01.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2525	13 unit
02.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2300	13 unit
03.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2170	12 unit
04.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2170	12 unit
05.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2470	12 unit
06.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2250	12 unit
07.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	1940	12 unit
08.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2065	12 unit
09.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2190	12 unit
10.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2190	12 unit
11.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2210	12 unit
12.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2165	12 unit
13.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2140	12 unit
14.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2190	12 unit
15.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2265	12 unit
16.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2130	12 unit
17.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2197	15 unit
18.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2849	15 unit
19.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2989	15 unit
20.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2934	15 unit
21.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2914	15 unit
22.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2582	15 unit
23.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0	2375	13 unit
24.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 1 0 0	2300	12 unit

Keterangan : 1 = *ON* dan 0 = *OFF*

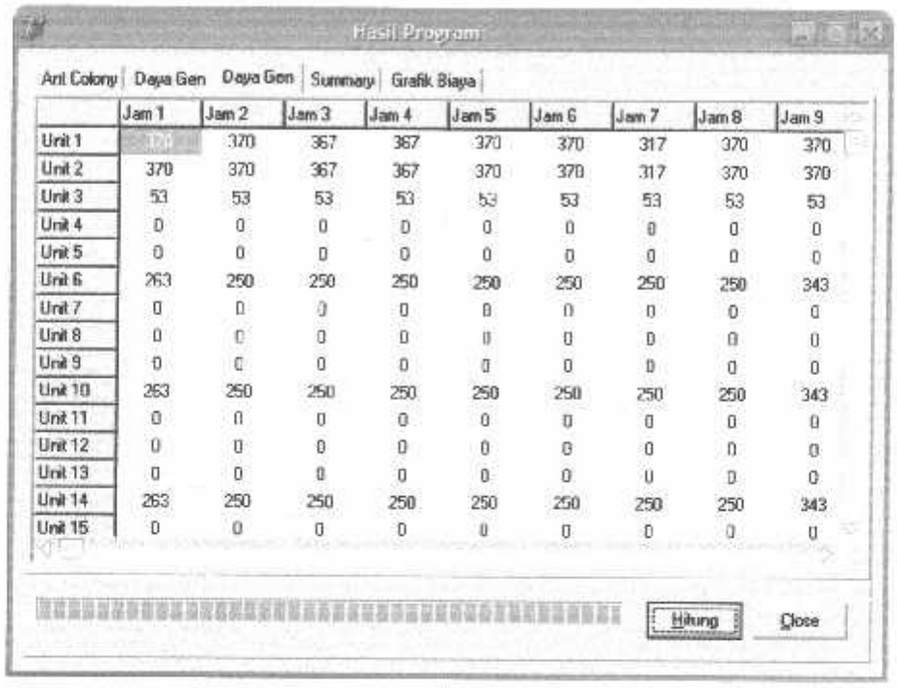
Tabel 4.5
Kombinasi Unit pembangkit Termal
Pada Minggu, 31 Juli 2005

Jam	Status ON dan OFF Unit Pembangkit	Beban (MW)	Status ON Unit Pembangkit
01.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2275	12 unit
02.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1755	12 unit
03.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1755	12 unit
04.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1740	12 unit
05.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1895	12 unit
06.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1970	12 unit
07.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1642	12 unit
08.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1565	12 unit
09.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1615	12 unit
10.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1675	12 unit
11.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1625	12 unit
12.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1575	12 unit
13.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1575	12 unit
14.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1575	12 unit
15.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	1575	12 unit
16.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 1 10	1575	15 unit
17.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 1 1 1 10	1689	14 unit
18.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2689	14 unit
19.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2929	14 unit
20.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2924	14 unit
21.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2904	14 unit
22.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2632	14 unit
23.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2330	12 unit
24.00	1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 10	2215	12 unit

Keterangan : 1 = *ON* dan 0 = *OFF*

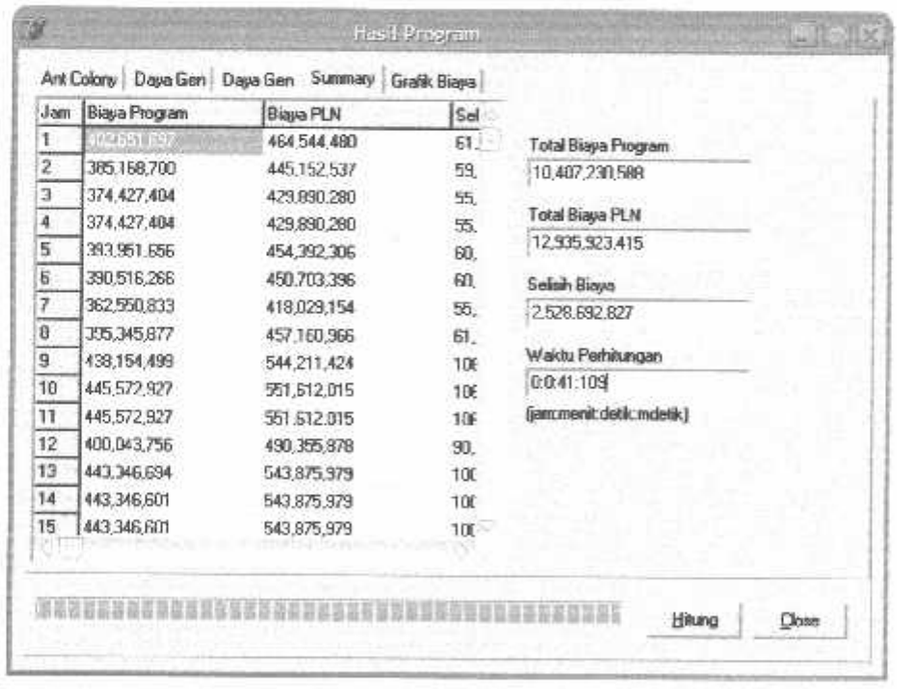


GAMBAR 4.9
tampilan program eksekusi *Ant Colony Search Algorithm*



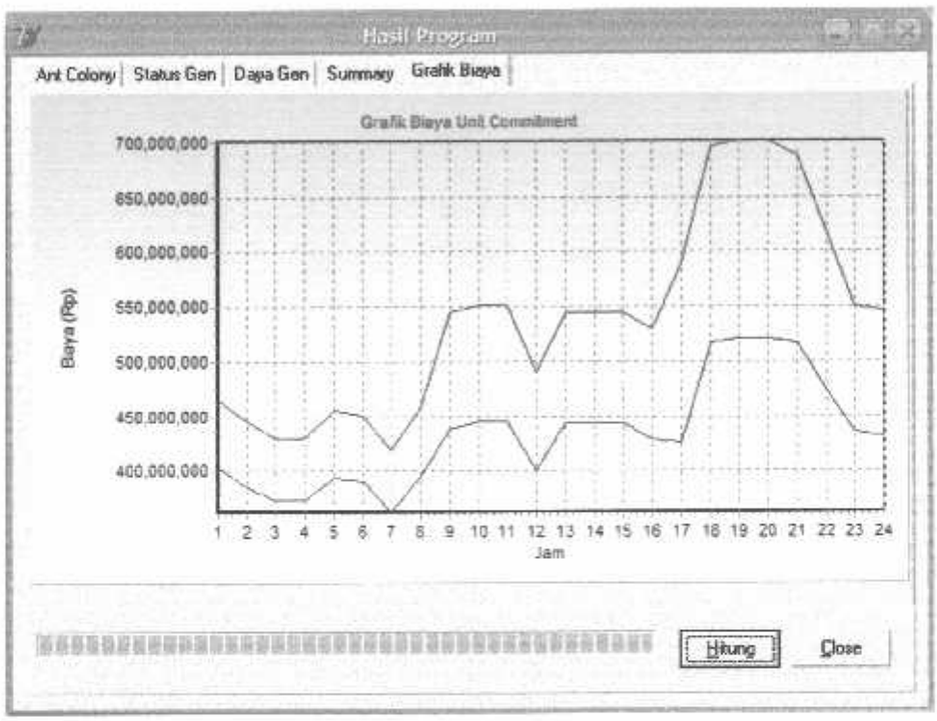
	Jam 1	Jam 2	Jam 3	Jam 4	Jam 5	Jam 6	Jam 7	Jam 8	Jam 9
Unit 1	370	370	367	367	370	370	317	370	370
Unit 2	370	370	367	367	370	370	317	370	370
Unit 3	53	53	53	53	53	53	53	53	53
Unit 4	0	0	0	0	0	0	0	0	0
Unit 5	0	0	0	0	0	0	0	0	0
Unit 6	263	250	250	250	250	250	250	250	343
Unit 7	0	0	0	0	0	0	0	0	0
Unit 8	0	0	0	0	0	0	0	0	0
Unit 9	0	0	0	0	0	0	0	0	0
Unit 10	263	250	250	250	250	250	250	250	343
Unit 11	0	0	0	0	0	0	0	0	0
Unit 12	0	0	0	0	0	0	0	0	0
Unit 13	0	0	0	0	0	0	0	0	0
Unit 14	263	250	250	250	250	250	250	250	343
Unit 15	0	0	0	0	0	0	0	0	0

GAMBAR 4.10
Daya Generator Pada Tanggal 27 Juli 2005



Jam	Biaya Program	Biaya PLN	Sel
1	[illegible]	464,544,480	61
2	385,168,700	445,152,537	59
3	374,427,404	429,890,280	55
4	374,427,404	429,890,280	55
5	393,951,656	454,392,306	60
6	390,516,266	450,703,396	60
7	362,550,833	418,029,154	55
8	395,345,877	457,160,966	61
9	438,154,499	544,211,424	10
10	445,572,927	551,512,015	10
11	445,572,927	551,512,015	10
12	400,043,756	490,355,878	90
13	443,346,694	543,875,979	10
14	443,346,601	543,875,979	10
15	443,346,601	543,875,979	10

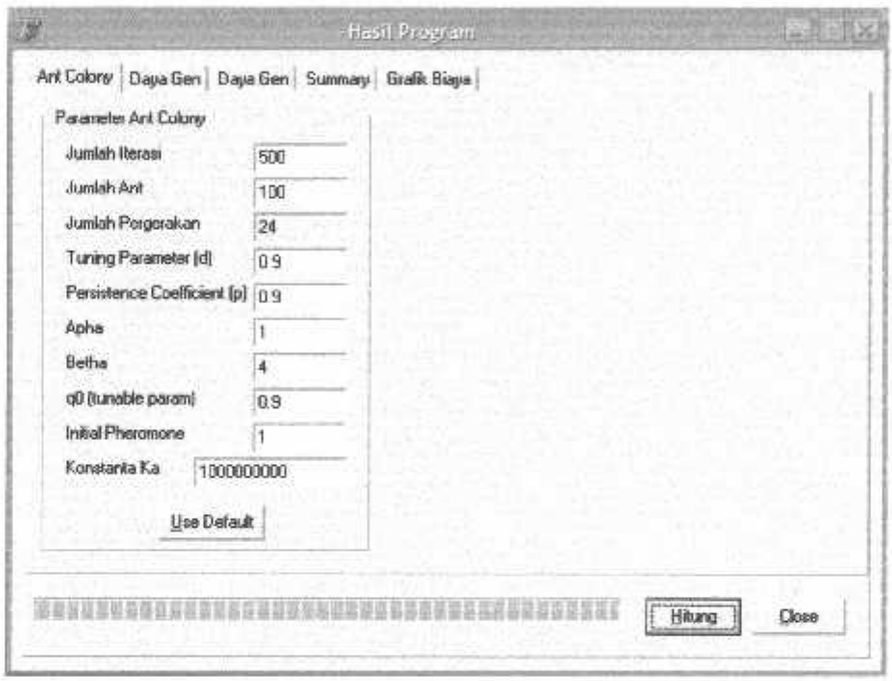
GAMBAR 4.11
Hasil Perhitungan Optimasi Pada Tanggal 27 Juli 2005



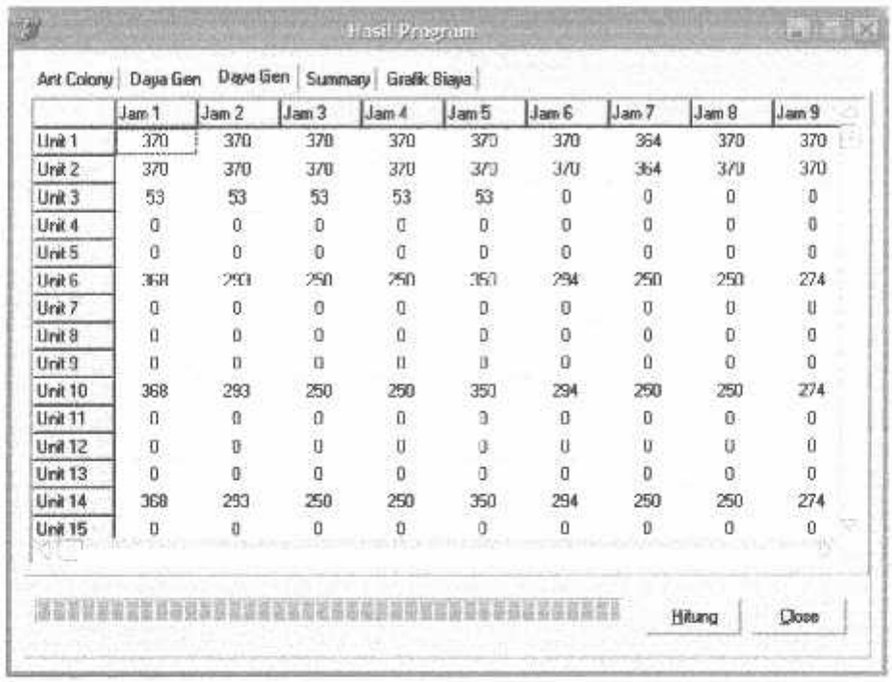
GAMBAR 4.12
Kurva Hasil Perhitungan Optimasi Pada Tanggal 27 Juli 2005

Tabel 4.6
Kombinasi Unit pembangkit Termal
Dengan Metode *Ant Colony Search Algorithm*
Pada Rabu, 27 Juli 2005

Jam	Sistem ON dan OFF Unit Pembangkitan																																					Beban (MW)	Status ON Unit Pembangkit
01.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2300	9
02.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2175	9
03.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2090	9
04.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2090	9
05.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2240	9
06.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2215	9
07.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1990	9
08.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2250	9
09.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2540	9
10.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2590	9
11.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2590	9
12.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2340	9
13.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2575	9
14.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2575	9
15.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2575	9
16.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2475	9
17.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2457	9
18.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2951	11
19.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2981	11
20.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2981	11
21.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2951	11
22.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2664	11
23.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2430	10
24.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2405	10

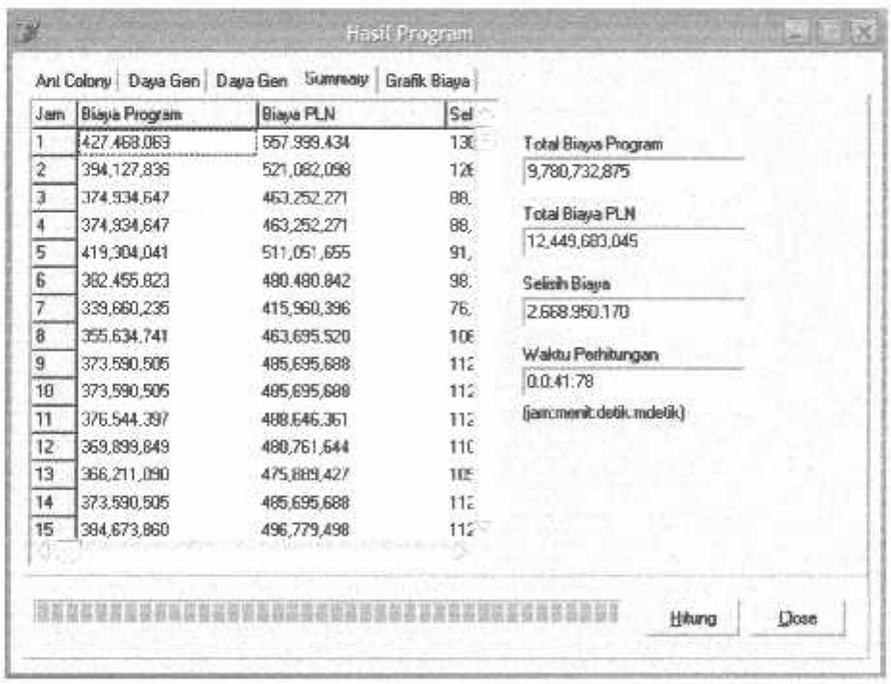


GAMBAR 4.13
tampilan program eksekusi *Ant Colony Search Algorithm*



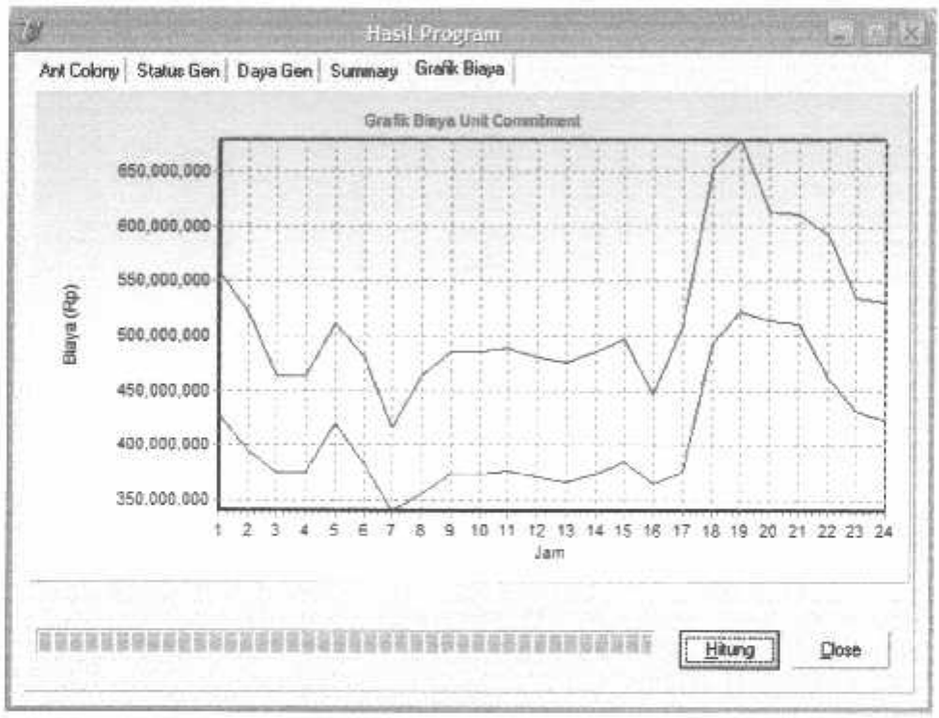
	Jam 1	Jam 2	Jam 3	Jam 4	Jam 5	Jam 6	Jam 7	Jam 8	Jam 9
Unit 1	370	370	370	370	370	370	364	370	370
Unit 2	370	370	370	370	370	370	364	370	370
Unit 3	53	53	53	53	53	0	0	0	0
Unit 4	0	0	0	0	0	0	0	0	0
Unit 5	0	0	0	0	0	0	0	0	0
Unit 6	368	293	250	250	350	294	250	250	274
Unit 7	0	0	0	0	0	0	0	0	0
Unit 8	0	0	0	0	0	0	0	0	0
Unit 9	0	0	0	0	0	0	0	0	0
Unit 10	368	293	250	250	350	294	250	250	274
Unit 11	0	0	0	0	0	0	0	0	0
Unit 12	0	0	0	0	0	0	0	0	0
Unit 13	0	0	0	0	0	0	0	0	0
Unit 14	368	293	250	250	350	294	250	250	274
Unit 15	0	0	0	0	0	0	0	0	0

GAMBAR 4.14
Daya Generator Pada Tanggal 30 Juli 2005



Jam	Biaya Program	Biaya PLN	Sel
1	427,468,063	557,399,434	13
2	394,127,836	521,082,098	12
3	374,934,647	463,252,271	88.
4	374,934,647	463,252,271	88.
5	419,304,041	511,051,655	91.
6	382,455,823	480,480,842	98.
7	339,660,235	415,960,396	76.
8	355,634,741	463,695,520	10
9	373,590,505	485,695,688	11
10	373,590,505	485,695,688	11
11	376,544,397	488,646,361	11
12	369,899,849	480,761,644	11
13	366,211,090	475,889,427	10
14	373,590,505	485,695,688	11
15	384,673,860	496,779,498	11

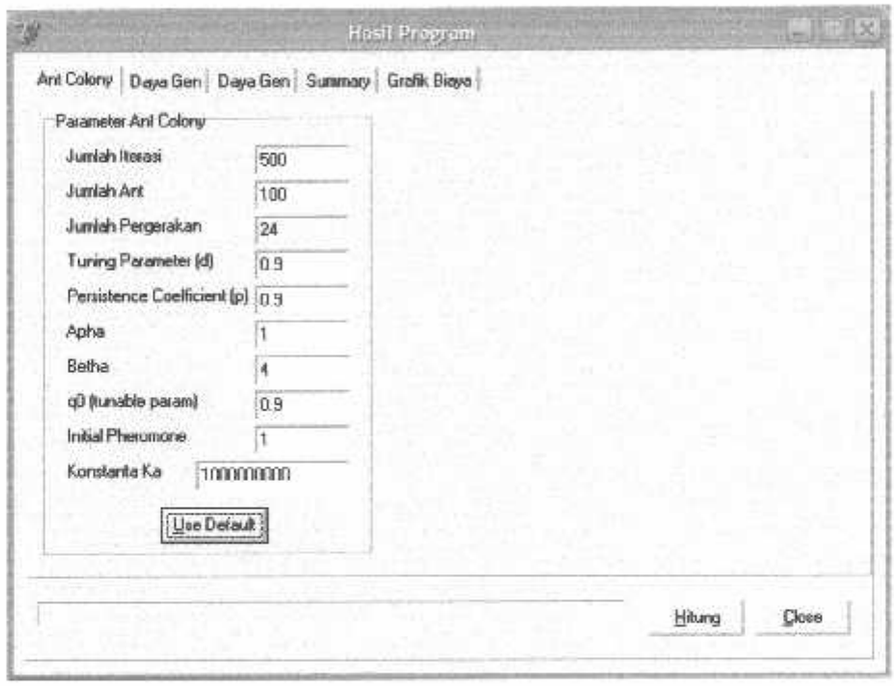
GAMBAR 4.15
Hasil Perhitungan Optimasi Pada Tanggal 30 Juli 2005



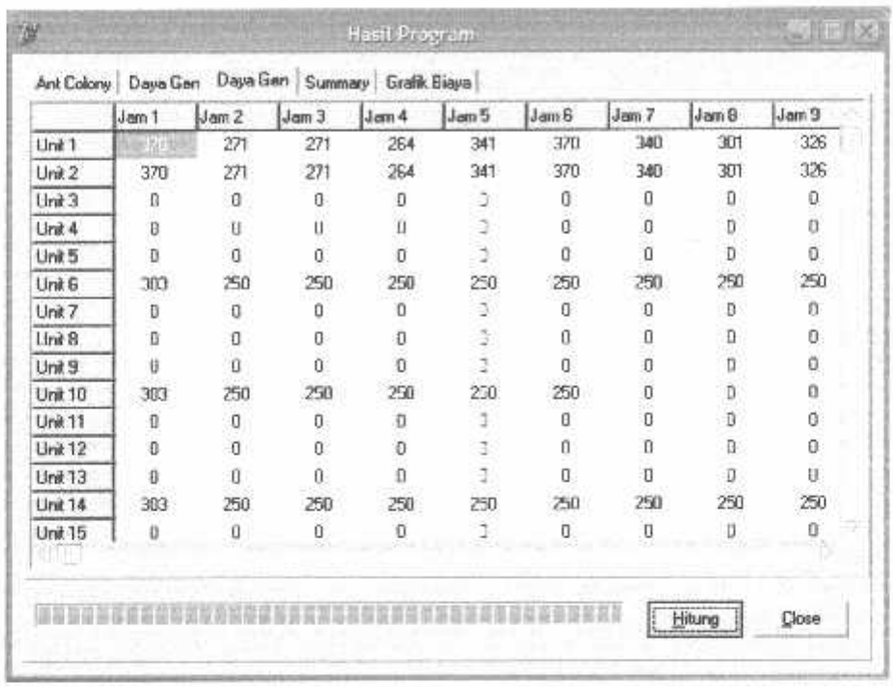
GAMBAR 4.16
Kurva Hasil Perhitungan Optimasi Pada Tanggal 30 Juli 2005

Tabel 4.7
Kombinasi Unit pembangkit Termal
Dengan Metode *Ant Colony Search Algorithm*
Pada sabtu, 30 Juli 2005

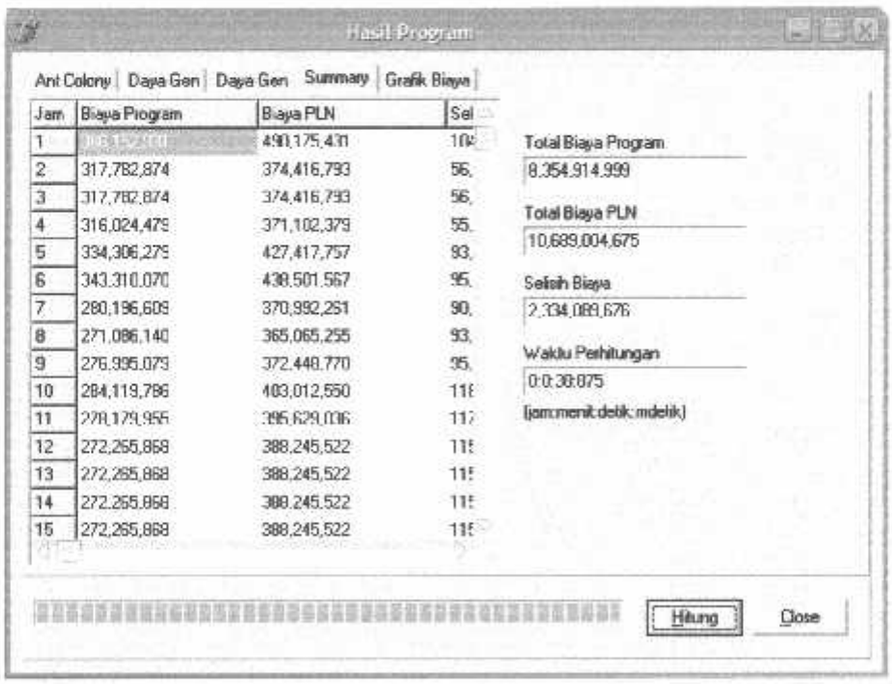
Jam	Sistem ON dan OFF Unit Pembangkitan																																					Beban (MW)	Status ON Unit Pembangkit
01.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2525	8
02.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2300	8
03.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2170	8
04.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2170	8
05.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2470	8
06.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2250	7
07.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1940	7
08.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2065	7
09.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2190	7
10.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2190	7
11.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2210	7
12.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2165	7
13.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2140	7
14.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2190	7
15.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2265	7
16.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2130	7
17.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2197	7
18.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2849	10
19.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2989	11
20.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2934	11
21.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2914	11
22.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2582	11
23.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2375	11
24.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2300	11



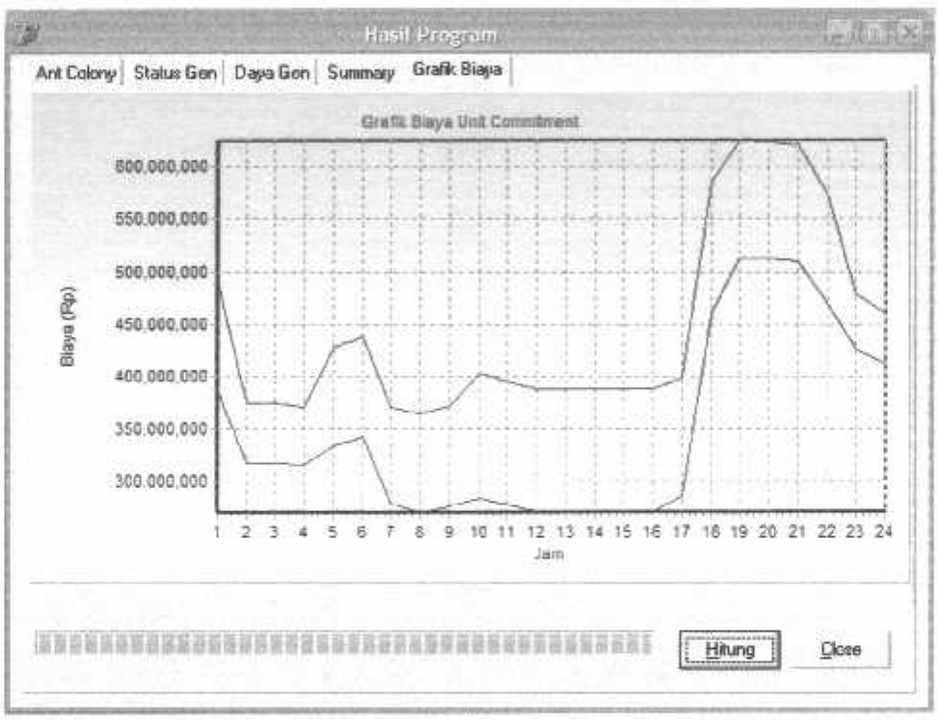
GAMBAR 4.17
tampilan program eksekusi *Ant Colony Search Algorithm*



GAMBAR 4.18
Daya Generator Pada Tanggal 31 Juli 2005



GAMBAR 4.19
Hasil Perhitungan Optimasi Pada Tanggal 31 Juli 2005



GAMBAR 4.20
Grafik Hasil Perhitungan Optimasi Pada Tanggal 31 Juli 2005

Tabel 4.8
Kombinasi Unit pembangkit Termal
Dengan Metode *Ant Colony Search Algorithm*
Pada minggu, 31 Juli 2005

Jam	Sistem ON dan OFF Unit Pembangkitan																																					Beban (MW)	Status ON Unit Pembangkit
01.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2275	7
02.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1755	7
03.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1755	7
04.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1740	7
05.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1895	7
06.00	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1970	7
07.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1642	6
08.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1565	6
09.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1615	6
10.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1675	6
11.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1625	6
12.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1575	6
13.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1575	6
14.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1575	6
15.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1575	6
16.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1575	6
17.00	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1689	6
18.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	2689	9
19.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2929	11
20.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2924	11
21.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2904	11
22.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2632	11
23.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2330	11
24.00	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	2215	11

Setelah mendapatkan hasil yang paling optimal seperti pada penjelasan diatas, maka dilanjutkan dengan melakukan perhitungan total biaya operasi unit-unit pembangkit tiap periode jamnya dan perhitungan biaya operasi dalam system PT. Pembangkitan Jawa-Bali sebagai perbandingan yang diperoleh dari hasil perhitungan fungsi objektif dengan memperhatikan data pembebanan harian pada PT. Pembangkitan Jawa-Bali. Hasil perhitungan biaya operasi dapat dilihat pada tabel sebagai berikut:

Tabel 4.9
Perbandingan Total Biaya Operasional Per Jam PT. PJB
Dengan Metode *Ant Colony Search Algorithm*

Periode Waktu **(24 jam)**	**Total Biaya PT. PJB (Rupiah)**	**Total Biaya *ACSA*** ***(Rupiah)***
Rabu, 27 Juli 2005	12.935.923.415	10.407.230.588
Sabtu, 30 Juli 2005	12.449.683.045	9.780.732.875
Minggu, 31 Juli 2005	10.689.004.675	8.354.914.999

Tabel 4.9 menunjukkan bahwa dengan menggunakan *Ant Colony Search Algorithm* terdapat pengurangan total biaya operasional pembangkitan dalam tiap periode 24 jam (1 hari). Hal ini berarti bahwa proses pemrograman menyebabkan pengurangan biaya operasi pembangkit yang cukup besar di banding dengan biaya operasi PT. PJB.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada analisa program komputer dengan menggunakan softwere Borland Delphi 7.0 yang dijalankan pada komputer AMD Athlon™ XP 2400+, 1.53 GHz dengan *memory* 512 MHz dan hasil perhitungan terhadap penggunaan metode *Ant Colony Search Algorithm* untuk menyelesaikan masalah komitmen unit atau penjadwalan unit-unit pembangkit termal pada PT. Pembangkitan Jawa Bali (PT. PJB) pada tanggal 27, 30, dan 31 Juli 2005, maka dapat diambil kesimpulan sebagai berikut:

1. Pada *Ant Colony Search Algorithm* memberikan sebuah analisa perhitungan dan penyelesaian yang cukup efektif dalam mengoptimalkan pembebanan dan penghematan total biaya operasi dibandingkan dengan total biaya operasional dari PT. Pembangkitan Jawa Bali (PT. PJB). Adapun selisih hasil perhitungan total biaya operasi antara *Ant Colony Search Algorithm* dan PT. Pembangkitan Jawa Bali sebagai berikut:
   a. Pada tanggal 27 Juli 2005, total biaya operasional pada PT. Pembangkitan Jawa Bali (PT.PJB) sebesar Rp. 12.935.923.415,00 sedangkan total biaya hasil perhitungan operasional menggunakan metode *Ant Colony Search Algorithm*

sebesar Rp. 10.407.230.415 Sehingga terjadi penghematan biaya operasional sebesar Rp. 2.528.692.827

b. Pada tanggal 30 Juli 2005, total biaya operasional pada PT. Pembangkitan Jawa Bali (PT.PJB) sebesar Rp. 12.449.683.045 sedangkan total biaya hasil perhitungan operasional menggunakan metode *Ant Colony Search Algorithm* sebesar Rp. 9.780.732.875 Schingga terjadi penghematan biaya operasional sebesar Rp. 2.668.950.170

c. Pada tanggal 31 Juli 2005, total biaya operasional pada PT. Pembangkitan Jawa Bali (PT.PJB) sebesar Rp. 10.689.004.675,00 sedangkan total biaya hasil perhitungan operasional menggunakan metode *Ant Colony Search Algorithm* sebesar Rp.8.354.914.999,00. Sehingga terjadi penghematan biaya operasional sebesar Rp. 2.334.089.676,00

2. Proses eksekusi atau lamanya waktu perhitungan computer untuk metode Ant Colony Search Algorithm adalah sebagai berikut:

- Pada tanggal 27 Juli 2005 waktu eksekusi adalah

  41 Detik :109 *m*Detik

- Pada tanggal 30 Juli 2005 waktu eksekusi adalah

  41 Detik :078 *m*Detik

- Pada tanggal 31 Juli 2005 waktu eksekusi adalah

  38 Detik :875 *m*Detik

5.2 SARAN

berdasarkan kesimpulan diatas, dapat dibuat saran yang berhubungan dengan skripsi ini yaitu :

1. Metode Ant Colony Search Algorithm dapat diterapkan pada PT. Pembangkiatan Jawa-Bali yang sudah terinterkoneksi. Karena hasil perhitungan menunujukkan total biaya operasi pembangkitan yang lebih ekonomis, kombinasi penjadwalan unit-unit pembangkit yang lebih efisien dengan waktu eksekusi yang singkat.
2. untuk studi yang lebih luas, metode ant Colony Search Algorithm ini dapat dikembangkan dengan penambahan unit hidro artinya dapat dikembangkan untuk studi sistem penjadwalan hidrotermal.

DAFTAR PUSTAKA

1. T. Sum-im dan W.Ongsakul, " **ant colony search algorithm for Unit Commitment**", IEEE 2003.

2. Allen J. Wood dan W.F. Bruce. 1984. **Power Generation, Operation, and Control;** John Wiley and sons.

3. Djiteng Marsudi, Ir, " **Operasi Sistem Tenaga Listrik**", Balai Penerbit dan Humas ISTN, 1990.

4. Marco Dorigo dan Luca Maria Gambardella,"**Ant Colony System: A Cooperative Learning Approach to The Travelling Salesman Problem** " IEEE Transaction On Evolutionary Computation, Vol 1, no. 1, 1997.

5. In-Keun Yu, C S Chou, Y H Song" **Application of the ant colony search algorithm to short-term generation scheduling problem of thermal Units**" IEEE, 1998.

6. Marco Dorigo dan Gianni Di Caro,' **Ant Colony Optimization: A New Meta-heuristic'**, IEEE, 1999

LAMPIRAN :

- Berita acara ujian skripsi
- Lembar bimbingan skripsi
- Persetujuan perbaikan skripsi
- Formulir peerbaikan skripsi
- Lembar pengajuan judul skripsi
- Formulir bimbingan skripsi
- Data penawaran PT. PJB
- Data pembebanan PT. PJB
- Listing Ant Colony Search Algorithm

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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

1. **Nama** : HENDRA ANDRIANI
2. **NIM** : 99.12.129
3. **Jurusan** : Teknik Elektro S-1
4. **Konsentrasi** : Teknik Energi Listrik
5. **Judul Skripsi** : PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA-BALI

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

Hari : Sabtu
Tanggal : 18 Maret 2006
Dengan Nilai : 75,2 (B+)

Panitia Ujian Skripsi

Ir. Mochtar Asroni, MSME
Ketua

Ir. F. Yudi Limpraptono, MT
Sekretaris

Anggota Penguji

Ir. H. Almizan Abdullah, MSEE
Penguji Pertama

Ir. I Made Wartana, MT
Penguji Kedua

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

LEMBAR BIMBINGAN SKRIPSI

Nama	: **HENDRA ANDRIANI**
N.I.M	: 99.12.129
Jurusan	: TEKNIK ELEKTRO
Konsentrasi	: ENERGI LISTRIK
Judul Skripsi	.**PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE *ANT COLONY SEARCH ALGORITHM* PADA PT. PEMBANGKITAN JAWA-BALI**
Tanggal Mengajukan	: 14 September 2005
Tanggal Menyelesaikan	: 14 Maret 2006
Dosen Pembimbing	: **Ir. H. CHOIRI**
Telah Dievaluasi Dengan Nilai	: 80 (delapan Puluh)

Mengetahui,
Ketua Jurusan Teknik Elektro

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

Diperiksa dan Disetujui,
Dosen Pembimbing

Ir. H.Choiri
NIP.130703042

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

PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi Jurusan Teknik Elektro jenjang strata satu (S-1) yang diselenggarakan pada :

Hari : Sabtu
Tanggal : 18 Maret 2006

Telah dilakukan perbaikan skripsi oleh :

1. Nama : HENDRA ANDRIANI
2. NIM : 99.12.129
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA-BALI

Perbaikan meliputi :

No	Materi perbaikan	ket
1	Flowchart harus dilengkapi Economic Load Dispatch	
2	Tampilan program eksekusi harus ditampilkan	
3	Informasi biaya bahan bakar harus dicantumkan (sesudah atau sebelum kenaikan)	
4	Waktu perhitungan dicantumkan dalam kesimpulan	

Anggota Penguji

Ir. H. Almizan Abdullah, MSEE
Penguji Pertama

Ir. I Made Wartana, MT
Penguji Kedua

Dosen Pembimbing

Ir. H. Choiri

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : [illegible] DRES A
N I M : 99 [illegible] 129
Perbaikan meliputi :

Waktu perhitungan dicantumkan dlm. kesimpulan.

Malang, 18 Maret 2006

(_____)

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : HENDRA A
NIM : 9812129
Perbaikan meliputi :

1. Flow Chart harus dilengkapi dg. ELD (Economic Load Dispatch).
2. Tampilan Program Eksekusi juga harus di-tampilkan (Bukan hanya yg Validasi Program)
3. Informasi tentang biaya bahan bakar harus di-quantumkan dari keterangan narasumber tanpa in-formasi jelas, sehingga bermakna hanya Rbm akan Rendah?

Malang,

(_____)

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika *)

1	Nama Mahasiswa : HENDRA ANDRIANI		Nim : 99 12 129	
2	Waktu pengajuan	Tanggal : 08	Bulan : 08	Tahun : 2005
3	Spesifikasi judul (berilah tanda silang)			
	ⓐ Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri		e. Elektronika & Komponen f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainya	
4	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) : Ir. I Made Wartana, MT.		Mengetahui, Ketua Jurusan. Ir. F. Yudi Limpraptono, MT Nip. Y. 1039500274	
5	Judul yang diajukan mahasiswa :	PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PJB		
6	Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu			
7	Catatan : Kajian harus lebih akurat krn. yg pernah di bahas Karena Referensi cukup			
	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu		Disetujui, 09 Agustus, 2005 Dosen Ir. I Made Wartana, MT.	

Perhatian :

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

Form.S-2

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

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

BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

Malang, 17 Sept. 2005

Nomor : ITN-766/I.1.A/2/05
Lampiran : satu lembar
Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. **Ir. H. CHOIRI**
Dosen Institut Teknologi Nasional
di –
Malang

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

Nama	: HENDRA ANDRIAN
Nim	: 9912129
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro
Konsentrasi	: T. Energi Listrik (S-1)

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

14 Sept. 2005 s/d 14 Mar. 2006

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

Ketua
Jurusan Teknik Elektro S-1

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

Tindasan :
1. Mahasiswa yang bersangkutan
2. Arsip.

Form. S-4a

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : HENDRA ANDRIANI
Nim : 99.12.129
Masa Bimbingan : 14 September – 14 Maret 2006
Judul Skripsi : PENJADWALAN UNIT PEMBANGKIT TERMAL DENGAN METODE ANT COLONY SEARCH ALGORITHM PADA PT. PEMBANGKITAN JAWA BALI

No	Tanggal	Uraian	Paraf Pembimbing
1	15-10-05	Konsultasi BAB I (Revisi pencantuman keterangan gambar)	
2	22-12-05	Konsultasi BAB II (revisi)	
3	18-01-06	Konsultasi BAB I, BAB II (ACC)	
4	22-01-06	Konsultasi BAB III	
5	25-01-06	Konsultasi BAB IV (Revis)	
6	28-01-06	Konsultasi BAB V	
7	02-02-06	Konsultasi BAB V (ACC)	
8		ACC Di seminarkan	
9			
10			

Malang, Maret 2006
Dosen Pembimbing,

Ir. H. Choiri
NIP 130703042

Form.S-4b

PJB

DATA PENAWARAN
PT PLN PEMBANGKITAN JAWA BALI
AGUSTUS 2002

No.	NAMA PEMBANGKIT	KAPASITAS			LAMA WAKTU (JAM)				BIAYA START UP (JUTA Rp)		KOEFISIEN BIAYA BAHAN BAKAR		
		Daya Terpasang (MW)	MIN (MW)	MAX (MW)	MIN UP TIME	MIN DOWN TIME	COLD START UP	HOT START UP	COLD START UP	HOT START UP	a0	a1	a2
1	UP. PAITON												
	PLTU #1/2 (COAL)	2 x 400	225	370	72	48	17	4	882.98	149.68	3244978	111712.15	10.2971
2	UP. GRESIK												
	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	57.66	31.46	10836203.3	72527.004	368.874
	CC - 2.2.1 (GAS)		164	314	36	10	3	2	65.5	39.26	11795770.8	152515.737	6.831
	CC - 3.3.1 (GAS)	3 x 526	250	480	36	10	3	2	73.32	47.1	17177480.3	145165.581	4.554
	PLTU # 1/2 (GAS)	100	43	85	48	10	9	1	143.74	40.59	1327128.68	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												
	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	16010084	127206.655	35.26
	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	49.4505
	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	180.1	96.42	43043399	288609.995	7.6584
	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

Catatan :

Harga Batubara	253	Rp/Kg
Harga MFO	1595.5	Rp/liter
Harga HSD	1595.5	Rp/liter
Harga Gas UP. Gresik	2.53	US$/MMBTU
Harga Gas UP. M.Karang	2.45	US$/MMBTU
Nilai Tukar	9000	Rp/US$

SUB SISTEM REGION_1

RENCANA : HARI/TANGGAL : RABU, 27 JULI 2005

PT PLN (PEMBANGKITAN TENAGA LISTRIK JAWA-BALI)

		Jam	00.00	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	MKRNG1OC—GT		95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95
	MKRNG1OC1		95	95	95	95	95	95	95	95	95	95	95	95	95	95	90	95	95	95	95	95	95	95	95	95	95	95
	MKRNG2OC1		63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	95	95	95	95	95	95	95	95	63	63
	MKRNG2OC		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
	MKRNG3OC		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
	MKRNG1CC		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
	MKRNG2CC		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
	MKRNG3CC		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
PLTG	MKRNG	#1	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	80	85	85	85	85	85	85	80	80	
		#2	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	80	85	85	85	85	85	85	80	80	
		#3	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	80	85	85	85	85	85	85	80	80	
PLTU	MKRNG	#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	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	130	130	130	130	130	130	130	130
PLTGU	MTWAR1OC—GT		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
	MTWAR2OC1		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
	MTWAR1OC1		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
	MTWAR2OC		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
	MTWAR1OC		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
	MTWAR1CC		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
	MTWAR2CC		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
	MTWAR3CC		400	400	400	400	325	315	315	315	315	315	390	450	390	370	310	400	420	500	550	550	550	550	550	550	550	550
	MTWAR	GT2.1	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	0
	MTWAR	GT2.2	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	0
PLTG	MTWAR	#3.1	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	0
		#3.2	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	0
		#3.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	80	80	80	50	50	40	40		
PLTG	MTWAR	#4.1	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	0
		#4.2	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	0
		#4.3	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	0
PLTP	GSLAR	#4-#6	183	183	183	183	182	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183
	DISARANG		110	110	110	110	110	110	110	110	110	110	110	110	75	75	75	120	120	120	120	120	120	120	90	90	90	
PLTU	Krakatau 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	0
(*)	Pembangkitan Area-1		5000	5000	4950	5000	4950	4950	4850	4850	4850	4850	5000	5000	4750	4850	4900	5000	5150	5300	5400	5400	5400	5400	5400	5400	5400	5400
(**)	Beban Area-1		4934	4958	4931	4978	4902	4908	4928	4925	4920	4919	4980	4950	4743	4877	4969	5098	5346	5506	5614	5706	5739	5703	5687	5468	5496	
	Selisih (*) - (**)		85	127	109	175	133	287	211	196	285	311	310	184	19	33	23	-133	75	7	37	-129	162	-155	-141	-143	-176	
	Cadangan Sekonka		72	24	21	21	55	55	116	117	54	30	18	50	139	117	126	74	81	43	27	27	27	27	27	27	27	27
	Cadangan Area-1		672	652	727	681	709	832	843	828	812	892	937	997	710	850	900	817	790	740	700	700	700	650	630	322	322	

SUB SISTEM REGION_1

RENCANA : HARI/TANGGAL : RABU , 27 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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	Rata-2
PLTGU	MKRNG1OC –GT 1 1	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95
	MKRNG1OC1	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95
	MKRNG2OC1	63	95	95	95	63	63	63	63	63	63	95	95	95	95	95	75	63	63	63	63	63	63	63	73
	MKRNG2OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG3OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG1CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG2CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG3CC	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 #1	80	80	80	80	80	80	80	80	80	80	85	85	85	85	85	80	80	80	80	80	80	80	80	70
	#2	80	80	80	80	80	80	80	80	80	80	85	85	85	85	85	80	80	80	80	80	80	80	80	70
	#3	80	80	80	80	80	80	80	80	80	80	85	85	85	85	85	80	80	80	80	80	80	80	80	70
PLTU	MKRNG #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
	#5	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	130	120	120	120	120	120	120	125
PLTGU	MTWAR1OC –GT 1 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
	MTWAR2OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR3OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR1OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR3OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR1CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR2CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR3CC	550	550	550	550	550	550	550	500	450	550	560	560	560	560	560	550	560	550	500	475	450	450	475	525
	MTWAR GT 2.1	0	0	0	0	0	0	0	65	80	80	130	135	135	135	135	130	120	120	65	0	0	0	0	28
	MTWAR GT 2.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	65	65	65	5
PLTG	MTWAR #3.1	0	0	0	0	0	0	0	40	50	60	120	120	140	140	140	100	80	80	50	50	50	50	50	25
	#3.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	#3.3	40	120	80	60	40	40	40	40	50	60	120	120	140	120	120	100	80	80	50	50	0	0	0	42
PLTG	MTWAR #4.1	0	0	0	0	0	0	0	50	50	60	120	120	140	140	140	100	80	80	50	50	0	0	0	24
	#4.2	0	0	0	0	0	0	0	0	50	60	120	120	140	140	140	100	80	80	50	50	50	50	0	26
	#4.3	0	0	0	0	0	0	0	40	50	60	120	130	140	140	140	100	80	80	50	50	50	50	50	29
	PEMBELIAN DARI LUAR PLN																								
PLTP	Gas4K #4-#6	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	191
	CIKARANG	120	120	120	120	120	120	120	120	120	135	135	135	135	135	135	135	135	135	135	110	110	110	110	115
PLTU	Krakatau 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
(*) Pembangkitan Area-1		5421	5455	5508	5466	5456	5447	5418	5529	5567	5593	5636	5715	5715	5645	5625	5705	5718	5933	5815	5574	5337	5318	5243	5542
(**) Beban Area-1		5371	5455	5508	5466	5456	5447	5418	5597	5593	5646	5797	5727	5704	5756	5730	5624	5723	5627	5412	5360	5231	5192	4965	5571
Selisih (*) - (**)		-236	298	292	282	320	334	-200	-69	85	107	438	548	501	589	595	919	245	306	262	214	212	257	418	102
Cadangan Sekeliling		37	44	36	35	37	33	61	94	74	77	76	53	22	29	20	62	-2	61	76	93	81	91	97	61
Cadangan Putar		243	115	165	185	217	237	252	704	692	642	450	120	30	33	50	290	417	462	717	722	637	647	572	392

SUB SISTEM REGION 4

RENCANA : HARI/TANGGAL : RABU, 27 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

		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	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
	PLTA	SUTAMI	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
	PLTA	BRANTAS	13	13	13	13	13	13	13	13	13	13	13	13	10	10	10	10	10	10	10	10	10	10	10	10	10
1	PLTU	PITON #1	360	350	350	350	360	360	360	360	360	360	360	360	310	310	310	360	360	350	367	360	350	360	350	310	310
2		PITON #2	360	350	350	350	360	360	360	360	360	360	360	360	310	310	310	360	360	360	360	360	350	350	350	310	310
3	PLTGU 1	GRSIK11OC - GT 1 A	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
		GRSIK11OC1 -	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
		GRSIK12OC1,	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
4		- GRSIK12OC	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		- GRSIK13OC	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
		GRSIK11CC	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
		GRSIK12CC	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		GRSIK13CC	300	300	275	250	250	250	250	250	250	275	300	250	250	250	250	250	275	300	300	325	325	325	325	275	275
	PLTGU 2	7 GRSIK21OC - GT 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
		GRSIK10CC2	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
		GRSIK20C2	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		- GRSIK22OC	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		- GRSIK23OC	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
		GRSIK21CC	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
		GRSIK22CC	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164
		GRSIK23CC	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 3	GRSIK31OC - GT	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
		GRSIK10C3	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
		GRSIK20C3	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
		- GRSIK32OC	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
		- GRSIK13OC	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
		GRSIK31CC	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
		GRSIK32CC	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
		GRSIK33CC	300	300	300	250	250	250	250	250	250	300	300	300	300	250	300	300	300	300	300	325	325	325	325	300	300
	PLTU	GRSIK #1	50	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	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	PLTU	GRSIK #3	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	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
	PLTG	GI EMR #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
		#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
		GRESIK #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
		#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
(*) Pembangkitan Area 4			4643	3613	4542	4513	4423	4127	4423	4162	4477	4508	4804	4493	4148	4010	4040	4496	4515	4503	4565	4648	4556	4610	4640	4415	4415
(**) Beban Area-4			2542	2458	2468	2400	2388	2375	2334	2445	2309	2525	2549	2396	2165	2135	2268	2376	2424	2498	2498	2545	2558	2617	2603	2387	2427
Selisih (*) - (**)			2101	2155	2075	2043	2035	2047	2089	1977	2037	1983	2053	2147	1983	1937	1972	2113	2040	2065	2068	2093	2092	2113	2137	2033	1988
Tabungan Sebelum			67	65	62	57	56	56	56	56	56	60	65	59	50	57	60	59	60	63	63	59	61	56	56	61	61
Cadangan PLTGU			370	400	420	520	510	540	540	540	540	445	360	420	420	890	720	470	445	395	395	320	310	320	320	545	545

SUB SISTEM REGION_4

RENCANA : HARI/TANGGAL : RABU , 27 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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	Rata-2
PLTA	Area 4	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
PLTA	SUTAMI	25	25	25	25	25	25	25	50	50	50	90	90	90	90	90	90	90	50	50	25	25	25	25	39
PLTA	BRANTAS	10	10	10	10	10	10	10	13	13	20	20	20	20	20	20	17	17	13	13	10	10	10	10	12
PLTU	PITON #1	360	360	360	360	360	360	360	310	310	360	370	370	370	370	370	370	370	370	370	360	360	360	360	355
	PITON #2	360	360	360	360	360	360	360	310	310	360	370	370	370	370	370	370	370	370	370	360	360	360	360	355
PLTGU 1	GRSIK11OC-CT #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
	GRSIK10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK12OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK13OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK11CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK12CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK13CC	325	325	325	325	325	325	250	250	250	250	325	350	350	350	350	350	350	325	275	275	275	275	275	292
PLTGU 2	GRSIK21OC-CT #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
	GRSIK10C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK22OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK23OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK21CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK22CC	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164	164
	GRSIK23CC	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 3	GRSIK31OC-CT #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
	GRSIK10C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK32OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK33OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK31CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK32CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK33CC	325	325	325	325	325	325	300	250	250	275	325	325	325	325	325	325	325	325	325	275	275	275	275	298
PLTU	GRSIK #1	50	50	50	50	50	50	50	50	50	50	80	80	80	80	80	80	80	50	50	50	50	50	50	54
	#2	50	50	50	50	50	50	50	50	50	50	80	80	80	80	80	80	80	50	50	50	50	50	50	54
PLTU	GRSIK #3	90	90	90	90	90	90	90	90	90	135	167	167	167	167	167	167	167	167	140	100	90	90	90	105
	#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
PLTG	GUAMR #1	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	0	0	0	4
	#2	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	0	0	0	4
	GRESIK #1	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	0	0	0	4
	#2	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	0	0	0	4
(*) Pembangkitan Area-4		4840	4680	4640	4540	4540	4505	4545	4473	4482	4651	5248	5273	5273	5233	5233	5230	5170	5103	5012	4829	4710	4635	4564	4837
(**) Beban Area-4		2774	2600	2593	2560	2505	2439	2431	2540	2734	3246	3523	3519	3543	3468	3444	3355	3222	3102	2953	2759	2711	2655	2650	2833
Selisih (*) - (**)		2066	2079	2046	1979	2031	2066	2073	1933	1748	1546	1724	1753	1731	1765	1789	1875	1948	1999	2073	2070	1999	2015	1870	2004
Cadangan Setelika		50	38	10	50	50	48	61	94	36	69	35	19	19	19	19	19	19	38	43	54	52	58	61	56
Cadangan Putar		320	280	170	320	320	370	529	570	[illegible]	561	734	209	209	209	209	209	209	224	411	440	460	515	490	440

RENCANA : HARI/TANGGAL : SABTU , 30 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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	MKRNG1OC – GT 1.1	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	55	95	95	95	95	95	95	95	95
	MKRNG1OC1	95	95	95	95	95	95	95	95	95	95	95	95	90	95	95	95	55	95	95	95	95	95	95	95	95
	MKRNG2OC1	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	95	55	95	95	95	95	95	95	95	65
	- MKRNG2OC	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
	- MKRNG3OC	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
	MKRNG1CC	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
	MKRNG2CC	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
	MKRNG3CC	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 #1	80	80	80	80	80	80	80	80	80	80	80	80	80	80	85	85	85	85	85	85	85	85	85	85	85
	#2	80	80	80	80	80	80	80	80	80	80	80	80	80	80	85	85	85	85	85	85	85	85	85	85	85
	#3	80	80	80	80	80	80	80	80	80	80	80	80	80	80	85	85	85	85	85	85	85	85	85	85	85
PLTU	MKRNG #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
	- #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
PLTGU	MTWAR1OC – GT 1.1	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
	MTWAR1OC1	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
	MTWAR2OC1	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
	- MTWAR1OC	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
	- MTWAR3OC	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
	MTWAR1CC	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
	- MTWAR2CC	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
	MTWAR3CC	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
	MTWAR 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
	MTWAR 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
	PEMBANGKITAN MURAH (MUST)																									
PLTG	MTWAR #3.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	60	100	100	110	140	140	100	50
	#3.2	50	50	50	50	50	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	#3.3	50	50	50	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTG	MTWAR #4.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	80	100	140	140	140	100	60
	#4.2	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
	#4.3	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
	PEMBELIAN DARI LUAR PLN																									
PLTP	GSLAK #5+6	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183
	CIKARANG	110	110	110	110	110	110	110	110	110	110	110	110	75	75	75	120	120	120	120	120	120	120	120	90	30
PLTU	Krakatau 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
Pembangkitan Jawa-Bali		4614	4809	4863	4714	4735	4595	4595	4545	4553	4765	4690	4710	4675	4810	4855	4970	5137	5340	5225	5265	5324	5329	5375	5160	5049
(*) Beban Area-1		4638	4867	4750	4700	4562	4573	4522	4504	4554	4664	4949	4839	4513	4477	4619	4735	4911	5053	5176	5196	6244	5363	5303	5108	5095
Selisih (*) - (**)		23	3	63	83	73	22	73	41	38	31	-79	-124	2	193	276	235	159	62	69	69	81	-18	2	52	76
Cadangan Sesaat		82	57	85	81	85	73	71	71	73	82	98	79	81	79	27	24	29	9	9	10	0	0	0	40	14
Cadangan Putar		320	551	613	715	710	545	595	505	486	285	150	135	400	345	120	90	130	120	115	95	15	15	15	150	255

SUB SISTEM REGION_1

RENCANA HARI/TANGGAL : SABTU , 30 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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	Rata-2
PLTGU	MKRNG1OC - #1	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95
	MKRNG1OC1	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95
	MKRNG2OC1	65	65	65	65	65	65	65	65	65	65	95	95	95	95	95	95	65	65	65	65	65	65	65	74
	- MKRNG2OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	- MKRNG3OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	- MKRNG1CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG2CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	- MKRNG3CC	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 #1	85	85	85	85	85	60	60	60	60	60	85	85	85	85	80	80	80	80	80	80	80	80	80	79
	#2	85	85	85	85	85	60	60	60	60	60	85	85	85	85	80	80	80	80	80	80	80	80	80	79
	#3	85	85	85	85	85	60	60	60	60	60	85	85	85	85	80	80	80	80	80	80	80	80	80	79
PLTU	MKRNG #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
	#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
PLTGU	MTWAR1OC - #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
	MTWAR1OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR2OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	- MTWAR1OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	- MTWAR1OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR1CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR2CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR3CC	500	500	490	550	550	490	490	420	420	490	490	490	550	550	490	550	550	550	475	500	420	425	425	480
	MTWAR GT #1	0	0	0	0	0	0	0	65	80	80	100	120	120	120	100	100	100	100	80	80	80	60	80	39
	MTWAR GT #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
PLTG	MTWAR #3.1	60	140	130	80	80	40	40	40	40	40	60	130	130	120	80	60	60	60	60	60	60	60	60	81
	#3.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	#3.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
PLTG	MTWAR #4.1	60	140	130	80	80	40	40	40	40	40	60	130	130	110	80	60	60	60	60	60	60	0	0	49
	#4.2	0	0	0	0	0	0	0	40	40	40	60	130	130	110	80	60	60	60	60	60	0	0	0	19
	#4.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	PEMBELIAN DARI LUAR PLN																								
PLTP	GSLAK #4-#6	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183
	CIKARANG	120	120	120	120	120	120	120	120	120	120	135	135	135	135	135	135	135	135	135	110	110	110	110	115
PLTG	Krakatau Steel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
(*) Pembangkitan Jawa-1		5703	5789	5776	5734	5812	4814	4814	4760	4734	4659	5324	5273	5529	5529	5194	5334	5264	5234	5184	5185	4985	4835	4835	5074
(**) Beban Area-1		5672	5784	5956	5924	5917	4953	5001	4991	4975	5040	5466	5508	5639	5583	5511	5483	5387	5324	5084	5016	4824	4729	4685	5064
Selisih (*) - (**)		7	90	21	-5	-68	-139	-187	-192	-181	-141	-142	-109	-60	-54	-117	-149	-139	-78	31	95	111	106	150	5
Cadangan Sekotika		34	3	59	34	34	83	83	83	92	87	40	26	26	23	29	29	92	92	100	98	93	89	85	54
Cadangan Putar		355	45	215	115	215	530	530	525	825	725	315	60	80	110	245	305	385	385	480	505	475	415	415	353

SUB SISTEM REGION_4

RENCANA : HARI/TANGGAL : SABTU, 30 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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
PLTA	SUTAmi	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
PLTA	BRANTAS	11	11	11	11	11	11	11	11	11	11	11	11													
PLTU	PITON #1	350	350	350	350	350	350	350	350	350	350	350	350	350	300	300	300	300	300	300	300	300	300	300	300	300
	PITON #2	350	350	350	350	350	350	350	350	350	350	350	300	250	250	250	250	250	250	250	250	250	250	250	250	250
PLTGU 1	GRSIK11GC + GT 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
	GRSIK11GC1	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
	GRSIK12GC1	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
	GRSIK12GC	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
	GRSIK13GC	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
	GRSIK11GC	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
	GRSIK12CC	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
	GRSIK13CC	400	400	350	325	300	300	300	300	350	400	400	325	250	250	250	250	250	250	250	250	250	250	250	250	250
PLTGU 2	GRSIK21GC + GT 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
	GRSIK21GC2	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
	GRSIK22GC2	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
	[illegible]	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
	GRSIK23GC	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
	GRSIK21GC	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
	GRSIK22CC	250	250	225	185	185	164	164	164	184	184	224	164	164	164	164	164	164	164	164	164	164	164	164	164	164
	GRSIK23CC	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 3	GRSIK31GC + GT 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
	GRSIK31GC3	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
	GRSIK32GC3	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
	GRSIK22GC	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
	GRSIK33GC	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
	GRSIK31CC	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
	GRSIK32CC	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
	GRSIK33CC	350	350	325	300	300	300	300	300	350	400	400	325	250	250	250	250	250	300	300	300	300	300	300	300	300
PLTU	GRSIK #1	50	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	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
PLTU	GRSIK #3	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	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
PLTG	GETMR #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
	#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
	GRESIK #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
	#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
(*) Pembangkitan Area-4		4803	4803	4803	4731	4686	4667	4667	4667	4787	4827	4852	4342	4350	3350	3398	3398	3598	3848	3948	3948	3848	3998	3998	3998	3998
(**) Beban Area-4		2507	2503	2403	2795	2407	2759	2391	2412	2428	2513	2440	2240	2025	2003	2108	2208	2259	2343	2364	2367	2312	2383	2385	2421	2220
Selisih (*) - (**)		2296	2400	2337	2314	2280	2208	2276	2254	2359	2399	2512	2302	1943	1532	1492	1390	1438	1506	1505	1482	1496	1523	1534	1571	1518
Cadangan Sekunder		80	80	75	70	69	68	68	68	74	79	82	100	87	83	83	83	83	66	66	66	66	66	66	66	66
Cadangan Putar		495	495	595	665	710	721	731	731	811	811	436	858	1238	1601	1601	1601	1501	1351	1351	1351	1351	1351	1351	1351	1408

SUB SISTEM REGION_4

RENCANA : HARI/TANGGAL : SABTU , 30 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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	Rata ?
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
PLTA	SUTAMI	25	25	25	25	25	25	25	50	50	90	90	90	90	90	90	90	90	80	50	25	25	25	25	38
PLTA	BRANTAS						14	14	14	14	14	22	22	22	22	22	19	19	19	18	14	14	14	14	12
PLTU	PITON #1	300	300	350	350	300	300	300	300	300	310	360	370	370	370	370	360	360	360	360	360	360	350	360	333
	PITON #2	250	250	300	350	350	250	250	250	300	310	360	370	370	370	370	360	360	360	360	360	360	350	360	309
PLTGU 1	GRSIK11OC - GT 1.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK1OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK2OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK12OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK13OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK11CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK12CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK13CC	250	250	250	300	325	325	325	325	300	325	400	425	425	425	425	425	425	400	325	300	300	300	300	318
PLTGU 2	GRSIK21OC - 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
	GRSIK1OC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK2OC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK22OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK23OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK21CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK22CC	164	164	164	164	164	164	164	164	164	164	250	250	250	225	200	164	164	164	164	164	164	164	164	179
	GRSIK23CC	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 3	GRSIK31OC - GT 3.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK1OC3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK2OC3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK32OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK33OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK31CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK32CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK33CC	300	300	300	300	300	300	300	300	300	300	375	425	425	425	425	425	425	400	325	300	300	300	300	323
PLTU	GRSIK #1	50	50	50	50	50	50	50	50	50	50	90	90	90	80	80	80	80	80	50	50	50	50	50	56
	#2	50	50	50	50	50	50	50	50	50	50	90	90	80	80	80	80	80	80	50	50	50	50	50	56
PLTU	GRSIK #3	90	90	90	50	90	50	90	90	90	135	167	167	167	167	167	167	167	167	140	90	50	50	90	105
	#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
PLTG	GLTMR #1	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	0	0	0	4
	#2	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	0	0	0	4
	GRESIK #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
	#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
(*) Pembangkitan Area-4		2811	2949	3242	3622	3829	3835	4035	4250	4284	4541	5475	5580	5579	5525	5499	5421	5421	5755	5346	4952	4664	4715	4555	4458
(**) beban Area-4		2731	2893	2790	2823	2691	2232	2388	2321	2423	2706	3139	3291	3282	3211	3246	3166	3035	3022	2781	2686	2273	2486	2462	2768
Selisih (*) - (**)		1508	3456	1488	1573	1638	1753	1753	1719	1811	1878	2285	2319	2308	2286	2254	2225	2386	2333	2158	2123	2106	2081	2093	1960
Cadangan Sekelika		55	56	86	98	89	21	94	94	98	107	59	21	21	27	28	25	27	82	76	71	62	72	62	74
Cadangan P u l a r		1351	1351	1351	1301	1279	1335	1302	1119	1430	1095	777	150	133	222	247	323	323	448	445	810	870	771	727	913

SUB SISTEM REGION_1

RENCANA : HARI/TANGGAL : MINGGU , 31 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

		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	MKRNG1OC-CT 1		95	95	95	95	95	95	95	80	80	95	95	95	95	80	95	95	95	95	95	95	95	95	95	95	95
	MKRNG1OC1		95	95	95	95	95	95	95	80	80	95	95	95	95	80	95	95	95	95	95	95	95	95	95	95	95
	MKRNG2OC1		65	65	65	65	65	65	65	65	65	65	65	55	55	65	65	65	65	65	65	65	65	65	65	65	65
	MKRNG2OC		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
	MKRNG3OC		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
	MKRNG1CC		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
	MKRNG2CC		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
	MKRNG3CC		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	#1	80	80	80	50	50	50	50	50	50	80	80	80	50	80	80	80	80	80	80	80	80	80	80	80	80
		#2	80	80	80	50	50	50	50	50	50	80	80	80	50	80	80	80	80	80	80	80	80	80	80	80	80
		#3	80	80	80	50	50	50	50	50	50	80	80	80	50	80	80	80	80	80	80	80	80	80	80	80	80
PLTU	MKRNG	#4	50	50	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
		#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
PLTGU	MTWAR1OC-CT 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
	MTWAR1OC1		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
	MTWAR2OC1		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
	MTWAR12OC		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
	MTWAR13OC		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
	MTWAR13CC		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
	MTWAR12CC		0	0	0	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	330	330	330	320	300
	MTWAR13CC		420	420	350	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR	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
	MTWAR	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
			PEMBANGKITAN MUARA TAWAR																								
PLTU	MTWAR	#3.1	50	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		#3.2	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
		#3.3	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
PLTG	MTWAR	#4.1	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
		#4.2	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
		#4.3	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
			PEMBELIAN DARI LUAR PLN																								
PLTP	DIELAK	#4-#6	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183
	CIKARANG		75	75	75	75	75	75	75	75	75	75	75	75	50	50	50	50	50	50	50	50	50	50	50	50	50
PLTU	Krakatau 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
(*) Pembangkit Area-1			4805	4665	4675	4485	4495	4435	4425	4465	4465	4565	4605	4575	4505	4595	4633	4895	4993	4093	4263	4203	4263	4243	4223	4183
(**) Beban Area-1			4510	4585	4480	4422	4439	4284	4327	4307	4204	4315	4572	4405	4223	4106	4044	4011	4153	4281	4357	4325	4388	4159	4358	4326	4247
Selisih (*) - (**)			295	80	95	73	106	111	188	158	181	250	33	-72	-130	-103	11	18	-139	-174	-154	-125	-125	-126	-115	-143	-56
Cadangan Seketika			83	78	69	85	69	85	65	73	59	69	79	160	170	174	189	164	164	169	175	175	178	177	177	176	174
Cadangan Putar			590	580	540	445	445	445	445	415	555	355	335	607	807	912	892	922	922	822	742	712	652	677	672	632	722

SUB SISTEM REGION_1

RENCANA : HARI/TANGGAL : MINGGU, 31 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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	Rata-2
PLTGU	MKRNG1OC - Gt	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	94
	MKRNG1OC1	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	95	94
	MKRNG2OC1	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65	65
	MKRNG2OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG3OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG1CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG2CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MKRNG3CC	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	GRNG #1	80	80	80	80	80	80	80	60	60	80	80	80	80	80	80	80	80	80	80	80	60	60	60	71
	#2	80	80	80	80	80	80	80	60	60	80	80	80	80	80	80	80	80	80	80	80	60	60	60	70
	#3	80	80	80	80	80	80	80	60	60	80	80	80	80	80	80	80	80	80	80	80	60	60	60	71
PLTU	MKRNG #4	100	100	100	100	100	100	100	100	100	100	120	120	120	120	120	120	120	120	120	120	120	100	100	103
	#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
PLTGU	MTWAR1OC-G1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR1OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR2OC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR2OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR1OC	0	0	0	0	0	0	72	72	72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
	MTWAR1CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MTWAR2CC	425	425	325	325	325	250	250	250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	196
	MTWAR3CC	0	0	0	0	0	0	0	0	0	350	550	550	550	550	550	550	550	550	550	550	400	350	350	163
	MTWAR 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	2
	MTWAR 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
PLTG	MTWAR #3.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	#3.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	#3.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PLTG	MTWAR #4.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	#4.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	#4.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	PEMBELIAN DARI LUAR PLN																								
PLTP	GSLAK #4-#8	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183	183
PLTU	C.KARANG	50	50	50	50	50	50	50	50	75	120	120	120	120	120	120	120	120	120	120	120	120	100	100	76
	Krakatau 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
(*) Pembangkitan Area-1		4164	4152	4152	4237	4235	4215	4216	4156	4174	4727	5061	5109	5104	5104	5104	5159	5123	5070	4975	4844	4706	4205	4805	4781
(**) Beban Area-1		4241	4236	4151	4232	4195	4363	4208	4317	4500	4716	5156	5274	5239	5319	5307	5307	5263	5310	5002	4964	4729	4576	4553	4927
Selisih (*) - (**)		-76	-83	001	-95	223	-38	-31	-158	-326	-339	-152	-165	-140	-130	-148	-208	-152	-31	-87	-124	-34	-29	-52	-146
Cadangan Seketika		193	178	178	178	177	177	175	172	172	180	37	37	37	12	12	12	37	35	38	82	79	71	69	115
Cadangan [illegible]		742	657	657	657	677	677	848	898	898	792	785	80	30	30	30	30	80	110	270	345	420	540	540	546

[illegible]

SUB SISTEM REGION_4

RENCANA : HARI/TANGGAL : MINGGU, 31 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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
PLTA	SUTAMI	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25
PLTA	BRANTAS	11	11	11	11	11	11	11	11	11	11	11	11	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
PLTU	PITON #1	360	360	360	360	360	360	360	360	360	360	360	360	350	296	250	225	225	225	225	225	225	225	225	225	225
	PITON #2	360	360	360	360	360	360	360	360	360	360	360	360	350	296	250	225	225	225	225	225	225	225	225	225	225
PLTGU 1	GRSIK11OC GT 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
	GRSIK12OC1	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
	GRSIK13OC1	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
	GRSIK11OC	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
	GRSIK12OC	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
	GRSIK13OC	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
	GRSIK11CC	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
	GRSIK12CC	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
	GRSIK13CC	300	300	250	250	250	250	250	250	250	300	325	325	250	250	250	300	300	300	300	300	300	300	300	250	250
PLTGU 2	GRSIK21OC GT 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
	GRSIK10C2	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
	[illegible]	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
	GRSIK22OC	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
	GRSIK23OC	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
	GRSIK21CC	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
	GRSIK22CC	164	164	164	164	164	164	164	164	164	164	164	164	164	161	164	164	164	164	[illegible]	164	164	164	164	164	164
	GRSIK23CC	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 3	GRSIK31OC GT 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
	GRSIK10C3	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
	[illegible]	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
	GRSIK32OC	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
	GRSIK33OC	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
	GRSIK31CC	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
	GRSIK32CC	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
	GRSIK33CC	300	300	250	250	250	250	250	250	250	250	325	300	250	250	250	250	300	300	300	300	250	250	250	250	250
PLTU	GRSIK #1	50	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	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
PLTU	GRSIK #3	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	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
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
	#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
	GRESIK #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
	#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
(*) Pembangkitan Area 4		4502	4502	4442	4392	4342	4291	4342	4342	4342	4392	4392	4212	3958	3584	3346	3248	3318	3368	3468	3456	3414	3414	3416	3358	3358
(**) Beban Area 4		2323	2246	2268	2247	2254	2253	2334	2296	2267	2239	2263	2016	1931	1864	1846	1825	1924	1973	2016	2013	2011	2018	2039	1979	1969
Selisih (*) - (**)		2179	2255	2174	2145	2088	2024	2008	2047	2115	2150	2129	2201	1937	1897	1428	1342	1395	1396	1453	1458	1405	1400	1384	1380	1389
Cadangan Setotika		84	84	87	87	52	52	52	52	52	54	59	58	87	52	77	77	79	79	79	79	77	77	77	74	74
Cadangan Putar		624	624	574	574	724	724	724	724	724	674	674	648	1134	1382	1854	1854	1801	1554	1454	1454	1504	1504	1504	1554	1554

SUB SISTEM REGION_4

RENCANA : HARI/TANGGAL : MINGGU, 31 JULI 2005

PT PLN PEMBANGKITAN TENAGA LISTRIK JAWA-BALI

	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	Rata-2
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
PLTA	SUTAMI	25	25	25	25	25	25	25	50	50	90	90	90	90	90	90	90	90	50	50	25	25	25	25	38
PLTA	BRANTAS	7	7	7		7	14	14	14	14	22	22	22	22	22	22	19	19	18	18	14	14	14	14	12
PLTU	PITON #1	225	225	225	225	225	225	225	250	296	305	350	370	370	370	370	370	370	360	360	350	360	350	350	304
	PITON #2	225	225	225	225	225	225	225	250	296	305	350	370	370	370	370	370	370	360	360	350	360	350	350	304
PLTGU 1	GRSIK11OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK10C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK12OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK13OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK11CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK12CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK13CC	250	250	250	250	250	250	250	250	250	300	375	425	425	425	420	415	400	400	350	350	325	300	300	300
PLTGU 2	GRSIK21OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK10C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK22OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK23OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK21CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK22CC	164	164	164	164	164	164	164	164	164	164	185	225	250	250	225	225	164	164	164	164	164	164	164	172
	GRSIK23CC	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 3	GRSIK31OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK10C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK20C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK32OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK33OC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK31CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK32CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	GRSIK33CC	250	250	250	250	250	250	250	250	250	300	375	400	400	400	400	400	400	400	325	325	300	300	300	291
PLTU	GRSIK #1	50	50	50	50	50	50	50	50	50	50	80	80	80	80	80	80	80	80	50	50	50	50	50	55
	#2	50	50	50	50	50	50	50	50	50	50	80	80	80	80	80	80	80	80	50	50	50	50	50	55
PLTU	GRSIK #3	90	90	90	90	90	90	90	90	90	135	167	167	167	167	167	167	167	167	150	140	90	90	90	106
	#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
PLTG	GLTMR #1	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	16	16	16	0	0	0	4
	#2	0	0	0	0	0	0	0	0	16	16	16	16	15	16	16	16	16	16	16	16	0	0	0	4
	GRESIK #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
	#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
(*) Pembangkitan Area-4		2082	2079	2078	2050	2068	3470	3470	3555	4056	4497	5155	5385	5475	5410	5380	5372	5276	5145	4911	4887	4640	4503	4505	4155
(**) Beban Area-4		1950	2005	2030	1954	1965	3111	3108	3114	3329	3601	3121	3183	3211	3190	3163	3147	3019	2981	2704	2830	2502	2432	2405	2320
Selisih (*) - (**)		1419	1363	1358	1435	1404	1319	1362	1531	1727	1898	2034	2223	2224	2221	2217	2225	2257	2284	2727	2229	2139	2103	2127	1835
Cadangan Sekatika		74	74	74	74	74	78	78	81	50	99	111	44	26	48	44	44	60	57	67	74	68	97	57	68
Cadangan Putar		1554	1554	1554	1554	1554	1813	1672	1553	1202	1115	457	227	177	202	232	237	333	423	825	836	729	684	884	913

```
unit uMenu;

interface

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

type
  TfrmMenu = class(TForm)
    Panel1: TPanel;
    btnNew: TButton;
    btnOpen: TButton;
    btnExit: TButton;
    StatusBar1: TStatusBar;
    Panel2: TPanel;
    OpenDialog1: TOpenDialog;
    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);
begin
  frmInput.Caption:='Input Data';
  frmInput.btnNext.Caption:='&Save';
  frmInput.Show;
end;

procedure TfrmMenu.btnOpenClick(Sender: TObject);
var NamaFile,Nama:string;
```

```
    output:TextFile;
    Pmin,Pmax,a2,a1,a0,Sh,Sc,Ramp,Load,Res:double;
    i,j,Ngen,Njam,Tup,Tdown,Tcold,InitSt:integer;
    aLoad,aRes:dArr1;
    aPLN:dArr2;
    aGen:TGenArr;
begin
  try
    if OpenDialog1.Execute then
    begin
      NamaFile:=OpenDialog1.FileName;
      AssignFile(output,NamaFile);
      Reset(output);
      Readln(output,Ngen);
      Readln(output,Njam);
      frmInput.edtNGen.Text:=IntToStr(Ngen);
      frmInput.edtNjam.Text:=IntToStr(Njam);
      frmInput.fgGen.RowCount:=Ngen+1;
      SetLength(aGen,Ngen+1);
      for i:=1 to Ngen do
      begin
        Readln(output,Pmax,Pmin,a0,a1,a2,Tup,Tdown,Sh,Sc,tcold,InitSt,
        Ramp,Nama);
        aGen[i]:=TPembangkit.Create(Nama,Pmin,Pmax,a2,a1,a0,Sh,Sc,Ramp,
              Tup,Tdown,Tcold,InitSt);
        frmInput.fgGen.Cells[1,i]:=Nama;
        frmInput.fgGen.Cells[2,i]:=FloatToStr(Pmax);
        frmInput.fgGen.Cells[3,i]:=FloatToStr(Pmin);
        frmInput.fgGen.Cells[4,i]:=FloatToStr(a0);
        frmInput.fgGen.Cells[5,i]:=FloatToStr(a1);
        frmInput.fgGen.Cells[6,i]:=FloatToStr(a2);
        frmInput.fgGen.Cells[7,i]:=IntToStr(Tup);
        frmInput.fgGen.Cells[8,i]:=IntToStr(Tdown);
        frmInput.fgGen.Cells[9,i]:=FloatToStr(Sh);
        frmInput.fgGen.Cells[10,i]:=FloatToStr(Sc);
        frmInput.fgGen.Cells[11,i]:=IntToStr(Tcold);
        frmInput.fgGen.Cells[12,i]:=IntToStr(InitSt);
        frmInput.fgGen.Cells[13,i]:=FloatToStr(Ramp);
      end;
      frmInput.fgLoad.RowCount:=Njam+1;
      SetLength(aLoad,Njam+1);
      SetLength(aRes,Njam+1);
      for i:=1 to Njam do
      begin
        Readln(output,Load,Res);
        aLoad[i]:=Load;
        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;
       ReadIn(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;


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;


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.
```

```
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;
  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;

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.FImag)/denot
e),
      ((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.toString(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;


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;


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;


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;

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[0]) 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;

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;

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;
 for i:=0 to Nmat-1 do
 begin
  tmp[i,Nmat]:=mat2[i];
 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;
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;
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.
```

```
unit uGenerator;

interface

type
 TPembangkit=class
 private
  FNama:string;
  FPmax,FPmin,Fa2,Fa1,Fa0,FSh,FSc,FDaya,FRamp:double;
  FTup,FTdown,FTcold,FInitSt:integer;
  function GetAFLC:double;
 public
  constructor Create;overload;
  constructor Create(const rNama:string;
          const rPmin,rPmax,ra2,ra1,ra0,rSh,rSc,rRamp:double;
          const rTup,rTdown,rTcold,rInitSt:integer);overload;
  constructor Create(const rPembangkit:TPembangkit);overload;
  procedure Assign(const rPembangkit:TPembangkit);
  function GetBiaya(const rDaya:double):double;
  function GetDaya(const rLamda:double):double;
  function GetLamda(const rDaya:double):double;
  property Nama:string read FNama write FNama;
  property Pmax:double read FPmax write FPmax;
  property Pmin:double read FPmin write FPmin;
  property a2:double read Fa2 write Fa2;
  property a1:double read Fa1 write Fa1;
  property a0:double read Fa0 write Fa0;
  property Sh:double read FSh write FSh;
  property Sc:double read FSc write FSc;
  property Ramp:double read FRamp write FRamp;
  property Tup:integer read FTup write FTup;
  property Tdown:integer read FTdown write FTdown;
  property Tcold:integer read FTcold write FTcold;
  property InitSt:integer read FInitSt write FInitSt;
  property Daya:double read FDaya write FDaya;
  property AFLC:double read GetAFLC;
 end;

 TGenArr=array of TPembangkit;

implementation

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

```
end;

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

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

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

procedure TPembangkit.Assign(const rPembangkit:TPembangkit);
```

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

//data operation

function TPembangkit.GetBiaya(const rDaya:double):double;
begin
  result:=0;
  if rDaya<>0 then
  begin
    result:=Fa2*sqr(rDaya)+Fa1*rDaya+Fa0;
  end;
end;

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

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

end.
```

```
unit uHasil;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, TeEngine, Series, TeeProcs, Chart, Grids, ComCtrls,
  StdCtrls;

type
  TfrmHasil = class(TForm)
    TabSheet5: TTabSheet;
    TabSheet6: TTabSheet;
    TabSheet7: TTabSheet;
    Panel1: TPanel;
    btnClose: TButton;
    btnHitung: TButton;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Chart1: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    TabSheet4: TPageControl;
    fgCostPerJam: TStringGrid;
    Label8: TLabel;
    edtTotalProgram: TEdit;
    Label9: TLabel;
    edtTotalPLN: TEdit;
    Label1: TLabel;
    edtSelisih: TEdit;
    fgDaya: TStringGrid;
    fgStatus: TStringGrid;
    Label3: TLabel;
    edtTime: TEdit;
    Label4: TLabel;
    pbIterasi: TProgressBar;
    GroupBox1: TGroupBox;
    Label2: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label10: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label12: TLabel;
    Label15: TLabel;
```

```
    edtMaxIterasi: TEdit;
    edtNAnt: TEdit;
    edtD: TEdit;
    edtP: TEdit;
    edtAlpha: TEdit;
    edtBetha: TEdit;
    edtQ0: TEdit;
    edtNMovc: TEdit:
    edtInitPhe: TEdit;
    btnUseDefault: TButton;
    edtKa: TEdit;
    lblKa: TLabel;
    procedure btnCloseClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnHitungClick(Sender: TObject);
    procedure btnUseDefaultClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmHasil: TfrmHasil;

implementation

uses uObjFunc, uUtils, uAntColony, uDataAnt;

{$R *.dfm}

procedure TfrmHasil.btnCloseClick(Sender: TObject);
begin
  Close;
end;

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
  fgCostPerJam.Cells[0,0]:='Jam';
  fgCostPerJam.Cells[1,0]:='Biaya Program';
  fgCostPerJam.Cells[2,0]:='Biaya PLN';
  fgCostPerJam.Cells[3,0]:='Selisih Biaya';
end;

procedure TfrmHasil.btnHitungClick(Sender: TObject);
var i,j:integer;
```

```
    CostTotal,CostPLN:double;
    chrom:bArr2;
    PL:dArr2;
    CostPerJam,CostPerJamPLN:dArr1;
    jam,menit,detik,mdetik:word;
    mulai,selesai,selang:TDateTime;
    ant:TAntColony;
    MaxIterasi,Nmove,Nparam,Nant:integer;
    q0,Alpha,Betha,d,p,InitPhe,Ka:double;
    InitState:iArr1;
begin
  mulai:=time;
  MaxIterasi:=StrToInt(edtMaxIterasi.Text);
  pbIterasi.Max:=MaxIterasi;
  Nmove:=gObjFunc.Njam;
  Nparam:=gObjFunc.Ngen;
  Nant:=StrToInt(edtNant.Text);
  q0:=StrToFloat(edtq0.Text);
  Alpha:=StrToFloat(edtAlpha.Text);
  Betha:=StrToFloat(edtBetha.Text);
  d:=StrToFloat(edtD.Text);
  p:=StrToFloat(edtP.Text);
  InitPhe:=StrToFloat(edtInitPhe.Text);
  Ka:=StrToFloat(edtKa.Text);
  InitState:=gObjFunc.InitState;
  ant:=TAntColony.Create(MaxIterasi,Nmove,Nparam,Nant,q0,Alpha,
     Betha,d,p,InitPhe,Ka,InitState);
  ant.doHitung(chrom);
  ant.Free;
  //gObjFunc.doExecute(chrom,PL,CostPerJam,CostTotal);
  selesai:=time;
  selang:=selesai-mulai;
  DecodeTime(selang,jam,menit,detik,mdetik);
  edtTime.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
          IntToStr(detik)+':'+IntToStr(mdetik);
  gObjFunc.doHitungChrom(chrom,PL,CostPerJam,CostTotal);
  for i:=1 to high(chrom) do
  begin
    for j:=1 to high(chrom[0]) do
    begin
      fgDaya.Cells[j,i]:=RealToStr(PL[i,j],0);
      if chrom[i,j]=true then
      begin
        fgStatus.Cells[j,i]:='1';
      end
      else
```

```
    begin
      fgStatus.Cells[j,i]:='0';
    end;
  end;
end;
edtTotalProgram.Text:=FormatFloat('#,##0',CostTotal);
gObjFunc.doHitungPLN(CostPerJamPLN,CostPLN);
Series1.Clear;
Series2.Clear;
for i:=1 to high(CostPerJam) do
begin
  fgCostPerJam.Cells[0,i]:=IntToStr(i);
  fgCostPerJam.Cells[1,i]:=FormatFloat('#,#00',CostPerJam[i]);
  fgCostPerJam.Cells[2,i]:=FormatFloat('#,#00',CostPerJamPLN[i]);
  fgCostPerJam.Cells[3,i]:=FormatFloat('#,#00',
                  (CostPerJamPLN[i]-CostPerJam[i]));
  Series1.Add(CostPerJam[i],IntToStr(i));
  Series2.Add(CostPerJamPLN[i],IntToStr(i));
end;
edtTotalPLN.Text:=FormatFloat('#,##0',CostPLN);
edtSelisih.Text:=FormatFloat('#,##0',(CostPLN-CostTotal));
end;

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
 edtMaxIterasi.Text:='500';
 edtNAnt.Text:='100';
 edtD.Text:='0.9';
 edtP.Text:='0.9';
 edtAlpha.Text:='1';
 edtBetha.Text:='4';
 edtQ0.Text:='0.9';
 edtInitPhe.Text:='1';
 edtKa.Text:='1000000000';
 btnHitung.Enabled:=true;
end;

end.
```

```
unit uInputGen;

interface

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

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

var
  frmInput: TfrmInput;

implementation

uses uObjFunc, uHasil;

{$R *.dfm}
```

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

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

procedure TfrmInput.edtNGenChange(Sender: TObject);
var i:integer;
begin
  if edtNgen.Text='' then
  begin
    fgGen.RowCount:=2;
    fgPLN.RowCount:=2;
  end
  else
  begin
    fgGen.RowCount:=StrToInt(edtNgen.Text)+1;
    fgPLN.RowCount:=StrToInt(edtNgen.Text)+1;
    for i:=1 to StrToInt(edtNgen.Text) do
    begin
      fgGen.Cells[0,i]:=IntToStr(i);
      fgPLN.Cells[0,i]:='Gen '+IntToStr(i);
    end;
  end;
end;
```

```
procedure TfrmInput.edtNjamChange(Sender: TObject);
var i:integer;
begin
 if edtNjam.Text='' then
 begin
  fgLoad.RowCount:=2;
  fgPLN.ColCount:=2;
 end
 else
 begin
  fgLoad.RowCount:=StrToInt(edtNjam.Text)+1;
  fgPLN.ColCount:=StrToInt(edtNjam.Text)+1;
  for i:=1 to StrToInt(edtNjam.Text) do
  begin
   fgLoad.Cells[0,i]:=IntToStr(i);
   fgPLN.Cells[i,0]:='Jam '+IntToStr(i);
  end;
 end;
end;

procedure TfrmInput.btnNextClick(Sender: TObject);
var input:TextFile;
  NamaFile,Nama:string;
  Pmin,Pmax,a2,a1,a0,Sh,Sc,Ramp,Load,Res:double;
  i,j,Tup,Tdown,Tcold,InitSt,Ngen,Njam:integer;
begin
 if btnNext.Caption='&Save' then
 begin
  if SaveDialog1.Execute then
  begin
   NamaFile:=SaveDialog1.FileName;
   AssignFile(input,NamaFile+'.txt');
   Rewrite(input);
   Ngen:=StrToInt(edtNgen.Text);
   Njam:=StrToInt(edtNjam.Text);
   Writeln(input,Ngen);
   Writeln(input,Njam);
   for i:=1 to Ngen do
   begin
    Nama:=fgGen.Cells[1,i];
    Pmax:=StrToFloat(fgGen.Cells[2,i]);
    Pmin:=StrToFloat(fgGen.Cells[3,i]);
    a0:=StrToFloat(fgGen.Cells[4,i]);
    a1:=StrToFloat(fgGen.Cells[5,i]);
    a2:=StrToFloat(fgGen.Cells[6,i]);
    Tup:=StrToInt(fgGen.Cells[7,i]);
```

```
    Tdown:=StrToInt(fgGen.Cells[8,i]);
    Sh:=StrToFloat(fgGen.Cells[9,i]);
    Sc:=StrToFloat(fgGen.Cells[10,i]);
    Tcold:=StrToInt(fgGen.Cells[11,i]);
    InitSt:=StrToInt(fgGen.Cells[12,i]);
    Ramp:=StrToFloat(fgGen.Cells[13,i]);
    Writeln(input,Pmax:7:0,' ',Pmin:7:0,' ',
    a0:9:4,' ',a1:9:4,' ',a2:9:5,' ',Tup,' ',Tdown,' ',
    Sh:7:0,' ',Sc:7:0,' ',tcold,' ',InitSt,' ',Ramp:7:0,' ',Nama);
   end;
   for i:=1 to Njam do
   begin
    Load:=StrToFloat(fgLoad.Cells[1,i]);
    Res:=StrToFloat(fgLoad.Cells[2,i]);
    Writeln(input,Load,Res);
   end;
   fgPLN.RowCount:=Ngen+1;
   fgPLN.ColCount:=Njam+1;
   for i:=1 to Ngen do
   begin
    for j:=1 to Njam do
    begin
     Load:=StrToFloat(fgPLN.Cells[j,i]);
     Write(input,Load:7:2,' ');
    end;
    Writeln(input,'');
   end;
   CloseFile(input);
  end;
end
else if btnNext.Caption='&Next' then
begin
 frmHasil.fgStatus.RowCount:=gObjFunc.Ngen+1;
 frmHasil.fgStatus.ColCount:=gObjFunc.Njam+1;
 frmHasil.fgDaya.RowCount:=gObjFunc.Ngen+1;
 frmHasil.fgDaya.ColCount:=gObjFunc.Njam+1;
 frmHasil.fgCostPerJam.RowCount:=gObjFunc.Njam+1;
 frmHasil.edtNMove.Text:=IntToStr(gObjFunc.Njam);
 for i:=1 to gObjFunc.Ngen do
 begin
  frmHasil.fgStatus.Cells[0,i]:='Unit '+IntToStr(i);
  frmHasil.fgDaya.Cells[0,i]:='Unit '+IntToStr(i);
 end;
 for i:=1 to gObjFunc.Njam do
 begin
  frmHasil.fgStatus.Cells[i,0]:='Jam '+IntToStr(i);
```

```
    frmHasil.fgDaya.Cells[i,0]:='Jam '+IntToStr(i);
   end;
   frmHasil.Show;
  end;
end;

end.

unit uObjFunc;

interface

uses uUtils,uGenerator,SysUtils,uDataAnt;

type
 TObjFunc=class
 private
   FNgen,FNjam:integer;
   FBeban,FRes,FAFLC:dArr1;
   FPLN:dArr2;
   FGen:TGenArr;
   function GetBeban:dArr1;
   function GetRes:dArr1;
   function GetPLN:dArr2;
   function GetGen:TGenArr;
   procedure SetGen(const rGen:TGenArr);
   procedure SetBeban(const rBeban:dArr1);
   procedure SetRes(const rRes:dArr1);
   procedure SetPLN(const rPLN:dArr2);
   function GetInitSt:iArr1;
   function isON(const rFlip:double):boolean;
   function isServe(const rJam:integer;const rChrom:bArr1):boolean;
   function isRampRate(const rJam:integer;const rPL:dArr2):boolean;
   function CalcAFLC(const rXs:integer;
         const rNo:integer):double;
   procedure RepairAFLC(var rChrom:bArr2);
   procedure UpdateXs(const rChrom:bArr1;
         var rXs:iArr1);
   function CreateChromBase:bArr2;
   function CreateChromONOFF:bArr2;
   function HitungEcoDis(const rJam:integer;
         const rChrom1:bArr1):dArr1;
   function GetSortAFLC1:iArr1;
   function GetSortAFLC(const rXs:iArr1):iArr1;
   function GetSortChrom(const rXs:iArr1;
         const rRank:integer):bArr1;
```

```
  function HitungCostGen(const rPL:dArr2):dArr2;
  function HitungCostSUC(const rPL:dArr2):dArr2;
  procedure GetSwap(var rChrom:bArr2);
  function doCariGreyZone(const rChrom:bArr2):bArr2;
  procedure UpdateChrom(var rChrom:bArr2);
 public
  constructor Create;overload;
  constructor Create(const rBeban,rRes:dArr1;
          const rPLN:dArr2;
          const rGen:TGenArr);overload;
  function getRandomChrom(const rFlip:double):bArr2;
  function getChromFromAnother(const rChrom:bArr2;
        const rPflip:double):bArr2;
  function getConstructSolution(const rFlip:double):bArr2;
  procedure setLocalSearch(var rChrom:bArr2);
  procedure doHitungChrom(var rChrom:bArr2;
        var rCostTotal:double);overload;
  procedure doHitungChrom(var rChrom:bArr2;
        var rPL:dArr2;
        var rCostPerJam:dArr1;
        var rCostTotal:double);overload;
  procedure doHitungPLN(
        var rCostPerJam:dArr1;
        var rCostTotal:double);
  procedure doExecute(var rChrom:bArr2;
        var rPL:dArr2;
        var rCostPerJam:dArr1;
        var rCostTotal:double);
  procedure doCalcState(const rJam:integer;
        var rState1,rState2:TState);
  destructor Destroy;override;
  property Ngen:integer read FNgen write FNgen;
  property Njam:integer read FNjam write FNjam;
  property Gen:TGenArr read GetGen write SetGen;
  property Beban:dArr1 read GetBeban write SetBeban;
  property PLN:dArr2 read GetPLN write SetPLN;
  property Res:dArr1 read GetRes write SetRes;
  property SortAFLC:iArr1 read GetSortAFLC1;
  property InitState:iArr1 read GetInitSt;
 end;

var gObjFunc:TObjFunc;

implementation

//constructor
```

```
constructor TObjFunc.Create;
begin
 inherited Create;
 FNgen:=0;
 FNjam:=0;
end;

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

//data accessing
function TObjFunc.GetInitSt:iArr1;
var i:integer;
begin
 SetLength(result,FNgen+1);
```

```
 for i:=1 to FNgen do
 begin
  result[i]:=FGen[i].InitSt;
 end;
end;

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

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

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

function TObjFunc.GetGen:TGenArr;
var i:integer;
begin
 SetLength(result,FNgen+1);
 for i:=1 to FNgen do
 begin
  result[i]:=TPembangkit.Create(FGen[i]);
```

```
  end;
end;

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

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

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

```
  for i:=1 to FNjam do
  begin
    FRes[i]:=rRes[i];
  end;
end;

procedure TObjFunc.SetPLN(const rPLN:dArr2);
var i,j,Ncek:integer;
begin
  if FNgen>0 then
  begin
    Ncek:=high(rPLN);
    if FNgen<>Ncek then raise Exception.Create('Dimensi matrik tidak sama!');
  end
  else
  begin
    FNgen:=high(rPLN);
  end;
  if FNjam>0 then
  begin
    Ncek:=high(rPLN[0]);
    if FNjam<>Ncek then raise Exception.Create('Dimensi matrik tidak sama!');
  end
  else
  begin
    FNjam:=high(rPLN[0]);
  end;
  SetLength(FPLN,FNgen+1,FNjam+1);
  for i:=1 to FNgen do
  begin
    for j:=1 to FNjam do
    begin
      FPLN[i,j]:=rPLN[i,j];
    end;
  end;
end;

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

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

```
  load,sBebanMin,sBebanMax:double;
begin
 result:=true;
 sBebanMin:=0;
 sBebanMax:=0;
 for i:=1 to FNgen do
 begin
  if rChrom[i]=true then
  begin
   sBebanMin:=sBebanMin+FGen[i].Pmin;
   sBebanMax:=sBebanMax+FGen[i].Pmax;
  end;
 end;
 load:=FBeban[rJam]+FRes[rJam];
 if load<sBebanMin then result:=false;
 if load>sBebanMax then result:=false;
end;

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

function TObjFunc.GetSortAFLC1:iArr1;
var i,j,tmp:integer;
  tmpAFLC:double;
begin
 SetLength(result,FNgen-1);
 for i:=1 to FNgen do
```

```
  begin
    result[i]:=i;
  end;
  for i:=1 to fNgen-1 do
  begin
    for j:=i to fNgen do
    begin
      if FAFLC[i]>FAFLC[j] then
      begin
        tmpAFLC:=FAFLC[i];
        FAFLC[i]:=FAFLC[j];
        FAFLC[j]:=tmpAFLC;
        tmp:=result[i];
        result[i]:=result[j];
        result[j]:=tmp;
      end;
    end;
  end;
  for i:=1 to fNgen do
  begin
    fAFLC[i]:=fGen[i].AFLC;
  end;
end;

function TObjFunc.CalcAFLC(const rXs:integer;
        const rNo:integer):double;
var SUC:double;
begin
  SUC:=0;
  if rXs<0 then
  begin
    if abs(rXs)>=FGen[rNo].Tdown then
    begin
      if abs(rXs)>(FGen[rNo].Tdown+FGen[rNo].Tcold) then
      begin
        SUC:=FGen[rNo].Sc;
      end
      else
      begin
        SUC:=FGen[rNo].Sh;
      end;
    end;
  end;
  result:=(FGen[rNo].GetBiaya(FGen[rNo].Pmax)+SUC)/FGen[rNo].Pmax;
end;
```

```
function TObjFunc.GetSortAFLC(const rXs:iArr1):iArr1;
var i,j:integer;
   AFLC:dArr1;
begin
 SetLength(result,FNgen+1);
 SetLength(AFLC,FNgen+1);
 for i:=1 to FNgen do
 begin
  result[i]:=i;
  AFLC[i]:=CalcAFLC(rXs[i],i);
 end;
 for i:=1 to FNgen-1 do
 begin
  for j:=i to FNgen do
  begin
   if AFLC[i]>AFLC[j] then
   begin
    Swap(AFLC[i],AFLC[j]);
    Swap(result[i],result[j]);
   end;
  end;
 end;
end;

function TObjFunc.GetSortChrom(const rXs:iArr1;
      const rRank:integer):bArr1;
var i:integer;
   SortAFLC:iArr1;
begin
 SetLength(result,FNgen+1);
 for i:=1 to FNgen do
 begin
  result[i]:=false;
 end;
 SortAFLC:=GetSortAFLC(rXs);
 for i:=1 to rRank do
 begin
  result[SortAFLC[i]]:=true;
 end;
end;

procedure TObjFunc.UpdateXs(const rChrom:bArr1;
       var rXs:iArr1);
var i:integer;
begin
 for i:=1 to FNgen do
```

```
 begin
  if rChrom[i]=true then
  begin
   if rXs[i]<0 then
   begin
    rXs[i]:=1;
   end
   else
   begin
    rXs[i]:=rXs[i]+1;
   end;
  end
  else
  begin
   if rXs[i]<0 then
   begin
    rXs[i]:=rXs[i]-1;
   end
   else
   begin
    rXs[i]:=-1;
   end;
  end;
 end;
end;

function TObjFunc.CreateChromBase:bArr2;
var i,j,k:integer;
   chrom1:bArr1;
   Xs:iArr1;
   serve:boolean;
begin
 SetLength(result,FNgen+1,FNjam+1);
 SetLength(chrom1,FNgen+1);
 SetLength(Xs,FNGen+1);
 for i:=1 to FNgen do
 begin
  Xs[i]:=FGen[i].InitSt;
 end;
 for i:=1 to FNjam do
 begin
  for j:=1 to FNgen do
  begin
   chrom1:=GetSortChrom(Xs,j);
   serve:=isServe(i,chrom1);
   if serve=true then
```

```
      begin
       for k:=1 to FNgen do
       begin
        result[k,i]:=chrom1[k];
       end;
       UpdateXs(chrom1,Xs);
       break;
      end;
     end;
    end;
  end;

function TObjFunc.CreateChromONOFF:bArr2;
var i,j,k:integer;
    chrom1:bArr1;
    Xs:iArr1;
    serve:boolean;
begin
  SetLength(result,FNgen+1,FNjam+1);
  SetLength(chrom1,FNgen+1);
  SetLength(Xs,FNgen+1);
  for i:=1 to FNgen do
  begin
    Xs[i]:=FGen[i].InitSt;
  end;
  for i:=1 to FNjam do
  begin
    for j:=FNgen downto 1 do
    begin
      chrom1:=GetSortChrom(Xs,j);
      serve:=isServe(i,chrom1);
      if serve=true then
      begin
        for k:=1 to FNgen do
        begin
          result[k,i]:=chrom1[k];
        end;
        UpdateXs(chrom1,Xs);
        break;
      end;
    end;
  end;
end;

procedure TObjFunc.RepairAFLC(var rChrom:bArr2);
var i,j,NUnitON:integer;
```

```
    Xs,SortAFLC:iArr1;
    Chrom1:bArr1;
begin
  SetLength(Xs,FNgen+1);
  SetLength(Chrom1,FNgen+1);
  for i:=1 to FNgen do
  begin
    Xs[i]:=FGen[i].InitSt;
  end;
  for i:=1 to FNjam do
  begin
    NUnitON:=0;
    for j:=1 to FNgen do
    begin
      if rChrom[j,i]=true then
      begin
        inc(NUnitON);
      end;
    end;
    for j:=1 to FNgen do
    begin
      rChrom[j,i]:=false;
      Chrom1[j]:=false;
    end;
    SortAFLC:=GetSortAFLC(Xs);
    for j:=1 to NUnitON do
    begin
      rChrom[SortAFLC[j],i]:=true;
      Chrom1[SortAFLC[j]]:=true;
    end;
    UpdateXs(Chrom1,Xs);
  end;
end;

function TObjFunc.getRandomChrom(const rFlip:double):bArr2;
var i,j:integer;
    chromBase,chromONOFF:bArr2;
begin
  chromBase:=CreateChromBase;
  chromONOFF:=CreateChromONOFF;
  SetLength(result,Ngen+1,Njam+1);
  for i:=1 to Ngen do
  begin
    for j:=1 to Njam do
    begin
      if chromBase[i,j]=true then
```

```
      begin
       result[i,j]:=true;
      end
      else
      begin
       if chromONOFF[i,j]=true then
       begin
        if isON(rFlip)=true then
        begin
         result[i,j]:=true;
        end
        else
        begin
         result[i,j]:=false;
        end;
       end;
      end;
     end;
    end;
    RepairAFLC(result);
    GetSwap(result);
  end;

  function TObjFunc.getChromFromAnother(const rChrom:bArr2;
        const rPflip:double):bArr2;
  var i,j:integer;
  begin
    SetLength(result,FNgen+1,FNjam+1);
    for i:=1 to FNgen do
    begin
     for j:=1 to FNjam do
     begin
      result[i,j]:=rChrom[i,j];
     end;
    end;
    for i:=1 to FNgen do
    begin
     for j:=1 to FNjam do
     begin
      if isON(rPflip)=true then
      begin
       result[i,j]:=not result[i,j];
      end;
     end;
    end;
    RepairAFLC(result);
```

```
  GetSwap(result);
end;

function TObjFunc.getConstructSolution(const rFlip:double):bArr2;
var i,j:integer;
    chromBase,chromONOFF:bArr2;
begin
  chromBase:=CreateChromBase;
  chromONOFF:=CreateChromONOFF;
  SetLength(result,Ngen+1,Njam+1);
  for i:=1 to Ngen do
  begin
    for j:=1 to Njam do
    begin
      if chromBase[i,j]=true then
      begin
        result[i,j]:=true;
      end
      else
      begin
        if chromONOFF[i,j]=true then
        begin
          if isON(rFlip)=true then
          begin
            result[i,j]:=true;
          end
          else
          begin
            result[i,j]:=false;
          end;
        end;
      end;
    end;
  end;
  RepairAFLC(result);
  GetSwap(result);
end;

procedure TObjFunc.setLocalSearch(var rChrom:bArr2);
var i,j:integer;
    CostBest,CostCek:double;
    greyChrom:bArr2;
begin
  doHitungChrom(rChrom,CostBest);
  greyChrom:=doCariGreyZone(rChrom);
  for i:=1 to FNgen do
```

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

procedure TObjFunc.GetSwap(var rChrom:bArr2);
var i,j,k,init,pos:integer;
begin
 for i:=1 to FNgen do
 begin
  init:=FGen[i].InitSt;
  for j:=1 to FNjam do
  begin
   if rChrom[i,j]=true then
   begin
    if init<0 then
    begin
     if abs(init)>=FGen[i].Tdown then
     begin
      init:=1;
     end
     else
     begin
      pos:=j+init;
      if pos<1 then
      begin
       pos:=1;
      end;
      for k:=pos to j-1 do
      begin
       rChrom[i,k]:=true;
```

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

procedure TObjFunc.UpdateChrom(var rChrom:bArr2);
var i,j:integer;
   chromBase:bArr2;
begin
 chromBase:=CreateChromBase;
 for i:=1 to FNgen do
 begin
  for j:=1 to FNjam do
  begin
   if (rChrom[i,j]=false) and (chromBase[i,j]=true) then
   begin
    rChrom[i,j]:=true;
   end;
  end;
 end;
```

```
  RepairAFLC(rChrom);
  GetSwap(rChrom);
end;

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

function TObjFunc.HitungEcoDis(const rJam:integer;
      const rChrom1:bArr1):dArr1;
var i,j:integer;
   Status:bArr1;
   LoadCek,Pa,Pb,Lmd,LoadSplit:double;
   aBeban,diffa2,diffa1,Cek,tes:double;
begin
 SetLength(Status,FNgen+1);
 for i:=1 to FNgen do
 begin
  Status[i]:=rChrom1[i];
  FGen[i].Daya:=0;
 end;
 aBeban:=FBeban[rJam];
 LoadCek:=aBeban;
 LoadSplit:=aBeban;
 for i:=1 to 15 do
 begin
  Pa:=0;
  Pb:=0;
  for j:=1 to FNgen do
  begin
   if Status[j] then
   begin
    diffa2:=FGen[j].a2*2;
    diffa1:=FGen[j].a1;
    Pa:=Pa+1/diffa2;
    Pb:=Pb+diffa1/diffa2;
```

```
  end;
end;
if Pa<>0 then
begin
  Lmd:=(LoadSplit+Pb)/Pa;
end
else
begin
  Lmd:=LoadSplit+Pb;
end;
Cek:=0;
for j:=1 to FNgen do
begin
  if Status[j] then
  begin
    diffa2:=2*FGen[j].a2;
    diffa1:=FGen[j].a1;
    FGen[j].Daya:=(Lmd-diffa1)/diffa2;
    if FGen[j].Daya<FGen[j].Pmin then
    begin
      FGen[j].Daya:=FGen[j].Pmin;
    end;
    if FGen[j].Daya>FGen[j].Pmax then
    begin
      FGen[j].Daya:=FGen[j].Pmax;
    end;
  end;
  Cek:=Cek+FGen[j].Daya;
end;
tes:=LoadCek-Cek;
if (tes<0.0001) and (tes>-0.0001) then
begin
  break;
end
else if tes>0 then
begin
  for j:=1 to fNgen do
  begin
    if Status[j] then
    begin
      if FGen[j].Daya=FGen[j].PMax then
      begin
        Status[j]:=false;
        LoadSplit:=LoadSplit-FGen[j].Daya;
        if LoadSplit<0 then
        begin
```

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

   function TObjFunc.HitungCostGen(const rPL:dArr2):dArr2;
   var i,j:integer;
   begin
    SetLength(result,FNgen+1,FNjam+1);
    for i:=1 to FNgen do
    begin
     for j:=1 to FNjam do
     begin
```

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

function TObjFunc.HitungCostSUC(const rPL:dArr2):dArr2;
var i,j,init,tcold:integer;
begin
 SetLength(result,FNgen+1,FNjam+1);
 for i:=1 to FNgen do
 begin
  init:=FGen[i].InitSt;
  tcold:=FGen[i].Tdown+FGen[i].Tcold;
  for j:=1 to FNjam do
  begin
   result[i,j]:=0;
   if rPL[i,j]<>0 then
   begin
    if init>0 then
    begin
     init:=init+1;
    end
    else if init<0 then
    begin
     if abs(init)<=tcold then
     begin
      result[i,j]:=FGen[i].Sh;
     end
     else
     begin
      result[i,j]:=FGen[i].Sc;
     end;
     init:=1;
    end;
   end
   else if rPL[i,j]=0 then
   begin
    if init>0 then
    begin
     init:=-1;
    end
    else if init<0 then
    begin
     init:=init-1;
    end;
   end;
```

```
    end;
  end;
end;

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

procedure TObjFunc.doHitungChrom(var rChrom:bArr2;
        var rPL:dArr2;
        var rCostPerJam:dArr1;
        var rCostTotal:double);
var i,j:integer;
    PLa:dArr1;
    CostGen,CostSUC:dArr2;
```

```
  chrom1:bArr1;
begin
 UpdateChrom(rChrom);
 SetLength(rPL,FNgen+1,FNjam+1);
 SetLength(rCostPerJam,FNjam+1);
 SetLength(chrom1,FNgen+1);
 SetLength(PLa,FNgen+1);
 for i:=1 to FNjam do
 begin
  for j:=1 to FNgen do
  begin
   chrom1[j]:=rChrom[j,i];
  end;
  PLa:=HitungEcoDis(i,chrom1);
  for j:=1 to FNgen do
  begin
   rPL[j,i]:=PLa[j];
  end;
 end;
 CostGen:=HitungCostGen(rPL);
 CostSUC:=HitungCostSUC(rPL);
 SetLength(rCostPerJam,FNjam+1);
 rCostTotal:=0;
 for i:=1 to FNjam do
 begin
  rCostPerJam[i]:=0;
  for j:=1 to FNgen do
  begin
   rCostPerJam[i]:=rCostPerJam[i]+CostGen[j,i]+CostSUC[j,i];
  end;
  rCostTotal:=rCostTotal+rCostPerJam[i];
 end;
end;

procedure TObjFunc.doHitungPLN(var rCostPerJam:dArr1;
      var rCostTotal:double);
var i,j:integer;
  CostGen,CostSUC:dArr2;
begin
 SetLength(rCostPerJam,FNjam+1);
 CostGen:=HitungCostGen(FPLN);
 CostSUC:=HitungCostSUC(FPLN);
 rCostTotal:=0;
 for i:=1 to FNjam do
 begin
  rCostPerJam[i]:=0;
```

```
    for j:=1 to FNgen do
    begin
      rCostPerJam[i]:=rCostPerJam[i]+CostGen[j,i]+CostSUC[j,i];
    end;
    rCostTotal:=rCostTotal+rCostPerJam[i];
  end;
end;

//data output
function TObjFunc.doCariGreyZone(const rChrom:bArr2):bArr2;
var i,j,init,tcold:integer;
begin
  SetLength(result,FNgen+1,FNjam+1);
  for i:=1 to FNgen do
  begin
    for j:=1 to FNjam do
    begin
      result[i,j]:=false;
    end;
  end;
  for i:=1 to FNgen do
  begin
    init:=FGen[i].InitSt;
    tcold:=FGen[i].Tdown+FGen[i].Tcold;
    for j:=1 to FNjam do
    begin
      if rChrom[i,j]=true then
      begin
        if init<0 then
        begin
          if abs(init)=(tcold+1) then
          begin
            result[i,j]:=true;
          end;
          init:=1;
        end
        else if init>0 then
        begin
          init:=init+1;
        end;
      end
      else if rChrom[i,j]=false then
      begin
        if init<0 then
        begin
          init:=init-1;
```

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

procedure TObjFunc.doExecute(var rChrom:bArr2;
       var rPL:dArr2;
       var rCostPerJam:dArr1;
       var rCostTotal:double);
var i,j:integer;
   CostBest,CostCek:double;
   greyChrom:bArr2;
begin
 rChrom:=CreateChromBase;
 GetSwap(rChrom);
 doHitungChrom(rChrom,CostBest);
 greyChrom:=doCariGreyZone(rChrom);
 for i:=1 to FNgen do
 begin
  for j:=1 to FNjam do
  begin
   if greyChrom[i,j]=true then
   begin
    rChrom[i,j-1]:=true;
    doHitungChrom(rChrom,CostCek);
    if CostBest>CostCek then
    begin
     CostBest:=CostCek;
    end
    else
    begin
     rChrom[i,j-1]:=false;
    end;
   end;
  end;
 end;
 doHitungChrom(rChrom,rPL,rCostPerJam,rCostTotal);
end;

//function Ant Colony
procedure TObjFunc.doCalcState(const rJam:integer;
```

```
        var rState1,rState2:TState);
var i:integer;
  status:boolean;
  PLa:dArr1;
  sumCostGen,sumSUC:double;
begin
 status:=true;
 for i:=1 to FNgen do
 begin
  if rState2.chrom[i]=true then
  begin
   if rState1.chrom[i]=false then
   begin
    if abs(rState1.stat[i])<FGen[i].Tdown then
    begin
     status:=false;
     break;
    end;
   end;
  end
  else if rState2.chrom[i]=false then
  begin
   if rState1.chrom[i]=true then
   begin
    if rState1.stat[i]<FGen[i].Tup then
    begin
     status:=false;
     break;
    end;
   end;
  end;
 end;
 if status=true then
 begin
  if isServe(rJam,rState2.chrom)=true then
  begin
   PLa:=HitungEcoDis(rJam,rState2.chrom);
   sumCostGen:=0;
   for i:=1 to FNgen do
   begin
    if PLa[i]<>0 then
    begin
     sumCostGen:=sumCostGen+FGen[i].GetBiaya(PLa[i]);
    end;
   end;
   sumSUC:=0;
```

```
    for i:=1 to FNgen do
    begin
     if rState2.chrom[i]=true then
     begin
      if rState1.chrom[i]=true then
      begin
       rState2.stat[i]:=rState1.stat[i]+1;
      end
      else if rState1.chrom[i]=false then
      begin
       if abs(rState1.stat[i])>(FGen[i].Tdown+FGen[i].Tcold) then
       begin
        sumSUC:=sumSUC+FGen[i].Sc;
       end
       else
       begin
        sumSUC:=sumSUC+FGen[i].Sh;
       end;
       rState2.stat[i]:=1;
      end;
     end
     else if rState2.chrom[i]=false then
     begin
      if rState1.chrom[i]=true then
      begin
       rState2.stat[i]:=-1;
      end
      else if rState1.chrom[i]=false then
      begin
       rState2.stat[i]:=rState1.stat[i]-1;
      end;
     end;
    end;
    rState2.food:=rState1.food+sumCostGen+sumSUC;
   end
   else
   begin
    rState2.food:=0;
   end;
  end
  else
  begin
   rState2.food:=0;
  end;
end;
end.
```

```
unit uDataAnt;

interface

uses uUtils;

type
 TState=record
  chrom:bArr1;
  stat:iArr1;
  food:double;
 end;

 TStateArr1=array of TState;
 TStateArr2=array of array of TState;

 TCity=record
  count:integer;
  state:TStateArr1;
 end;

 TCityArr1=array of TCity;

 TAnt=record
  Edge:TStateArr1;
  sumFood:double;
  status:boolean;
 end;

 TAntArr1=array of TAnt;

 TJam=record
  pos:dArr2;
 end;

 TJamArr1=array of TJam;

 TPheromone=record
  jam:TJamArr1;
 end;

 TPheromoneArr1=array of TPheromone;

implementation

end.
```

```
unit uAntColony;

interface

uses uUtils,uDataAnt,uObjFunc,uHasil;

type
 TAntColony=class
 private
  FMaxIterasi,FNmove,FNparam,FNAnt:integer;
  Fq0,FAlpha,FBetha,Fd,Fp,FInitPhe,FKa:double;
  FCities:TCity;
  FPhe:TPheromone;
  FAnts:TAntArr1;
  FInitState:iArr1;
  function getAnt(var rAnt:TAnt):TAnt;
  procedure FillStateToAnt(const rNmove:integer;
        var rAnt:TAnt;
        var rState:TState);
  procedure ResetAnts;
  procedure ResetCities;
  procedure CalcCities(const rNmove:integer;
        var rAnt:TAnt);
  function CheckSameChrom(const rChrom1,rChrom2:bArr1):boolean;
  function FindPosAnt(const rNmove:integer;
       var rAnt:TAnt):integer;
  function FindPosFirst:integer;
  procedure Statistik(const rNmove,rPosAnt:integer;
        var rSumFood:double);
  procedure Seleksi(const rNmove:integer;
        var rAnt:TAnt;
        var rDest:integer);
  procedure AntMove(const rNmove:integer;
        var rAnt:TAnt);
  procedure UpdateGlobalPheromone(var rAnt:TAnt);
  function FindBestAnt(var rAnts:TAntArr1):TAnt;
 public
  constructor Create(const rMaxIterasi,rNmove,rNparam,rNAnt:integer;
         const rq0,rAlpha,rBetha,rd,rp,rInitPhe,rKa:double;
         const rInitState:iArr1);
  procedure dolHitung(var rChrom:bArr2);
  property MaxIterasi:integer read FMaxIterasi write FMaxIterasi;
  property Nmove:integer read FNmove write FNmove;
  property Nparam:integer read FNparam write FNparam;
  property NAnt:integer read FNAnt write FNAnt;
 end;
```

```
implementation

//constructor
constructor TAntColony.Create(const
rMaxIterasi,rNmove,rNparam,rNAnt:integer;
        const rq0,rAlpha,rBetha,rd,rp,rInitPhe,rKa:double;
        const rInitState:iArr1);
var i,j,k:integer;
  sortAFLC:iArr1;
begin
 inherited Create;
 FMaxIterasi:=rMaxIterasi;
 FNmove:=rNmove;
 FNparam:=rNparam;
 FNAnt:=rNAnt;
 Fq0:=rq0;
 FAlpha:=rAlpha;
 FBetha:=rBetha;
 Fd:=rd;
 Fp:=rp;
 FInitPhe:=rInitPhe;
 FKa:=rKa;
 SetLength(FCities.state,FNparam+1);
 FCities.count:=FNparam;
 sortAFLC:=gObjFunc.SortAFLC;
 SetLength(FInitState,FNparam+1);
 for i:=1 to FNparam do
 begin
  FInitState[i]:=rInitState[i];
 end;
 for i:=1 to FCities.count do
 begin
  SetLength(FCities.state[i].chrom,FNparam+1);
  SetLength(FCities.state[i].stat,FNparam+1);
  for j:=1 to FNparam do
  begin
   FCities.state[i].chrom[j]:=false;
   FCities.state[i].stat[j]:=0;
   FCities.state[i].food:=0;
  end;
  for j:=1 to i do
  begin
   FCities.state[i].chrom[sortAFLC[j]]:=true;
  end;
 end;
 SetLength(FPhe.jam,FNmove+1);
```

```
  for i:=1 to FNmove do
  begin
   SetLength(FPhe.jam[i].pos,FNparam+1,FNparam+1);
   for j:=1 to FNparam do
   begin
    for k:=1 to FNparam do
    begin
     FPhe.jam[i].pos[j,k]:=1;
    end;
   end;
  end;
  SetLength(FAnts,FNAnt+1);
  for i:=0 to FNAnt do
  begin
   SetLength(FAnts[i].Edge,FNmove+1);
   for j:=0 to FNmove do
   begin
    SetLength(FAnts[i].Edge[j].chrom,FNparam+1);
    SetLength(FAnts[i].Edge[j].stat,FNparam+1);
    for k:=1 to FNparam do
    begin
     FAnts[i].Edge[j].chrom[k]:=false;
     FAnts[i].Edge[j].stat[k]:=0;
     FAnts[i].Edge[j].food:=0;
    end;
   end;
   for j:=1 to FNparam do
   begin
    if FInitState[j]<0 then
    begin
     FAnts[i].Edge[0].chrom[j]:=false;
     FAnts[i].Edge[0].stat[j]:=FInitState[j];
    end
    else
    begin
     FAnts[i].Edge[0].chrom[j]:=true;
     FAnts[i].Edge[0].stat[j]:=FInitState[j];
    end;
   end;
   FAnts[i].status:=true;
   FAnts[i].sumFood:=0;
  end;
 end;

function TAntColony.getAnt(var rAnt:TAnt):TAnt;
var i,j:integer;
```

```
begin
  SetLength(result.Edge,FNmove+1);
  for i:=1 to FNmove do
  begin
    SetLength(result.Edge[i].chrom,FNparam+1);
    SetLength(result.Edge[i].stat,FNparam+1);
    for j:=1 to FNparam do
    begin
      result.Edge[i].chrom[j]:=rAnt.Edge[i].chrom[j];
      result.Edge[i].stat[j]:=rAnt.Edge[i].stat[j];
      result.Edge[i].food:=rAnt.Edge[i].food;
    end;
  end;
  result.status:=rAnt.status;
  result.sumFood:=rAnt.sumFood;
end;

procedure TAntColony.FillStateToAnt(const rNmove:integer;
        var rAnt:TAnt;
        var rState:TState);
var i:integer;
begin
  for i:=1 to FNparam do
  begin
    rAnt.Edge[rNmove].chrom[i]:=rState.chrom[i];
    rAnt.Edge[rNmove].stat[i]:=rState.stat[i];
  end;
  rAnt.Edge[rNmove].food:=rState.food;
  if rAnt.status=true then
  begin
    rAnt.sumFood:=rAnt.sumFood+rState.food;
  end
  else
  begin
    rAnt.sumFood:=0;
  end;
end;

procedure TAntColony.ResetAnts;
var i,j,k:integer;
begin
  for i:=1 to FNAnt do
  begin
    for j:=1 to FNmove do
    begin
      for k:=1 to FNparam do
```

```
      begin
        FAnts[i].Edge[j].chrom[k]:=false;
        FAnts[i].Edge[j].stat[k]:=0;
      end;
      FAnts[i].Edge[j].food:=0;
    end;
    FAnts[i].status:=true;
    FAnts[i].sumFood:=0;
  end;
end;

procedure TAntColony.ResetCities;
var i,j:integer;
begin
  for i:=1 to FCities.count do
  begin
    for j:=1 to FNparam do
    begin
      FCities.state[i].stat[j]:=0;
    end;
    FCities.state[i].food:=0;
  end;
end;

procedure TAntColony.CalcCities(const rNmove:integer;
        var rAnt:TAnt);
var i:integer;
begin
  for i:=1 to FCities.count do
  begin
    gObjFunc.doCalcState(rNmove,rAnt.Edge[rNmove-1],
    FCities.State[i]);
    if FCities.State[i].food<>0 then
    begin
      FCities.State[i].food:=FKa/FCities.State[i].food;
    end;
  end;
end;

function TAntColony.CheckSameChrom(const
rChrom1,rChrom2:bArr1):boolean;
var i:integer;
begin
  result:=true;
  for i:=1 to FNparam do
  begin
```

```
    if rChrom1[i]<>rChrom2[i] then
    begin
      result:=false;
      break;
    end;
  end;
end;

function TAntColony.FindPosAnt(const rNmove:integer;
        var rAnt:TAnt):integer;
var i:integer;
begin
  result:=0;
  for i:=1 to FCities.count do
  begin
    if CheckSameChrom(rAnt.Edge[rNmove].chrom,
      FCities.state[i].chrom)=true then
    begin
      result:=i;
      break;
    end;
  end;
end;

function TAntColony.FindPosFirst:integer;
var i:integer;
    chromFirst:bArr1;
begin
  result:=0;
  SetLength(chromFirst,FNparam+1);
  for i:=1 to FNparam do
  begin
    if FInitState[i]<0 then
    begin
      chromFirst[i]:=false;
    end
    else
    begin
      chromFirst[i]:=true;
    end;
  end;
  for i:=1 to FCities.count do
  begin
    if CheckSameChrom(chromFirst,FCities.state[i].chrom)=true then
    begin
      result:=i;
```

```
      break;
    end;
  end;
end;

procedure TAntColony.Statistik(const rNmove,rPosAnt:integer;
        var rSumFood:double);
var i:integer;
    phe,Tij:double;
begin
  rSumFood:=0;
  for i:=1 to FCities.count do
  begin
    phe:=FPhe.jam[rNmove].pos[rPosAnt,i];
    Tij:=FCities.state[i].food;
    if Tij<>0 then
    begin
      rSumFood:=rSumFood+pangkat(phe,FAlpha)*pangkat(Tij,FBetha);
    end;
  end;
end;

procedure TAntColony.Seleksi(const rNmove:integer;
        var rAnt:TAnt;
        var rDest:integer);
var i,maxPos,posAnt:integer;
    SumFood,rand,partFood,maxPhe,pheCity,phe,Tij,cek:double;
begin
  posAnt:=FindPosAnt(rNmove-1,rAnt);
  Statistik(rNmove,posAnt,SumFood);
  if SumFood=0 then rAnt.status:=false;
  phe:=FPhe.jam[rNmove].pos[PosAnt,1];
  Tij:=FCities.state[1].food;
  maxPhe:=0;
  if Tij<>0 then
  begin
    maxPhe:=pangkat(phe,FAlpha)*pangkat(Tij,FBetha);
  end;
  maxPos:=1;
  for i:=2 to FCities.count do
  begin
    phe:=FPhe.jam[rNmove].pos[PosAnt,i];
    Tij:=FCities.state[i].food;
    pheCity:=0;
    if Tij<>0 then
    begin
```

```
    pheCity:=pangkat(phe,FAlpha)*pangkat(Tij,FBetha);
   end;
   if maxPhe<pheCity then
   begin
    maxPhe:=PheCity;
    maxPos:=i;
   end;
  end;
  rand:=random;
  if rand<=Fq0 then
  begin
   rDest:=maxPos;
  end
  else
  begin
   rDest:=0;
   partFood:=0;
   rand:=random*SumFood;
   repeat
    inc(rDest);
    phe:=FPhe.jam[rNmove].pos[PosAnt,rDest];
    Tij:=FCities.state[rDest].food;
    if Tij<>0 then
    begin
     pheCity:=pangkat(phe,FAlpha)*pangkat(Tij,FBetha);
     partFood:=partFood+pheCity;
    end;
   until (partFood>rand) or (rDest=FCities.count);
  end;
  //update local pheromone menurut jurnal Rendra Salasa
  cek:=Fd*FPhe.jam[rNmove].pos[posAnt,rDest]+(1-Fd)*FInitPhe;
  FPhe.jam[rNmove].pos[posAnt,rDest]:=cek;

//FPhe[rNoWaduk].jam[rJam].pos[PosAnt,rDest]:=Fd*FPhe[rNoWaduk].jam[rJa
m].pos[posAnt,rDest]+
  //                          (1-Fd)*FInitPhe;
  //update local pheromone menurut jurnal Herman
  //FPhe[rNoWaduk].jam[rJam].pos[PosAnt,rDest]:=(1-
Fd)*FPhe[rNoWaduk].jam[rJam].pos[posAnt,rDest]+
  //                          Fd*FInitPhe;
end;

procedure TAntColony.AntMove(const rNmove:integer;
        var rAnt:TAnt);
var posCity:integer;
begin
```

```
 CalcCities(rNmove,rAnt);
 Seleksi(rNmove,rAnt,posCity);
 FillStateToAnt(rNmove,rAnt,FCities.State[posCity]);
end;

procedure TAntColony.UpdateGlobalPheromone(var rAnt:TAnt);
var i,posPrev,posNext,posFirst:integer;
   dPhe,dPh:double;
begin
 dPhe:=rAnt.sumFood;
 posFirst:=FindPosFirst;
 for i:=1 to FNmove do
 begin
  if i=1 then
  begin
   posPrev:=posFirst;
  end
  else
  begin
   posPrev:=FindPosAnt(i-1,rAnt);
  end;
  posNext:=FindPosAnt(i,rAnt);
  //update global pheromone menurut jurnal Hendra Salasa
  dPh:=Fp*FPhe.jam[i].pos[posPrev,posNext]+(1-Fp)*dPhe;
  FPhe.jam[i].pos[posPrev,posNext]:=dPh;
  //update global pheromone menutur jurnal Herman
  //FPhe[i].jam[j].pos[posPrev,posNext]:=(1-
Fp)*FPhe[i].jam[j].pos[posPrev,posNext]+
  //                          Fp*dPhe;
  //Phe[i].pos[posPrev,posNext]:=(1-Fp)*FPhe[i].pos[posPrev,posNext]+
  //                        Fp*dPhe;
 end;
end;

function TAntColony.FindBestAnt(var rAnts:TAntArr1):TAnt;
var i,posMax:integer;
   maxFood:double;
begin
 maxFood:=rAnts[1].sumFood;
 posMax:=1;
 for i:=2 to FNAnt do
 begin
  if maxFood<rAnts[i].sumFood then
  begin
   maxFood:=rAnts[i].sumFood;
   posMax:=i;
```