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

Analisa Penempatan Kapasitor Dengan Menggunakan Hybrid Particle Swarm Optimazer With Mutation (HPSOM) Pada Saluran Transmisi 150 kV Sub Sistem Paiton - Bali

SKRIPSI

Disusun Oleh :
IRWAN ZAINUDIN YAZID
NIM. 01.12.103

SEPTEMBER 2006

LEMBAR PERSETUJUAN

ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN

HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION (HPSOM)

PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON–BALI

SKRIPSI

Disusun Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik

Disusun Oleh :
Irwan Zainudin Yazid
NIM. 01.12.103

Mengetahui,	Diperiksa dan Disetujui,
Ketua Jurusan Teknik Elektro S-1	Dosen Pembimbing
Ir. F Yudi Limpraptono, MT	Ir. Eko Nurcahyo
Nip. Y. 103 9500 274	Nip. P. 102 8700 172

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

ABSTRAKSI

ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON BALI

(Irwan Zainudin Yazid, NIM 01.12.103, Teknik Elektro/ Teknik Energi Listrik)
(Dosen Pembimbing : Ir Eko Nurcahyo)

Kata Kunci : Optimal Power Flow, Minimalisasi Rugi Daya, Particle Swarm Optimation.

Dengan semakin meningkatnya kebutuhan energi listrik baik untuk industri maupun rumah tangga, yang sebagian besar didominasi oleh beban induktif. Dengan meningkatnya beban induktif , maka daya reaktif dalam jaringan akan semakin besar sehingga memperbesar komponen rugi-rugi daya dan memperburuk profil tegangan pada jaringan transmisi.

Permasalahan Optimal power Flow *telah menjadi perhatian utama khususnya pada saluran transmisi. Alternatif yang bisa dipakai untuk memperbaiki kondisi jaringan antara lain adalah dengan pengaturan tegangan generator, pemasangan kapasitor dan penambahan daya aktif generator Pada skripsi ini digunakan* Hybrid Particle Swarn Optimizer With Mutation (HPSOM)*, yang dapat dijadikan sebagai salah satu alternatif dalam* meminimalisasi rugi-rugi daya *pada saluran transmisi 150 kV Paiton –Bali. Dalam skripsi ini akan dibahas tentang penentuan kapasitas kapasitor, sehingga akan dihasilkan kualitas tenaga listrik yang baik.*

Analisa dilakukan dengan bantuan program komputer dengan menggunakan bahasa pemrograman Delphi versi 7.0 dan telah sukses diuji cobakan pada sub sistem 150 kV Paiton dan Bali yang terdiri dari 25 bus.

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT atas segala limpahan Rahmat, Hidayah dan Ma'unah-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi yang berjudul

"ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN *HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION (HPSOM)* PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON–BALI"

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

Selama penulisan skripsi ini, penulis telah banyak mendapatkan bimbingan, pengarahan dan bantuan yang diberikan dari berbagai pihak, sehingga skripsi ini dapat terselesaikan, oleh karena itu, pada kesempatan kali ini penulis 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. Eko Nurcahyo , selaku Dosen Pembimbing.
4. PT. PLN (Persero) P3B Region Jawa Timur dan Bali yang telah memberikan data yang di butuhkan dalam penulisan skripsi ini.

DAFTAR ISI

HALAMAN JUDUL i

LEMBAR PERSETUJUAN ii

ABSTRAKSI iii

KATA PENGANTAR iv

DAFTAR ISI vii

DAFTAR GAMBAR x

DAFTAR TABEL xii

DAFTAR GRAFIK xiii

BAB I PENDAHULUAN 1

1.1. Latar Belakang 1

1.2. Rumusan Masalah 2

1.3. Tujuan 3

1.4. Batasan Masalah 3

1.5. Metodologi Penelitian 4

1.6. Sistematika Penulisan 5

1.7. Kontribusi Penelitian 6

BAB II SISTEM TENAGA LISTRIK, OPERASI EKONOMIS DAN KARAKTERISTIK UNIT PEMBANGKIT 7

2.1. Sistem Tenaga Listrik 7

2.2. Saluran Transmisi 8

2.2.1. Saluran Transmisi Pendek 9

2.2.2. Saluran Transmisi Menengah 9

2.2. Daya Dalam Sistem Tenaga Listrik 12

2.3.1. Daya Aktif 12

2.3.2. Daya Reaktif 13

2.3.3. Daya Semu 13

2.4. Sistem Per-Unit 14

2.5. Mengubah Dasar Sistem Per-Unit 15

2.6. Kapasitor Shunt 16

2.7. Pengurangan Rugi-rugi Daya Dengan Kapasitor Shunt 17

2.8. Perbaikan Tegangan 18

2.9. Perbaikan Faktor Daya dan Kenaikan Kapasitas sistem 19

2.10 Perhitungan Pengaruh Perbaikan Faktor Daya 21

2.11. Penentuan Rating Kapasitor Untuk Perbaikan Faktor Daya Beban 22

BAB III ANALISA PENEMPATAN KAPASITOR MENGGUNAKAN METODE HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION (HPSOM) 24

3.1. Analisa Aliran Daya 24

3.1.1. Klasifikasi Bus 25

3.1.2. Metode Newton Raphson 26

3.2. Hybrid Particle Swarm Optimazer With Mutation 28

3.2.1. Teori Dasar Particle Swarm Optimazer 28

3.2.2. Parameter PSO 28

3.2.3. Hybrid Particle Swarm Optimazer With Mutation 31

3.2.4. Mutation (Mutasi) 32

3.3. Formulasi Permasalahan 33

BAB IV HASIL DAN ANALISA HASIL 35

4.1. Program Komputer Analisa Penempatan Kapasitor Menggunakan Metode Hybrid Particle Swarm Optimazer With Mutation 35

4.2. Algoritma Program 35

4.2.1. Algoritma Aliran Daya Newton Raphson 35

4.2.2. Algoritma Pemecahan Masalah 36

4.2.3. Algoritma Program Analisa Penempatan Kapasitor Menggunakan Metode Hybrid Particle Swarm Optimazer With Mutation 37

4.3. Diagram Alir 37

4.3.1. Diagram Alir Aliran Daya Newton Rapshon 38

4.3.2. Diagram Alir Pemecahan Masalah 39

4.3.3. Diagram Alir Analisa Penempatan Kapasitor Menggunakan Metode Hybrid Particle Swarm Optimazer With Mutation 40

4.4. Validasi Data IEEE 30 41

4.4.1. Hasil Validasi IEEE 30 Dengan Menggunakan Hybrid Particle Swarm Optimazer With Mutation 43

4.5. Data Pembangkitan dan Pembebanan 150 kV Sub Sistem Paiton dan Bali 46

4.6. Data Saluran Transmisi 150 kV Sub Sistem Paiton dan Bali 47

4.7. Prosedur Pelaksanaan Program Perhitungan 49

4.8. Hasil dan Analisis Hasil Perhitungan Analisa Penempatan Kapasitor Menggunakan Metode Hybrid Particle SwarmOptimazer With Mutation Pada Saluran Transmisi 150 kV Sub Sistem Paiton dan Bali..... 57

4.8.1. Hasil Perhitungan Sebelum Optimasi 57

4.8.2. Hasil Perhitungan Setelah Optimasi 60

4.9. Perbandingan Hasil Perhitungan Sebelum dan Setelah Optimasi Analisa Penempatan Kapasitor Menggunakan Metode Hybrid Particle SwarmOptimazer With Mutation 62

4.9.1. Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus 63

4.9.2. Perbandingan Tingkat Rugi-rugi Daya Pada Saluran 65

4.10. Hasil Analisa 66

BAB V KESIMPULAN DAN SARAN 69

5.1. Kesimpulan 69

5.2. Saran..... 70

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1. Rangkaian Setara Saluran Transmisi 8

Gambar 2.2. Rangkaian Setara Saluran Transmisi Pendek 9

Gambar 2.3. Rangkaian Setara Saluran Transmisi Menengah 10

Gambar 2.4. Diagram Skema Saluran Transmisi Panjang 11

Gambar 2.5. Representasi Segitiga Daya 14

Gambar 2.6. Saluran Primer Dengan Beban Terpusat 17

Gambar 2.7. Vektor Diagram sebelum dan Sesudah Pemasangan Kapasitor 19

Gambar 2.8. Perbaikan Faktor Daya 20

Gambar 2.9. Diagram Fasor Dan Sudut Daya Beban Transmisi 21

Gambar 4.1. Diagram Alir Aliran Daya NewtonRapshon 38

Gambar 4.2. Diagram Alir Pemecahan Masalah 39

Gambar 4.3. Diagram Alir Analisa Penempatan Kapasitor Menggunakan Metode Hybrid Particle SwarmOptimazer With Mutation 40

Gambar 4.4. Tampilan Parameter Validasi 43

Gambar 4.5. Tampilan Hasil Validasi 44

Gambar 4.6. Diagram Single Line Sub Sistem Paiton Bali 45

Gambar 4.7. Menu Utama Program 49

Gambar 4.8. Membuka File Optimasi 50

Gambar 4.9. Tampilan Data General 50

Gambar 4.10. Tampilan Data Bus 51

Gambar 4.11. Tampilan Data Saluran 51

Gambar 4.12. Tampilan Data Generator 52

Gambar 4.13. Tampilan LoadFlow Awal 52

Gambar 4.14. Tampilan Aliran Daya Awal 53

Gambar 4.15. Tampilan Summary Hasil Perhitungan Awal 53

Gambar 4.16. Tampilan Parameter Untuk Optimasi 54

Gambar 4.17. Tampilan Hasil Perhitungan Program 54

Gambar 4.18. Tampilan Hasil LoadFlow Setelah Optimasi 55

Gambar 4.19. Tampilan Hasil Aliran Daya Setelah Optimasi 55

Gambar 4.20. Tampilan Summary Setelah Optimasi 56

Gambar 4.21. Tampilan Grafik Perbandingan Tegangan 56

Gambar 4.22. Diagram lokasi Penempatan Kapasitor 68

DAFTAR TABEL

Tabel 4.1. Data Pembangkitan IEEE 30 Bus 41

Tabel 4.2. Data Saluran Impedansi IEEE 30..... 42

Tabel 4.3. Perbandingan Hasil Data Referensi Jurnal Dengan Data Optimasi 44

Tabel 4.4. Data Pembangkitan Dan Pembebanan 150 kV Sub Sistem Paiton- Bali ... 46

Tabel 4.5. Data Saluran Transmisi 150 kV sub Sistem Paiton dan Bali 48

Tabel 4.6. Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Sebelum Optimasi 57

Tabel 4.7. Hasil Perhitungan Aliran Daya Antar Saluran Sebelum Optimasi 58

Tabel 4.8. Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum Optimasi 59

Tabel 4.9. Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Setelah Optimasi 60

Tabel 4.10. Hasil Perhitungan Aliran Daya Antar Saluran Setelah Optimasi 61

Tabel 4.11. Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Setelah Optimasi 62

Tabel 4.12. Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus 63

Tabel 4.13. Perbandingan Aliran Daya Tiap Saluran..... 65

Tabel 4.14. Penempatan Dan Kapasitas Kapasitor Pada Tiap Bus 66

DAFTAR GRAFIK

Grafik 4.1. Hasil Perhitungan Daya Yang Dibangkitkan Sebelum Optimasi 59

Grafik 4.2. Hasil Perhitungan Daya Yang Dibangkitkan Setelah Optimasi 62

Grafik 4.3. Perbandingan Profil Tegangan 64

Grafik 4.4. Perbandingan Total Daya Aktif Pembangkitan, Pembebanan,Rugi-Rugi Pada Kondisi Awal dan Kondisi Akhir 66

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dengan meningkatnya usaha di sektor industri dan meningkatnya taraf hidup masyarakat maka kebutuhan akan energi listrik semakin meningkat pula, sehingga diperlukan penyediaan energi listrik yang baik dan keandalan penyaluran tegangan. Bertambahnya industri – industri menyebabkan peran penggunaan alat – alat listrik akan semakin luas, misalnya motor – motor listrik, trafo, AC, lampu TL, dan lain – lain. Beban industri sangat membutuhkan daya reaktif induktif. Dengan meningkatnya beban – beban induktif, maka daya reaktif yang ada di jaringan transmisi akan semakin besar, sehingga memperbesar komponen rugi – rugi daya dan dapat memperburuk profil tegangan pada jaringan transmisi.

Alokasi daya reaktif yang tepat dalam sistem bisa menghasilkan kondisi optimum, yaitu kondisi dengan rugi-rugi transmisi yang minimum dengan biaya investasi yang murah dan mampu memperbaiki profil tegangan sistem. Permasalahan besarnya rugi-rugi daya di saluran menjadi perhatian utama khususnya pada saluran transmisi 150 KV.

Permasalahan tersebut telah menjadi perhatian utama khususnya pada saluran transmisi. Alternatif yang bisa dipakai untuk memperbaiki kondisi jaringan antara lain adalah dengan pengaturan tegangan generator, pemasangan kapasitor dan penambahan daya aktif generator.

Pada skripsi ini digunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)*, yang dapat dijadikan sebagai salah satu alternatif dalam mengurangi rugi-rugi pada saluran transmisi 150 kV Paiton –Bali. Dalam skripsi ini akan dibahas tentang penentuan kapasitas kapasitor, sehingga akan dihasilkan kualitas tenaga listrik yang baik.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan yaitu :

- Bagaimana optimalisasi yang akan dilakukan pada saluran transmisi 150 kV menggunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)* .
- Bagaimana kondisi profil tegangan setelah dilakukan optimasi.
- Bagaimana rugi-rugi daya pada saluran transmisi 150 kV setelah optimasi menggunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM).*

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

"ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN *HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION (HPSOM)* PADA SALURAN TRANSMISI 150 KV SUB SISTEM PAITON–BALI"

1.3. Tujuan

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

- Menganalisis optimasi yang akan digunakan pada saluran transmisi 150 kV menggunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM).*
- Menganalisis profil tegangan setelah proses optimalisasi.

  Menganalisis rugi daya dan profil tegangan pada saluran transmisi 150 kV setelah optimalisasi menggunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM).*

1.4. Batasan Masalah

Dalam skripsi ini akan dilakukan analisis mengenai optimalisasi menggunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)* pada saluran 150 kV sub sistem Paiton-Bali. Agar pembahasan mengarah sesuai tujuan, maka pembahasan dalam proposal skripsi ini dibatasi oleh hal sebagai berikut :

- Analisa dilakukan dengan asumsi bahwa sistem berada dalam operasi normal.
- Tidak membahas masalah pengaruh kontrol tegangan terhadap alat proteksi.
- Tidak membahas kabel laut, masalah peralatan kompensasi dan penempatannya.
- Tidak membahas metode Newton Rapshon secara mendetail.

1.5. Metodologi Penelitian

Metode yang digunakan dalam pembahasan proposal ini dilakukan dengan langkah – langkah sebagai berikut :

- Studi Literatur Yaitu kajian pustaka yang mempelajari teori – teori yang terkait melalui literatur yang telah ada, yang berhubungan dengan permasalahan.
- Pengumpulan data lapangan yang dipakai dalam objek penelitian yakni data impedansi saluran transmisi, data pembebanan, data tap kapasitor serta data kapasitas kapasitor yang akan digunakan pada saluran 150 kV sub sistem Paiton - Bali.
- Simulasi dan pembahasan masalah

  Simulasi dan pembahasan masalah dengan menggunakan bahasa pemrograman Borland Delphi versi 7.0 .
- Pengambilan kesimpulan dari hasil analisis.

1.6. Sistimatika Penulisan

Sistimatika dari pembahasan di dalam skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Berisikan Latar belakang, Rumusan Masalah, Tujuan, Batasan Masalah, Metodologi Penelitian, Sistematika Penulisan dan Konstribusi.

BAB II : SISTEM TRANSMISI DAN APLIKASI KAPASITOR

Disini akan dibahas masalah sistem jaringan transmisi, pengertian jatuh tegangan, pengaruh rendahnya jatuh daya, penjelasan teori tentang kapasitor serta faktor daya .

BAB III : **ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN *HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION (HPSOM)***

Pada bab ini akan dibahas metode aliran daya *Newton Raphson*, teori *Particle Swarm Optimizer* dan teori-teori mengenai metode *Hybrid Particle Swarn Optimizer With Mutation (HPSOM).*

BAB IV : HASIL DAN ANALISA HASIL

Menguraikan alur program, hasil validasi, serta hasil perhitungan OPF menggunakan metode *Hybrid Particle Swarn Optimizer With Mutation (HPSOM).*

BAB V : KESIMPULAN

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan, yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

1.7 Konstribusi Penelitian

Penggunaan dari metode *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)*, dalam menganalisa permasalahan OPF dapat membantu pemerintah khususnya PT. PLN (persero) dalam mengatasi masalah pembangkitan dalam sistem tenaga listrik guna menghasilkan operasi yang andal dan ekonomis khususnya dalam biaya pembangkitan. Maka kami berharap agar metode ini dapat dijadikan sebagai bahan pertimbangan oleh PT. PLN (persero) dalam pemecahan masalah OPF.

BAB II
SISTEM TENAGA LISTRIK

2.1. Sistem Tenaga Listrik

Tiga bagian utama sistem tenaga listrik yaitu : pusat pembangkit tenaga listrik, saluran transmisi dan sistem distribusi yang berhubungan langsung dengan konsumen. Saluran transmisi merupakan penghubung antara pusat pembangkit melalui hubungan antar sistem yang menuju sistem pada sistem yang lain.

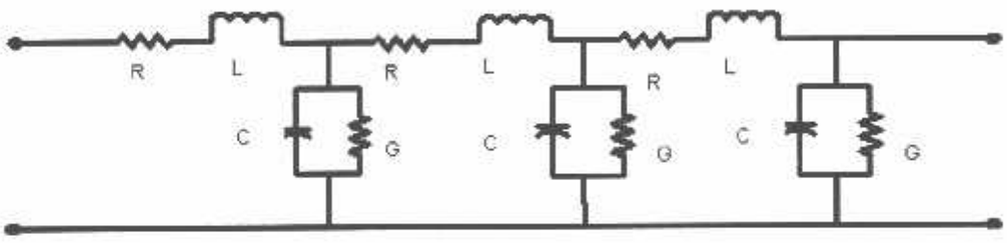
Saluran transmisi mempunyai empat parameter yang mempengaruhi kemampuannya dalam menyalurkan daya listrik. Keempat parameter tersebut yaitu : resistansi (R), induktansi (L), kapasitansi (C), serta konduktansi (G).

Resistansi umumnya tergantung pada jenis penghantar sedangkan konduktansi menyatakan besarnya arus bocor antar penghantar, antar penghantar dengan tanah, tetapi harganya relatif kecil maka dapat diabaikan. Induktansi adalah parameter rangkaian yang menghubungkan tegangan yang diimbaskan oleh perubahan fluksi akibat perubahan arus, sedangkan kapasitansi suatu saluran transmisi timbul akibat adanya beda potensial antara penghantar dengan tanah, dalam hal ini kapasitansi menyebabkan penghantar bermuatan seperti yang terjadi pada pelat kapasitor.

Impedansi seri terbentuk dari resistansi dan induktansi yang terbagi merata sepanjang saluran. Sedangkan konduktansi dan kapasitansi terdapat diantara penghantar-penghantar dari saluran fasa tunggal atau diantar penghantar dengan netral dari suatu saluran berfasa tiga membentuk admitansi paralel.

2.2. Saluran Transmisi

Tenaga listrik yang dibangkitkan disalurkan melalui saluran transmisi. Saluran–saluran transmisi ini membawa tenaga listrik dari pusat–pusat tenaga listrik ke pusat–pusat beban. Suatu saluran transmisi tenaga listrik mempunyai empat parameter yang mempengaruhi kemampuan untuk berfungsi sebagai bagian dari sistem tenaga, yaitu resistansi, induktansi, kapasitansi, dan konduktansi. Keempat parameter saluran transmisi tersebut merata disepanjang saluran transmisi. Parameter–parameter tersebut sangat berpengaruh terhadap tegangan bus dan aliran daya yang mengalir pada saluran tersebut.



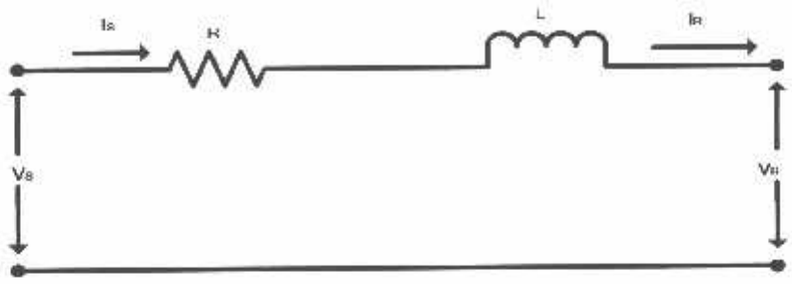
Gambar 2.1.
Rangkaian Setara Saluran Transmisi[2]

Menurut panjangnya, saluran transmisi dapat dikasifikasikan menjadi 3 golongan, yaitu :

1. Saluran transmisi pendek, adalah saluran yang panjangnya < 80 km.
2. Saluran transmisi menengah, adalah saluran yang panjangnya 80-240 km.
3. Saluran transmisi panjang, adalah saluran yang panjangnya > 240 km.

2.2.1. Saluran Transmisi Pendek

Rangkaian ekivalen untuk saluran transmisi pendek diperlihatkan pada gambar 2.2 dimana I_S dan I_g merupakan arus pada ujung pengiriman dan ujung penerimaan. Sedangkan V_S dan V_R adalah tegangan saluran terhadap netral pada ujung pengiriman dan ujung penerimaan.



Gambar 2.2.
Rangkaian Setara Saluran Transmisi Pendek [2]

Karena tidak ada cabang pararel (shunt), arus pada ujung ujung pengiriman dan penerimaan akan sama besar :

$I_S = I_R$ .....(2.1.)

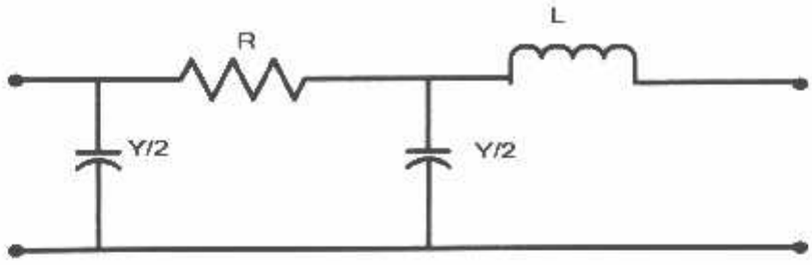
Bila kondisi tegangan pada ujung penerima diketahui, maka tegangan pada ujung sisi kirim adalah :

$V_S = V_R + I$ ....(2.2.)

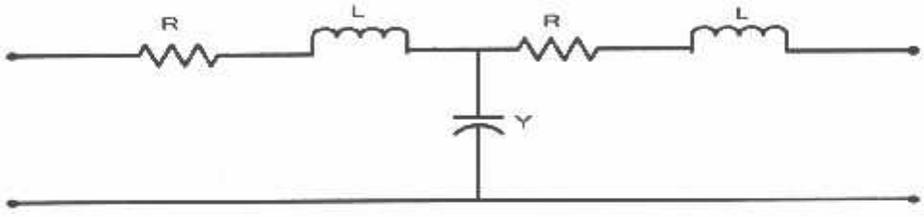
2.2.2. Saluran Transmisi Menengah

Bertambahnya saluran menyebabkan kapasitansi shunt bertambah besar dan tidak dapat diabaikan. Saluran transmisi jarak menengah pada umumnya digambarkan dengan rangkaian π pada atau rangkaian T, dapat dilihat pada

gambar 2.3(a) dan gambar 2.3(b). Dari dua versi ini rangkaian π lebih umum dipakai dari pada rangkaian T.



(a) Rangakaian Ekivalen π



(b) Rangkaian Ekivalen T

Gambar 2.3.
Rangkaian Setara Saluran Transmisi Menengah[2]

Untuk rangkaian π berlaku :

$$V_s = \left[\frac{ZY}{2}+1\right]V_R + ZI_R \quad \text{.....(2.3.)}$$

$$I_S = \left[\frac{ZY}{4}+1\right]YV_R + \left[\frac{ZY}{2}+1\right]I_R \quad \text{.....(2.4.)}$$

Untuk rangkaian T berlaku :

$$V_S = \left[\frac{ZY}{2}+1\right]V_R + \left[\frac{ZY}{4}+1\right]ZI_R \quad \text{.....(2.5.)}$$

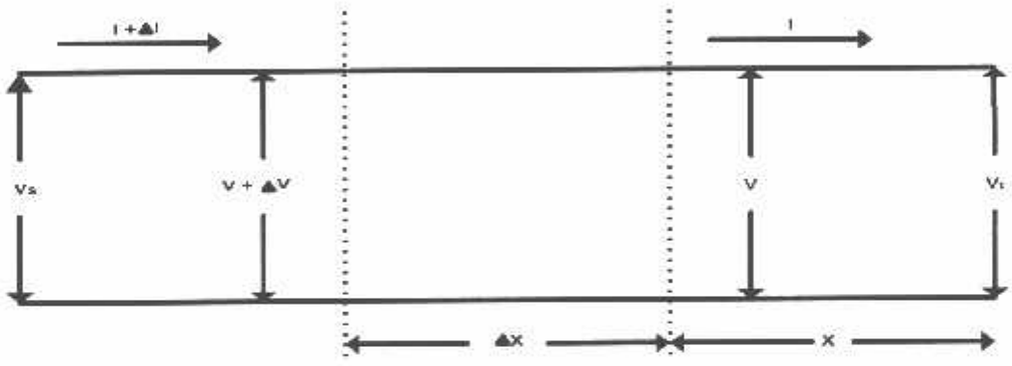
$$I_S = YV_R + \left[\frac{ZY}{2}+1\right]I_R \quad \text{.....(2.6.)}$$

Dimana :

V_S dan I_S	=	Tegangan dan Arus sisi kirim
V_R dan I_R	=	Tegangan dan Arus sisi terima
Z	=	Impedansi seri total saluran transmisi
Y	=	Admitansi shunt total saluran

2.2.3. Saluran Transmisi Panjang

Pada saluran panjang parameter–parameter saluran tidak terpusat menjadi satu, melainkan tersebar merata diseluruh panjang saluran, dapat dilihat pada gambar 2.4.



Gambar 2.4.

Diagram Skema Saluran Transmisi Panjang[2]

Persamaan tegang dan arus pada setiap titik sepanjang saluran transmisi dengan jarak x dari ujung sisi terima dapat ditulis sebagai berikut :

$$V = \frac{V_R - I_R Z_C}{2} e^{yx} + \frac{V_R + I_R Z_C}{2} e^{-yx} \quad \text{.....(2.7.)}$$

$$I = \frac{\frac{V_R}{Z_C} + I_R}{2} e^{yx} + \frac{\frac{V_R}{Z_C} - I_R}{2} e^{-yx} \quad \text{.....(2.8.)}$$

Persamaan untuk saluran transmisi panjang dapat ditulis dalam bentuk hiperbola sebagai berikut :

$V_{(x)} = V_R \cosh \tau x + I_R Z_c \sinh \tau x$ .....(2.9.)

$I_{(x)} = I_R \cosh \tau x + \frac{V_R}{Z_c} \sinh \tau x$ .....(2.10.)

Dimana :

τ = Konstanta rambatan pada saluran $= \sqrt{ZY}$

Z_c = Impedansi karakteristik saluran $= \sqrt{\frac{Z}{Y}}$

2.3. Daya Dalam Sistem Tenaga Listrik

Dalam sistem tenaga listrik, pembangkit–pembangkit tenaga listrik harus mampu menyediakan tenaga listrik sesuai dengan permintaan beban listrik yang ada. Hal yang harus diperhatikan adalah kondisi sistem yang tetap konstan, dalam hal ini tegangan dan frekwensi harus tetap konstan karena berhubungan dengan daya. Daya listrik yanga dibangkitkan dikenal dengan istilah:

2.3.1 Daya aktif (*Active power*)

Secara umum daya aktif dinyatakan oleh persamaan :

$P = [\ V\ ]\ [\ I\ ] \cos \varphi$ .....(2.11.)

Dimana :

V dan I nilai efektifnya.

P adalah daya rata–rata yang disebut juga daya aktif.

2.3.2 Daya Reaktif (*Reactive power*)

Daya reaktif adalah daya yang timbul karena adanya pembentukan medan magnet pada beban–beban induktif (KVAR).

Persamaan daya reaktif adalah :

$$Q = [\,V\,]\,[\,I\,] \sin \varphi \qquad (2.12.)$$

Dua macam daya reaktif yaitu : daya reaktif induktif dan daya reaktif kapasitif, dimana keduanya memiliki tanda yang berlawanan. Daya reaktif kapasitif adalah daya yang dibutuhkan oleh kapasitor yang tidak menghasilkan kerja, tetapi tersimpan dalam bentuk energi magnetis atau energi kapasitif.

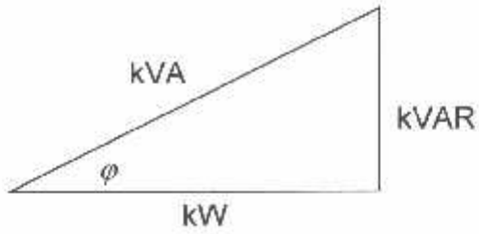
Daya reaktif induktif adalah daya listrik yang dibutuhkan untuk menghasilkan medan magnet yang dibutuhkan oleh alat–alat seperti motor induksi, transformator dan sebagainya.

2.3.3. Daya Semu (*Apparent power*)

Daya semu merupakan penjumlahan secara vektoris antara daya aktif dan daya reaktif.

Selain diatas dikenal juga istilah faktor daya (*power factor*) adalah perbandingan antara daya aktif dan daya semu, sehingga dapat dirumuskan pada persamaan berikut :

$$Cos\ \varphi = \frac{P}{S} = \frac{P}{\sqrt{P^2 + Q^2}} = \frac{kW}{kVA} \qquad (2.13.)$$



Gambar 2.5.
Representasi Segitiga Daya[6]

Dimana :

Cos φ = Faktor daya
KW = Daya aktif
KVA = Daya semu
KVAR = Daya reaktif

2.4. Sistem Per-Unit

Untuk memudahkan proses perhitungan, dalam sistem tenaga listrik digunakan sistem Per-Unit (pu).

$$\text{Besaran Per-Unit} = \frac{\text{Besaran sebenarnya}}{\text{Besaran dasar}} \quad \text{.....(2.14.)}$$

Rumus–rumus yang digunakan untuk penentuan arus dasar dan impedansi dasar adalah :

- Untuk data 1 phasa

Arus dasar

$$I_d = \frac{\text{kVA dasar 1 fasa}}{\text{kV dasar L - N}} \quad \text{.....(2.15.)}$$

Impedansi dasar

$$Z_d = \frac{(\text{kV dasar L - N})^2 \times 10^{-3}}{\text{kVA dasar 1 fasa}} \quad \text{.....(2.16.)}$$

$$Z_d = \frac{(\text{kV dasar L - N})^2}{\text{MVA dasar 1 fasa}} \quad \text{.....(2.17.)}$$

➢ Untuk data 3 fasa

Arus dasar

$$I_d = \frac{\text{kVA dasar 3 fasa}}{\sqrt{3}\ \text{kV dasar L - L}} \quad \text{.....(2.18.)}$$

Impedansi dasar

$$Z_d = \frac{(\text{kV dasar L - L})^2 \times 10^{-3}}{\text{kVA dasar3 fasa}} \quad \text{.....(2.19.)}$$

$$Z_d = \frac{(\text{kV dasar L - L})^2}{\text{MVA dasar 3 fasa}} \quad \text{.....(2.20.)}$$

Dalam persamaan di atas nilai–nilai besaran diberikan untuk rangkaian satu fasa. Jadi tegangannya adalah tegangan antara fasa dengan tanah dan daya setiap fasa.

Setelah besaran–besaran dasar telah ditentukan maka besaran– besaran itu dinormalisasikan terhadap besaran dasar. Dengan demikian impedansi per satuan dari suatu elemen rangkaian didefinisikan sebagai berikut :

$$Z = \frac{\text{Impedansi sebenarnya } Z\ (\Omega)}{\text{Impedansi dasar } Z_d} \quad \text{.....(2.21.)}$$

2.5 Mengubah Dasar Sistem Per-Unit

Kadang-kadang impedansi per-unit untuk suatu komponen dari suatu sistem dinyatakan menurut dasar yang berbeda dengan dasar yang dipilih untuk bagian dari sistem dimana komponen tersebut berada. Karena semua impedansi dalam bagian manapun dari suatu sistem harus dinyatakan dengan dasar

impedansi yang sama, maka dalam perhitungannya kita perlu mempunyai cara untuk dapat mengubah impedansi per-unit dari suatu dasar ke dasar yang lain. Dengan mensubstitusikan impedansi dasar yang diberikan dalam persamaan (2.17.) dan (2.20.) ke dalam persamaan (2.21.) maka diperoleh:

$$Z_u = \frac{(\text{Impedansi sebenarnya}, \Omega) \cdot (\text{MVA dasar})}{(\text{Tegangan dasar, kV})^2} \quad \text{.....(2.22.)}$$

Persamaan (2.22.) memperlihatkan bahwa impedansi per-unit berbanding lurus dengan MVA dasar serta berbanding terbalik dengan kuadrat tegangan dasar. Untuk mengubah dari impedansi per-unit menurut suatu dasar yang diberikan menjadi impedansi per-unit menurut suatu dasar yang baru, dapat dipakai persamaan berikut:

$$Z_{baru}\,\text{per-unit} = Z_{diberikan}\,\text{per-unit}\left(\frac{\text{kV}_{diberikan}\,\text{dasar}}{\text{kV}_{baru}\,\text{dasar}}\right)^2 \times \left(\frac{\text{MVA}_{baru}\,\text{dasar}}{\text{MVA}_{diberikan}\,\text{dasar}}\right) \quad \text{..... (2.23.)}$$

Persamaan ini tidak ada hubungannya dengan pemindahan nilai impedansi dalam ohm dari salah satu sisi transformator ke sisi yang lain.

2.6 Kapasitor Shunt

Kapasitor shunt adalah kapasitor yang dihubungkan secara paralel dengan saluran transmisi untuk kompensasi daya reaktif yang dilengkapi alat penghubung dan pemutus secara otomatis. Hal ini bertujuan agar kapasitor tersebut dapat terhubung dengan sistem pada saat – saat tertentu sesuai dengan besarnya kVar yang dibutuhkan, sehingga pada saat beban induktif pada titik terendah sistem tidak terjadi kelebihan daya reaktif kapasitif yang dihasilkan oleh kapasitor daya. Dengan memperhatikan besarnya variasi beban dlam sistem transmisi, maka

kapasitor digunakan untuk mengurangi jatuh tegangan di sepanjang saluran dan memperbaiki faktor daya.

2.7 Pengurangan Rugi – rugi Daya Dengan Kapasitor Shunt

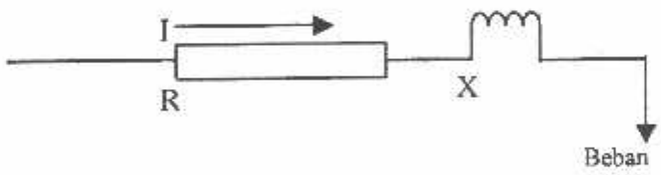
Rugi-rugi saluran per fasa dari saluran 3 fasa seimbang dengan beban terpusat seperti pada gambar 3. adalah I^2 (R-jX) atau dapat dibedakan menjadi :

- Rugi daya aktif (I^2R) = $(I_R{}^2 + I_X{}^2)$ R
- Rugi daya reaktif (I^2 X) = $(I_R{}^2 + I_X{}^2)$

dimana :

I_R = komponen arus aktif

I_X = komponen arus reaktif



Gambar 2.6
Saluran Primer Dengan Beban Terpusat[6]

Rugi-rugi saluran daya (I^2R) dapat dibagi menjadi dua komponen yaitu komponen arus aktif dan komponen arus reaktif. Rugi-rugi karena komponen arus aktif tidak akan mempengaruhi penempatan kapasitor *shunt* pada saluran, hal ini dapat dijelaskan sebagai berikut :

Diasumsikan bahwa rugi-rugi daya (I^2R) disebabkan oleh arus saluran I *(lagging)* yang mengalir pada resistansi R, sehingga :

$$I^2R=(I\cos\theta)^2R + (I \sin \theta)^2 R \qquad (2.24)$$

Setelah dipasang kapasitor shunt dengan arus Ic, didapat arus saluran baru I_1, dan rugi-rugi daya I_1^2 R sebagai berikut :

$$I_1^2R=(I\cos\theta)^2 R + (I \sin \theta - Ic)^2 R \quad \text{.....(2.25)}$$

Sehingga pengurangan rugi-rugi daya sebagai akibat pemasangan kapasitor didapat :

$$\Delta_{pls} = I^2 R - I_1^2 R$$

$$= (I \cos \theta)^2 R + (I \sin \theta)^2 R - (I \sin \theta - Ic)^2 R$$

$$=2(I\sin\theta)IcR - Ic^2 R \quad \text{.....(2.26)}$$

maka hanya komponen arus reaktif (I sin θ) saja yang berpengaruh terhadap pengurangan rugi daya I^2R akibat pemasangan kapasitor *shunt* pada saluran transmisi. Sehingga pengurangan rugi daya saluran 3 fasa adalah :

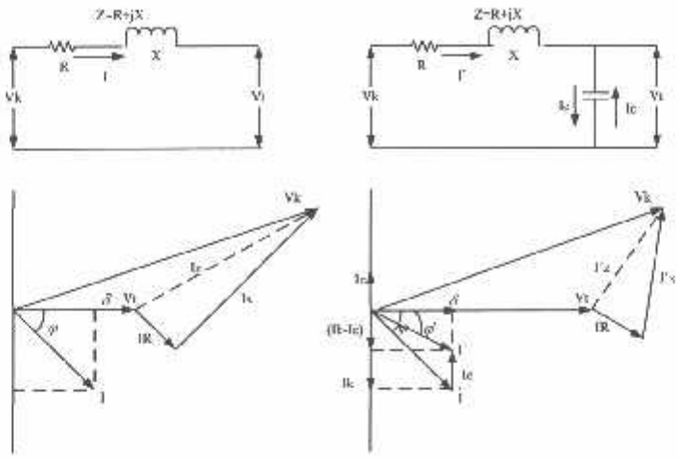
$$3R[2(I\sin\theta)Ic-Ic^2] \text{ Watt} \quad \text{.....(2.27)}$$

Bila suatu kapasitor ckVar dipasang pada beban, maka faktor daya menjadi :

$$Cos\theta_1 = \frac{kW}{KVA_1} \quad \text{.....(2.28)}$$

2.8. Perbaikan Tegangan

Pemakaian kapasitor shunt dalam sistem tenaga listrik selain untuk perbaikan faktor daya juga bertujuan untuk menaikkan tegangan. Dan vektor dapat dilihat pada gambar 2.7, sebagai berikut :



Gambar 2.7
Vektor Diagram sebelum (a) dan Sesudah(b) pemasangan kapasitor *shunt* pada jaringan[6]

Kerugian jaringan disebabkan arus beban I sebelum kapasitor dipasang :

$E=RI_r+XI_L$ .....(2.29)

Kerugian tegangan setelah kapasitor dipasang :

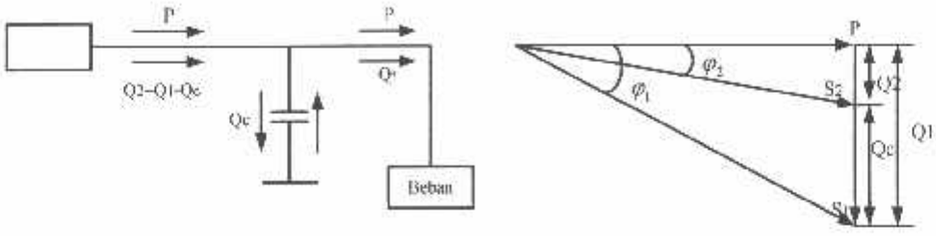
$E=RI_r+XI_L-X_{ic}$ .....(2.30)

2.9. Perbaikan Faktor Daya dan Kenaikan Kapasitas Sistem

Manfaat terbesar yang diperoleh perbaikan faktor daya berasal dari pengurangan daya reaktif dalam sistem. Hal ini menghasilkan pengurangan biaya pemakaian daya yang lebih rendah, kenaikan kapasitas sistem, perbaikan tegangan dan pengurangan rugi-rugi dalam sistem. Satu-satunya jalan untuk memperbaiki faktor daya adalah mengurangi daya reaktif di jaringan, jika komponen arus reaktif dapat dikurangi, maka total arus akan berkurang sedangkan komponen daya aktif tidak berubah, maka faktor daya akan lebih besar sebagai akibat

berkurangnya daya reaktif. Faktor daya akan mencapai 100% jika komponen daya reaktif sama dengan nol (0).

Dengan menambah kapasitor, daya reaktif Q akan berkuran, gambar 5 menunjukkan perbaikan faktor daya pada sistem, kapasitor mensuplai daya reaktif ke beban.



Gambar 2.8
Perbaikan Faktor Daya[6]

Diasumsikan bahwa beban disuplai oleh daya nyata P, daya reaktif *(lagging)* Q dan daya semu S, pada faktor daya tertinggal cos θ

$$Cos\,\theta = \frac{P}{S} \quad \text{.....(2.31)}$$

Bila suatu kapasitor Qc kVAR dipasang pada beban, faktor daya dapat diperbaiki dari Cos θ_1 menjadi cos θ_2 dimana :

$$Cos\,\theta = \frac{P}{S_1}$$

$$Cos\,\theta = \frac{P}{\sqrt{(P^2+Q^2)}}$$

$$Cos\,\theta = \frac{P}{\sqrt{P^2+(Q_1-Q_c)^2}} \quad \text{.....(2.32)}$$

Sehingga daya semu dan daya reaktif berkurang dari S_1 (kVA) ke S_2 (kVA) dari Q_1 (kVAR) Q_2 (kVAR) sehingga kapasitas beban akan meningkat. Dengan demikian dapat diambil kesimpulan bahwa presentase pengurangan rugi-rugi daya jaringan dapat dihitung menggunakan rumus berikut :

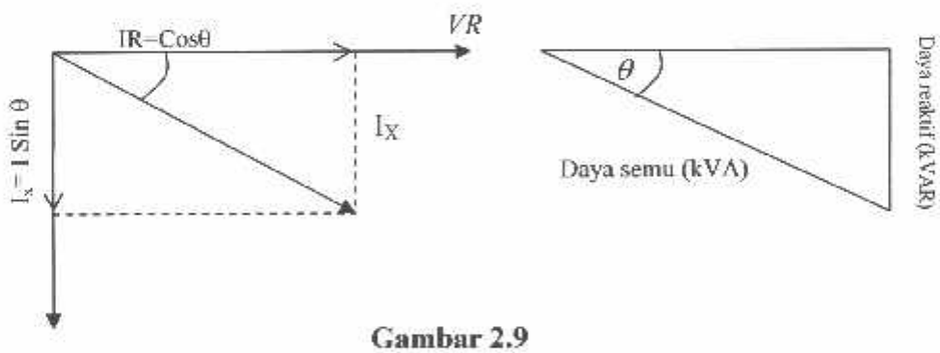
$$\%\text{Rugidaya}=100\left(\frac{Faktor\ daya\ mula-mula(\cos\theta_1)}{Faktor\ daya\ baru(\cos\theta_2)}\right)\ldots\ldots\ldots\ldots\ldots\ldots(2.33)$$

$$\%\ \text{Daya}=100\left(1-\left(\frac{Faktor\ daya\ mula-mula(\cos\theta_1)}{Faktor\ daya\ baru(\cos\theta_2)}\right)^2\right)\ldots\ldots\ldots\ldots\ldots(2.34)$$

2.10. Perhitungan Pengaruh Perbaikan Faktor Daya

Diagram phase dari dua komponen arus nyata itu arus aktif dan reaktif dapat dilihat pada gambar 6 berikut :

dilihat pada gambar 2-11 berikut;



Gambar 2.9
Diagram Fasor dan Sudut Daya Beban Transmisi [6]

Penjumlahan secara vektor dari arus aktif dan reaktif menghasilkan arus total yang dapat dinyatakan dengan persamaan :

$$I = \text{arus semu} = \sqrt{(arus\ aktif)^2+(arus\ reaktif)^2}$$

$$=\sqrt{(I\,Cos\,\theta)^2+(I\,Sin\theta)^2}\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.35)$$

pada suatu tegangan V daya aktif, daya reaktif dan daya nyata adalah sebanding dengan arus, dimana hubungannya dapat dinyatakan sebagai berikut :

$$\text{Daya semu } (kVA) = \sqrt{(Daya\,aktif)^2 + (Datya\,\mathrm{Re}\,aktif)^2}$$

$$(VI) = \sqrt{(VI\cos\theta)^2 + (VI\sin\theta)^2} \quad \text{.....(2.36)}$$

$$\text{Faktor Daya} = \frac{Daya\,aktif}{Daya\,semu} = \frac{kW}{kVA}$$

Daya Aktif = Daya semu * Faktor daya

kW = kVA * Faktor daya

kW = kVA Cos θ

2.11. Penentuan Rating Kapasitor untuk Perbaikan Faktor Daya Beban

Dari hubungan fasor diagram daya aktif dan daya reaktif dapat ditulis beberapa persamaan matematis sebagai berikut :

$$Cos\,\theta = \frac{daya\,aktif}{Daya\,semu} = \frac{(kW)}{(kVA)} \quad \text{.....(2.37)}$$

$$Sin\,\theta = \frac{Daya\,reaktif}{Daya\,semu} = \frac{(kVAR)}{(kVA)} \quad \text{.....(2.38)}$$

$$Tan\,\theta = \frac{Daya\,reaktif}{Daya\,aktif} = \frac{(kVAR)}{(kW)} \quad \text{.....(2.39)}$$

Karena komponen daya aktif biasanya konstan, dan daya semu serta kompnen daya reaktif berubah sesuai dengan faktor daya, maka persamaan ini dapat ditulis sebagai berikut :

Daya reaktif pada daya mula-mula = Daya aktif x tan θ_1

= (kW) x Tan θ_1

Daya reaktif pada faktor daya baru = Daya aktif x tan θ_2

= (kW) x Tan θ_2

Dengan θ_1 = sudut dari faktor daya mula-mula

θ_2 = sudut dari faktor daya yang telah diperbaiki

Rating kapasitor yang dibutuhkan perbaikan faktor daya :

Daya reaktif (kVAR) = Daya aktif x (tan θ_1 - θ_2)

(kVAR) = (kW) x (tan θ_1 - θ_2)

Untuk penyederhanaan (tan θ_1 - θ_2) sering ditulis Δ tan, yang merupakan suatu faktor pengali untuk menentukan daya reaktif.

Daya reaktif (kVAR) = Daya aktif x Δ tan

(kVAR) = (kW) x Δ tan

BAB III

ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN *HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION (HPSOM)*

3.1. Analisa Aliran Daya

Dalam melayani beban yang dibutuhkan oleh konsumen dan pengoperasian tenaga listrik perlu menganalisa aliran daya, sehingga sistem yang dioperasikan dapat memenuhi persyaratan teknis maupun ekonomisnya. Dalam analisa aliran daya dilakukan perhitungan terhadap tegangan, arus, daya aktif dan reaktif yang terdapat dalam berbagai titik dalam jala–jala jaringan transmisi tenaga listrik.

Tujuan dari analisa aliran daya adalah:

1. Mencari harga magnitude tegangan $|v|$ dan sudut fasa tegangan δ bus beban.
2. Mencari besar daya reaktif Q dan sudut fasa tegangan δ dari generator bus.
3. Untuk mengetahui besar daya aktif dan daya reaktif pada bus slack.
4. Untuk mengetahui apakah semua peralatan pada sistem memenuhi batas-batas yang telah ditetapkan untuk operasi penyaluran daya.
5. Untuk mengetahui kondisi awal pada perencanaan sistem yang baru.
6. Untuk menentukan besar daya yang mengalir disetiap saluran jaringan tenaga listrik.

3.1.1. Klasifikasi Bus

Pada setiap bus dari jaringan terdapat parameter–parameter yaitu daya aktif (P), daya reaktif (Q), besar tegangan $|v|$ dan sudut fasa tegangan δ .

Dengan melihat parameter diatas setiap bus dapat diklasifikasikan menjadi 3 bagian:

1. Bus Beban (PQ) atau *Load Bus*

Pada bus ini hanya terdapat kebutuhan daya untuk beban dimana P daya aktif dan Q daya reaktif diketahui, sementara $|v|$ dan δ berubah–ubah sesuai dengan kebutuhan. Oleh karena itu $|v|$ dan δ harus dicari.

2. Bus Generator (PV)

Pada bus ini hanya terdapat daya pembangkitan dimana $|v|$ diatur menggunakan regulator tegangan dan P diatur dengan governor. Sehingga untuk bus ini P dan $|v|$ diketahui sementara itu Q (daya reaktif) dan δ (sudut fasa) dicari.

3. Bus Slack

Pada bus ini $|v|$ dan δ sudah ditentukan besarnya sementara P dan Q dihitung. Biasanya nilai $|v|$ adalah 1 pu, sedangkan sudut fasa tegangan δ berharga nol, karena itu fasor tegangan dari bus dipakai sebagai referensi.

Daya total yang mengalir pada setiap bus dituliskan sebagai berikut :

$$S_k=P_k+jQ_k=V_k^{*}I_k \qquad (3.1.)$$

Atau

$$P_k jQ_k = V_k^* \sum_{n=1}^{N} Y_{kn} V_n \quad \text{.....(.3.2.)}$$

Dari persamaan $V_k = v_k + j\delta_k$ dan $Y_{k11} = G_{k11} - jB_{k11}$, maka persamaan 3.2. menjadi :

$$P_k jQ_k (v_k + j\delta_k) \sum_{n=1}^{N} (G_{k11} - jB_{k11})(V_n + j\delta_n) \quad \text{.....(3.3)}$$

Bila dituliskan dalam bentuk real dan imajiner maka persamaan di atas menjadi :

$$P_k = \sum_{n=1}^{N} \{v_k (v_k G_{kn} + \delta_k B_{kn}) + \delta_k (\delta_{kn} G_{kn} - v_k B_{kn})\} \quad \text{.....(3.4.)}$$

$$Q_k = \sum_{n=1}^{N} \{\delta_k (v_n G_{kn} + \delta_n B_{kn}) - v_k (\delta_n G_{kn} - v_n B_{kn})\} \quad \text{.....(3.5.)}$$

3.1.2 Metode Newton Rapshon

Proses yang dilakukan adalah membandingkan antara daya yang ditempatkan berdasarkan data ($P_{k,\ sched}$ dan $Q_{k,\ sched}$) dengan daya hasil perhitungan ($P_{k,\ calc}$ dan $Q_{k,\ calc}$). Selisih daya yang diterapkan dan perhitungan (ΔP_k dan ΔQ_k) dihitung dengan persamaan :

$$\Delta P_k = P_{k,\ sched} - P_{k,\ calc} \quad \text{.....(3.6.)}$$

$$\Delta Q_k = Q_{k,\ sched} - Q_{k,\ cal} \quad \text{.....(3.7.)}$$

Selisih daya diatas digunakan untuk menghitung nilai perubahan parameter tegangan bus, yaitu $\Delta |V_k|$ dan $\Delta\delta_k$, dengan menggunakan elemen jacobian, sehingga koreksi terhadap nilai parameter tegangan yang telah ditetapkan nilai awal sebelumnya. Elemen Jacobian sendiri merupakan turunan parsial P dan Q terhadap masing-masing variabel pada persamaan (3.4.) dan (3.5.), yang dalam bentuk matriks dituliskan sebagai berikut:

$$\begin{bmatrix} \Delta P_1 \\ \cdots \\ \Delta p_{n-1} \\ \Delta Q_1 \\ \cdots \\ \Delta Q_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{\partial P_1}{\partial v_1} & \cdots & \frac{\partial P_1}{\partial v_{n-1}} & \frac{\partial P_1}{\partial \delta_1} & \cdots & \frac{\partial P_1}{\partial \delta_{n-1}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial P_{n-1}}{\partial v_1} & \cdots & \frac{\partial P_{n-1}}{\partial v_{n-1}} & \frac{\partial P_{n-1}}{\partial \delta_1} & \cdots & \frac{\partial P_{n-1}}{\partial \delta_{n-1}} \\ \frac{\partial Q_1}{\partial v_1} & \cdots & \frac{\partial Q_1}{\partial v_{n-1}} & \frac{\partial Q_1}{\partial \delta_1} & \cdots & \frac{\partial Q_1}{\partial \delta_{n-1}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial Q_{n-1}}{\partial v_1} & \cdots & \frac{\partial Q_{n-1}}{\partial v_{n-1}} & \frac{\partial Q_{n-1}}{\partial \delta_1} & \cdots & \frac{\partial Q_{n-1}}{\partial \delta_{n-1}} \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \cdots \\ \Delta v_{n-1} \\ \Delta \delta_1 \\ \cdots \\ \Delta \delta_{n-1} \end{bmatrix}$$

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

.....(3.8.)

Dimana elemen-elemen jacobian dapat dihitung dengan menurunkan persamaan- persamaan (3.4.) dan 3.5.).

Perubahan nilai tegangan bus dijumlahkan dengan nilai tegangan bus sebelumnya, yang kemudian nilai tegangan bus terbaru ini digunakan untuk menghitung kembali daya $P_{k,\ calc}$ dan $Q_{k,\ calc}$ menurut persamaan 3.4.) dan (3.5.). Proses ini berulang, yang disebut iterasi hingga mencapai kondisi dimana nilai perubahan daya ΔP dan ΔQ konvergen mencapai suatu nilai minimum yang telah ditentukan (berkisar 0,001 hingga 0,0001 pu).

3.2. Hybrid Particle Swarn Optimizer With Mutation

3.2.1. Teori Dasar Particle Swarn Optimizer (PSO)[2]

Pada dasarnya *Hybrid Particle Swarn Optimizer With Mutation* (HPSOM) adalah merupakan pengembangan dari metode PSO. Sebelum memasuki pembahasan tentang *Hybrid Particle Swarn Optimizer With Mutation* (HPSOM) ada baiknya kita mereview kembali tentang PSO terlebih dahulu. PSO adalah suatu metode yang diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995, yang metode ini di ilhami dari perilaku sosial dari koloni burung. Dalam PSO terdapat teori bahwa proses adaptasi terdiri dari tiga prinsip utama: mengevaluasi , membandingkan dan meniru. Jadi suatu partikel dalam PSO akan mengevaluasi sekitarnya dan membandingkan dirinya dengan yang lain yang ada dalam koloni tersebut dan kemudian meniru dari sekitarnya yang paling kuat.

3.2.2. Parameter PSO

Beberapa parameter yang digunakan dalam PSO antara lain :

1. Partikel *(X)*

   Merupakan kandidat solusi yang direpresentasikan dengan vektor m-dimensional, dimana waktu m adalah jumlah parameter yang dioptimalkan. Dalam fungsi waktu *t*, partikel ke-*i* dapat dirumuskan sebagai : $X_i(t) = [X_{i1}(t), X_{i2}(t), ..., X_{in}(t)]$ dimana Xs adalah parameter yang dioptimalkan dan $X_{ik}(t)$ adalah posisi dari partikel ke-*i* yang berhubungan dengan dimensi *k*.

2. Populasi , *pop* (*t*)

   Merupakan set dari n buah partikel pada waktu t , seperti :

   $pop(t) = [X_1(t), X_2(t), ..., X_n(t)$ .....(3.9)

3. Swarm

   Populasi dari pergerakan partikel yang bergerak bersama secara berdekatan dimana setiap partikel bergerak dalam arah yang acak

4. Kecepatan Partikel *(Particle Velocity)*

   Merupakan kecepatan pergerakan yang direpresentasiakan dengan vektor m-dimensional . Dalam fungsi waktu t, kecepatan partikel ($V_i(t)$) ke-*i* dapat dituliskan sebagai $V_i(t) = [V_{i1}(t), V_{i2}(t), ...V_{in}(t)]$ dimana $V_{ik}(t)$ adalah kecepatan komponen dari partikel ke-*i* yang berhubungan dengan dimensi *k*. Tingkat kecepatan diperbaharui dengan ketetapan secara terpisah untuk tiap dimensi $j \in 1,...,n$, karena itu V_{ij} menotasikan dimensi *j* dari vektor kecepatan yang berhubungan dengan pertikel ke-*i*. Persamaan kecepatannya menjadi :

   $V_{ij}(t+1) = x(V_{ij}(t) + \varphi_1 rand(Y_{ij}(t)) + \varphi_2 rand(G_1(t) - X_{ij}(t))$ .....(3.10)

   Dari definisi kecepatan tersebut , φ_2 mengatur nilai step maksimum partikel pada *global best*, sedangkan φ_1 mengatur ukuran step pertikel pada *personal best*. Nilai V_{ij} diantara range $[-V_{max}, V_{max}]$ untuk mengurangi kemungkinan partikel meninggalkan daerah pencarian . posisi dari tiap partikel diperbaharui dengan menggunakan vektor kecepatan

yang baru sehingga menjadi :

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad \text{.....(3.11)}$$

5. *Personal Best (P-best)*

$$Y_i(t+1) =$$

$$\begin{matrix} Y_i(t) if F_T(X_i(t+1) \geq F_T * (Y_i(t)) \\ X_i(t+1) if F_T * (X_i(t+1)) < F_T * (Y_i(t)) \end{matrix} \quad \text{.....(3.12)}$$

6. *Global Best (G-best)*

G-best menawarkan satu tingkat lebih cepat atas pemusatan biaya ketahanan. G-best hanya menawarkan satu "*bestsolution*", yang dinamakan *partikel global best*. Partikel ini hanya bertindak sebagai suatu penarik , menarik semua partikel kearahnya. Semua partikel akan konvergen pada posisi tersebut dengan cepat jadi jika tidak diperbaharui secara reguler sekumpulan partikel tersebut akan konvergen secara prematur

7. *Contriction Factor*

Faktor ini mungkin membantu dalam mencapai konvergen. Faktor konstriktion pada persamaan kecepatan yang diperbaharui yang dipresentasikan dengan χ :

$$\chi = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad \text{.....(3.13)}$$

dimana :

$\varphi = \varphi_1 + \varphi_2$

$\varphi > 4$ .....(3.14)

$\varphi_1 = \varphi_2 = 2.05$

$\varphi = \varphi_1 + \varphi_2 = 4.1$

8. *Stopping Criteria (Kriteria Berhenti)*

   Ini adalah dimana proses pencarian akan berhenti. Pencarian akan berhenti jika satu dari kriteria yang diberikan sangat memuaskan , yaitu :

   a. jumlah dari interasi sejak pergantian terakhir dari best solution lebih besar dari jumlah sebelumnya .

   b. jumlah interasi sudah mencapai nilai yang maksimum.

3.2.3. Hybrid Particle Swarn Optimizer With Mutation (HPSOM)

Dalam PSO akan di peroleh *G – Best* (solusi terbaik) , aspek ini sangat penting dalam kecepatan untuk memperoleh hasil akhir. Jika suatu pertikel mempunyai posisi yang baik yakni sama dengan posisi *G –Best,* partikel hanya akan bergerak dari titik ini jika *inertia weigh* dan nilai percepatan sebelumnya adalah nol. Jika nilai percepatan sebelumnya sangat dekat dengan nol , kemudian semua partikel akan berhenti bergerak ketika mereka menyamakan dengan partikel *G-Best.* Yang hal ini mungkin akan membawa nilai konvergen dari algoritma yang terlalu awal (prematur). Pada kenyataannya , hal ini tidak memberikan jaminan bahwa algoritma telah konvergen pada nilai daerah minimal – hal ini tidak hanya berarti bahwa semua partikel telah konvergen pada nilai yang telah dihasilkan oleh PSO. Hal ini dikenal sebagai stagnasi. Solusi yang diusulkan berdasarkan pada penambahan parameter dan

penambahan rumus. Solusi yang lain adalah dengan menggunakan a *breeding and subpopulation*. Untuk mengatasi masalah ini dalam tulisan ini akan diperkenalkan sebuah metode baru yang dikenal dengan *HPSOM (Hybrid Particle Swarm Optimizer with Mutation)* dengan menggunakan proses mutasi yang sering digunakan pada GA kedalam PSO

Proses ini akan mengikuti proses pencarian dari daerah optimal dan mencari di daerah yang lain. Proses ini berawal dari pemilihan secara acak patrikel dalam swarm dan bergerak dalam posisi yang berbeda dalam daerah pencarian. Dalam tulisan ini proses mutasi akan berdasarkan rumus berikut :

$$mut(p[k]) = p([k]x - 1) + \omega \quad \text{.....(3.15)}$$

Dimana :

- $p[k]$ adalah pemilihan secara acak pertikel dari swarm
- ω adalah nilai secara acak yang diperoleh dari batasan [0,01 x ($x_{max} - x_{min}$)] yang menampilkan panjang daerah pencarian

3.2.4. Mutation (Mutasi)

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai partikel dalam satu individu. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana k = 1,2,.., *NVAR* (jumlah partikel). Probabilitas mutasi *(Pm)* ditentukan dan digunakan untuk mengendalikan frekuensi operator mutasi. Apabila nilai random r_k, P_m maka gen ke-k kromosom tersebut terpilih untuk mengalami mutasi. Proses

mutasi sama dengan *Evolutionary Strategies* yaitu menggunakan operator *Gaussian mutation*, dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak Gaussian untuk menciptakan partikel baru (*offspring*)

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

3.3.Formulasi Permasalahan

Mengurangi kerugian dalam optimalisasi aliran daya dapat di formulasikan sebagai berikut :

Minimize $f(x,\sigma)$
Subject to $g(x,\sigma) = 0$
Dan $h(x) < 0$, $\quad x_l < x < x_u$

Dimana :

- f_x adalah fungsi skalar dan jumlah kerugian dalam suatu daerah pengamatan.
- g_x adalah batasan persamaan fungsional dan persamaan aliran daya.
- h_x adalah batas ketidaksamaan fungsional dan batas variabel kendali

- x adalah variabel tetap vektor yang terdiri dari kedua kontrol dan bergantung pada variabel-variabel tersebut (sudut dan magnitude tegangan, suseptansi shunt dan daya aktif)

Proses meminimalisasi kerugian terdiri atas optimasi objek function, membuat batasan pada persamaan aliran daya, batas aliran cabang , batas tegangan bus dan batas variabel control.

BAB IV

HASIL DAN ANALISA HASIL

4.1. Program Komputer *Analisa penempatan Kapasitor Dengan Menggunakan Hybrid Particle Swarm Optimazer With Mutation (HPSOM)* Pada Saluran Transmisi 150 kV Sub Sistem Paiton - Bali

Dalam penyelesaian masalah ini maka diperlukan bantuan program komputer dalam perhitungan yang membutuhkan ketelitian dan keakuratan.

Program komputer dalam skripsi ini dijalankan dengan menggunakan bahasa pemrograman Borland Delphi versi 7.0 dan diaplikasikan pada komputer Pentium 4. dengan Prosesor 2.4 GHz, dengan Memori 256 Mb

4.2. Algoritma Program

4.2.1. Algoritma Aliran Daya Newton Raphson

1. Bentuk matriks admitansi bus :[Y_{bus}]
2. Penetapan harga awal tegangan dan sudut fasa untuk semua bus kecuali bus slack [$V_i(0)$, $\delta(0)$]
3. Menentukan nomor iterasi awal, k=0
4. Menghitung injeksi daya aktif dan reaktif pada setiap bus dengan persamaan kecuali bus slack
5. Menghitung selisih daya yang dijadwalkan dengan injeksi daya bus dari perhitungan
6. Menentukan perubahan maksimum pada daya aktif dan daya reaktif

7. Membandingkan apakah selisih daya sudah sama atau lebih kecil dari ε
8. Jika "ya" hitung daya aktif dan reaktif, tegangan dan sudut fasa tegangan pada setiap bus, serta aliran daya pada saluran dan perhitungan selesai, jika "tidak" lanjutkan ke langkah berikutnya
9. Membentuk elemen matriks Jacobian
10. Menghitung faktor koreksi tegangan dan sudut fasa setiap bus kecuali bus slack dan bus generator
11. Menghitung nilai tegangan dan sudut fasa yang baru
12. Mengganti nilai sudut fasa yang lama dengan sudut fasa yang baru, tegangan yang lama dengan tegangan yang baru
13. Perhitungan dilanjutkan ke langkah 4 dengan nilai iterasi yang baru sampai hasil yang didapatkan konvergen.

4.2.2. Algoritma Pemecahan Masalah

1. Memasukkan data saluran dan data pembebanan
2. Menghitung loadflow dengan menggunakan metode NewtonRaphson.
3. Memeriksa apakah tegangan variabel sudah berada pada batas-batas yang telah ditentukan.
4. Jika memenuhi batasan di atas **(Ya)** , hitung semua variabel dan rugi daya dan perhitungan selesai.
5. Jika tidak memenuhi (**Tidak**), maka di lanjutkan dengan menggunakan *Hybrid Particle Swarn Optimizer With Mutation* (HPSOM)

4.2.3. Algoritma Program *Analisa Penempatan Kapasitor Dengan Menggunakan Hybrid Particle Swarm Optimazer With Mutation (HPSOM)*

1. Memasukkan data beban yang meliputi data saluran R dan X , dan data pembebanan P dan Q.
2. Baca data parameter PSO
3. Inisialisasi Partikel
4. Periksa, apakah partikel baru lebih besar dari partikel sebelumnya
5. Jika **Ya**, ganti nilai partikel sebelumya dengan nilai partikel baru
6. Jika **Tidak**, periksa apakah partikel baru lebih besar dari partikel best
7. Jika **Ya** , ganti nilai partikel best dengan nilai partikel baru.
8. Jika **Tidak** , hitung kecepatan maksimum dengan persamaan :

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)[y_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,j}(t)[y_i(t) - x_{i,j}(t)$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

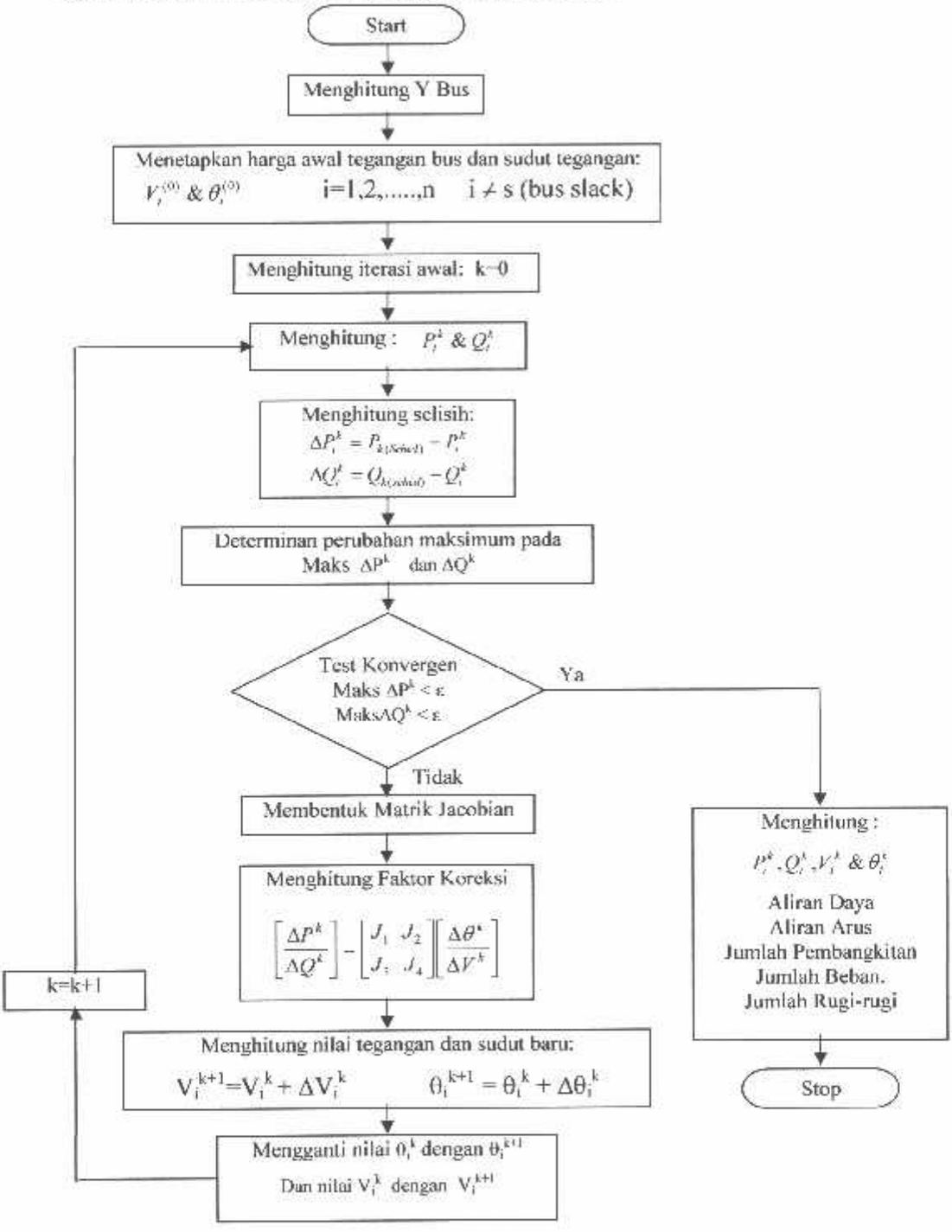
9. Menentukan posisi dengan persamaan :

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iterasi_{\max}} .iter$$

10. Melakukan proses mutasi
11. Periksa, Apakah iterasi sudah maksimal
12. Jika **Ya**, cetak hasil dan proses telah selesai
13. Jika **Tidak**, tambah interasi dengan satu dan kembali ke-langkah no 4 .

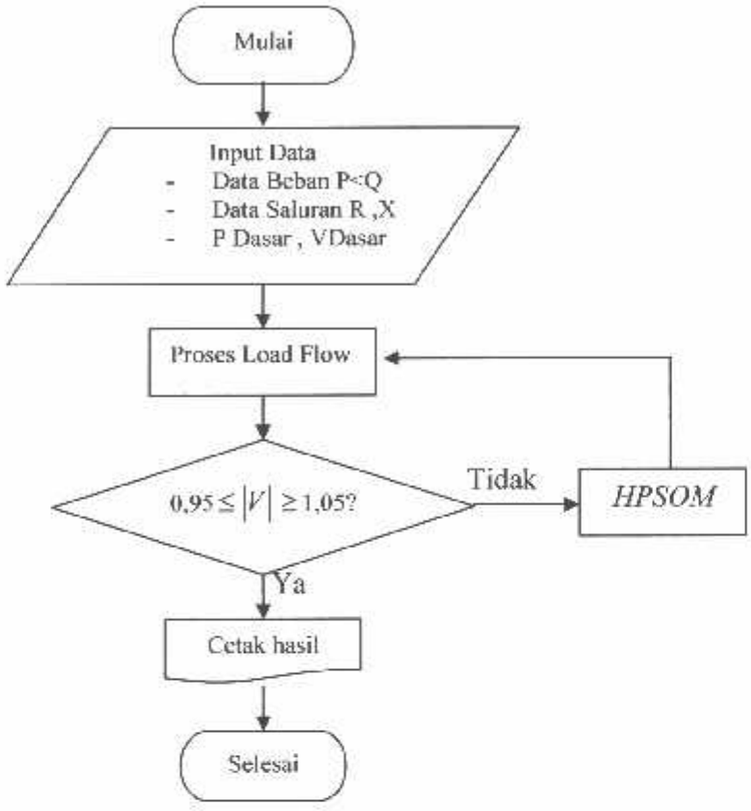
4.3. Diagram Alir

4.3.1. Diagram Alir Aliran Daya Newton Raphson



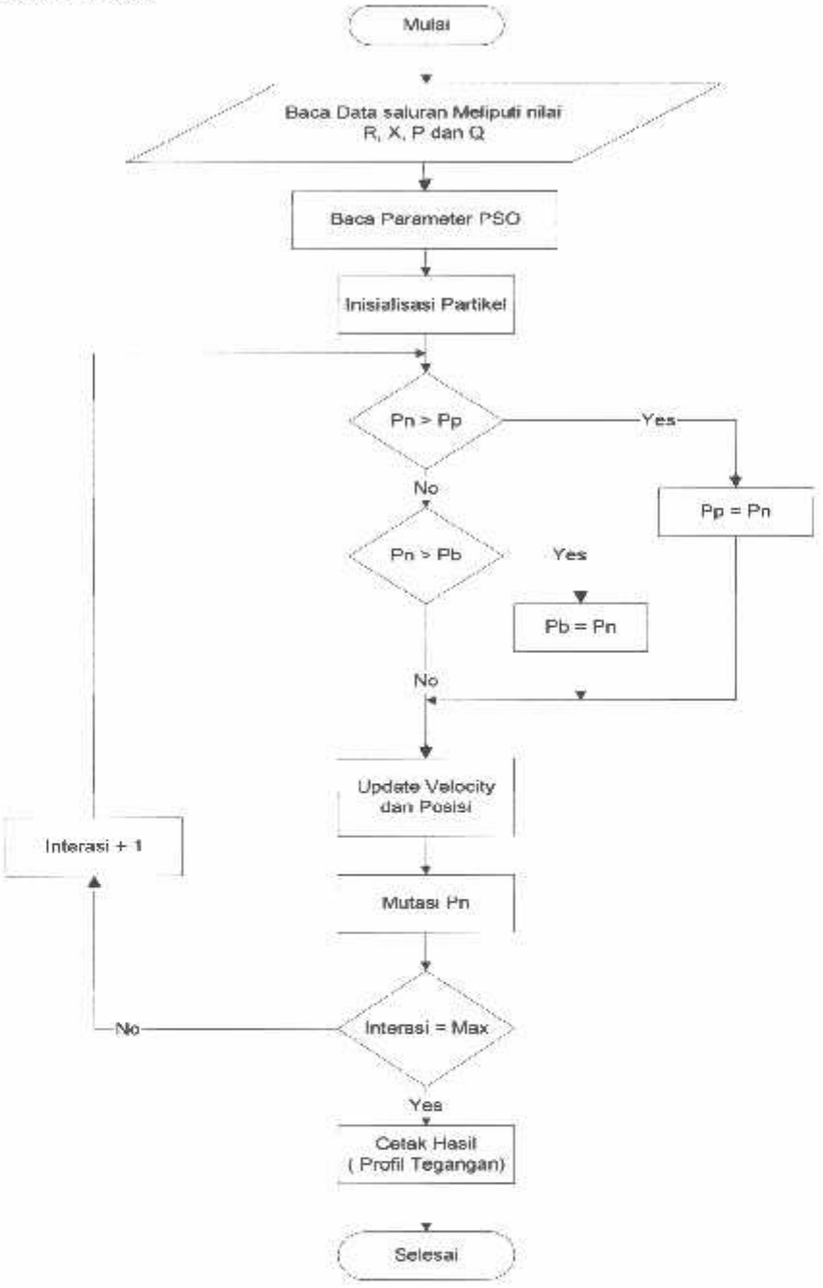
Gambar 4.1.
Flowchart Perhitungan Aliran Daya Metode *Newton Rapshon*

4.3.2.Diagram Alir Pemecahan Masalah



Gambar 4.2
Flowchart Perhitungan Program Optimasi

4.3.2. Diagram Alir *Analisa Penempatan Kapasitor Dengan Metode Particle Swarm Optimazer With Mutation* Pada Saluran Transmisi 150 kV sub Sistem Paiton-Bali



Gambar 4.3
Flowchart Program Analisa Penempatan Kapasitor Menggunakan Metode Particle Swarm Optimazer With Mutation

4.4. Validasi Data IEEE 30

Validasi Data IEEE 30 untuk optimal power flow dengan menggunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)*

Tabel 4.1
Data Pembangkitan IEEE 30 bus

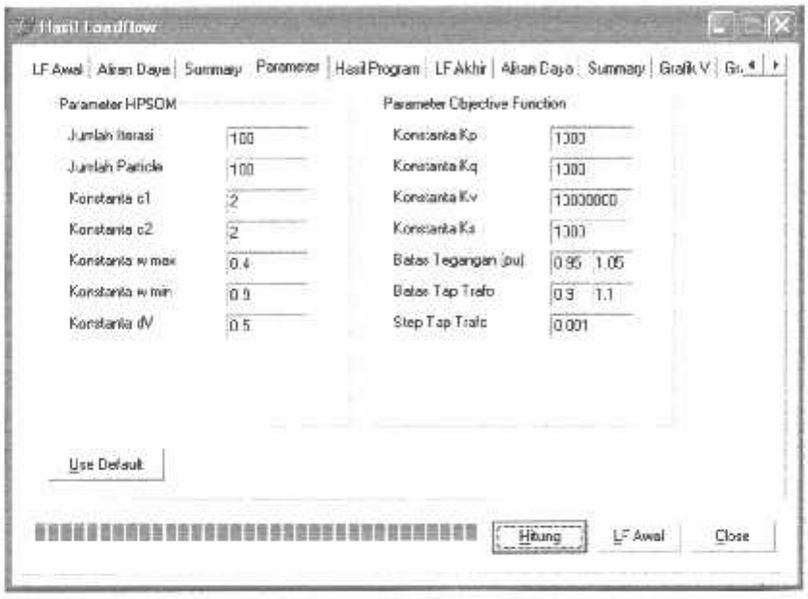
No. Bus	Tegangan		Pembangkitan		Pembebanan	
	Magnitude (pu)	Sudut (deg)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1,05	0	0	0	0	0
2	1,0338	0	57,56	0	27,7	12,7
3	1	0	0	0	2,4	1,2
4	1	0	0	0	7,6	1,6
5	1,058	0	24,56	0	94,2	19
6	1	0	0	0	0	0
7	1	0	0	0	22,8	10,9
8	1,023	0	35	0	30	30
9	1	0	0	0	0	0
10	1	0	0	0	5,8	2
11	1,0913	0	17,93	0	0	0
12	1	0	0	0	11,2	7,5
13	1,0883	0	16,91	0	0	0
14	1	0	0	0	6,2	1,6
15	1	0	0	0	8,2	2,5
16	1	0	0	0	3,5	1,8
17	1	0	0	0	9	5,8
18	1	0	0	0	3,2	0,9
19	1	0	0	0	9,5	3,4
20	1	0	0	0	2,2	0,7
21	1	0	0	0	17,5	11,2
22	1	0	0	0	0	0
23	1	0	0	0	3,2	1,6
24	1	0	0	0	8,7	6,7
25	1	0	0	0	0	0
26	1	0	0	0	3,5	2,3
27	1	0	0	0	0	0
28	1	0	0	0	0	0
29	1	0	0	0	2,4	0,9
30	1	0	0	0	10,6	1,9

Tabel 4.2.
Data Saluran Impedansi IEEE 30

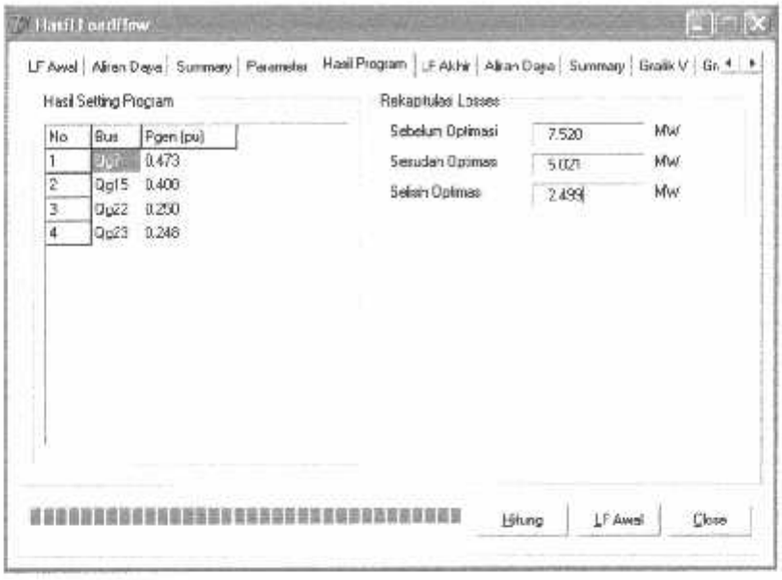
No. Saluran	Hubungan Bus	R (pu)	X (pu)	B (pu)
1	1-2	0,0192	0,0575	0,0264
2	1-3	0,0452	0,1852	0,0204
3	2-4	0,057	0,1737	0,0184
4	3-4	0,0132	0,0379	0,0042
5	3-5	0,0472	0,1983	0,0209
6	2-6	0,0581	0,1763	0,0187
7	4-6	0,0119	0,0414	0,0045
8	5-7	0,046	0,0102	0,0102
9	6-7	0,0267	0,082	0,0085
10	6-8	0,012	0,042	0,0045
11	6-9	0,000	0,208	0,0000
12	6-10	0.000	0,556	0,0000
13	9-11	0,000	0,208	0,0000
14	9-10	0,000	0,11	0,0000
15	4-12	0,000	0,256	0,0000
16	12-13	0,000	0,14	0,0000
17	12-14	0,1231	0,2559	0,0000
18	12-15	0,0602	0,1304	0,0000
19	12-16	0,0945	0,1987	0,0000
20	14-15	0,221	0,1997	0,0000
21	16-17	0,0824	0,1932	0,0000
22	15-18	0,107	0,2185	0,0000
23	18-19	0,0639	0,1292	0,0000
24	19-20	0,034	0,068	0,0000
25	10-20	0,0936	0,209	0,0000
26	10-17	0,0324	0,0845	0,0000
27	10-21	0,0348	0,0749	0,0000
28	10-22	0,0727	0,1499	0,0000
29	21-22	0,0116	0,0236	0,0000
30	15-23	0,1	0,202	0,0000
31	22-24	0,115	0,179	0,0000
32	23-24	0,132	0,27	0,0000
33	24-25	0,1885	0,3292	0,0000
34	25-26	0,2544	0,38	0,0000
35	25-27	0,1093	0,2087	0,0000
36	28-27	0	0,396	0,0000
37	27-29	0,2198	0,4153	0,0000

38	27-30	0,3202	0,6027	0,0000
39	29-30	0,2399	0,4533	0,0000
40	8-28	0,0636	0,2	0,0210
41	6-28	0,0169	0,0599	0,0065

4.4.1.Hasil validasi data IEEE 30 dengan menggunakan *Hybrid Particle Swarm Optimizer With Mutation (HPSOM)*



Gambar 4.4
Tampilan Parameter Validasi



Gambar 4.5
Tampilan Hasil Validasi

Tabel 4.3
perbandingan data jurnal dengan hasil validasi

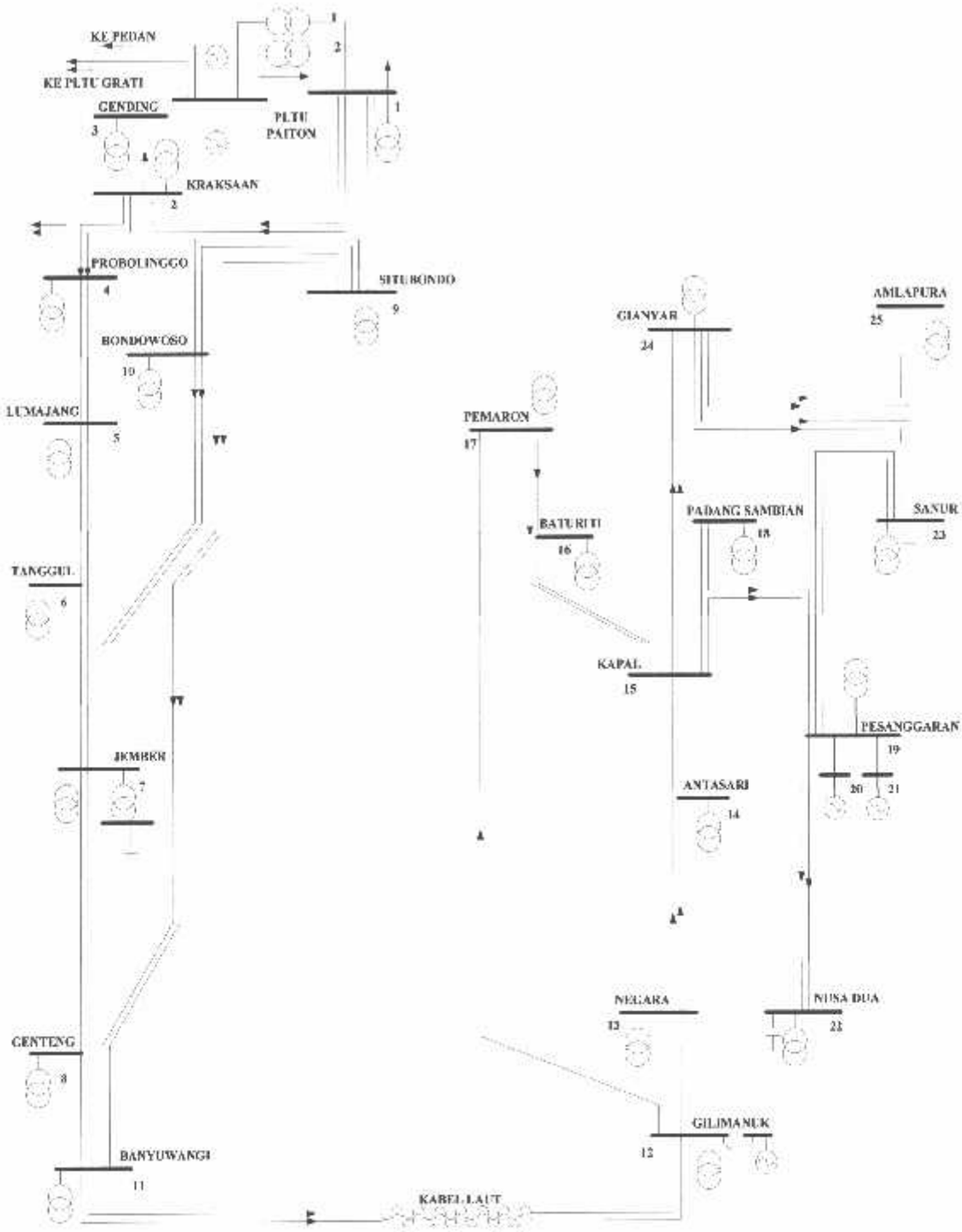
Data Jurnal	Hasil Pengujian
Rugi Daya	Rugi Daya
5.025	5.021

$$\%error = \frac{hasil\ jurnal - hasil\ pengujian}{hasil\ jurnal} x100\%$$

$$\%error = \frac{5.025 - 5.021}{5.025} x100\%$$

$$\%error = 0{,}0796\%$$

Dari proses validasi di atas maka didapatkan *error* sebesar 0,0796 %.



Gambar 4.6
Diagram single line sub sistem paiton Bali
Sumber : PT. PLN (Persero) P3B Jawa-Bali

4.5. Data Pembangkitan dan Pembebanan 150 kV Sub Sistem Paiton-Bali

Berdasarkan data pada referensi [4] maka dilakukan pengolahan data untuk mengubah dasar tegangan ke dalam pu. Serta perhitungan P_{load} dan Q_{load}.

Pada bus nomor 1, maka:

Tegangan Dasar = 150 kV

Tegangan Sebenarnya = 155 kV

$$\text{Tegangan (pu)} = \frac{\text{Tegangan sebenarnya}}{\text{Tegangan dasar}}$$

$$\text{Tegangan (pu)} = \frac{155}{150} = 1{,}033 \text{ pu}$$

P_{beban} = 11,7+ 27,6 = 39,30 MW

Q_{beban} = 7,9 MVAR

Selanjutnya dilakukan perhitungan P_{load} dan Q_{load} , sehingga didapatkan hasil yang ditunjukkan pada tabel 4.4.

Tabel 4.4
Data Pembangkitan dan Pembebanan 150 kV Sub Sistem Paiton dan Bali

No.	Nama Bus	Tegangan (pu)	Pgenerator (MW)	Qgenerator (MVAR)	Pload (MW)	Qload (MVAR)	Tipe
1	PAITON	1,033	-	-	39.3	7.9	1
2	KRAKSAN	-	-	-	14	6	3
3	GENDING	-	-	-	18	8.5	3
4	PROBOLINGGO	-	-	-	36.1	16.5	3
5	LUMAJANG	-	-	-	39	16.6	3
6	TANGGUL	-	-	-	22.5	9.7	3
7	JEMBER	-	-	50	62.2	30	3
8	GENTENG	-	-	-	41	21.3	3

9	SITUBONDO	-	-	-	23	5.9	3
10	BONDOWOSO	-	-	-	17.2	7	3
11	BANYUWANGI	-	-	-	37,8	18.7	3
12	GILIMANUK	0,992	99,9	-	5.4	2.7	2
13	NEGARA	-	-	-	11.9	4.4	3
14	ANTASARI	-	-	-	6.2	2.9	3
15	KAPAL	-	-	-	63.6	22.8	3
16	BATURITI	-	-	-	4.1	0.8	3
17	PEMARON	-	-	-	23.6	4.3	3
18	PDG SAMBIAN	-	-	-	16.4	7	3
19	PESANGGARAN	-	-	-	73.3	28,9	3
20	PLTD PSGRAN	0,95	32,5	-	0	0	2
21	PLTG PSGRAN	0,95	67,3	-	0	0	2
22	NUSADUA	-	-	25	45.5	13.6	3
23	SANUR	-	-	25	50.3	13.5	3
24	GIANYAR	-	-	-	35.4	7.9	3
25	AMLAPURA	-	-	-	14.2	3.3	3

Keterangan: 1 = bus slack
2 = bus generator
3 = bus beban

4.6. Data Saluran Transmisi 150 kV Sub Sistem Paiton-Bali

Sub sistem Paiton dan Bali terdiri dari 25 bus dan 34 saluran transmisi, dalam hal ini saluran transmisi yang dibahas adalah saluran transmisi 150 kV.

Berdasarkan data, maka dilakukan pengolahan data untuk saluran dengan jumlah sirkuit 2. Sebagai contoh pada hubungan saluran 1-2 dilakukan perhitungan sebagai berikut:

$$R = \frac{R_1 \times R_2}{R_1 + R_2} = \frac{0{,}0105 \times 0{,}0105}{0{,}0105 + 0{,}0105} = \frac{1{,}1025 \times 10^{-4}}{0{,}021} = 0{,}0053\,\text{pu}$$

$$X = \frac{X_1 \times X_2}{X_1 + X_2} = \frac{0{,}0359 \times 0{,}0359}{0{,}0359 + 0{,}0359} = \frac{1{,}2888 \times 10^{-3}}{0{,}0718} = 0{,}0179\,\text{pu}$$

$$B = \frac{B_1 \times B_2}{B_1 + B_2} = \frac{0{,}0130 \times 0{,}0130}{0{,}0130 + 0{,}0130} = \frac{1{,}69 \times 10^{-4}}{0{,}026} = 0{,}0065 \text{ pu}$$

Untuk selanjutnya dilakukan perhitungan data saluran, sehingga didapatkan hasil data yang ditunjukkan pada tabel 4.5.

Tabel 4.5
Data Saluran Transmisi 150 kV Sub Sistem Paiton-Bali

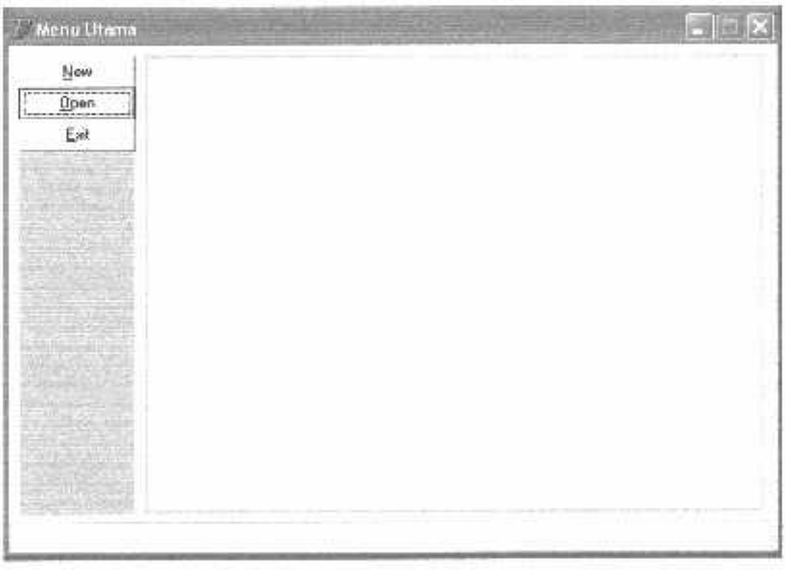
No.	Hubungan Bus	Jumlah Sirkuit	R (pu)	X (pu)	B (pu)
1	1 – 2	2	0,0053	0,0179	0,0065
2	1 – 9	2	0,0048	0,0346	0,0251
3	2 – 3	1	0,01	0,0340	0,0123
4	2 – 4	2	0,0079	0,0269	0,0097
5	4 – 5	2	0,0135	0,0460	0,0166
6	5 – 6	1	0,0167	0,0571	0,0206
7	5 – 7	1	0,0315	0,1077	0,0389
8	6 – 7	1	0,0148	0,0506	0,0183
9	7 – 8	1	0,0337	0,1151	0,0415
10	7 – 10	2	0,0104	0,0353	0,0128
11	7 – 11	1	0,0430	0,1469	0,0530
12	8 – 11	1	0,0177	0,0604	0,0218
13	9 – 10	2	0,0091	0,0310	0,0112
14	9 – 11	2	0,0966	0,0457	0,0340
15	11 – 12	2	0,0016	0,0035	-
16	12 – 13	2	0,0116	0,0034	0,0124
17	12 – 17	1	0,0399	0,1314	0,0502
18	13 – 14	1	0,0270	0,0783	0,0288
19	13 – 15	1	0,0412	0,1194	0,0439
20	14 – 15	1	0,0142	0,0411	0,0151
21	15 – 16	1	0,0362	0,0693	0,0240
22	15 – 17	1	0,0556	0,1054	0,0369
23	15 – 18	1	0,0105	0,0304	0,0112
24	15 – 19	1	0,0105	0,0304	0,0112
25	15 – 22	1	0,0293	0,0561	0,0195
26	15 – 24	2	0,0059	0,0170	0,0063
27	16 – 17	1	0,0194	0,0371	0,0129
28	18 – 19	1	0,0045	0,0130	0,0048

29	19 – 20	1	0,0020	0,0044	-
30	19 – 21	1	0,0020	0,0056	-
31	19 – 22	1	0,0127	0,0243	0,0084
32	19 – 23	2	0,0037	0,0070	0,0025
33	23 – 24	2	0,0078	0,0149	0,0052
34	24 – 25	2	0,0103	0,0298	0,0110

4.7. Prosedur pelaksanaan Program Perhitungan.

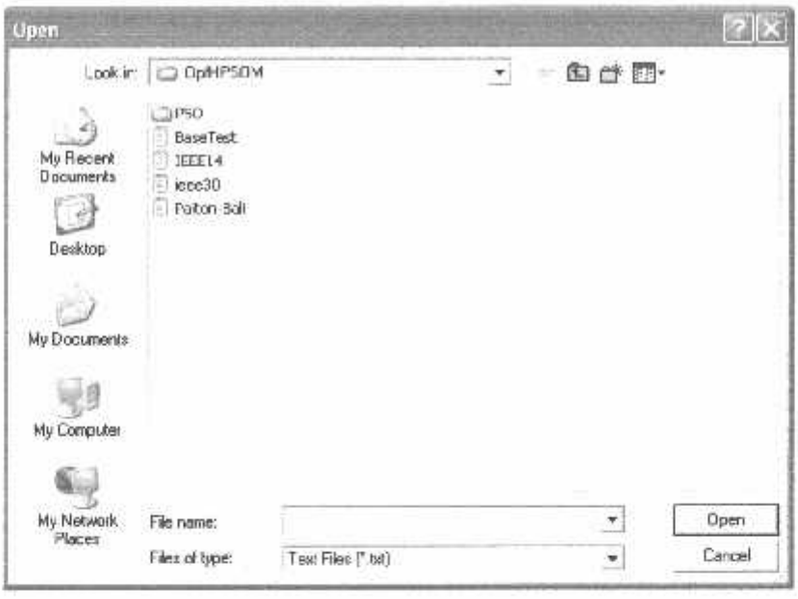
Prosedur menjalankan program perhitungan dilakukan setelah memasukkan seluruh data ke dalam program dengan menggunakan bahasa Pemrograman Borland Delphi versi 7.0. Mengenai prosedur jalannya program dapat dilakukan sebagai berikut:

1. Menu utama program, tekan tombol ***Open File*** untuk membuka data yang sudah tersimpan, setelah menekan tombol ***Open File*** maka akan keluar tambilan data seperti pada gambar berikut.



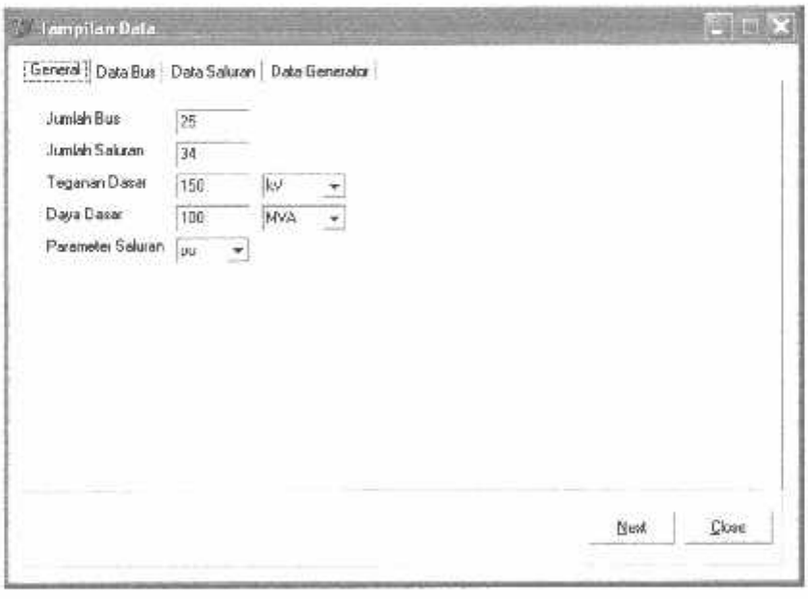
Gambar 4.7. Menu Utama Program

2. Kemudian membuka file dengan menekan ***open*** dan pilih file ***Paiton-Bali***.



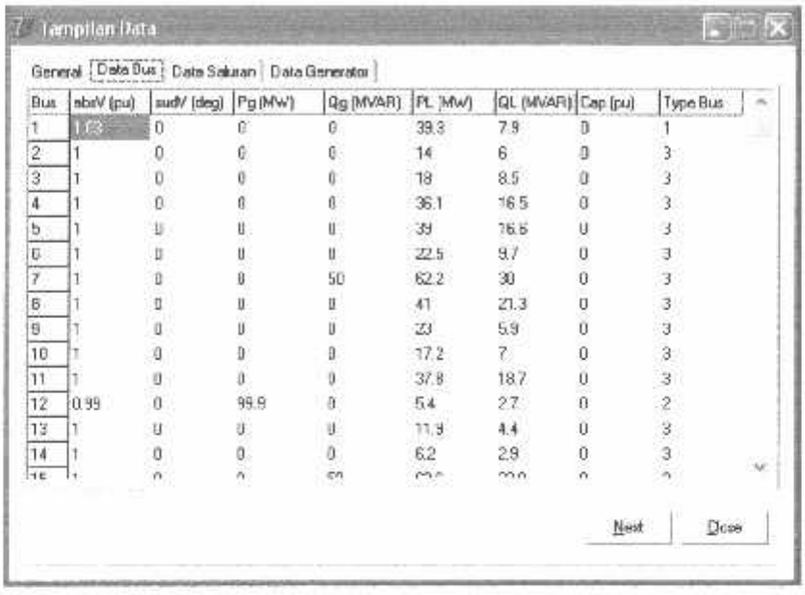
Gambar 4.8 Membuka File Optimasi

3. Maka kita akan masuk pada halaman tampilan data, yang pertama adalah tampilan ***Data General (Utama)***



Gambar 4.9 Tampilan data General (Umum)

4. Tampilan data yang kedua adalah ***Data Bus.***



Tampilan Data

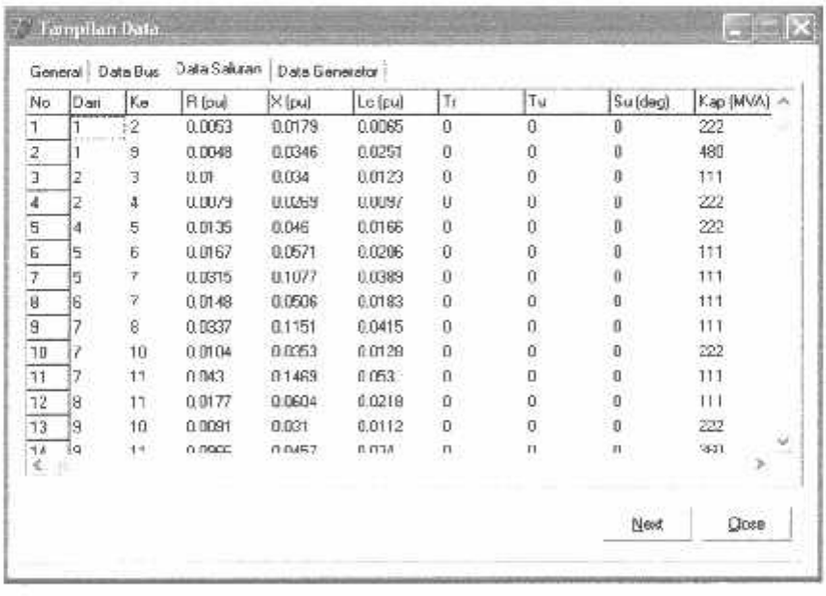
General | Data Bus | Data Saluran | Data Generator

Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Cap (pu)	Type Bus
1	1.03	0	0	0	39.3	7.9	0	1
2	1	0	0	0	14	6	0	3
3	1	0	0	0	18	8.5	0	3
4	1	0	0	0	36.1	16.5	0	3
5	1	0	0	0	39	16.5	0	3
6	1	0	0	0	22.5	9.7	0	3
7	1	0	0	50	62.2	30	0	3
8	1	0	0	0	41	21.3	0	3
9	1	0	0	0	23	5.9	0	3
10	1	0	0	0	17.2	7	0	3
11	1	0	0	0	37.8	18.7	0	3
12	0.99	0	99.9	0	5.4	2.7	0	2
13	1	0	0	0	11.9	4.4	0	3
14	1	0	0	0	6.2	2.9	0	3

Next | Close

Gambar 4.10 Tampilan Data Bus

5. Tampilan data yang ketiga adalah ***Data Saluran.***



Tampilan Data

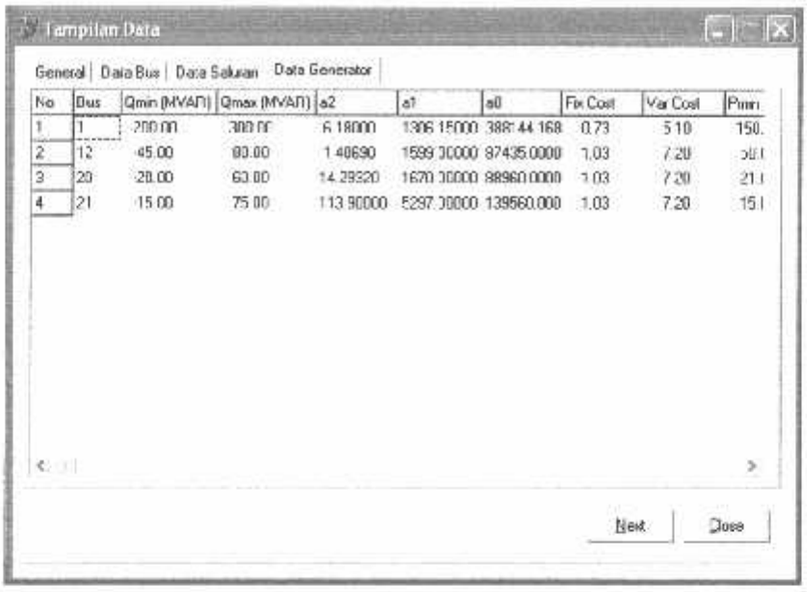
General | Data Bus | Data Saluran | Data Generator

No	Dari	Ke	R (pu)	X (pu)	Lc (pu)	Tr	Tu	Su (deg)	Kap (MVA)
1	1	2	0.0053	0.0179	0.0065	0	0	0	222
2	1	9	0.0048	0.0346	0.0251	0	0	0	480
3	2	3	0.01	0.034	0.0123	0	0	0	111
4	2	4	0.0079	0.0359	0.0097	0	0	0	222
5	4	5	0.0135	0.046	0.0166	0	0	0	222
6	5	6	0.0167	0.0571	0.0206	0	0	0	111
7	5	7	0.0315	0.1077	0.0389	0	0	0	111
8	6	7	0.0148	0.0506	0.0183	0	0	0	111
9	7	8	0.0337	0.1151	0.0415	0	0	0	111
10	7	10	0.0104	0.0353	0.0129	0	0	0	222
11	7	11	0.043	0.1469	0.053	0	0	0	111
12	8	11	0.0177	0.0604	0.0218	0	0	0	111
13	9	10	0.0091	0.031	0.0112	0	0	0	222

Next | Close

Gambar 4.11 Tampilan Data Saluran

6. Tampilan data yang keempat adalah ***Data Generator.***

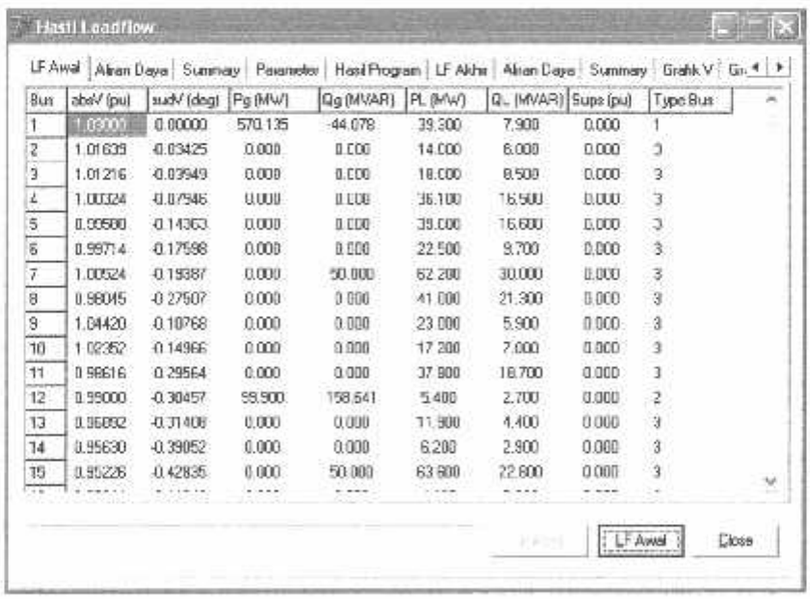


No	Bus	Qmin (MVAR)	Qmax (MVAR)	a2	a1	a0	Fix Cost	Var Cost	Pmin
1	1	200.00	300.00	6.18000	1306.15000	38844.168	0.73	5.10	150.
2	12	-45.00	80.00	1.40690	1599.30000	87435.0000	1.03	7.20	50.1
3	20	-28.00	60.00	14.29320	1670.30000	88960.0000	1.03	7.20	21.1
4	21	-15.00	75.00	113.90000	5297.30000	139660.000	1.03	7.20	15.1

Gambar 4.12 Tampilan Data Generator

7. kemudian tekan ***next*** dan tekan ***LFAwal*** untuk mengetahui hasil perhitungan

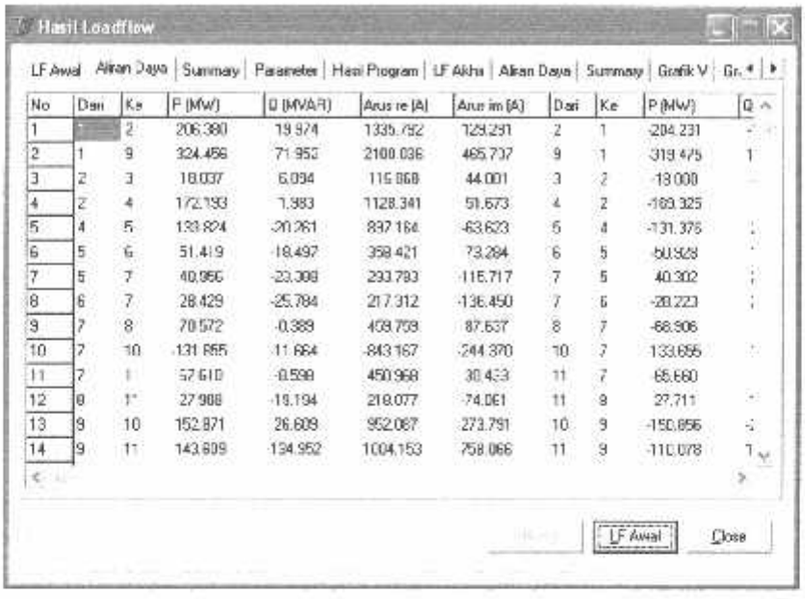
Load Flow Awal



Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Susp (pu)	Type Bus
1	1.03000	0.00000	570.135	-44.078	39.300	7.900	0.000	1
2	1.01639	-0.03425	0.000	0.000	14.000	6.000	0.000	3
3	1.01216	-0.03949	0.000	0.000	18.000	8.500	0.000	3
4	1.00324	-0.07546	0.000	0.000	36.100	16.500	0.000	3
5	0.99588	-0.14363	0.000	0.000	39.000	16.600	0.000	3
6	0.99714	-0.17598	0.000	0.000	22.500	9.700	0.000	3
7	1.00524	-0.19387	0.000	50.000	62.200	30.000	0.000	3
8	0.98045	-0.27507	0.000	0.000	41.000	21.300	0.000	3
9	1.04420	-0.10768	0.000	0.000	23.000	5.900	0.000	3
10	1.02352	-0.14966	0.000	0.000	17.200	7.000	0.000	3
11	0.98616	-0.29564	0.000	0.000	37.900	18.700	0.000	3
12	0.99000	-0.30457	59.900	158.641	5.400	2.700	0.000	2
13	0.96892	-0.31408	0.000	0.000	11.900	4.400	0.000	3
14	0.95630	-0.39052	0.000	0.000	6.200	2.900	0.000	3
15	0.95226	-0.42835	0.000	50.000	63.600	22.600	0.000	3

Gambar 4.13 Tampilan Loadflow Awal

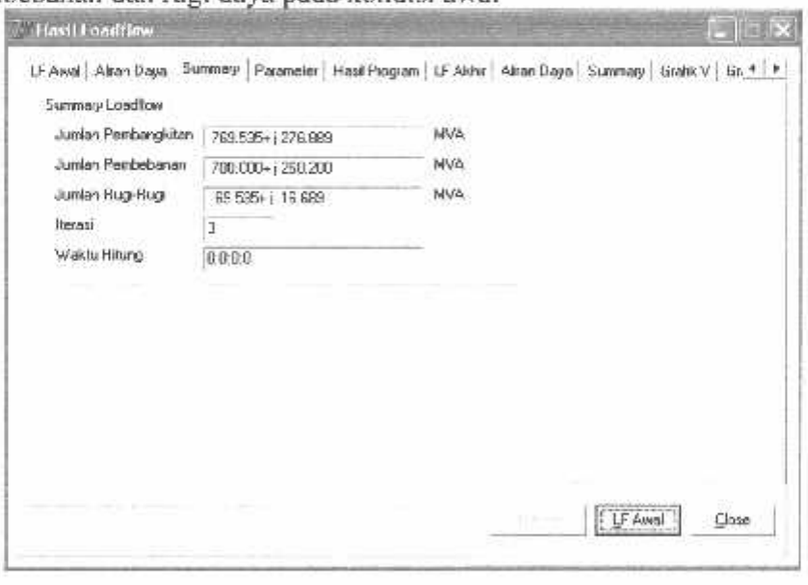
8. Tekan ***Aliran Daya*** untuk mengetahui nilai aliran daya dari perhitungan loadflow awal.



No	Dari	Ke	P (MW)	Q (MVAR)	Arus re (A)	Arus im (A)	Dari	Ke	P (MW)	Q
1	1	2	206.380	19.974	1335.792	729.291	2	1	-204.231	[illegible]
2	1	9	324.456	71.952	2100.036	465.737	9	1	-319.475	1
3	2	3	18.037	6.094	116.868	44.001	3	2	-18.000	[illegible]
4	2	4	172.193	1.383	1128.341	51.673	4	2	-169.325	[illegible]
5	4	5	133.824	-20.261	837.164	-63.623	5	4	-131.375	[illegible]
6	5	6	51.419	-18.497	358.421	73.284	6	5	-50.928	[illegible]
7	5	7	40.956	-23.308	293.793	-116.717	7	5	40.302	[illegible]
8	6	7	28.429	-25.784	217.312	-136.450	7	6	-28.223	[illegible]
9	7	8	70.572	-0.389	459.759	87.637	8	7	-68.906	[illegible]
10	7	10	-131.865	11.664	-843.167	-244.370	10	7	133.655	[illegible]
11	7	11	57.610	-0.598	450.968	30.433	11	7	-55.660	[illegible]
12	8	11	27.908	-19.194	218.077	-74.061	11	8	27.711	[illegible]
13	9	10	152.871	26.609	952.087	273.791	10	9	-150.856	[illegible]
14	9	11	143.609	134.952	1004.153	758.066	11	9	-110.078	1

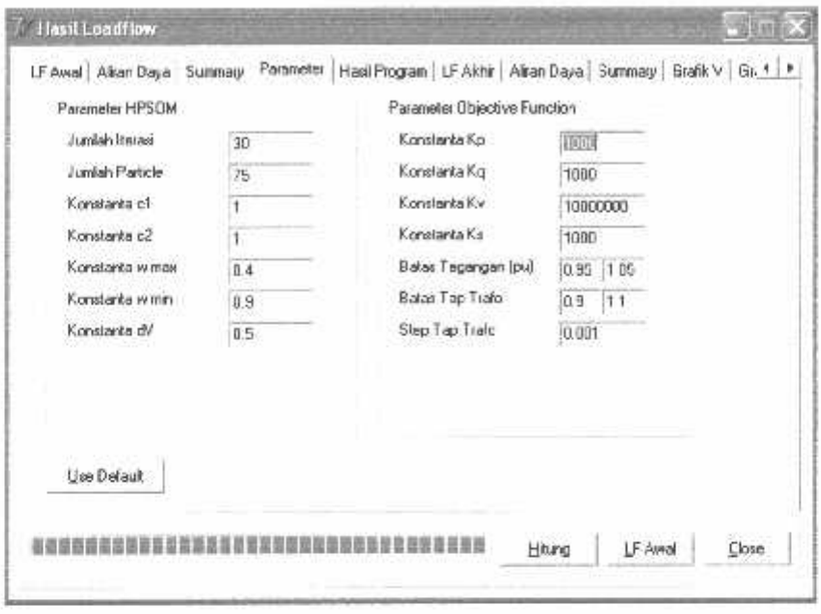
Gambar 4.14 Tampilan Aliran Daya Awal

9. Tekan tombol ***Summary*** untuk melihat hasil total pembangkitan, pembebanan dan rugi daya pada kondisi awal



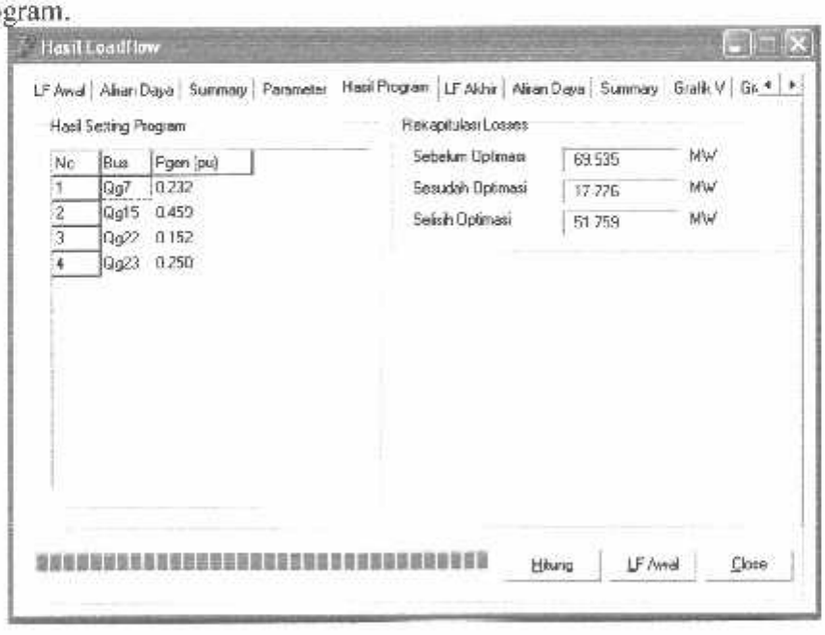
Gambar 4.15 Tampilan Summary Hasil Perhitungan Awal

10. Tekan tombol ***Parameter*** untuk mengetahui parameter yang digunakan dalam program, kemudian tekan ***Hitung***.



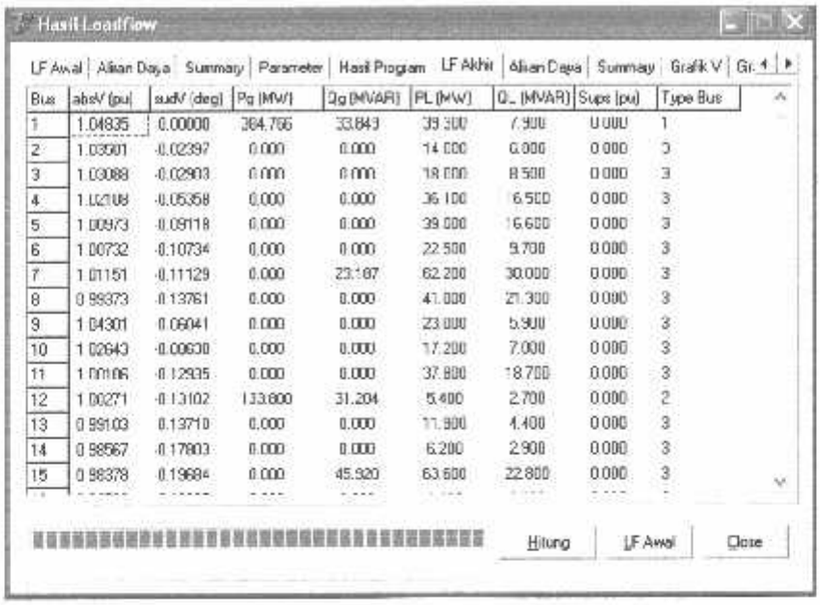
Gambar 4.16 Tampilan Parmeter Untuk Optimasi

11. Kemudian tekan ***Hasil Program*** untuk mengetahui hasil perhitungan program.



Gambar 4.17 Tampilan Hasil Perhitungan Program

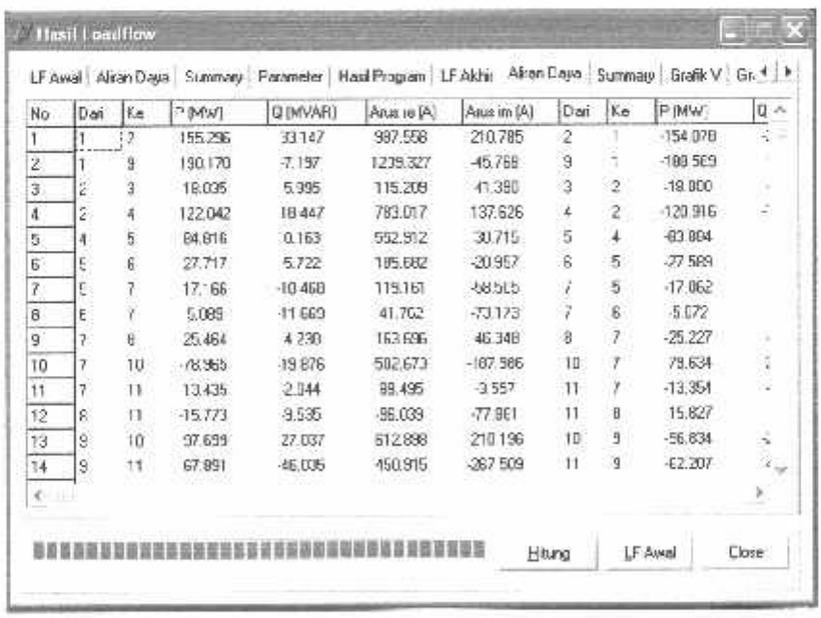
12. Tekan tombol ***LF Akhir*** untuk mengetahui nilai loadflow setelah optimasi



Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	Q_ (MVAR)	Sups (pu)	Type Bus
1	1.04835	0.00000	364.766	33.843	39.300	7.900	0.000	1
2	1.03501	-0.02397	0.000	0.000	14.000	0.000	0.000	3
3	1.03088	-0.02903	0.000	0.000	18.000	8.500	0.000	3
4	1.02108	-0.05358	0.000	0.000	36.100	6.500	0.000	3
5	1.00973	-0.09118	0.000	0.000	39.000	16.600	0.000	3
6	1.00732	-0.10734	0.000	0.000	22.500	9.700	0.000	3
7	1.01151	-0.11129	0.000	23.187	62.200	30.000	0.000	3
8	0.99373	-0.13761	0.000	0.000	41.000	21.300	0.000	3
9	1.04301	0.06041	0.000	0.000	23.000	5.900	0.000	3
10	1.02643	-0.09630	0.000	0.000	17.200	7.000	0.000	3
11	1.00106	-0.12935	0.000	0.000	37.800	18.700	0.000	3
12	1.00271	-0.13102	133.800	31.204	5.400	2.700	0.000	2
13	0.99103	0.13710	0.000	0.000	11.300	4.400	0.000	3
14	0.98667	-0.17803	0.000	0.000	6.200	2.900	0.000	3
15	0.98378	0.19684	0.000	45.920	63.600	22.800	0.000	3

Gambar 4.18 Tampilan Laod Flow Setelah Optimasi

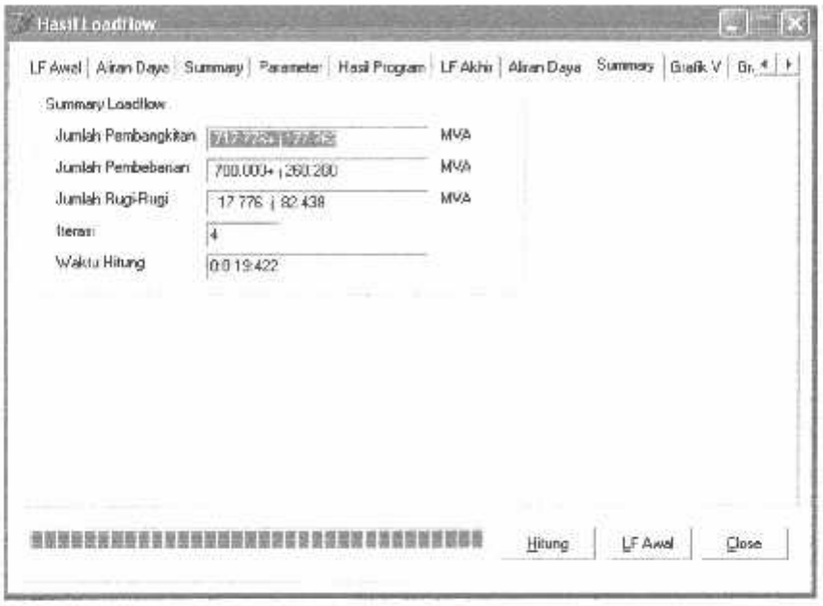
13. Tekan ***Aliran Daya*** untuk mengetahui nilai aliran daya setelah optimasi



No	Dari	Ke	P (MW)	Q (MVAR)	Arus re (A)	Arus im (A)	Dari	Ke	P (MW)
1	1	2	155.296	33.147	987.558	210.785	2	1	-154.078
2	1	9	190.170	-7.197	1239.327	-45.768	9	1	-189.569
3	2	3	18.035	5.395	115.209	41.380	3	2	-18.000
4	2	4	122.042	18.447	783.017	137.626	4	2	-120.916
5	4	5	84.816	0.163	552.912	-30.715	5	4	-83.804
6	5	6	27.717	5.722	185.682	-20.957	6	5	-27.589
7	5	7	17.166	-10.460	115.161	-38.515	7	5	-17.062
8	6	7	5.089	-11.669	41.702	-73.173	7	6	-5.072
9	7	8	25.464	4.230	163.696	46.348	8	7	-25.227
10	7	10	-78.965	-19.876	502.673	-187.586	10	7	79.634
11	7	11	13.435	2.044	89.495	-3.557	11	7	-13.354
12	8	11	-15.773	-9.535	-96.039	-77.861	11	8	15.827
13	9	10	97.699	27.037	612.898	210.196	10	9	-96.834
14	9	11	67.891	-46.035	450.915	-267.509	11	9	-62.207

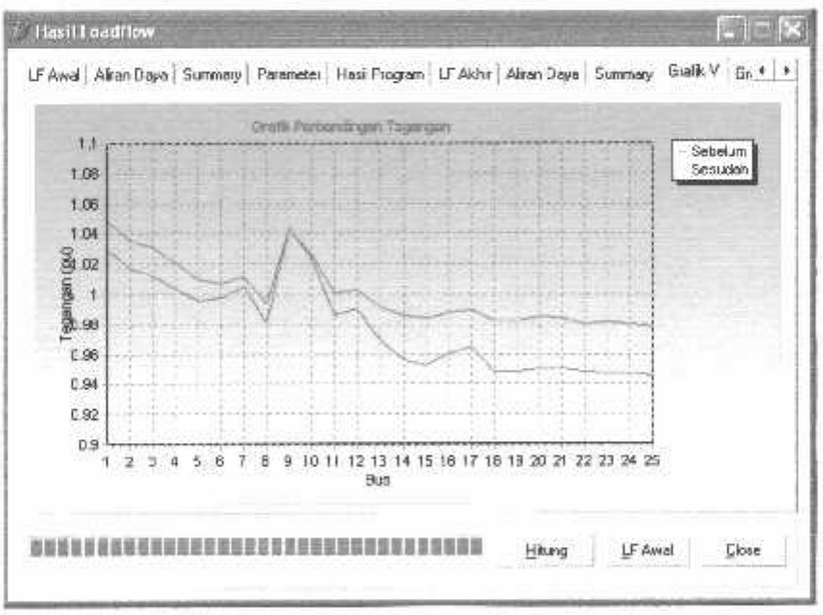
4.19 Gambar Aliran Daya Setelah Optimasi

14. Kemudian tekan tombol ***Summary***, untuk melihat hasil jumlah pembangkitan, pembebanan dan rugi daya pada kondisi akhir



Gambar 4.20 Tampilan Summary Setelah Optimasi

15. Kemudian tekan tombol ***Grafik V*** untuk melihat perbandingan tegangan pada tiap bus.



4.21 Tampilan Grafik Perbandingan Tegangan

4.8. Hasil Perhitungan *Analisa Penempatan Kapasitor Menggunakan Hybrid Particle Swarm Optimazer With Mutation (HPSOM)* Pada Saluran Transmisi 150 kV Sub Sistem Paiton – Bali.

4.8.1. Hasil Perhitungan sebelum Optimasi

Tabel 4.6
Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Sebelum Optimasi

Bus No.	Tegangan		Pembangkit		Pembebanan	
	Magnitude (pu)	Sudut (deg)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1.03333	0.00000	570.135	-44.078	39.300	7.900
2	1.01639	-0.03425	0.000	0.000	14.000	6.000
3	1.01216	-0.03949	0.000	0.000	18.000	8.500
4	1.00324	-0.07946	0.000	0.000	36.100	16.500
5	0.99580	-0.14363	0.000	0.000	39.000	16.600
6	0.99714	-0.17598	0.000	0.000	22.500	9.700
7	1.00524	-0.19387	0.000	50.000	62.200	30.000
8	0.98045	-0.27507	0.000	0.000	41.000	21.300
9	1.04420	-0.10768	0.000	0.000	23.000	5.900
10	1.02352	-0.14966	0.000	0.000	17.200	7.000
11	0.98616	-0.29564	0.000	0.000	37.800	18.700
12	0.99000	-0.30457	99.900	158.641	5.400	2.700
13	0.96892	-0.31408	0.000	0.000	11.900	4.400
14	0.95630	-0.39052	0.000	0.000	6.200	2.900
15	0.95226	-0.42835	0.000	50.000	63.600	22.800
16	0.96044	-0.41346	0.000	0.000	4.100	0.800
17	0.96476	-0.40340	0.000	0.000	23.600	4.300
18	0.94867	-0.43636	0.000	0.000	16.400	7.000
19	0.94868	-0.43771	0.000	0.000	73.300	28.900
20	0.95000	-0.43645	32.200	13.850	0.000	0.000
21	0.95000	-0.43350	67.300	-1.524	0.000	0.000
22	0.94794	-0.44481	0.000	25.000	45.500	13.600
23	0.94767	-0.44121	0.000	25.000	50.300	13.500
24	0.94779	-0.43924	0.000	0.000	35.400	7.900
25	0.94551	-0.44370	0.000	0.000	14.200	3.300

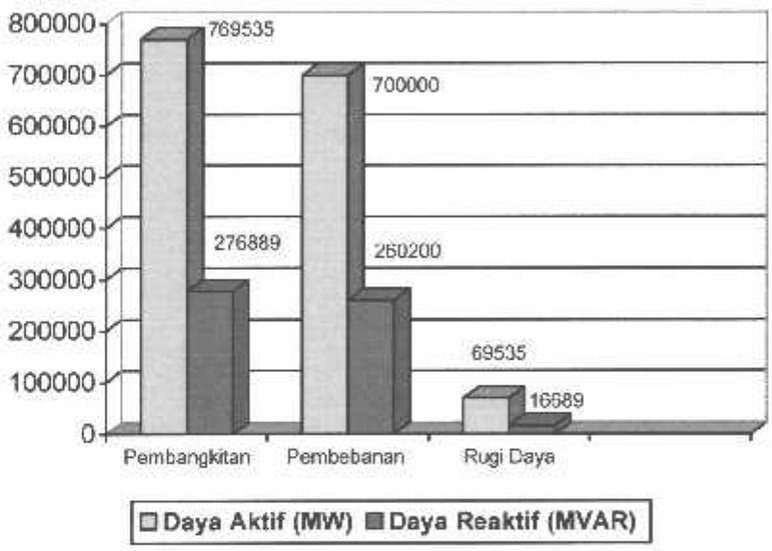
Tabel 4.7
Hasil Perhitungan Aliran Daya Antar Saluran Sebelum Optimasi

No.	Saluran		Daya	
	Dari	Ke	Aktif (MW)	Reaktif (MVAR)
1	1	2	206.380	19.974
2	1	9	324.456	-71.952
3	2	3	18.037	6.094
4	2	4	172.193	1.983
5	4	5	133.824	-20.261
6	5	6	51.419	-18.497
7	5	7	40.956	-23.388
8	6	7	28.429	-25.784
9	7	8	70.572	-0389
10	7	10	-131.855	-11.664
11	7	11	67.610	-8.598
12	8	11	27.908	-19.194
13	9	10	152.871	26.609
14	9	11	143.609	-134.952
15	11	12	165.658	-182.898
16	12	13	187.121	-24.949
17	12	17	71.936	-4.200
18	13	14	86.646	-13.504
19	13	15	84.460	-14.700
20	14	15	78.253	-17.426
21	15	16	-19.999	-2.834
22	15	17	-21.561	-2.997
23	15	18	24.777	1.767
24	15	19	28.354	0.516
25	15	22	23.907	-6.714
26	15	24	59.465	4.107
27	16	17	-24.259	0.450
28	18	19	8.305	-3.418
29	19	20	-32.173	-13.790
30	19	21	-67.200	1.805
31	19	22	21.863	-9.200
32	19	23	40.771	-8.011
33	23	24	-9.600	3.804
34	24	25	14.224	1.397

Tabel 4.8
Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Sebelum Optimasi

Pembangkitan		Pembebanan		Rugi-rugi		Waktu komputasi
Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	
769.535	276.889	700.000	260200	69.535	16.689	0 :0:0:16

Grafik 4.1
Hasil Perhitungan Daya Sebelum Optimasi



4.8.2. Hasil Perhitungan Setelah Optimasi

Tabel 4.9
Hasil Perhitungan Tegangan, Sudut Tegangan, Pembangkitan dan Pembebanan Setelah Optimasi

Bus No.	Tegangan		Pembangkit		Pembebanan	
	Magnitude (pu)	Sudut (deg)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)
1	1.04835	0.00000	384.766	33.849	39.300	7.900
2	1.03501	-0.02397	0.000	0.000	14.000	6.000
3	1.03088	-0.02903	0.000	0.000	18.000	8.500
4	1.02108	-0.05358	0.000	0.000	36.100	16.500
5	1.00973	-0.09118	0.000	0.000	39.000	16.600
6	1.00732	-0.10734	0.000	0.000	22.500	9.700
7	1.01151	-0.11129	0.000	23.000	62.200	30.000
8	0.99373	-0.13761	0.000	0.000	41.000	21.300
9	1.04301	-0.06041	0.000	0.000	23.000	5.900
10	1.02643	-0.08630	0.000	0.000	17.200	7.000
11	1.00106	-0.12935	0.000	0.000	37.800	18.700
12	1.00271	-0.13102	133.800	31.204	5.400	2.700
13	0.99103	-0.13710	0.000	0.000	11.900	4.400
14	0.98567	-0.17803	0.000	0.000	6.200	2.900
15	0.98378	-0.19684	0.000	45.920	63.600	22.800
16	0.98786	-0.19305	0.000	0.000	4.100	0.800
17	0.98982	-0.18892	0.000	0.000	23.600	4.300
18	0.98208	-0.19616	0.000	0.000	16.400	7.000
19	0.98283	-0.19391	0.000	0.000	73.300	28.900
20	0.98514	-0.19089	74.824	18.025	0.000	0.000
21	0.98449	-0.18641	124.386	-14.659	0.000	0.000
22	0.97971	-0.20231	0.000	15.235	45.500	13.600
23	0.98126	-0.19931	0.000	25.000	50.300	13.500
24	0.98014	-0.20204	0.000	0.000	35.400	7.900
25	0.97795	-0.20621	0.000	0.000	14.200	3.300

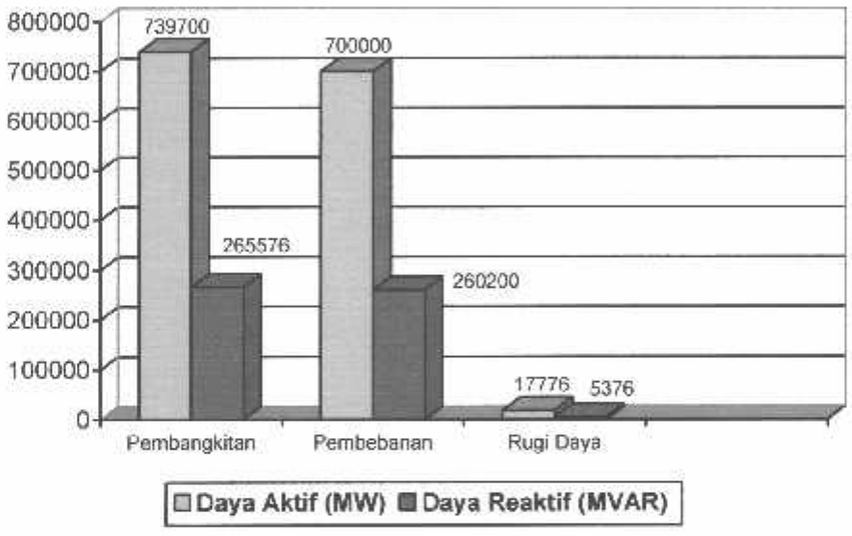
Tabel 4.10
Hasil Perhitungan Aliran Daya Antar Saluran Setelah Optimasi

No.	Saluran		Daya	
	Dari	Ke	Aktif (MW)	Reaktif (MVAR)
1	1	2	155.296	33.0147
2	1	9	190.170	-7.197
3	2	3	18.035	5.995
4	2	4	122.042	18.447
5	4	5	84.816	0.163
6	5	6	27.717	-5.722
7	5	7	17.166	-10.468
8	6	7	5.089	-11.669
9	7	8	25.464	4.230
10	7	10	-78.965	-19.876
11	7	11	13.435	-2.044
12	8	11	-15.773	-9.535
13	9	10	97.699	27.037
14	9	11	67.891	-46.035
15	11	12	21.935	-57.287
16	12	13	107.172	-21.886
17	12	17	43.104	-7.029
18	13	14	48.031	-11.562
19	13	15	45.867	-12.662
20	14	15	41.176	-10.735
21	15	16	-6.550	-4.689
22	15	17	-8.035	-4.938
23	15	18	-.0221	4.503
24	15	19	-7.344	4.555
25	15	22	11.540	-0.742
26	15	24	32.881	9.127
27	16	17	-10.668	0.858
28	18	19	-16.624	-0.342
29	19	20	-74.702	-17.756
30	19	21	-124.062	15.565
31	19	22	34.157	-5.891
32	19	23	67.317	-13.577
33	23	24	16.836	-1.936
34	24	25	14.222	1.256

Tabel 4.11
Total Pembangkitan, Pembebanan dan Rugi-rugi Saluran Setelah Optimasi

Pembangkitan		Pembebanan		Rugi-rugi		Waktu komputasi
Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	Aktif (MW)	Reaktif (MVAR)	
717.776	265.576	700.000	260200	17.776	5.376	0 :0:0:938

Grafik 4.2
Hasil Perhitungan Daya Setelah Optimasi



4.9. Perbandingan Hasil Perhitungan Sebelum dan Setelah Optimasi *Analisa Penempatan Kapasitor* Menggunakan *Metode Hybrid Particle Swarm Optimazer With Muatation (HPSOM)*

Dari semua hasil perhitungan diatas maka bisa dibuat tabel-tabel perbandingan sesuai dengan tujuan yang ingin dicapai.

4.9.1. Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus

Berikut ini adalah tabel perbandingan tingkat tegangan dan sudut tegangan pada tiap bus.

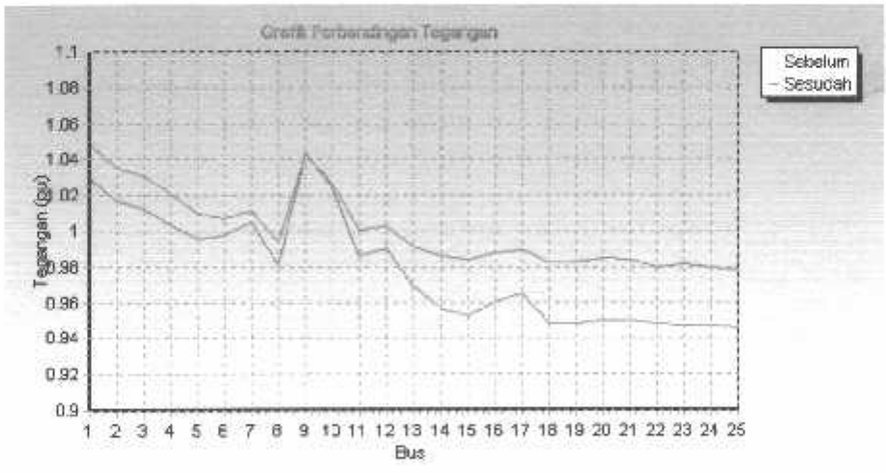
Tabel 4.12
Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus

Bus No	Sebelum Optimasi		Sesudah Optimasi	
	Tegangan (pu)	Sudut Tegangan (deg)	Tegangan (pu)	Sudut Tegangan (deg)
1	1.03333	0.00000	1.04835	0.00000
2	1.01639	-0.03425	1.03501	-0.02397
3	1.01216	-0.03949	1.03088	-0.02903
4	1.00324	-0.07946	1.02108	-0.05358
5	0.99580	-0.14363	1.00973	-0.09118
6	0.99714	-0.17598	1.00732	-0.10734
7	1.00524	-0.19387	1.01151	-0.11129
8	0.98045	-0.27507	0.99373	-0.13761
9	1.04420	-0.10768	1.04301	-0.06041
10	1.02352	-0.14966	1.02643	-0.08630
11	0.98616	-0.29564	1.00106	-0.12935
12	0.99000	-0.30457	1.00271	-0.13102
13	0.96892	-0.31408	0.99103	-0.13710
14	0.95630	-0.39052	0.98567	-0.17803
15	0.95226	-0.42835	0.98378	-0.19684
16	0.96044	-0.41346	0.98786	-0.19305
17	0.96476	-0.40340	0.98982	-0.18892
18	0.94867	-0.43636	0.98208	-0.19616
19	0.94868	-0.43771	0.98283	-0.19391
20	0.95000	-0.43645	0.98514	-0.19089
21	0.95000	-0.43350	0.98449	-0.18641
22	0.94794	-0.44481	0.97971	-0.20231
23	0.94767	-0.44121	0.98126	-0.19931
24	0.94779	-0.43924	0.98014	-0.20204
25	0.94551	-0.44370	0.97795	-0.20621

Dari tabel diatas dapat di lihat bahwa tegangan pada tiap bus masih dalam batas yang di perbolehkan. Setelah optimasi maka terjadi perubahan profil tegangan yang sebelumnya tidak sesuai batasan yang telah ditetapkan yaitu pada :

- Bus 18 yang semula 0.94867 p.u menjadi 0.98208 p.u
- Bus 19 yang semula 0.94868 p.u menjadi 0.98283 p.u
- Bus 22 yang semula 0.94794 p.u menjadi 0.97971 p.u
- Bus 23 yang semula 0.94767 p.u menjadi 0.98126 p.u
- Bus 24 yang semula 0.94779 p.u menjadi 0.98014 p.u
- Bus 25 yang semula 0.94551 p.u menjadi 0.97795 p.u

Grafik 4.3
Perbandingan Profil Tegangan

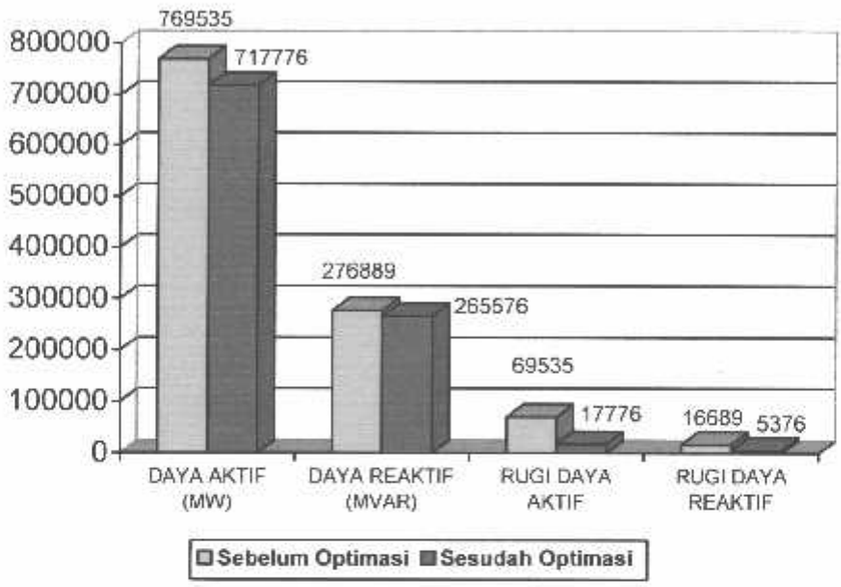


4.9.2. Perbandingan Aliran Daya Pada Saluran

Tabel 4.13
Perbandingan aliran daya tiap saluran

No.	Saluran		Kondisi Awal		Kondisi Akhir	
			Daya		Daya	
	Dari	Ke	P(MW)	Q(MVAR)	P(MW)	Q(MVAR)
1	1	2	206.380	19.974	155.296	33.0147
2	1	9	324.456	-71.952	190.170	-7.197
3	2	1	18.037	6.094	18.035	5.995
4	2	3	172.193	1.983	122.042	18.447
5	2	4	133.824	-20.261	84.816	0.163
6	3	2	51.419	-18.497	27.717	-5.722
7	4	2	40.956	-23.388	17.166	-10.468
8	4	5	28.429	-25.784	5.089	-11.669
9	5	4	70.572	-0389	25.464	4.230
10	5	6	-131.855	-11.664	-78.965	-19.876
11	5	7	67.610	-8.598	13.435	-2.044
12	6	5	27.908	-19.194	-15.773	-9.535
13	6	7	152.871	26.609	97.699	27.037
14	7	5	143.609	-134.952	67.891	-46.035
15	7	6	165.658	-182.898	21.935	-57.287
16	7	8	187.121	-24.949	107.172	-21.886
17	7	10	71.936	-4.200	43.104	-7.029
18	7	11	86.646	-13.504	48.031	-11.562
19	8	7	84.460	-14.700	45.867	-12.662
20	8	11	78.253	-17.426	41.176	-10.735
21	9	1	-19.999	-2.834	-6.550	-4.689
22	9	10	-21.561	-2.997	-8.035	-4.938
23	9	11	24.777	1.767	-.0221	4.503
24	10	7	28.354	0.516	-7.344	4.555
25	10	9	23.907	-6.714	11.540	-0.742
26	11	7	59.465	4.107	32.881	9.127
27	11	8	-24.259	0.450	-10.668	0.858
28	11	9	8.305	-3.418	-16.624	-0.342
29	11	12	-32.173	-13.790	-74.702	-17.756
30	12	11	-67.200	1.805	-124.062	15.565
31	12	13	21.863	-9.200	34.157	-5.891
32	12	17	40.771	-8.011	67.317	-13.577
33	13	12	-9.600	3.804	16.836	-1.936
34	13	14	14.224	1.397	14.222	1.256

Grafik 4.4
Perbandingan Total Daya Aktif Pembangkitan, Pembebanan dan Rugi-rugi Saluran Pada Kondisi Awal dan Kondisi Akhir



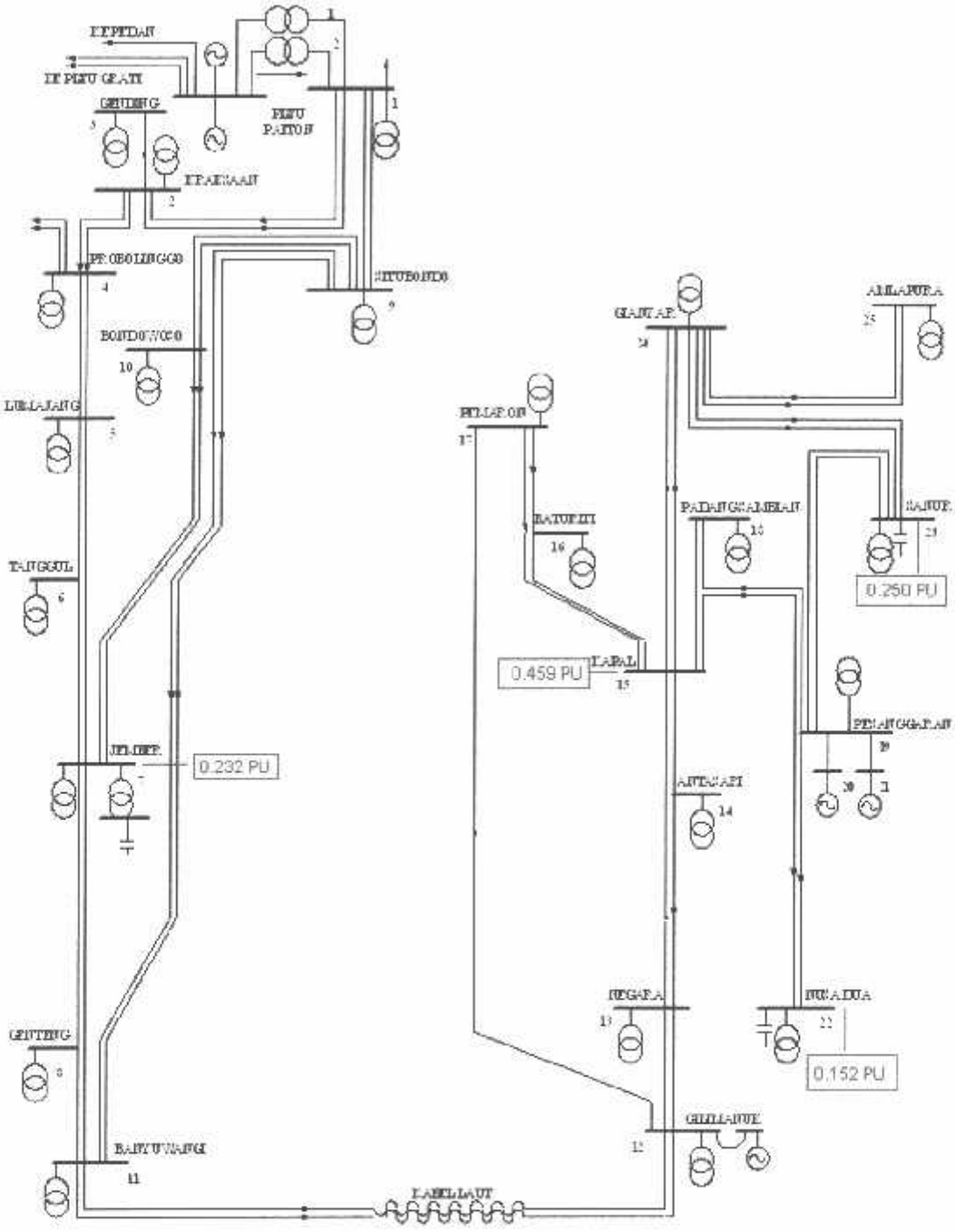
4.10. Hasil Analisa

Tabel 4.14

Penempatan dan Kapasitas Kapasitor Pada Tiap Bus

No.	Bus	Kapasitor (pu)
1	7	0.232
2	15	0.459
3	22	0.152
4	23	0.250

Setelah dilakukan hasil perhitungan, maka bus yang perlu dipasangi kapasitor adalah bus # 7 dengan kapasitas 0.232 pu, bus #15 dengan kapasitas 0.459 pu, bus #22 dengan kapasitas 0.152 pu, dan bus #23 dengan kapasitas 0.250 pu.



Gambar 4.22
Diagram lokasi penempatan kapasitor
Sumber : PT. PLN (Persero) P3B Jawa-Bali

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan.

Setelah dilakukan optimasi pada saluran transmisi 150 kV menggunakan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)*. Maka dapat di ambil kesimpulan sebagai berikut :

1. Setelah di lakukan analisa loadflow sebelum optimasi dapat diketahui ada beberapa bus yang tegangannya berada diluar batas yang telah di tentukan. Yaitu pada bus # 18, bus #19, bus # 22, bus 23, bus # 24, bus # 25.
2. Setelah dilakukan optimasi dengan *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)*, maka bus yang perlu dipasangi kapasitor adalah : bus # 7 sebesar 0.232 pu, bus # 15 sebesar 0.459 pu, bus # 22 sebesar 0.152 pu, bus # 23 sebesar 0.250 pu.
3. Setelah dilakukan pemasangan kapasitor maka terjadi perubahan profil tegangan pada beberapa bus, antaralain :
   - Bus 18 yang semula 0.94867 p.u menjadi 0.98208 p.u
   - Bus 19 yang semula 0.94868 p.u menjadi 0.98283 p.u
   - Bus 22 yang semula 0.94794 p.u menjadi 0.97971 p.u
   - Bus 23 yang semula 0.94767 p.u menjadi 0.98126 p.u
   - Bus 24 yang semula 0.94779 p.u menjadi 0.98014 p.u
   - Bus 25 yang semula 0.94551 p.u menjadi 0.97795 p.u
4. Rugi daya aktif saluran pada kondisi awal adalah 69.535 MW dan rugi daya aktif pada kondisi akhir setelah pemasangan kapasitor adalah 17.776 MW,

dari perhitungan rugi daya aktif turun sebesar 51.759 MW. Sedang rugi daya reaktif semula 16.689 MVAR menjadi 5.376 MVAR.

5.2. Saran.

Penggunaan metode *Hybrid Particle Swarn Optimizer With Mutation (HPSOM)*, masih perlu dikembangkan lagi terhadap sistem jaringan yang lebih luas. schingga nantinya dapat digunakan lebih efektif dan efisien dalam pemecahan permasalahan optimal power flow.

Daftar Pustaka

[1]. Ahmed A.A.Esmin, Germano Lambert-Torres, Antonio C . Zombroni de Souza, "A Hybrid Particle Swarm Optimization Applied toLoss Power Minimization", IEEE 2005

[2]. Rohit Kumar Pancholi, " Particle Swarm Optimation For Economic Dispatch with Line Flow and Voltage Constrain", IEEE 2003

[3]. Marsudi ,Ir. Djiteng, " *Operasi Sistem Tenaga Listrik* ", Balai Penerbit dan humas Institut Sains dan Teknologi Nasional, Jakarta, 1990.

[4]. Stevenson, William D, Jr, "*Analisa System Tenaga Listrik*", Penerbit Erlangga edisi keempat, 1996.

[5]. Zuhal, "*Dasar Teknik Tenaga Listrik dan Elektronika Daya*", PT. Gramedia Utama, Jakarta, 1995.

[6]. Bisri ,Hasan, " *Sistem Distribusi Tenaga Listrik* ".

LAMPIRAN

- Berita Acara Ujian Skripsi.
- Formulir Perbaikan Skripsi.
- Lembar Bimbingan Skripsi.
- Formulir Bimbingan Skripsi.
- Surat Survey di PT.PLN (Persero) PENYALURAN DAN PUSAT PENGATUR BEBAN JAWA BALI REGION JAWA TIMUR DAN BALI.
- Data Loadflow Region Jawa Timur dan Bali, Sub Sistem 150 kV Paiton dan Bali . Rabu 23 maret 2005 pukul 19.00 WIB. PT. PLN (Persero) Region Jawa Timur dan Bali
- Data saluran Transmisi Sub Sistem 150 kV Paiton – Bali.
- Listing Program Borland Delphi 7.0 Analisa Penempatan Kapasitor DenganMenggunakan Hybrid Particle Swarm Optimazer With Mutation (HPSOM) Pada Saluran Transmisi 150 kV Sub Sistem Paiton - Bali

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** : IRWAN ZAINUDIN YAZID
2. **NIM** : 01.12.103
3. **Jurusan** : Teknik Elektro S-1
4. **Konsentrasi** : Teknik Energi Listrik
5. **Judul Skripsi** : ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON - BALI

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

Hari : Jum'at
Tanggal : 22 september 2006
Dengan Nilai : 74 (B−)

Panitia Ujian Skripsi

Ketua

Ir. Mochtar Asroni, MSME
NIP. 101 8100 036

Sekretaris

Ir. F. Yudi Limpraptono, MT
NIP. 103 9500 274

Anggota Penguji

Penguji Pertama

Ir. Yusuf Ismail Nakhoda, MT
NIP.101 8800 189

Penguji Kedua

Ir. Djojo Priatmono,MT
NIP.101 8500 107

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

PERSETUJUAN PERBAIKAN SKRIPSI

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

Hari : Jum'at
Tanggal : 22 september 2006
Telah dilakukan perbaikan skripsi oleh :

1. Nama : Irwan Zainudn Yazid
2. NIM : 01.12.103
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : **ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN *HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION (HPSOM)* PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON–BALI**

Perbaikan meliputi :

No	Materi Perbaikan	Ket
1.	Ralat judul skripsi	
2	Perbaikan profil tegangan	

Anggota Penguji

Penguji Pertama

Ir. Yusuf Ismail Nakhoda, MT
NIP.101 8800 189

Penguji Kedua

Ir. Djojo Priatmono
NIP. 101 8500 107

Dosen Pembimbing

Ir. Eko Nurcahyo
NIP. 102 8700 172

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA : IRWAN ZAINUDIN
NIM : 0112103
Perbaikan meliputi :

- Ralat judul skripsi
- Perbaikan profil tegangan.

Malang,

( )

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

LEMBAR BIMBINGAN SKRIPSI

1. Nama : IRWAN ZAINUDIN YAZID
2. NIM : 01.12.103
3. NIRM : 0132021102103
4. Jurusan : TEKNIK ELEKTRO S-1
5. Konsentrasi : TEKNIK ENERGI LISTRIK
6. Judul Skripsi :

ANALISA PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN PARTICLE SWARM OPTIMAZER WITH MUTATION PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON-BALI

7. Tanggal Mengajukan Skripsi : 30 Juni 2006
8. Tanggal Menyelesaikan Skripsi : 27 September 2006
9. Dosen Pembimbing : Ir. Eko Nurcahyo
10. Telah dievaluasi dengan nilai : 80

Malang, September 2006

Mengetahui Ketua Jurusan Teknik Elektro	Disetujui Dosen Pembimbing
Ir. F. Yudi Limpraptono, MT NIP. 103 9500 274	Ir. Eko Nurcahyo NIP.102 8700 107

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : IRWAN ZAINUDIN YAZID
Nim : 01.12.103
Masa Bimbingan : 30 Juni 2006 s/d 30 Desember 2006
Judul Skripsi : ANALISA PENEMPATAN KAPASITOR DENGAN MENNGUNAKAN HYBRID PARTICLE SWARM OPTIMAZER WITH MUTATION PADA SALURAN TRANSMISI 150 kV SUB SISTEM PAITON - BALI

No.	Tanggal	Uraian	Parap Pembimbing
1.	9-08-2006	Konsultasi jurnal dan makalah seminar proposal	
2.	16-08-2006	Konsultasi BAB I dan BAB II (pemberian sumber reverensi gambar yang digunakan)	
3.	25-08-2006	Konsultasi BAB III	
4.	7-09-2006	BAB IV, dan BAB V (perbaikan kesimpulan)	
5.	9-09-2006	Konsultasi makalah seminar hasil	
6.	18-09-2006	Konsultasi akhir keseluruhan BAB	
7.	29-09-2006	Acc ujian Skripsi	
8.			

Malang, September 2006
Dosen Pembimbing,

Ir. Eko Nurcahyo
Nip. P. 1028 700 172

Form.S-4b

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

Jalan Suningrat No. 45 Taman Sidoarjo 61257
Telepon : (031) 7882113, 7882114 Facsimile : (031) 7882578, 7881024 Website : www.pln-jawa-bali.co.id
Kotak Pos : 4119 SBS E-mail : region4@pln-jawa-bali.co.id

Nomor : 248 / 330/ RJTB/ 2006
Surat Sdr. No. : ITN-1824/III.TA/2/06
Lampiran : 1 (satu) lampiran.
Perihal : Ijin Survey/ Pengambilan Data.

18 SEP 2006

Kepada

Yth. Dekan Fakultas Teknologi Industri.
INSTITUT TEKNOLOGI NASIONAL MALANG
Di
MALANG

Menunjuk surat Saudara nomor : ITN-1824/III.TA/2/06 tanggal 03 Juli 2006 perihal : Survey/ Permintaan Data, dengan ini diberitahukan bahwa kami tidak keberatan untuk memberikan ijin kepada Mahasiswa Saudara, bernama :

- IRWAN ZAINUDIN NIM : 01.12.103

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

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

Demikian harap makum dan terima kasih atas perhatian saudara.

MANAGER BIDANG SDM & ADMINISTRASI,

SRI WIRATMO

Tembusan Yth. :

0. M.SDMO PLN P3B.
0. M.OPHAR PLN P3B RJTB.
0. AsMan Transmisi & Gardu Induk
0. Sdr. Irwan Zainudin

MODEL 1003

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini, saya :

Nama : IRWAN ZAINUDIN YAZID

Pria/ Wanita : Pria

Tempat / Tanggal lahir : Ponorogo / 17- september - 1982

Alamat / No. telepon : Jl. Cangar Agung No. 25. Malang

Pekerjaan : Mahasiswa. ITN

Dengan ini saya menerangkan bahwa :

1. Saya bersedia dan setuju menanggung semua akibat yang ditimbulkan karena kesalahan maupun kelalaian saya dan semua akibat lainnya yang terjadi pada instalasi peralatan milik PLN selama melakukan Training/ Praktek Kerja/ Riset pada PT PLN (Persero) P3B Region Jawa Timur dan Bali, yang telah mendapat ijin dari PT PLN (Persero) P3B Region Jawa Timur dan Bali ;

2. Saya atas peringatan pertama akan membayar sepenuhnya, semua biaya yang langsung menimbulkan kerugian atau kecelakaan , karena kelalaian saya ;

3. Saya akan segera mematuhi semua petunjuk –petunjuk yang diberikan oleh Petugas PT PT PLN (Persero) P3B Region Jawa Timur dan Bali ;

4. Saya sanggup tidak membocorkan hal – hal yang bersifat rahasia perusahaan PT PLN (Persero) P3B Region Jawa Timur dan Bali dan bahan yang saya peroleh dalam Training/ Praktek Kerja/ Riset, dan tidak saya pergunakan untuk hal – hal yang dapat merugikan PT PLN (Persero) P3B Region Jawa Timur dan Bali

5. Saya sanggup menanggung sendiri segala sesuatu untuk keperluan Training/ Praktek Kerja/ Riset termasuk biaya perjalanan , penginapan makan dan sebagainya ;

6. Saya sanggup menyerahkan 1 (satu) buah buku laporan Training/ Praktek Kerja/ Riset kepada PT PLN (Persero) P3B Region Jawa Timur dan Bali, setelah saya presentasikan kepada Manager Bidang SDMAD PT PLN (Persero) P3B Region Jawa Timur dan Bali mengenai tugas Training/ Praktek Kerja/ Riset.

7. Saya tunduk dan akan mentaati semua peraturan yang berlaku di PT PLN (Persero) P3B Region Jawa Timur dan Bali, dan saya sanggup tidak meninggalkan tugas kedinasan selama Training/ Praktek Kerja/ Riset.

Surabaya,

Yang membuat pernyataan

6000

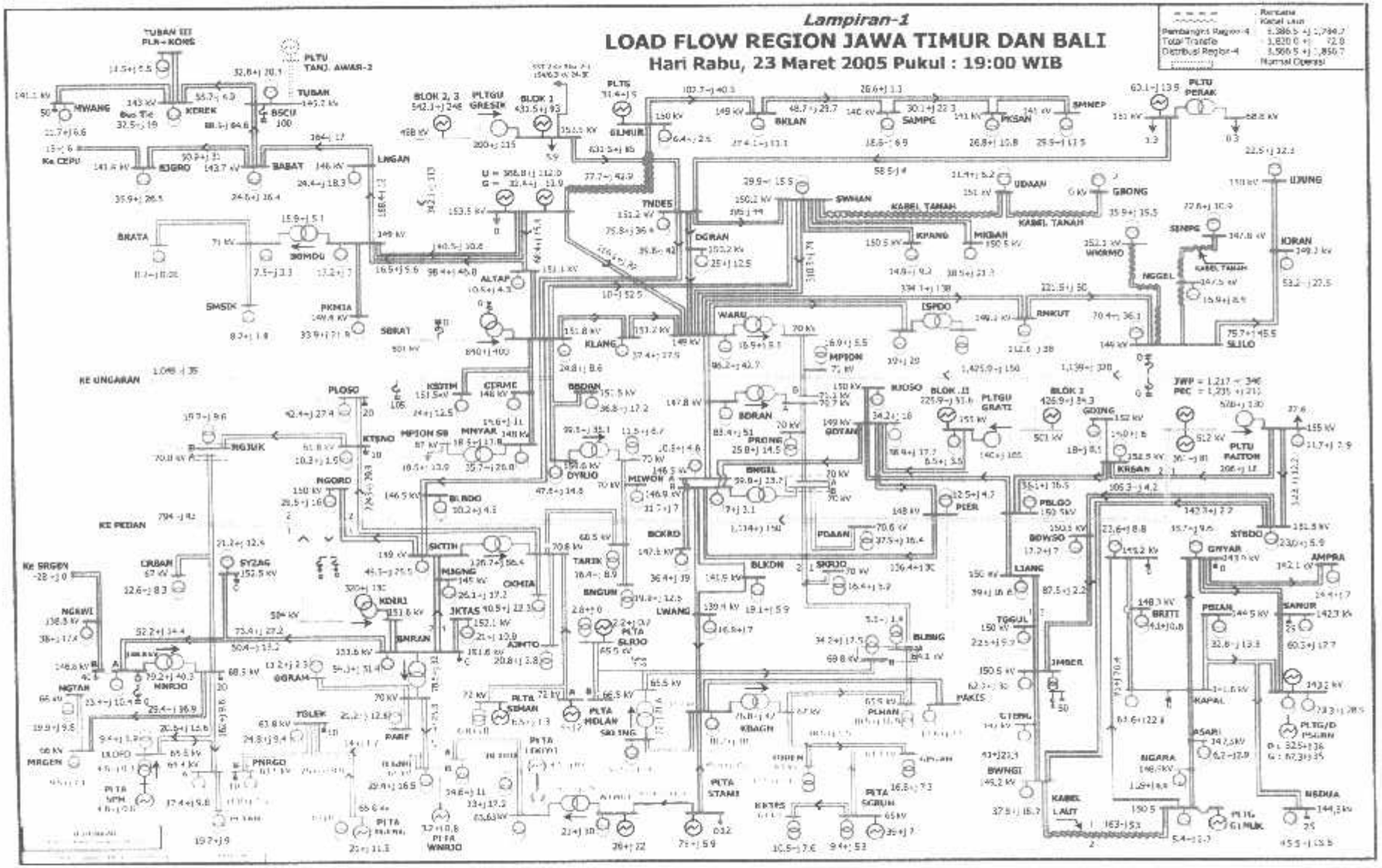
METERAI TEMPEL

Irwan Zainudin Yazid

Lampiran-1

LOAD FLOW REGION JAWA TIMUR DAN BALI

Hari Rabu, 23 Maret 2005 Pukul : 19:00 WIB



DATA KARAKTERISTIK TRANSMISI UPT & SRB

Jml Sirkit	Dari	Ke	No. Sirkit	Data_Penghantar Teg (kV)	Jarak (Km)	Jenis		I Nom (Amp)	Tap CT Dari	Tap CT Ke	Tap WT Dari	Tap WT Ke	R (Ohm/km)	X (Ohm/km)	B (MyS/km)	R (Ohm)	X (Ohm)	B (Ohm)	Z act (Ohm)	Z base (Ohm)	Z pu (Ohm)	R pu (Ohm)	X pu (Ohm)	B pu (Ohm)	Daya (MVA)
				(1)	(2)								(3)	(4)	(5)	(6)=(2)x(3)	(7)=(2)x(4)	(8)=(2)x(5)	(9)=sqrt $(6)^2+(7)^2$	(10)=(1)²/100	(11)=(9)/(10)	(12)=(6)/(10)	(13)=(7)/(10)	(14)=(8)(kV)²/0.1875	(15)=(1)x(Ax)1.732/1000
6. UPT Probolinggo																									
1	PITON7	GRATI7	1	500	55.433	GANNET	4x392,6	2900	2000	2000			0.0251	0.2808	4.0490	1.3914	15.5656	224.4482	15.6276	2,500.000	0.0063	0.0006	0.0062	0.0045	2,424.8000
1	PITON7	GRATI7	2	500	55.433	GANNET	4x392,6	2900	2000	2000			0.0251	0.2808	4.0490	1.3914	15.5656	224.4482	15.6276	2,500.000	0.0063	0.0006	0.0062	0.0045	2,424.8000
1	BNGIL5	GDTAN5	1	150	16.805	ACSR. AW	330 mm²	740	1000	1000	1250	1600	0.1172	0.4003	2.8530	1.9696	6.7270	47.9447	7.0094	225.0000	0.0312	0.0088	0.0299	0.0108	192.2520
1	BNGIL5	GDTAN5	2	150	16.805	ACSR. AW	330 mm²	740	1000	1000	1250	1600	0.1172	0.4003	2.8530	1.9696	6.7270	47.9447	7.0094	225.0000	0.0312	0.0088	0.0299	0.0108	192.2520
1	BNGIL5	EDRAN5	1	150	9.800	ACSR. AW	330 mm²	740	1000	1000	1600	1600	0.1172	0.4003	2.8530	1.1486	3.9229	27.9594	4.0876	225.0000	0.0182	0.0051	0.0174	0.0063	192.2520
1	BNGIL5	WARU5	1	150	9.800	ACSR. AW	330 mm²	740	1000	1000	1600	1600	0.1172	0.4003	2.8530	1.1486	3.9229	27.9594	4.0876	225.0000	0.0182	0.0051	0.0174	0.0063	192.2520
1	BNGIL5	PIER5	1	150	5.150	ACSR. ZEBRA	2 x 435 mm²	1620	2000	1000	1250	1250	0.3870	0.2807	4.0203	1.993[illegible]	1.4456	20.7045	2.4621	225.0000	0.0109	0.0089	0.0064	0.0047	420.8760
1	BNGIL5	PIER5	2	150	5.150	ACSR. ZEBRA	2 x 435 mm²	1620	2000	1000	1250	1250	0.3870	0.2807	4.0203	1.993[illegible]	1.4456	20.7045	2.4621	225.0000	0.0109	0.0089	0.0064	0.0047	420.8760
1	BNGIL5	BCKRO5	1	150	6.200	ACSR. ZEBRA	2 x 435 mm²	1620	1000	1000	1250	1250	0.3870	0.2807	4.0203	2.3994	1.7403	24.9259	2.9641	225.0000	0.0132	0.0107	0.0077	0.0056	420.8760
1	BNGIL5	BCKRO5	2	150	6.200	ACSR. ZEBRA	2 x 435 mm²	1620	1000	1000	1250	1250	0.3870	0.2807	4.0203	2.3994	1.7403	24.9259	2.9641	225.0000	0.0132	0.0107	0.0077	0.0056	420.8760
1	BNGIL5	PNMAS5	1	150	27.770	ACSR. AW	330 mm²	740	1000	1000	1600	1600	0.1172	0.4003	2.8530	3.2546	11.1163	79.2278	11.5830	225.0000	0.0515	0.0145	0.0494	0.0178	192.2520
1	BNGIL5	LWANG5	1	150	34.680	ACSR. AW	330 mm²	740	1000	1000	1600	1600	0.1172	0.4003	2.8530	4.0645	13.8824	98.9420	14.4652	225.0000	0.0643	0.0181	0.0617	0.0223	192.2520
1	GDTAN5	RJOSO5	1	150	10.487	ACSR. AW	330 mm²	740	1000	800	1250	1250	0.1172	0.4003	2.8530	1.2291	4.1979	29.9194	4.3742	225.0000	0.0194	0.0055	0.0187	0.0067	192.2520
1	GDTAN5	RJOSO5	2	150	10.487	ACSR. AW	330 mm²	740	1000	800	1250	1250	0.1172	0.4003	2.8530	1.2291	4.1979	29.9194	4.3742	225.0000	0.0194	0.0055	0.0187	0.0067	192.2520
1	PIER5	GRATI5	1	150	31.935	TACSR	410 mm²	1620	1000	1000	1250	2000	0.3870	0.2807	4.0203	12.3588	8.9642	128.3883	15.2675	225.0000	0.0679	0.0549	0.0398	0.0289	420.8760
1	PIER5	GRATI5	2	150	31.935	TACSR	410 mm²	1620	1000	1000	1250	2000	0.3870	0.2807	4.0203	12.3588	8.9642	128.3883	15.2675	225.0000	0.0679	0.0549	0.0398	0.0289	420.8760
1	PBLGO5	GDTAN5	1	150	33.627	ACSR. AW	330 mm²	740	800	1000	1250	1250	0.1172	0.4003	2.8530	3.9645	13.5409	96.5084	14.1094	225.0000	0.0627	0.0176	0.0602	0.0217	192.2520
1	PBLGO5	GDTAN5	2	150	33.627	ACSR. AW	330 mm²	740	800	1000	1250	1250	0.1172	0.4003	2.8530	3.9645	13.5409	96.5084	14.1094	225.0000	0.0627	0.0176	0.0602	0.0217	192.2520
1	PBLGO5	LJANG5	1	150	51.692	ACSR. AW	330 mm²	740	600	600	800	800	0.1172	0.4003	2.8530	6.0583	20.6923	147.4773	21.5610	225.0000	0.0958	0.0269	0.0920	0.0332	192.2520
1	PBLGO5	LJANG5	2	150	51.692	ACSR. AW	330 mm²	740	600	600	800	800	0.1172	0.4003	2.8530	6.0583	20.6923	147.4773	21.5610	225.0000	0.0958	0.0269	0.0920	0.0332	192.2520
1	KRSAN5	PBLGO5	1	150	30.239	ACSR. AW	330 mm²	740	1000	1000	1250	1250	0.1172	0.4003	2.8530	3.5440	12.1047	86.2719	12.6128	225.0000	0.0561	0.0158	0.0538	0.0194	192.2520
1	KRSAN5	PBLGO5	2	150	30.239	ACSR. AW	330 mm²	740	1000	1000	1250	1250	0.1172	0.4003	2.8530	3.5440	12.1047	86.2719	12.6128	225.0000	0.0561	0.0158	0.0538	0.0194	192.2520
1	KRSAN5	GDING5	1	150	19.137	ACSR. AW	330 mm²	740	1000	400	1250	1250	0.1172	0.4003	2.8530	2.2429	7.6605	54.5979	7.9821	225.0000	0.0355	0.0100	0.0340	0.0123	192.2520
1	PITON5	KRSAN5	1	150	20.204	ACSR. AW	330 mm²	740	1000	1000	1250	1250	0.1172	0.4003	2.8530	2.3679	8.0877	57.6420	8.4272	225.0000	0.0375	0.0105	0.0358	0.0130	192.2520

DATA KARAKTERISTIK TRANSMISI UPT & SRB

Jml Sirkit	Dari	Ke	No. Sirkit	Data Penghantar Tag (kV)	Jarak (Km)	Jenis	I Nom (Amp)	Tap CT Dari	Tap CT Ke	Tap WT Dari	Tap WT Ke	R (Ohm/km)	X (Ohm/km)	B (MiS/km)	R (Ohm)	X (Ohm)	B (Ohm)	Z act (Ohm)	Z base (Ohm)	Z pu (Ohm)	R pu (Ohm)	X pu (Ohm)	B pu (Ohm)	Daya (MVA)
				(1)	(2)							(3)	(4)	(5)	(6)=(2)x(3)	(7)=(2)x(4)	(8)=(2)x(5)	(9)=sqrt [(6)^2+(7)^2]	(10)=(1)^2/100	(11)=(9)/(10)	(12)=(6)/(10)	(13)=(7)/(10)	(14)=(8)x(10)^2/(0.1973)	(15)=(1)x(A)x1.732/1000
6. UPT Probolinggo																								
1	PITON5	KRSAN5	2	150	20.204	ACSR. AW	330 mm²	740	1000	1000	1250	1250	0.1172	0.4003	2.8530	2.3679	8.0877	57.6420	8.4272	225.0000	0.0375	0.0105	0.0359	0.0130	192.2520
1	PITON5	STBOO5	1	150	55.433	ACSR. ZEBRA	2 x 435 mm²	1900	2000	2000	1250	1250	0.3870	0.2807	4.0203	21.4525	15.5600	222.3573	26.5015	225.0000	0.1178	0.0953	0.0692	0.0502	415.6800
1	PITON5	STBOO5	2	150	55.433	ACSR. ZEBRA	2 x 435 mm²	1900	2000	2000	1250	1250	0.3870	0.2807	4.0203	21.4525	15.5600	222.3573	26.5015	225.0000	0.1178	0.0953	0.0692	0.0502	415.6800
1	BNGIL4	GDTAN4	1	70	4.452	ACSR. OSTRICH	300 MCM	440							-	-	-	-				#DIV/0!	#DIV/0!	-	53.3456
1	BNGIL4	GDTAN4	2	70	4.452	ACSR. OSTRICH	300 MCM	400				0.2175	0.3871	2.9560	0.9683	1.7234	13.1601	1.9768	49.0000	0.0403	0.0198	0.0352	0.0006	48.4960
1	BNGIL4	PDAAN4	1	70	9.700	ACSR. OSTRICH	300 MCM	440	250	400		0.2175	0.3871	2.9560	2.1098	3.7549	28.6732	4.3070	49.0000	0.0879	0.0431	0.0766	0.0014	53.3456
1	BNGIL4	PDAAN4	2	70	9.700	ACSR. OSTRICH	300 MCM	440	250	400	800	800	0.2175	0.3871	2.9560	2.1098	3.7549	28.6732	4.3070	49.0000	0.0879	0.0431	0.0766	0.0014	53.3456
1	BNGIL4	BDRAN4	1	70	11.700	ACSR. PIPER	300 MCM	440	400	400	400	400	0.2180	0.3859	2.9660	2.5506	4.5150	34.8905	5.1857	49.0000	0.1058	0.0521	0.0921	0.0017	53.3456
1	BNGIL4	PRONG4	1	70	11.700	ACSR. PIPER	300 MCM	440	400	400	400	400	0.2180	0.3859	2.9660	2.5506	4.5150	34.8905	5.1857	49.0000	0.1058	0.0521	0.0921	0.0017	53.3456
1	BNGIL4	BLBNG4	1	70	40.000	ACSR. PIGEON	300 AWG	300	200	200	800	800	0.3866	0.3584	3.2020	15.4640	14.3360	128.0800	21.0869	49.0000	0.4303	0.3156	0.2926	0.0063	36.3720
1	BNGIL4	BLBNG4	2	70	40.000	ACSR. PIGEON	300 AWG	300	200	200	400	800	0.3866	0.3584	3.2020	15.4640	14.3360	128.0800	21.0869	49.0000	0.4303	0.3156	0.2926	0.0063	36.3720
Modifikasi TI By Grat-pPier 1 sirkit menjadi (Grati-Gdtan-Pier) dan Probolinggo - Gondangwetan 1 sirkit menjadi (Pblgo-Grato-Gdtar)																								
1	GRATI5	GDTAN5	1	150	10.743	TACSR	410 mm²	2800	1000	1000	1250	2000	0.3870	0.2807	4.0203	4.1575	3.0156	43.1931	5.1360	225.0000	0.0228	0.0185	0.0134	0.0097	727.4400
1	GRATI5	GDTAN5	2	150	10.743	TACSR	410 mm²	2800	1000	1000	1250	2000	0.3870	0.2807	4.0203	4.1575	3.0156	43.1931	5.1360	225.0000	0.0228	0.0185	0.0134	0.0097	727.4400
1	GDTAN5	PIER5	1	150	21.183	TACSR	410 mm²	2800	1000	1000	1250	2000	0.3870	0.2807	4.0203	8.1978	5.9461	85.1620	10.1272	225.0000	0.0450	0.0364	0.0264	0.0192	727.4400
1	GDTAN5	PIER5	2	150	21.183	TACSR	410 mm²	2800	1000	1000	1250	2000	0.3870	0.2807	4.0203	8.1978	5.9461	85.1620	10.1272	225.0000	0.0450	0.0364	0.0264	0.0192	727.4400

DATA KARAKTERISTIK TRANSMISI UPT & SRB

Jml Sirkit	Dari	Ke	No. Sirkit	Data _ Penghantar				I Nom	Tap CT		Tap WT		R	X	B	R	X	B	Z act	Z base	Z pu	R pu	X pu	B pu	Daya
				Teg (kV)	Jarak (Km)	Jenis		(Amp)	Dari	Ke	Dari	Ke	(Ohm/km)	(Ohm/km)	(MyS/km)	(Ohm)	(Ohm)	(Ohm)	(Ohm)	(Ohm)	(Ohm)	(Ohm)	(Ohm)	(Ohm)	(MVA)
				(1)	(2)								(3)	(4)	(5)	(6)=(2)x(3)	(7)=(2)x(4)	(8)=(2)x(5)	(9)=akar $(6)^2+(7)^2$	(10)=$(1)^2$/100	(11)=(9)/(10)	(12)=(6)/(10)	(13)=(7)/(10)	(14)=(8)/$(kV)^2$/0.1875	(15)=(1)x(A)x1.732/1000
7. UPT Jember																									
1	PBLGO5	LJANG5	-	150	51.892	ACSR. DOVE	330 mm²	740	600	600	800	800	0.1172	0.4003	2.8530	6.0583	20.6923	147.4773	21.5610	225.0000	0.0958	0.0269	0.0920	0.0332	192.2520
1	PBLGO5	LJANG5	2	150	51.892	ACSR. DOVE	330 mm²	740	600	600	800	800	0.1172	0.4003	2.8530	6.0583	20.6923	147.4773	21.5610	225.0000	0.0958	0.0269	0.0920	0.0332	192.2520
1	LJANG5	TGGJL5	-	150	32.080	ACSR. AW	330 mm²	740	800	800	1250	1250	0.1172	0.4003	2.8530	3.7598	12.8416	91.5242	13.3807	225.0000	0.0595	0.0167	0.0571	0.0206	192.2520
1	LJANG5	JMBER5	-	150	60.530	ACSR. AW	330 mm²	740	800	800	1250	800	0.1172	0.4003	2.8530	7.0941	24.2302	172.6921	25.2473	225.0000	0.1122	0.0315	0.1077	0.0389	192.2520
1	TNGJL5	JMBER5	1	150	28.450	ACSR. AW	330 mm²	740	1000	800	1250	800	0.1172	0.4003	2.8530	3.3343	11.3885	81.1679	11.8996	225.0000	0.0527	0.0148	0.0506	0.0183	192.2520
1	JMBER5	BWNGI5	1	150	82.595	ACSR. AW	330 mm²	740	600	600	800	800	0.1172	0.4003	2.8530	9.6801	33.0626	235.6435	34.4507	225.0000	0.1531	0.0430	0.1469	0.0530	192.2520
1	JMBER5	STENG5	1	150	64.708	ACSR. AW	330 mm²	740	600	1000	1250	1250	0.1172	0.4003	2.8530	7.5838	25.9026	184.6119	26.9900	225.0000	0.1200	0.0337	0.1151	0.0415	192.2520
1	BDWSC5	JMBER5	1	150	39.662	ACSR. AW	330 mm²	740	800	800	800	800	0.1172	0.4003	2.8530	4.6484	15.8767	113.1557	15.5432	225.0000	0.0735	0.0207	0.0706	0.0255	192.2520
1	BDWSC5	JMBER5	2	150	39.662	ACSR. AW	330 mm²	740	800	800	1600	1600	0.1172	0.4003	2.8530	4.6484	15.8767	113.1557	15.5432	225.0000	0.0735	0.0207	0.0706	0.0255	192.2520
1	STBDO5	PITON5	1	150	55.433	ACSR. ZEBRA	2 x 435 mm²	1620	2000	2000	1250	1250	0.3870	0.2807	4.0203	21.4526	15.5600	222.8573	23.5015	225.0000	0.1178	0.0953	0.0692	0.0502	420.8760
1	STBDO5	PITON5	2	150	55.433	ACSR. ZEBRA	2 x 435 mm²	1620	2000	2000	1250	1250	0.3870	0.2807	4.0203	21.4526	15.5600	222.8573	23.5015	225.0000	0.1178	0.0953	0.0692	0.0502	420.8760
1	STBDO5	BDWSO5	1	150	34.786	ACSR. AW	330 mm²	740	800	800	800	800	0.1172	0.4003	2.8530	4.0769	13.9248	95.2445	14.5094	225.0000	0.0645	0.0181	0.0619	0.0223	192.2520
1	STBDO5	BDWSO5	2	150	34.786	ACSR. AW	330 mm²	740	800	800	1600	1600	0.1172	0.4003	2.8530	4.0769	13.9248	95.2445	14.5094	225.0000	0.0645	0.0181	0.0619	0.0223	192.2520
1	STBDO5	BWNGI5	1	150	74.200	ACSR. DOVE	2 x 340 mm²	1200	800	800	1600	1600	0.0586	0.2815	4.0740	4.3481	20.8873	302.2908	21.3351	225.0000	0.0948	0.0193	0.0928	0.0680	311.7600
1	STBDO5	BWNGI5	2	150	74.200	ACSR. DOVE	2 x 340 mm²	1200	800	800	1600	1600	0.0586	0.2815	4.0740	4.3481	20.8873	302.2908	21.3351	225.0000	0.0948	0.0193	0.0928	0.0680	311.7600
1	GTENG5	BWNGI5	1	150	33.850	ACSR. DOVE	330 mm²	740	1000	800	1250	800	0.1172	0.4003	2.8530	3.9789	13.5902	96.8694	14.1607	225.0000	0.0629	0.0177	0.0604	0.0218	192.2520
1	BWNGI5	KTPNG5	1	150	7.990	ACSR. HAWK	477 MM	666	800	800			0.1370	0.3996	2.8800	1.0946	3.1888	23.0112	3.3526	225.0000	0.0149	0.0049	0.0141	0.0052	170.1690
	BWNGI5	KTP-GLM	I (11)	150	4.471	SUBMARINE CABLE	330 mm²	500	800	800			0.1613	0.3482		0.7212	1.5568	-	1.7157	225.0000	0.0076	0.0032	0.0069	-	129.9000
	BWNGI5	KTP-GLM	I (lama)	150	4.829	OSS	300 mm²	465	800	800						-	-	-				#DIV/0!	#DIV/0!	-	120.8070
1	BWNGI5	KTPNG5	2	150	7.990	ACSR. HAWK	477 MM	666	800	800			0.1370	0.3996	2.8800	1.0946	3.1888	23.0112	3.3526	225.0000	0.0149	0.0049	0.0141	0.0052	170.1690
	BWNGI5	KTP-GLM	II (10)	150	4.389	SUBMARINE CABLE	300 mm²	500	800	800			0.1643	0.3546		0.7211	1.5563		1.7153	225.0000	0.0076	0.0032	0.0069	-	129.9000
	BWNGI5	KTP-GLM	II (lama)	150	4.260	OSS	300 mm²	465	800	800						-	-	-				#DIV/0!	#DIV/0!	-	120.8070

DATA KARAKTERISTIK TRANSMISI UPT & SRB

Jml Sirkit	Dari	Ke	No. Sirkit	Data _ Penghantar Teg (kV)	Jarak (Km)	Jenis		I Nom (Amp)	Tap CT Dari	Tap CT Ke	Tap WT Dari	Tap WT Ke	R (Ohm/km)	X (Ohm/km)	B (MyS/km)	R (Ohm)	X (Ohm)	B (Ohm)	Z act (Ohm)	Z base (Ohm)	Z pu (Ohm)	R pu (Ohm)	X pu (Ohm)	B pu (Ohm)	Daya (MVA)
				(1)	(2)								(3)	(4)	(5)	(6)=(2)x(3)	(7)=(2)x(4)	(8)=(2)x(5)	(9)=akar (6)²+(7)²	(10)=(1)²/100	(11)=(9)/(10)	(12)=(6)/(10)	(13)=(7)/(10)	(14)=((9)/(kV)²)x1.1975	(15)=(1)x(A)x1.732/100
Sub Region Bali																									
1	GLNUK5	CH G_NUK	1	150	1.720	ACSR. HAWK	477 MCM	645	800	800			0.1370	0.3966	2.8800	0.2356	0.6822	4.9536	0.7217	225.0000	0.0032	0.0010	0.0030	0.0011	167.5710
1	GLNUK5	CH G_NUK	2	150	1.720	ACSR. HAWK	477 MCM	645	800	800			0.1370	0.3966	2.8800	0.2356	0.6822	4.9536	0.7217	225.0000	0.0032	0.0010	0.0030	0.0011	167.5710
1	CH GLNUK	CH KTPNG	1	150	4.471	OFC	3 x 300 mm²	482	800	800			0.1613	0.3482		0.7212	1.5568	-	1.7157	225.0000	0.0076	0.0032	0.0069	-	120.0275
1	CH GLNUK	CH KTPNG	2	150	4.389	OFC	3 x 300 mm²	482	800	800			0.1613	0.3548		0.7079	1.5572	-	1.7106	225.0000	0.0076	0.0031	0.0069	-	120.0275
1	CH KTPNG	BWNG 5	1	150	7.990	ACSR. HAWK	477 MCM	645	800	800			0.1370	0.3966	2.8800	1.0946	3.1688	23.0112	3.3526	225.0000	0.0149	0.0049	0.0141	0.0052	167.5710
1	CH KTPNG	BWNG 5	2	150	7.990	ACSR. HAWK	477 MCM	645	800	800			0.1370	0.3966	2.8800	1.0946	3.1688	23.0112	3.3526	225.0000	0.0149	0.0049	0.0141	0.0052	167.5710
1	GLNUK5	NGARA5	1	150	38.040	ACSR. HAWK	477 MCM	645	800	800	830	830	0.1370	0.3966	2.8800	5.2115	15.0867	109.5552	15.9614	225.0000	0.0709	0.0232	0.0671	0.0247	167.5710
1	GLNUK5	NGARA5	2	150	38.040	ACSR. HAWK	477 MCM	645	800	800	830	830	0.1370	0.3966	2.8800	5.2115	15.0867	109.5552	15.9614	225.0000	0.0709	0.0232	0.0671	0.0247	167.5710
1	GLNUK5	PMRON5	1	150	75.900	ACSR. HAWK	2 x 477 MCM	1290	800	800	1250	1250	0.1163	0.3895	2.9380	8.9790	29.5631	222.9942	30.8965	225.0000	0.1373	0.0399	0.1314	0.0502	335.1420
1	NGARA5	ASARI5	1	150	67.710	ACSR. HAWK	477 MCM	645	800	800	830	830	0.1370	0.3966	2.8820	9.2763	26.8538	195.1402	28.4108	225.0000	0.1263	0.0412	0.1194	0.0439	167.5710
1	NGARA5	ASARI5	2	150	44.400	ACSR. HAWK	477 MCM	645	800	800	830	1250	0.1370	0.3966	2.8820	6.0828	17.6090	127.9903	18.6300	225.0000	0.0828	0.0270	0.0783	0.0288	167.5710
1	KAPAL5	ANSR15	1	150	23.310	ACSR. HAWK	477 MCM	645	800	800	830	1250	0.1370	0.3966	2.8800	3.1935	9.2447	67.1329	9.7808	225.0000	0.0435	0.0142	0.0411	0.0151	167.5710
1	KAPAL5	PMRON5	1	150	58.580	ACSR PARTRIDE	135 MCM	400	400	400	830	830	0.2136	0.4048	2.7970	12.5127	23.7132	163.8483	26.8120	225.0000	0.1192	0.0556	0.1054	0.0369	103.9200
1	KAPAL5	GNYAR5	1	150	19.210	ACSR. HAWK	477 MCM	645	800	800	830	830	0.1370	0.3966	2.8800	2.6318	7.6187	55.3243	8.0604	225.0000	0.0358	0.0117	0.0339	0.0125	167.5710
1	KAPAL5	GNYAR5	2	150	19.210	ACSR. HAWK	477 MCM	645	800	800	830	830	0.1370	0.3966	2.8800	2.6318	7.6187	55.3243	8.0604	225.0000	0.0358	0.0117	0.0339	0.0125	167.5710
1	KAPAL5	NSDUA5	1	150	30.910	ACSR. PARTRIDGE	135 mm²	400	400	400	830	830	0.2136	0.4080	2.7660	6.6024	12.6113	85.4971	14.2350	225.0000	0.0633	0.0293	0.0561	0.0192	103.9200
1	KAPAL5	PNGAN5	1	150	17.220	TACSR	477 mm²	645	800	800	1250	1250	0.1370	0.3970	2.8800	2.3591	6.8363	49.5936	7.2319	225.0000	0.0321	0.0105	0.0304	0.0112	167.5710
1	KAPAL5	BRTI5	1	150	38.170	TACSR	135 mm²	400	800	800	1250	1250	0.2136	0.4084	2.7970	8.1531	15.5886	106.7615	17.5920	225.0000	0.0782	0.0362	0.0693	0.0240	103.9200
1	GNYAR5	SANUR5	1	150	15.500	ACSR. PARTRIDGE	135 mm²	400	200	1000	830	830	0.2136	0.4080	2.7970	3.5244	6.7320	46.1505	7.5988	225.0000	0.0338	0.0157	0.0299	0.0104	103.9200
1	GNYAR5	SANUR5	2	150	15.380	ACSR. HAWK	477 MCM	400	800	1000	830	830	0.2136	0.4064	2.7970	3.4988	6.6896	45.8149	7.5493	225.0000	0.0336	0.0156	0.0297	0.0103	103.9200
1	GNYAR5	AMPRA5	1	150	33.760	ACSR. HAWK	477 MCM	645	400	1000	830	830	0.1370	0.3966	2.8800	4.6251	13.3892	97.2288	14.1668	225.0000	0.0630	0.0206	0.0595	0.0219	167.5710
1	GNYAR5	AMPRA5	2	150	33.760	ACSR. HAWK	477 MCM	645	1000	1000			0.1370	0.3966	2.8800	4.6251	13.3892	97.2288	14.1656	225.0000	0.0630	0.0206	0.0595	0.0219	167.5710
1	SANUR5	PGRAN5	1	150	7.740	ACSR. PARTRIDGE	135 mm²	400	400	400	830	1250	0.2136	0.4080	2.7970	1.6533	3.1579	21.6468	3.5645	225.0000	0.0158	0.0073	0.0140	0.0049	103.9200
1	SANUR5	PGRAN5	2	150	7.750	ACSR. HAWK	477 MCM	400	1000	1000	830	1250	0.2136	0.4080	2.7970	1.6554	3.1620	21.6768	3.5691	225.0000	0.0159	0.0074	0.0141	0.0049	103.9200

DATA KARAKTERISTIK TRANSMISI UPT & SRB

Jml Sirkit	Dari	Ke	No. Sirkit	Data Penghantar: Teg (kV)	Data Penghantar: Jarak (Km)	Data Penghantar: Jenis		I Nom (Amp)	Tap CT Dari	Tap CT Ke	Tap WT Dari	Tap WT Ke	R (Ohm/km)	X (Ohm/km)	B (MyS/km)	R (Ohm)	X (Ohm)	B (Ohm)	Z act (Ohm)	Z base (Ohm)	Z pu (Ohm)	R pu (Ohm)	X pu (Ohm)	B pu (Ohm)	Daya (MVA)
				(1)	(2)								(3)	(4)	(5)	(6)=(2)x(3)	(7)=(2)x(4)	(8)=(2)x(5)	(9)=akar $(6)^2+(7)^2$	(10)=$(1)^2$/100	(11)=(9)/(10)	(12)=(6)/(10)	(13)=(7)/(10)	(14)=(8)/(10)/$(6.1975)^2$	(15)=(1)x(A)x1.732/1000
Sub Region Bali																									
1	NSDUA5	PGRAN5	1	150	13.410	ACSR. PARTRIDGE	135 mm²	400	200	200	830	830	0.2136	0.4084	2.7970	2.8644	5.4766	37.5078	6.1805	225.0000	0.0275	0.0127	0.0243	0.0084	103.9200
1	BRITI5	PMRON5	1	150	20.430	ACSR. PARTRIDGE	135 mm²	400	800	200	830	830	0.2136	0.4084	2.7970	4.3638	8.3436	57.1427	9.4159	225.0000	0.0418	0.0194	0.0371	289.3302	0.6928
Pengembangan																									
1	BTRTI5	PMRON5	1	150	19.280	ACSR. HAWK	135 mm²	400	200	200	830	830	0.2136	0.4080	2.7560	4.1182	7.8662	53.3285	8.8790	225.0000	0.0395	0.0183	0.0350	0.0120	103.9200
1	BTRTI5	PMRON5	1	150	19.280	ACSR. HAWK	135 mm²	400	200	200	830	830	0.2136	0.4080	2.7560	4.1182	7.8662	53.3285	8.8790	225.0000	0.0395	0.0183	0.0350	0.0120	103.9200
1	KAPAL5	PBIAN5	1	150	9.120	ACSR. PARTRIDGE	477 mm2	645	400	1000	830	830	0.1370	0.3966	2.8300	1.2494	3.6170	26.2656	3.8267	225.0000	0.0170	0.0056	0.0161	0.0059	167.5710
1	PBIAN5	PGRAN5	1	150	7.600	ACSR. PARTRIDGE	477 MCM	645	400	1000	830	830	0.1370	0.3966	2.8300	1.0412	3.0142	21.8880	3.1889	225.0000	0.0142	0.0046	0.0134	0.0049	167.5710
Informasi																									
1	BTRTI5	PMRON5	1	150	19.280	ACSR. PARTRIDGE	135 mm²	400	200	200	830	830	0.2136	0.4080	2.7560	4.1182	7.8662	53.3285	8.8790	225.0000	0.0395	0.0183	0.0350	0.0120	103.9200
1	BTRTI5	PMRON5	1	150	19.280	ACSR. PARTRIDGE	135 mm²	400	200	200	830	830	0.2136	0.4080	2.7560	4.1182	7.8662	53.3285	8.8790	225.0000	0.0395	0.0183	0.0350	0.0120	103.9200
Rencana Up-rating																									
1	KAPAL5	PNGAN5	1	150	17.220	TACSR	477 mm²	645	800	800	1250	1250	0.1370	0.3970	2.8300	2.3591	6.8363	49.5936	7.2319	225.0000	0.0321	0.0105	0.0304	0.0112	167.5710
1	KAPAL5	BRITI5	1	150	38.170	TACSR	135 mm²	400	800	800	1250	1250	0.2136	0.4084	2.7970	8.1531	15.5886	106.7615	17.5920	225.0000	0.0782	0.0362	0.0693	0.0240	103.9200
1	BTRTI5	PNGAN5	1	150	28.010	TACSR	160 mm²	400	800	800	1250	1250	0.1370	0.3970	2.8300	3.8374	11.1200	80.6688	11.7635	225.0000	0.0523	0.0171	0.0494	0.0182	103.9200
1	KAPAL5	PNGAN5	1	150	17.220	TACSR	477 mm²	645	800	800	1250	1250	0.1370	0.3970	2.8300	2.3591	6.8363	49.5936	7.2319	225.0000	0.0321	0.0105	0.0304	0.0112	167.5710
1	KAPAL5	BRITI5	1	150	38.170	TACSR	135 mm²	400	800	800	1250	1250	0.2136	0.4084	2.7970	8.1531	15.5886	106.7615	17.5920	225.0000	0.0782	0.0362	0.0693	0.0240	103.9200
1	BTRTI5	PNGAN5	1	150	28.010	TACSR	160 mm²	400	800	800	1250	1250	0.1370	0.3970	2.8300	3.8374	11.1200	80.6688	11.7635	225.0000	0.0523	0.0171	0.0494	0.0182	103.9200

LISTING PROGRAM

```
program OpflIPSOM;

uses
  Forms,
  uAbout in 'uAbout.pas' {frmAbout},
  uHasil in 'uHasil.pas' {frmHasil},
  uInputLFChild in 'uInputLFChild.pas',
  uMenu in 'uMenu.pas' {frmMenu},
  uGenetic in 'PSO\uGenetic.pas',
  uIEP in 'PSO\uIEP.pas',
  uObjFunc in 'PSO\uObjFunc.pas',
  uPSO in 'PSO\uPSO.pas';

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TfrmMenu, frmMenu);
  Application.CreateForm(TfrmHasil, frmHasil);
  Application.CreateForm(TfrmAbout, frmAbout);
  Application.Run;
end.

unit uAbout;

interface

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

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

var
  frmAbout: TfrmAbout;

implementation

{$R * .dfm}
```

```
end

unit uGenetic;

interface

uses uUtils,uRandom,uObjFunc,uHasil;

type
 TIndividu=record
  chrom:bArr2;
  fitness:double;
 end;

 TPopulasi=array of TIndividu;

 TIndiDouble1=record
  chrom:dArr1;
  fitness:double;
 end;

 TPopDouble1=array of TIndiDouble1;

 TGenetic=class
 private
  FMaxGen,FPopSize,FLength:integer;
  FPcross,FPmutasi,FKa,FPflip:double;
  function getMin:dArr1;
  function getAvg:dArr1;
  function getMax:dArr1;
 protected
  FMin,FAvg,FMax:dArr1;
  FRandom:TRandomu;
 public
  constructor Create(const rMaxGen,rPopSize,rLength:integer;
          const rPcross,rPmutasi,rKa,rPflip:double);
  destructor Destroy;override;
  property MaxGen:integer read FMaxGen write FMaxGen;
  property PopSize:integer read FPopSize write FPopSize;
  property Length:integer read FLength write FLength;
  property Pcross:double read FPcross write FPcross;
  property Pmutasi:double read FPmutasi write FPmutasi;
  property Pflip:double read FPflip write FPflip;
  property Ka:double read FKa write FKa;
  property Min:dArr1 read getMin;
  property Avg:dArr1 read getAvg;
  property Max:dArr1 read getMax;
 end;

 TGenetic2=class(TGenetic)
```

```
  private
    FParam:integer;
    FParent,FChild:TPopulasi;
    FBestIndi:TIndividu;
    FMin1,FAvg1,FMax1,FSumFitness:double;
    function getIndividu(const rIndi:TIndividu):TIndividu;
    function FindIndiMax:TIndividu;
    procedure InitParent;
    procedure Statistik;
    function Seleksi:integer;
    function Mutasi(const rAllele:boolean):boolean;
    procedure doCrossover(const rType:integer;
            const rParent1,rParent2:bArr2;
            var rChild1,rChild2:bArr2);
    procedure Generasi(const rType:integer);
    procedure FindNewParent;
    procedure doHitung;
    function getBestChrom:bArr2;
  public
    constructor Create(const rMaxGen,rPopSize,rLength,rParam:integer;
            const rPcross,rPmutasi,rKa,rPflip:double);
    property Param:integer read FParam write FParam;
    property BestChrom:bArr2 read getBestChrom;
  end;

implementation

{ TGenetic }

//constructor
constructor TGenetic.Create(const rMaxGen,rPopSize,rLength:integer;
        const rPcross,rPmutasi,rKa,rPflip:double);
begin
  inherited Create;
  FMaxGen:=rMaxGen;
  FPopSize:=rPopSize;
  FLength:=rLength;
  FPcross:=rPcross;
  FPmutasi:=rPmutasi;
  FKa:=rKa;
  FPflip:=rPflip;
  FRandom:=TRandomu.Create;
end;

//destructor
destructor TGenetic.Destroy;
begin
  try
    FRandom.Free;
  finally
```

```
    inherited Destroy;
  end;
end;

function TGenetic.getAvg:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
  begin
    result[i]:=FMin[i];
  end;
end;

function TGenetic.getMax:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
  begin
    result[i]:=FAvg[i];
  end;
end;

function TGenetic.getMin: dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
  begin
    result[i]:=FMax[i];
  end;
end;

{ TGenetic2 }

//constructor
constructor TGenetic2.Create(const rMaxGen,rPopSize,rLength,rParam:integer;
        const rPcross,rPmutasi,rKa,rPflip:double);
begin
  inherited Create(rMaxGen,rPopSize,rLength,rPcross,rPmutasi,rKa,rPflip);
  FParam:=rParam;
end;

//data proccessing
function TGenetic2.getIndividu(const rIndi:TIndividu):TIndividu;
var i,j:integer;
begin
  SetLength(result.chrom,FParam,FLength);
  for i:=0 to FParam-1 do
```

```
  begin
    for j:=0 to FLength-1 do
    begin
      result.chrom[i,j]:=rIndi.chrom[i,j];
    end;
  end;
  result.fitness:=rIndi.fitness;
end;

function TGenetic2.FindIndiMax:TIndividu;
var i:integer;
begin
  result:=getIndividu(FParent[0]);
  for i:=0 to FPopSize-1 do
  begin
    if result.fitness<FParent[i].fitness then
    begin
      result:=getIndividu(FParent[i]);
    end;
  end;
end;

procedure TGenetic2.InitParent;
var i,j,k:integer;
begin
  SetLength(FParent,PopSize);
  SetLength(FChild,PopSize);
  SetLength(FMin,MaxGen);
  SetLength(FAvg,MaxGen);
  SetLength(FMax,MaxGen);
  for i:=0 to PopSize-1 do
  begin
    SetLength(FParent[i].chrom,Param,Length);
    SetLength(FChild[i].chrom,Param,Length);
    for j:=0 to Param-1 do
    begin
      for k:=0 to Length-1 do
      begin
        FParent[i].chrom[j,k]:=getFlip(Pflip);
      end;
    end;
   FParent[i].fitness:=Ka,///gObjFunc.doHitung(FParent[i].chrom);
  end;
  SetLength(FBestIndi.chrom,Param,Length);
end;

procedure TGenetic2.Statistik;
var i:integer;
begin
  FMin1:=FParent[0].fitness;
```

```
  FMax1:=FParent[0].fitness;
  FSumFitness:=FParent[0].fitness;
  for i:=1 to PopSize-1 do
  begin
    if FMin1>FParent[i].fitness then
    begin
      FMin1:=FParent[i].fitness;
    end;
    if FMax1<FParent[i].fitness then
    begin
      FMax1:=FParent[i].fitness;
    end;
    FSumFitness:=FSumFitness+FParent[i].fitness;
  end;
  FAvg1:=FSumFitness/PopSize;
end;

function TGenetic2.Seleksi:integer;
var rand,partsum:double;
   i:integer;
begin
  partsum:=0;
  i:=0;
  rand:=FRandom.NextDouble*FSumFitness;
  repeat
    i:=i+1;
    partsum:=partsum+FParent[i-1].fitness;
  until (partsum>rand) or (i=PopSize);
  Result:=i;
end;

function TGenetic2.Mutasi(const rAllele:boolean):boolean;
begin
  if GetFlip(Pmutasi)=true then
  begin
    result:=not rAllele;
  end
  else
  begin
    result:=rAllele;
  end;
end;

procedure TGenetic2.doCrossover(const rType:integer;
        const rParent1,rParent2:bArr2;
        var rChild1,rChild2:bArr2);
var i,j,point1,point2,count:integer;
begin
  SetLength(rChild1,Param.Length);
  SetLength(rChild2,Param.Length);
```

```
if GetFlip(Pcross)=true then
begin
 if rType=1 then
 begin
  point1:=FRandom.NextInt(1,Param*Length-1);
  count:=0;
  for i:=1 to Param do
  begin
   for j:=1 to Length do
   begin
    inc(count);
    if count<=point1 then
    begin
     rChild1[i-1,j-1]:=Mutasi(rParent1[i-1,j-1]);
     rChild2[i-1,j-1]:=Mutasi(rParent2[i-1,j-1]);
    end
    else
    begin
     rChild1[i-1,j-1]:=Mutasi(rParent2[i-1,j-1]);
     rChild2[i-1,j-1]:=Mutasi(rParent1[i-1,j-1]);
    end;
   end;
  end;
 end
 else if rType=2 then
 begin
  point1:=FRandom.NextInt(1,Param*Length-1);
  repeat
   point2:=FRandom.NextInt(1,Param*Length-1);
  until point2<>point1;
  if point2<point1 then
  begin
   Swap(point1,point2);
  end;
  count:=0;
  for i:=1 to Param do
  begin
   for j:=1 to Length do
   begin
    inc(count);
    if count<=point1 then
    begin
     rChild1[i-1,j-1]:=Mutasi(rParent1[i-1,j-1]);
     rChild2[i-1,j-1]:=Mutasi(rParent2[i-1,j-1]);
    end
    else if (count>point1) and (count<=point2) then
    begin
     rChild1[i-1,j-1]:=Mutasi(rParent2[i-1,j-1]);
     rChild2[i-1,j-1]:=Mutasi(rParent1[i-1,j-1]);
    end
```

```
        else if count>point2 then
        begin
          rChild1[i-1,j-1]:=Mutasi(rParent1[i-1,j-1]);
          rChild2[i-1,j-1]:=Mutasi(rParent2[i-1,j-1]);
        end;
      end;
    end;
  end;
 end
 else
 begin
  for i:=0 to Param-1 do
  begin
    for j:=0 to Length-1 do
    begin
      rChild1[i,j]:=Mutasi(rParent1[i,j]);
      rChild2[i,j]:=Mutasi(rParent2[i,j]);
    end;
  end;
 end;
end;

procedure TGenetic2.Generasi(const rType:integer);
var i,mate1,mate2:integer;
begin
 i:=1;
 repeat
  mate1:=Seleksi;
  mate2:=Seleksi;
  doCrossover(rType,FParent[mate1-1].chrom,FParent[mate2-1].chrom,
  FChild[i-1].chrom,FChild[i].chrom);
  FChild[i-1].fitness:=Ka;//gObjFunc.doHitung(FChild[i-1].chrom);
  FChild[i].fitness:=Ka;//gObjFunc.doHitung(FChild[i].chrom);
  i:=i+2;
 until i>PopSize;
end;

procedure TGenetic2.FindNewParent;
var i:integer;
begin
 for i:=0 to PopSize-1 do
 begin
  FParent[i]:=getIndividu(FChild[i]);
 end;
 FParent[PopSize-1]:=getIndividu(FBestIndi);
end;

procedure TGenetic2.doHitung;
var gen,typ,pos:integer;
   tmpIndi:TIndividu;
```

```
begin
  InitParent;
  Statistik;
  FBestIndi:=FindIndiMax;
  gen:=1;
  pos:=MaxGen div 2;
  repeat
    if gen<=pos then
    begin
      typ:=1;
    end
    else
    begin
      typ:=2;
    end;
    Generasi(typ);
    FindNewParent;
    Statistik;
    tmpIndi:=FindIndiMax;
    if FBestIndi.fitness<tmpIndi.fitness then
    begin
      FBestIndi:=getIndividu(tmpIndi);
    end;
    FMin[gen-1]:=FMin1;
    FAvg[gen-1]:=FAvg1;
    FMax[gen-1]:=FMax1;
    frmHasil.pbGen.StepBy(1);
    inc(gen);
  until (gen>MaxGen);
end;

function TGenetic2.getBestChrom:bArr2;
var i,j:integer;
begin
  doHitung;
  SetLength(result,Param,Length);
  for i:=0 to Param-1 do
  begin
    for j:=0 to Length-1 do
    begin
      result[i,j]:=FBestIndi.chrom[i,j];
    end;
  end;
end;

end.

unit uHasil;

interface
```

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

type
  TfrmHasil = class(TForm)
    PageControl1: TPageControl;
    TabSheet3: TTabSheet;
    TabSheet4: TTabSheet;
    TabSheet9: TTabSheet;
    Panel1: TPanel;
    btnClose: TButton;
    btnLFAwal: TButton;
    TabSheet1: TTabSheet;
    GroupBox6: TGroupBox;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    lblGen: TLabel;
    lblLoad: TLabel;
    lblLoss: TLabel;
    edtSumGen: TEdit;
    edtSumLoad: TEdit;
    edtSumLoss: TEdit;
    edtIterasi: TEdit;
    edtTime: TEdit;
    TabSheet2: TTabSheet;
    btnHitung: TButton;
    TabSheet5: TTabSheet;
    TabSheet6: TTabSheet;
    TabSheet7: TTabSheet;
    TabSheet8: TTabSheet;
    Chart1: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    fgBus2: TStringGrid;
    fgBranch: TStringGrid;
    fgBranch2: TStringGrid;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    lblGen2: TLabel;
    lblLoad2: TLabel;
```

lblLoss2: TLabel;
edtSumGen2: TEdit;
edtSumLoad2: TEdit;
edtSumLoss2: TEdit;
edtIterasi2: TEdit;
edtTime2: TEdit;
fgBus: TStringGrid;
GroupBox3: TGroupBox;
fgHasil: TStringGrid;
pbGen: TProgressBar;
GroupBox4: TGroupBox;
btnUseDefault: TButton;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
edtKp: TEdit;
edtKq: TEdit;
edtKv: TEdit;
Label24: TLabel;
edtVmin: TEdit;
edtVmax: TEdit;
TabSheet10: TTabSheet;
Chart2: TChart;
Series3: TLineSeries;
Series4: TLineSeries;
Series5: TLineSeries;
GroupBox7: TGroupBox;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
lblLoss21: TLabel;
lblLoss22: TLabel;
lblLoss23: TLabel;
edtBaseLoss: TEdit;
edtOptLoss: TEdit;
edtSelisihLoss: TEdit;
Label25: TLabel;
edtTapTrafoMin: TEdit;
edtTapTrafoMax: TEdit;
Label29: TLabel;
edtStepTrafo: TEdit;
Label30: TLabel;
edtKs: TEdit;
GroupBox2: TGroupBox;
Label17: TLabel;
Label15: TLabel;
Label16: TLabel;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;

```
    Label31: TLabel;
    Label32: TLabel;
    edtMaxIterasi: TEdit;
    edtPopSize: TEdit;
    edtc1: TEdit;
    edtc2: TEdit;
    edtwmin: TEdit;
    edtwmax: TEdit;
    edtdV: TEdit;
    procedure btnCloseClick(Sender: TObject);
    procedure btnLFAwalClick(Sender: TObject);
    procedure btnHitungClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnUseDefaultClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmHasil: TfrmHasil;

implementation

uses uUtils, uLoadflow, uNewtonRaphson, uComplex,
     uSensitivitas,uObjFunc,uPSO;

{$R *.dfm}

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

procedure TfrmHasil.btnLFAwalClick(Sender: TObject);
var i,Nbus,Nsal,ia,ja:integer;
    V,Sg,SL:CxArr1;
    Cap:dArr1;
    Lc,Tr:dArr2;
    TypBus:iArr1;
    Z,Tp,Alir,Arus:CxArr2;
    mulai,selesai,selang:TDateTime;
    jam,menit,detik,mdetik:word;
begin
  DecodeCommDataToLFData(gBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,
  gBranch,Z,Tp,Lc,Tr);
  mulai:=time;
  NewtonRaphson(gParamLF,V,Sg,SL,Cap,TypBus,Z,Tp,Alir,Arus,
  Lc,Tr);
```

```
  selesai :=time;
  selang:=selesai-mulai;
  Series1.Clear;
  Series2.Clear;
  for i:=0 to high(gBus) do
  begin
    fgBus.Cells[0,i+1]:=IntToStr(i+1);
    fgBus.Cells[1,i+1]:=RealToStr(V[i].real,5);
    fgBus.Cells[2,i+1]:=RealToStr(V[i].imag,5);
    fgBus.Cells[3,i+1]:=RealToStr(Sg[i].real,3);
    fgBus.Cells[4,i+1]:=RealToStr(Sg[i].imag,3);
    fgBus.Cells[5,i+1]:=RealToStr(SL[i].real,3);
    fgBus.Cells[6,i+1]:=RealToStr(SL[i].imag,3);
    fgBus.Cells[7,i+1]:=RealToStr(Cap[i],3);
    fgBus.Cells[8,i+1]:=IntToStr(TypBus[i]);
    Series1.Add(V[i].real,IntToStr(i+1));
  end;
  for i:=0 to high(gBranch) do
  begin
    ia:=gBranch[i].dari-1;
    ja:=gBranch[i].ke-1;
    fgBranch.Cells[0,i+1]:=IntToStr(i+1);
    fgBranch.Cells[1,i+1]:=IntToStr(gBranch[i].dari);
    fgBranch.Cells[2,i+1]:=IntToStr(gBranch[i].ke);
    fgBranch.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
    fgBranch.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
    fgBranch.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
    fgBranch.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
    fgBranch.Cells[7,i+1]:=IntToStr(gBranch[i].ke);
    fgBranch.Cells[8,i+1]:=IntToStr(gBranch[i].dari);
    fgBranch.Cells[9,i+1]:=RealToStr(Alir[ja,ia].real,3);
    fgBranch.Cells[10,i+1]:=RealToStr(Alir[ja,ia].imag,3);
    fgBranch.Cells[11,i+1]:=RealToStr(Arus[ja,ia].real,3);
    fgBranch.Cells[12,i+1]:=RealToStr(Arus[ja,ia].imag,3);
  end;
  edtSumGen.Text:=toStringJ(gParamLF.SumGen,3);
  edtSumLoad.Text:=toStringJ(gParamLF.SumLoad,3);
  edtSumLoss.Text:=toStringJ(gParamLF.SumLoss,3);
  edtBaseLoss.Text:=RealToStr(gParamLF.SumLoss.real,3);
  edtIterasi.Text:=IntToStr(gParamLF.Iterasi);
  DecodeTime(selang,jam,menit,detik,mdetik);
  edtTime.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
            IntToStr(detik)+':'+IntToStr(mdetik);
end;

procedure TfrmHasil.btnHitungClick(Sender: TObject);
var sa,i,Iterasi,c1,c2,PopSize,Length:integer;
   BatasV,BatasTrafo,w:TBatas;
   mulai,selesai,selang:TDateTime;
   jam,menit,detik,mdetik:word;
```

```
  BestChrom,Min,Avg,Max:dArr1;
  LBus:TBusArr1;
  LBranch:TBranchArr1;
  CostPLN,CostOpt,Kp,Kq,Kv,Ks,StepTrafo:double;
  Pmutasi,Ka,Pr,Lamda,A,dV:double;
  pso:THPSOM;
  BatasChrom:TBatasArr1;
begin
 BatasV.min:=StrToFloat(edtVmin.Text);
 BatasV.max:=StrToFloat(edtVmax.Text);
 BatasTrafo.min:=StrToFloat(edtTapTrafoMin.Text);
 BatasTrafo.max:=StrToFloat(edtTapTrafoMax.Text);
 StepTrafo:=StrToFloat(edtStepTrafo.Text);
 Kp:=StrToFloat(edtKp.Text);
 Kq:=StrToFloat(edtKq.Text);
 Kv:=StrToFloat(edtKv.Text);
 Ks:=StrToFloat(edtKs.Text);
 mulai:=time;
 NewtonRaphson(gBus,gBranch,gParamLF);
 gObjFunc:=TObjFunc.Create(Kp,Kq,Kv,Ks,StepTrafo,
        BatasV,BatasTrafo);
 CostPLN:=gObjFunc.FindTotalCostGen(gBus);
 Length:=gObjFunc.FindLength;
 BatasChrom:=gObjFunc.FindBatasChrom;
 Iterasi:=StrToInt(edtMaxIterasi.Text);
 pbGen.Max:=Iterasi;
 PopSize:=StrToInt(edtPopSize.Text);
 c1:=StrToInt(edtC1.Text);
 c2:=StrToInt(edtC2.Text);
 dV:=StrToFloat(edtdV.Text);
 w.min:=StrToFloat(edtwmin.Text);
 w.max:=StrToFloat(edtwmax.Text);
 pso:=THPSOM.Create(Iterasi,PopSize,c1,c2,Length,dV,w,BatasChrom);
 BestChrom:=PSO.BestSwarm;
 Min:=PSO.Global;
 Series3.Clear;
 for i:=0 to high(Min) do
 begin
  Series3.Add(Min[i],IntToStr(i));
 end;
 pso.Free;
 fgHasil.RowCount:=high(BestChrom)+2;
 sa:=0;
 for i:=0 to high(gBus) do
 begin
  if gBus[i].typeBus=2 then
  begin
   fgHasil.Cells[0,sa+1]:=IntToStr(sa+1);
   fgHasil.Cells[1,sa+1]:='Pg'+IntToStr(i+1);
   fgHasil.Cells[2,sa+1]:=FormatFloat('#,##0.000',BestChrom[sa]);
```

```
    inc(sa);
  end;
end;
for i:=0 to high(gBus) do
begin
  if gBus[i].typeBus<>3 then
  begin
    fgHasil.Cells[0,sa+1]:=IntToStr(sa+1);
    fgHasil.Cells[1,sa+1]:='Vg'+IntToStr(i+1);
    fgHasil.Cells[2,sa+1]:=FormatFloat('#,##0.000',BestChrom[sa]);
    inc(sa);
  end;
end;
for i:=0 to high(gBus) do
begin
  if gBus[i].typeBus=3 then
  begin
    if gBus[i].Qgen>0 then
    begin
      fgHasil.Cells[0,sa+1]:=IntToStr(sa+1);
      fgHasil.Cells[1,sa+1]:='Qg'+IntToStr(i+1);
      fgHasil.Cells[2,sa+1]:=FormatFloat('#,##0.000',BestChrom[sa]);
      inc(sa);
    end;
  end;
end;
for i:=0 to high(gBranch) do
begin
  if gBranch[i].Tr<>0 then
  begin
    fgHasil.Cells[0,sa+1]:=IntToStr(sa+1);
    fgHasil.Cells[1,sa+1]:='Tr'+IntToStr(gBranch[i].dari)+
                    IntToStr(gBranch[i].ke);
    fgHasil.Cells[2,sa+1]:=FormatFloat('#,##0.000',BestChrom[sa]);
    inc(sa);
  end;
end;
gObjFunc.doHitungAkhir(BestChrom,LBus,LBranch,CostOpt);
gObjFunc.Free;
selesai:=time;
selang:=selesai-mulai;
Series2.Clear;
for i:=0 to high(gBus) do
begin
  fgBus2.Cells[0,i+1]:=IntToStr(i+1);
  fgBus2.Cells[1,i+1]:=RealToStr(LBus[i].absV,5);
  fgBus2.Cells[2,i+1]:=RealToStr(LBus[i].sudV,5);
  fgBus2.Cells[3,i+1]:=RealToStr(LBus[i].Pgen,3);
  fgBus2.Cells[4,i+1]:=RealToStr(LBus[i].Qgen,3);
  fgBus2.Cells[5,i+1]:=RealToStr(LBus[i].PL,3);
```

```
    fgBus2.Cells[6,i+1]:=RealToStr(LBus[i].QL,3);
    fgBus2.Cells[7,i+1]:=RealToStr(LBus[i].Cap,3);
    fgBus2.Cells[8,i+1]:=IntToStr(LBus[i].typeBus);
    Series2.Add(LBus[i].absV,IntToStr(i+1));
  end;
  for i:=0 to high(LBranch) do
  begin
    fgBranch2.Cells[0,i+1]:=IntToStr(i+1);
    fgBranch2.Cells[1,i+1]:=IntToStr(LBranch[i].dari);
    fgBranch2.Cells[2,i+1]:=IntToStr(LBranch[i].ke);
    fgBranch2.Cells[3,i+1]:=RealToStr(LBranch[i].Sij.real,3);
    fgBranch2.Cells[4,i+1]:=RealToStr(LBranch[i].Sij.imag,3);
    fgBranch2.Cells[5,i+1]:=RealToStr(LBranch[i].Aij.real,3);
    fgBranch2.Cells[6,i+1]:=RealToStr(LBranch[i].Aij.imag,3);
    fgBranch2.Cells[7,i+1]:=IntToStr(LBranch[i].ke);
    fgBranch2.Cells[8,i+1]:=IntToStr(LBranch[i].dari);
    fgBranch2.Cells[9,i+1]:=RealToStr(LBranch[i].Sji.real,3);
    fgBranch2.Cells[10,i+1]:=RealToStr(LBranch[i].Sji.imag,3);
    fgBranch2.Cells[11,i+1]:=RealToStr(LBranch[i].Aji.real,3);
    fgBranch2.Cells[12,i+1]:=RealToStr(LBranch[i].Aji.imag,3);
  end;
  edtSumGen2.Text:=toStringJ(gParamLF.SumGen,3);
  edtSumLoad2.Text:=toStringJ(gParamLF.SumLoad,3);
  edtSumLoss2.Text:=toStringJ(gParamLF.SumLoss,3);
  edtOptLoss.Text:=RealToStr(gParamLF.SumLoss.real,3);
  edtSelisihLoss.Text:=RealToStr((strToFloat(edtBaseLoss.Text)-
                 StrToFloat(edtOptLoss.Text)),3);
  edtIterasi2.Text:=IntToStr(4);
  DecodeTime(selang,jam,menit,detik,mdetik);
  edtTime2.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
            IntToStr(detik)+':'+IntToStr(mdetik);
  //
end;

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
  fgHasil.Cells[0,0]:='No';
  fgHasil.Cells[1,0]:='Bus';
  fgHasil.Cells[2,0]:='Pgen';
end;

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
  edtMaxIterasi.Text:='100';
  edtPopSize.Text:='100';
  edtc1.Text:='2';
  edtc2.Text:='2';
  edtwmin.Text:='0.4';
  edtwmax.Text:='0.9';
  edtdV.Text:='0.5';
```

```
  edtKp.Text:='1000';
  edtKv.Text:='10000000';
  edtKq.Text:='1000';
  edtKs.Text:='1000';
  edtVmin.Text:='0.95';
  edtVmax.Text:='1.05';
  edtTapTrafoMin.Text:='0.9';
  edtTapTrafoMax.Text:='1.1';
  edtStepTrafo.Text:='0.001';
  btnHitung.Enabled:=true;
end;

end.

unit uIEP;

interface

uses uUtils,uGenetic,uLoadflow,uObjFunc,Math,uHasil;

type
  TPopDouble2=array of array of TIndiDouble1;

  TIEP=class(TGenetic)
  private
    FNM,FNgen:integer;
    FPr,FLamd,Fa,FMin1,FAvg1,FMax1,FSumFitness:double;
    FParent,FChild:TPopDouble2;
    FBestIndi:TIndiDouble1;
    FBatasChrom:TBatasArr1;
    function getIndividu(const rIndi:TIndiDouble1):TIndiDouble1;
    function getBatasChrom(const rNoM:integer):TBatasArr1;
    procedure InitParent;
    procedure Statistik;
    function getBestIndi:TIndiDouble1;
    function getInitTemp:dArr1;
    function getTempNow(const rR:integer;
          const rT0:dArr1):dArr1;
    function Mutasi(const rAllele:double;
          const rR:integer;
          const rBatasChrom:TBatas):double;
    procedure Generasi(const rR:integer);
    procedure Seleksi(const rTr:dArr1);
    procedure doHitung;
    function getBestChrom:dArr1;
  public
    constructor Create(const rMaxGen,rPopSize,rLength,rNM:integer;
            const rPmutasi,rKa,rPr,rLamd,rA:double;
            const rBatasChrom:TBatasArr1);
    property NM:integer read FNM write FNM;
```

```
  property BestChrom:dArr1 read getBestChrom;
 end;

implementation

//constructor
constructor TIEP.Create(const rMaxGen,rPopSize,rLength,rNM:integer;
        const rPmutasi,rKa,rPr,rLamd,rA:double;
        const rBatasChrom:TBatasArr1);
var i,j:integer;
   Pcross,Pflip:double;
begin
 Pcross:=0.75;
 Pflip:=0.5;
 inherited Create(rMaxGen,rPopSize,rLength,Pcross,rPmutasi,rKa,Pflip);
 FNM:=rNM;
 FPr:=rPr;
 FLamd:=rLamd;
 FA:=rA;
 SetLength(FBatasChrom,Length);
 for i:=0 to Length-1 do
 begin
  FBatasChrom[i].min:=rBatasChrom[i].min;
  FBatasChrom[i].max:=rBatasChrom[i].max;
 end;
 FNgen:=0;
 for i:=0 to high(gBus) do
 begin
  if gBus[i].typeBus=2 then
  begin
   inc(FNgen);
  end;
 end;
 SetLength(FParent,FNM,PopSize);
 SetLength(FChild,FNM,PopSize);
 for i:=0 to FNM-1 do
 begin
  for j:=0 to PopSize-1 do
  begin
   SetLength(FParent[i,j].chrom,Length);
   SetLength(FChild[i,j].chrom,Length);
  end;
 end;
 SetLength(FBestIndi.chrom,Length);
end;

//data processing
function TIEP.getIndividu(const rIndi:TIndiDouble1):TIndiDouble1;
var i:integer;
begin
```

```
  SetLength(result.chrom,Length);
  for i:=0 to Length-1 do
  begin
    result.chrom[i]:=rIndi.chrom[i];
  end;
  result.fitness:=rIndi.fitness;
end;

function TIEP.getBatasChrom(const rNoM:integer):TBatasArr1;
var i:integer;
begin
  SetLength(result,Length);
  for i:=0 to Length-1 do
  begin
    result[i].min :=FBatasChrom[i].min;
    result[i].max :=FBatasChrom[i].max;
  end;
  if FNM=2 then
  begin
    if rNoM=0 then
    begin
      result[0].min:=FBatasChrom[0].min;
      result[0].max:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    end
    else
    begin
      result[0].min:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
      result[0].max:=FBatasChrom[0].max;
    end;
  end
  else if FNM=4 then
  begin
    if rNoM=0 then
    begin
      result[0].min:=FBatasChrom[0].min;
      result[0].max:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
      result[1].min:=FBatasChrom[1].min;
      result[1].max:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    end
    else if rNoM=1 then
    begin
      result[0].min:=FBatasChrom[0].min;
      result[0].max:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
      result[1].min:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
```

```
    result[1].max:=FBatasChrom[1].max;
   end
   else if rNoM=2 then
   begin
    result[0].min:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[0].max:=FBatasChrom[0].max;
    result[1].min:=FBatasChrom[1].min;
    result[1].max:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
   end
   else
   begin
    result[0].min:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[0].max:=FBatasChrom[0].max;
    result[1].min:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[1].max:=FBatasChrom[1].max;
   end;
  end
  else if FNM=8 then
  begin
   if rNoM=0 then
   begin
    result[0].min:=FBatasChrom[0].min;
    result[0].max:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[1].min:=FBatasChrom[1].min;
    result[1].max:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[2].min:=FBatasChrom[2].min;
    result[2].max:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
   end
   else if rNoM=1 then
   begin
    result[0].min:=FBatasChrom[0].min;
    result[0].max:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[1].min:=FBatasChrom[1].min;
    result[1].max:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[2].min:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
    result[2].max:=FBatasChrom[2].max;
   end
   else if rNoM=2 then
   begin
    result[0].min:=FBatasChrom[0].min;
```

```
    result[0].max:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[1].min:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[1].max:=FBatasChrom[1].max;
    result[2].min:=FBatasChrom[2].min;
    result[2].max:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
  end
  else if rNoM=3 then
  begin
    result[0].min:=FBatasChrom[0].min;
    result[0].max:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[1].min:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[1].max:=FBatasChrom[1].max;
    result[2].min:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
    result[2].max:=FBatasChrom[2].max;
  end
  else if rNoM=4 then
  begin
    result[0].min:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[0].max:=FBatasChrom[0].max;
    result[1].min:=FBatasChrom[1].min;
    result[1].max:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[2].min:=FBatasChrom[2].min;
    result[2].max:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
  end
  else if rNoM=5 then
  begin
    result[0].min:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[0].max:=FBatasChrom[0].max;
    result[1].min:=FBatasChrom[1].min;
    result[1].max:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[2].min:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
    result[2].max:=FBatasChrom[2].max;
  end
  else if rNoM=6 then
  begin
    result[0].min:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[0].max:=FBatasChrom[0].max;
```

```
    result[1].min:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[1].max:=FBatasChrom[1].max;
    result[2].min:=FBatasChrom[2].min;
    result[2].max:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
   end
   else
   begin
    result[0].min:=FBatasChrom[0].min+(FBatasChrom[0].max-
FBatasChrom[0].min)/2;
    result[0].max:=FBatasChrom[0].max;
    result[1].min:=FBatasChrom[1].min+(FBatasChrom[1].max-
FBatasChrom[1].min)/2;
    result[1].max:=FBatasChrom[1].max;
    result[2].min:=FBatasChrom[2].min+(FBatasChrom[2].max-
FBatasChrom[2].min)/2;
    result[2].max:=FBatasChrom[2].max;
   end;
  end;
end;

procedure TIEP.InitParent;
var i,j,k:integer;
   LBatas:TBatasArr1;
begin
  SetLength(LBatas,Length);
  SetLength(FMin,MaxGen);
  SetLength(FAvg,MaxGen);
  SetLength(FMax,MaxGen);
  for i:=0 to FNM-1 do
  begin
   LBatas:=getBatasChrom(i);
   for j:=0 to PopSize-1 do
   begin
    for k:=0 to Length-1 do
    begin
     FParent[i,j].chrom[k]:=FRandom.NextDouble(LBatas[k].min,LBatas[k].max);
    end;
    FParent[i,j].fitness:=gObjFunc.doHitung(FParent[i,j].chrom);
   end;
  end;
end;

procedure TIEP.Statistik;
var i,j:integer;
begin
  FMin1:=FParent[0,0].fitness;
  FMax1:=FParent[0,0].fitness;
  FSumFitness:=0;
```

```
  for i:=0 to FNM-1 do
  begin
    for j:=0 to PopSize-1 do
    begin
      if FMin1>FParent[i,j].fitness then
      begin
        FMin1:=FParent[i,j].fitness;
      end;
      if FMax1<FParent[i,j].fitness then
      begin
        FMax1:=FParent[i,j].fitness;
      end;
      FSumFitness:=FSumFitness+FParent[i,j].fitness;
    end;
  end;
  FAvg1:=FSumFitness/(FNM*PopSize);
end;

function TIEP.getBestIndi:TIndiDouble1;
var i,j:integer;
    tmpIndi:TIndiDouble1;
begin
  result:=getIndividu(FParent[0,0]);
  for i:=0 to FNM-1 do
  begin
    tmpIndi:=getIndividu(FParent[i,0]);
    for j:=1 to PopSize-1 do
    begin
      if tmpIndi.fitness>FParent[i,j].fitness then
      begin
        tmpIndi:=getIndividu(FParent[i,j]);
      end;
    end;
    if result.fitness>tmpIndi.fitness then
    begin
      result:=getIndividu(tmpIndi);
    end;
  end;
end;

function TIEP.getInitTemp:dArr1;
var i,j:integer;
    Fmax,Fmin:double;
begin
  SetLength(result,FNM);
  for i:=0 to FNM-1 do
  begin
    Fmax:=FParent[i,0].fitness;
    Fmin:=FParent[i,0].fitness;
    for j:=0 to PopSize-1 do
```

```
    begin
      if Fmax<FParent[i,j].fitness then
      begin
        Fmax:=FParent[i,j].fitness;
      end;
      if Fmin>FParent[i,j].fitness then
      begin
        Fmin:=FParent[i,j].fitness;
      end;
    end;
    result[i]:=-(Fmax-Fmin)/ln(FPr);
  end;
end;

function TIEP.getTempNow(const rR:integer;
        const rT0:dArr1):dArr1;
var i:integer;
begin
  SetLength(result,FNM);
  for i:=0 to FNM-1 do
  begin
    result[i]:=power(FLamd,(rR-1))*rT0[i];
  end;
end;

function TIEP.Mutasi(const rAllele:double;
        const rR:integer;
        const rBatasChrom:TBatas):double;
var tho,rand:double;
begin
  //repeat
    tho:=power(Fa,(rR-1))*(rBatasChrom.max-rBatasChrom.min);
    if FRandom.NextBoolean(0.5)=true then
    begin
      rand:=FRandom.NextGaussian(0,1);
    end
    else
    begin
      rand:=FRandom.NextCauchy(0,1);
    end;
    result:=rAllele+tho*rand;
    if result>rBatasChrom.max then result:=rBatasChrom.max;
    if result<rBatasChrom.min then result:=rBatasChrom.min;
  //until (result>=rBatasChrom.max) and (result<=rBatasChrom.max);
end;

procedure TIEP.Generasi(const rR:integer);
var i,j,k:integer;
    LBatas:TBatasArr1;
begin
```

```
  SetLength(LBatas,Length);
  for i:=0 to FNM-1 do
  begin
    LBatas:=getBatasChrom(i);
    for j:=0 to PopSize-1 do
    begin
      for k:=0 to Length-1 do
      begin
        if FRandom.NextBoolean(PMutasi)=true then
        begin
          FChild[i,j].chrom[k]:=Mutasi(FParent[i,j].chrom[k],rR,LBatas[k]);
        end
        else
        begin
          FChild[i,j].chrom[k]:=FParent[i,j].chrom[k];
        end;
      end;
      FChild[i,j].fitness:=gObjFunc.doHitung(FChild[i,j].chrom);
    end;
  end;
end;

procedure TIEP.Seleksi(const rTr:dArr1);
var dF,pij,rand:double;
    i,j,no:integer;
    tmpParent:TPopDouble2;
begin
  SetLength(tmpParent,FNM,PopSize);
  for i:=0 to FNM-1 do
  begin
    for j:=0 to PopSize-1 do
    begin
      repeat
        no:=FRandom.NextInt(0,(PopSize-1));
      until no<>j;
      dF:=FChild[i,j].fitness-FParent[i,no].fitness;
      pij:=exp(-dF/rTr[i]);
      if pij>1 then pij:=1;
      rand:=FRandom.NextDouble(0,1);
      if pij=1 then
      begin
        tmpParent[i,j]:=getIndividu(FChild[i,j]);
      end
      else if pij>rand then
      begin
        tmpParent[i,j]:=getIndividu(FChild[i,j]);
      end
      else
      begin
        tmpParent[i,j]:=getIndividu(FParent[i,no]);
```

```
      end;
    end;
  end;
  for i:=0 to FNM-1 do
  begin
    for j:=0 to PopSize-1 do
    begin
      FParent[i,j]:=getIndividu(tmpParent[i,j]);
    end;
  end;
end;

procedure TIEP.doHitung;
var T0,Tr:dArr1;
    r:integer;
    tmpIndi:TIndiDouble1;
begin
  SetLength(Tr,FNM);
  InitParent;
  FBestIndi:=getBestIndi;
  T0:=getInitTemp;
  for r:=1 to MaxGen do
  begin
    Tr:=getTempNow(r,T0);
    Generasi(r);
    Seleksi(Tr);
    Statistik;
    tmpIndi:=getBestIndi;
    if FBestIndi.fitness>tmpIndi.fitness then
    begin
      FBestIndi:=getIndividu(tmpIndi);
    end;
    FMin[r-1]:=FMin1;
    FAvg[r-1]:=FAvg1;
    FMax[r-1]:=FMax1;
    frmHasil.pbGen.StepBy(1);
  end;
end;

function TIEP.getBestChrom:dArr1;
var i:integer;
begin
  doHitung;
  SetLength(result,Length);
  for i:=0 to Length-1 do
  begin
    result[i]:=FBestIndi.chrom[i];
  end;
end;
```

```
end.

unit uInputLFChild;

interface

uses uInputLF,uHasil,SysUtils,Forms;

type
  TfrmInputLFChild=class(TfrmInputLF)
  protected
    procedure ShowHasil;override;
  end;

var frmInput:TfrmInputLFChild;

implementation

uses uLoadflow;

procedure TfrmInputLFChild.ShowHasil;
begin
  try
    if frmHasil=nil then
    begin
      frmHasil:=TfrmHasil.Create(Application);
    end;
    frmHasil.fgBus.Cells[0,0]:='Bus';
    frmHasil.fgBus.Cells[1,0]:='absV (pu)';
    frmHasil.fgBus.Cells[2,0]:='sudV (deg)';
    frmHasil.fgBus.Cells[7,0]:='Sups (pu)';
    frmHasil.fgBus.Cells[8,0]:='Type Bus';
    frmHasil.fgBranch.Cells[0,0]:='No';
    frmHasil.fgBranch.Cells[1,0]:='Dari';
    frmHasil.fgBranch.Cells[2,0]:='Ke';
    frmHasil.fgBranch.Cells[5,0]:='Arus re (A)';
    frmHasil.fgBranch.Cells[6,0]:='Arus im (A)';
    frmHasil.fgBranch.Cells[7,0]:='Dari';
    frmHasil.fgBranch.Cells[8,0]:='Ke';
    frmHasil.fgBranch.Cells[11,0]:='Arus re (A)';
    frmHasil.fgBranch.Cells[12,0]:='Arus im (A)';
    frmHasil.fgBus2.Cells[0,0]:='Bus';
    frmHasil.fgBus2.Cells[1,0]:='absV (pu)';
    frmHasil.fgBus2.Cells[2,0]:='sudV (deg)';
    frmHasil.fgBus2.Cells[7,0]:='Sups (pu)';
    frmHasil.fgBus2.Cells[8,0]:='Type Bus';
    frmHasil.fgBranch2.Cells[0,0]:='No';
    frmHasil.fgBranch2.Cells[1,0]:='Dari';
    frmHasil.fgBranch2.Cells[2,0]:='Ke';
    frmHasil.fgBranch2.Cells[5,0]:='Arus re (A)';
```

```
 frmHasil.fgBranch2.Cells[6,0]:='Arus im (A)';
 frmHasil.fgBranch2.Cells[7,0]:='Dari';
 frmHasil.fgBranch2.Cells[8,0]:='Ke';
 frmHasil.fgBranch2.Cells[11,0]:='Arus re (A)';
 frmHasil.fgBranch2.Cells[12,0]:='Arus im (A)';
 if gParamLF.PKonst=1 then
 begin
  frmHasil.fgBus.Cells[3,0]:='Pg (W)';
  frmHasil.fgBus.Cells[4,0]:='Qg (VAR)';
  frmHasil.fgBus.Cells[5,0]:='PL (W)';
  frmHasil.fgBus.Cells[6,0]:='QL (VAR)';
  frmHasil.fgBranch.Cells[3,0]:='P (W)';
  frmHasil.fgBranch.Cells[4,0]:='Q (VAR)';
  frmHasil.fgBranch.Cells[9,0]:='P (W)';
  frmHasil.fgBranch.Cells[10,0]:='Q (VAR)';
  frmHasil.lblGen.Caption:='VA';
  frmHasil.lblLoad.Caption:='VA';
  frmHasil.lblLoss.Caption:='VA';
  //
  frmHasil.fgBus2.Cells[3,0]:='Pg (W)';
  frmHasil.fgBus2.Cells[4,0]:='Qg (VAR)';
  frmHasil.fgBus2.Cells[5,0]:='PL (W)';
  frmHasil.fgBus2.Cells[6,0]:='QL (VAR)';
  frmHasil.fgBranch2.Cells[3,0]:='P (W)';
  frmHasil.fgBranch2.Cells[4,0]:='Q (VAR)';
  frmHasil.fgBranch2.Cells[9,0]:='P (W)';
  frmHasil.fgBranch2.Cells[10,0]:='Q (VAR)';
  frmHasil.lblGen2.Caption:='VA';
  frmHasil.lblLoad2.Caption:='VA';
  frmHasil.lblLoss2.Caption:='VA';
  frmHasil.lblLoss21.Caption:='W';
  frmHasil.lblLoss22.Caption:='W';
  frmHasil.lblLoss23.Caption:='W';
  //
 end
 else if gParamLF.PKonst=1000 then
 begin
  frmHasil.fgBus.Cells[3,0]:='Pg (kW)';
  frmHasil.fgBus.Cells[4,0]:='Qg (kVAR)';
  frmHasil.fgBus.Cells[5,0]:='PL (kW)';
  frmHasil.fgBus.Cells[6,0]:='QL (kVAR)';
  frmHasil.fgBranch.Cells[3,0]:='P (kW)';
  frmHasil.fgBranch.Cells[4,0]:='Q (kVAR)';
  frmHasil.fgBranch.Cells[9,0]:='P (kW)';
  frmHasil.fgBranch.Cells[10,0]:='Q (kVAR)';
  frmHasil.lblGen.Caption:='kVA';
  frmHasil.lblLoad.Caption:='kVA';
  frmHasil.lblLoss.Caption:='kVA';
  //
  frmHasil.fgBus2.Cells[3,0]:='Pg (kW)';
```

```
 frmHasil.fgBus2.Cells[4,0]:='Qg (kVAR)';
 frmHasil.fgBus2.Cells[5,0]:='PL (kW)';
 frmHasil.fgBus2.Cells[6,0]:='QL (kVAR)';
 frmHasil.fgBranch2.Cells[3,0]:='P (kW)';
 frmHasil.fgBranch2.Cells[4,0]:='Q (kVAR)';
 frmHasil.fgBranch2.Cells[9,0]:='P (kW)';
 frmHasil.fgBranch2.Cells[10,0]:='Q (kVAR)';
 frmHasil.lblGen2.Caption:='kVA';
 frmHasil.lblLoad2.Caption:='kVA';
 frmHasil.lblLoss2.Caption:='kVA';
 frmHasil.lblLoss21.Caption:='kW';
 frmHasil.lblLoss22.Caption:='kW';
 frmHasil.lblLoss23.Caption:='kW';
 //
end
else if gParamLF.PKonst=1000000 then
begin
 frmHasil.fgBus.Cells[3,0]:='Pg (MW)';
 frmHasil.fgBus.Cells[4,0]:='Qg (MVAR)';
 frmHasil.fgBus.Cells[5,0]:='PL (MW)';
 frmHasil.fgBus.Cells[6,0]:='QL (MVAR)';
 frmHasil.fgBranch.Cells[3,0]:='P (MW)';
 frmHasil.fgBranch.Cells[4,0]:='Q (MVAR)';
 frmHasil.fgBranch.Cells[9,0]:='P (MW)';
 frmHasil.fgBranch.Cells[10,0]:='Q (MVAR)';
 frmHasil.lblGen.Caption:='MVA';
 frmHasil.lblLoad.Caption:='MVA';
 frmHasil.lblLoss.Caption:='MVA';
 //
 frmHasil.fgBus2.Cells[3,0]:='Pg (MW)';
 frmHasil.fgBus2.Cells[4,0]:='Qg (MVAR)';
 frmHasil.fgBus2.Cells[5,0]:='PL (MW)';
 frmHasil.fgBus2.Cells[6,0]:='QL (MVAR)';
 frmHasil.fgBranch2.Cells[3,0]:='P (MW)';
 frmHasil.fgBranch2.Cells[4,0]:='Q (MVAR)';
 frmHasil.fgBranch2.Cells[9,0]:='P (MW)';
 frmHasil.fgBranch2.Cells[10,0]:='Q (MVAR)';
 frmHasil.lblGen2.Caption:='MVA';
 frmHasil.lblLoad2.Caption:='MVA';
 frmHasil.lblLoss2.Caption:='MVA';
 frmHasil.lblLoss21.Caption:='MW';
 frmHasil.lblLoss22.Caption:='MW';
 frmHasil.lblLoss23.Caption:='MW';
 //
end;
frmHasil.fgBus.RowCount:=StrToInt(edtNbus.Text)+1;
frmHasil.fgBranch.RowCount:=StrToInt(edtNsal.Text)+1;
frmHasil.fgBus2.RowCount:=StrToInt(edtNbus.Text)+1;
frmHasil.fgBranch2.RowCount:=StrToInt(edtNsal.Text)+1;
frmHasil.ShowModal;
```

```
  finally
    frmHasil.Free;
  end;
end;

end.
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 uInputl,FChild,uComplex, uUtils, uLoadflow, uHasil;

{$R *.dfm}

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

procedure TfrmMenu.btnNewClick(Sender: TObject);
begin
  try
    if frmInput=nil then
```

```
    begin
      frmInput:=TfrmInputLFChild.Create(Application);
    end;
    frmInput.Caption:='Input Data';
    frmInput.btnNext.Caption:='&Save';
    frmInput.ShowModal;
  finally
    frmInput.Free;
  end;
end;

procedure TfrmMenu.btnOpenClick(Sender: TObject);
var NamaFile,Nama:string;
    output:TextFile;
    i,j,Typ,dari,ke,Nbus,Nsal,Param,Ngen,NCable:integer;
    Cap,absV,sudV,Pg,Qg,PL,QL,CapSal,Pmin,Pmax,Harga,Length:double;
    R,X,Lc,Tr,Tu,Su,VKonst,PKonst,Pbase,Vbase,dia:double;
begin
  try
    if OpenDialog1.Execute then
    begin
      NamaFile:=OpenDialog1.FileName;
      AssignFile(output,NamaFile);
      Reset(output);
      Readln(output,Nbus);
      Readln(output,Nsal);
      Readln(output,Vbase);
      Readln(output,VKonst);
      Readln(output,Pbase);
      Readln(output,PKonst);
      Readln(output,param);
      gParamLF.Vbase:=Vbase;
      gParamLF.VKonst:=VKonst;
      gParamLF.Pbase:=Pbase;
      gParamLF.PKonst:=PKonst;
      if Param=1 then
      begin
        gParamLF.ParamBranch:=pbPu;
      end
      else if Param=2 then
      begin
        gParamLF.ParamBranch:=pbOhm;
      end;
      gParamLF.MaxIterasi:=15;
      gParamLF.Toleransi:=0.0001;
      try
        frmInput:=TfrmInputLFChild.Create(Application);
        frmInput.edtNbus.Text:=IntToStr(Nbus);
        frmInput.edtNsal.Text:=IntToStr(Nsal);
        frmInput.edtVbase.Text:=FloatToStr(Vbase);
```

```
if VKonst=1 then
begin
  frmInput.cmbVKonst.Text:='V';
end
else if VKonst=1000 then
begin
  frmInput.cmbVKonst.Text:='kV';
end
else if VKonst=1000000 then
begin
  frmInput.cmbVKonst.Text:='MV';
end;
frmInput.edtPbase.Text:=FloatToStr(Pbase);
if PKonst=1 then
begin
  frmInput.cmbPKonst.Text:='VA';
  frmInput.fgBus.Cells[3,0]:='Pg (W)';
  frmInput.fgBus.Cells[4,0]:='Qg (VAR)';
  frmInput.fgBus.Cells[5,0]:='PL (W)';
  frmInput.fgBus.Cells[6,0]:='QL (VAR)';
  frmInput.fgBranch.Cells[9,0] :='Kap (VA)';
end
else if PKonst=1000 then
begin
  frmInput.cmbPKonst.Text:='kVA';
  frmInput.fgBus.Cells[3,0]:='Pg (kW)';
  frmInput.fgBus.Cells[4,0]:='Qg (kVAR)';
  frmInput.fgBus.Cells[5,0]:='PL (kW)';
  frmInput.fgBus.Cells[6,0]:='QL (kVAR)';
  frmInput.fgBranch.Cells[9,0] :='Kap (kVA)';
end
else if PKonst=1000000 then
begin
  frmInput.cmbPKonst.Text:='MVA';
  frmInput.fgBus.Cells[3,0]:='Pg (MW)';
  frmInput.fgBus.Cells[4,0]:='Qg (MVAR)';
  frmInput.fgBus.Cells[5,0]:='PL (MW)';
  frmInput.fgBus.Cells[6,0]:='QL (MVAR)';
  frmInput.fgBranch.Cells[9,0] :='Kap (MVA)';
end;
if param=1 then
begin
  frmInput.cmbParam.Text:='pu';
  frmInput.fgBranch.Cells[3,0]:='R (pu)';
  frmInput.fgBranch.Cells[4,0]:='X (pu)';
  frmInput.fgBranch.Cells[5,0]:='Lc (pu)';
  frmInput.fgBus.Cells[7,0]:='Cap (pu)';
end
else if param=2 then
begin
```

```
  frmInput.cmbParam.Text:='ohm';
  frmInput.fgBranch.Cells[3,0]:='R (ohm)';
  frmInput.fgBranch.Cells[4,0]:='X (ohm)';
  frmInput.fgBranch.Cells[5,0]:='Lc (ohm)';
  frmInput.fgBus.Cells[7,0]:='Cap (ohm)';
end;
SetLength(gBus,Nbus);
for i:=0 to Nbus-1 do
begin
  Readln(output,absV,sudV,Pg,Qg,PL,QL,Cap,Typ);
  gBus[i].absV:=absV;
  gBus[i].sudV:=sudV;
  gBus[i].Pgen:=Pg;
  gBus[i].Qgen:=Qg;
  gBus[i].PL:=PL;
  gBus[i].QL:=QL;
  gBus[i].Cap:=Cap;
  gBus[i].typeBus:=Typ;
  frmInput.fgBus.Cells[0,i+1]:=IntToStr(i+1);
  frmInput.fgBus.Cells[1,i+1]:=FloatToStr(absV);
  frmInput.fgBus.Cells[2,i+1]:=FloatToStr(sudV);
  frmInput.fgBus.Cells[3,i+1]:=FloatToStr(Pg);
  frmInput.fgBus.Cells[4,i+1]:=FloatToStr(Qg);
  frmInput.fgBus.Cells[5,i+1]:=FloatToStr(PL);
  frmInput.fgBus.Cells[6,i+1]:=FloatToStr(QL);
  frmInput.fgBus.Cells[7,i+1]:=FloatToStr(Cap);
  frmInput.fgBus.Cells[8,i+1]:=IntToStr(typ);
end;
SetLength(gBranch,Nsal);
for i:=0 to Nsal-1 do
begin
  Readln(output,dari,ke,R,X,Lc,Tr,Tu,Su,CapSal);
  gBranch[i].dari:=dari;
  gBranch[i].ke:=ke;
  gBranch[i].R:=R;
  gBranch[i].X:=X;
  gBranch[i].Lc:=Lc;
  gBranch[i].Tr:=Tr;
  gBranch[i].Tu:=Tu;
  gBranch[i].Su:=Su;
  gBranch[i].KapSal:=CapSal;
  frmInput.fgBranch.Cells[0,i+1]:=IntToStr(i+1);
  frmInput.fgBranch.Cells[1,i+1]:=IntToStr(dari);
  frmInput.fgBranch.Cells[2,i+1]:=IntToStr(ke);
  frmInput.fgBranch.Cells[3,i+1]:=FloatToStr(R);
  frmInput.fgBranch.Cells[4,i+1]:=FloatToStr(X);
  frmInput.fgBranch.Cells[5,i+1]:=FloatToStr(Lc);
  frmInput.fgBranch.Cells[6,i+1]:=FloatToStr(Tr);
  frmInput.fgBranch.Cells[7,i+1]:=FloatToStr(Tu);
  frmInput.fgBranch.Cells[8,i+1]:=FloatToStr(Su);
```

```
        frmInput.fgBranch.Cells[9,i+1]:=FloatToStr(CapSal);
      end;
      Readln(output,Ngen);
      if Ngen<>0 then
      begin
        frmInput.fgGen.RowCount:=Ngen+1;
        SetLength(gGenLF,Ngen);
        for i:=0 to Ngen-1 do
        begin
          Readln(output,dari,R,X,Lc,Tr,Tu,Su,CapSal,Pmin,Pmax);
          gGenLF[i].bus:=dari;
          gGenLF[i].Qmin:=R;
          gGenLF[i].Qmax:=X;
          gGenLF[i].a2:=Lc;
          gGenLF[i].a1:=Tr;
          gGenLF[i].a0:=Tu;
          gGenLF[i].FixCost:=Su;
          gGenLF[i].VarCost:=CapSal;
          gGenLF[i].Pmin:=Pmin;
          gGenLF[i].Pmax:=Pmax;
          frmInput.fgGen.Cells[0,i+1]:=IntToStr(i+1);
          frmInput.fgGen.Cells[1,i+1]:=IntToStr(gGenLF[i].bus);
          frmInput.fgGen.Cells[2,i+1]:=RealToStr(gGenLF[i].Qmin,2);
          frmInput.fgGen.Cells[3,i+1]:=RealToStr(gGenLF[i].Qmax,2);
          frmInput.fgGen.Cells[4,i+1]:=RealToStr(gGenLF[i].a2,5);
          frmInput.fgGen.Cells[5,i+1]:=RealToStr(gGenLF[i].a1,5);
          frmInput.fgGen.Cells[6,i+1]:=RealToStr(gGenLF[i].a0,5);
          frmInput.fgGen.Cells[7,i+1]:=RealToStr(gGenLF[i].FixCost,2);
          frmInput.fgGen.Cells[8,i+1]:=RealToStr(gGenLF[i].VarCost,2);
          frmInput.fgGen.Cells[9,i+1]:=RealToStr(gGenLF[i].Pmin,2);
          frmInput.fgGen.Cells[10,i+1]:=RealToStr(gGenLF[i].Pmax,2);
        end;
      end
      else
      begin
        frmInput.fgGen.RowCount:=2;
      end;
      CloseFile(output);
      frmInput.Caption:='Tampilan Data';
      frmInput.btnNext.Caption:='&Next';
      frmInput.ShowModal;
    finally
      frmInput.Free;
    end;
  end;
except
  MessageDlg('File Corrupt atau Error Program!',mtWarning,[mbOK],0);
end;
end;
```

```
end.

unit uObjFunc;

interface

uses uUtils,uNewtonRaphson,uLoadflow,uComplex;

type
  TObjFunc=class
  private
    FKp,FKq,FKv,FKs,FStepTrafo:double;
    FBatasV,FBatasTrafo:TBatas;
    FBatasChrom:TBatasArr1;
    function getBatasV:TBatas;
    procedure setBatasV(const rBatasV:TBatas);
    procedure DecodeChromToLFData(const rChrom:dArr1;
            var rLBus:TBusArr1;
            var rLBranch:TBranchArr1);
    function FindPinV(const rLBus:TBusArr1):double;
    function FindPinPslack(const rLBus:TBusArr1):double;
    function FindPinKapBranch(const rLBranch:TBranchArr1):double;
    function FindPinKapQgen(const rLBus:TBusArr1):double;
    function FindCostGen(const rNo:integer;
          const rP:double):double;
  public
    constructor Create(const rKp,rKq,rKv,rKs,rStepTrafo:double;
            const rBatasV,rBatasTrafo:TBatas);
    function FindLength:integer;
    function FindBatasChrom:TBatasArr1;
    function doHitung(const rChrom:dArr1):double;
    function FindTotalCostGen(const rLBus:TBusArr1):double;
    procedure doHitungAkhir(const rChrom:dArr1;
            var rLBus:TBusArr1;
            var rLBranch:TBranchArr1;
            var rCostTotal:double);
    property Kp:double read FKp write FKp;
    property Kq:double read FKq write FKq;
    property Kv:double read FKv write FKv;
    property Ks:double read FKs write FKs;
    property BatasV:TBatas read getBatasV write setBatasV;
  end;

var gObjFunc:TObjFunc;

implementation

//constructor
constructor TObjFunc.Create(const rKp,rKq,rKv,rKs,rStepTrafo:double;
          const rBatasV,rBatasTrafo:TBatas);
```

```
begin
  inherited Create;
  FKp:=rKp;
  FKq:=rKq;
  FKv:=rKv;
  FKs:=rKs;
  FStepTrafo:=rStepTrafo;
  FBatasV.min:=rBatasV.min;
  FBatasV.max:=rBatasV.max;
  FBatasTrafo.min:=rBatasTrafo.min;
  FBatasTrafo.max:=rBatasTrafo.max;
  FBatasChrom:=FindBatasChrom;
end;

//data accessing
function TObjFunc.getBatasV:TBatas;
begin
  result.min:=FBatasV.min;
  result.max:=FBatasV.max;
end;

procedure TObjFunc.setBatasV(const rBatasV:TBatas);
begin
  FBatasV.min:=rBatasV.min;
  FBatasV.max:=rBatasV.max;
end;

//data processing
function TObjFunc.FindLength:integer;
var i:integer;
begin
  result:=0;
  for i:=0 to high(gBus) do
  begin
    if gBus[i].typeBus=2 then
    begin
      inc(result);
    end;
  end;
  for i:=0 to high(gBus) do
  begin
    if gBus[i].typeBus<>3 then
    begin
      inc(result);
    end;
  end;
  for i:=0 to high(gBus) do
  begin
    if gBus[i].typeBus=3 then
    begin
```

```
      if gBus[i].Qgen>0 then
      begin
        inc(result);
      end;
    end;
  end;
  for i:=0 to high(gBranch) do
  begin
    if gBranch[i].Tr<>0 then
    begin
      inc(result);
    end;
  end;
end;

function TObjFunc.FindBatasChrom:TBatasArr1;
var i,sa,no:integer;
begin
  sa:=FindLength;
  SetLength(result,sa);
  sa:=0;
  no:=1;
  for i:=0 to high(gBus) do
  begin
    if gBus[i].typeBus=2 then
    begin
      result[sa].min:=gGenLF[no].Pmin;
      result[sa].max:=gGenLF[no].Pmax;
      inc(sa);
      inc(no);
    end;
  end;
  for i:=0 to high(gBus) do
  begin
    if gBus[i].typeBus<>3 then
    begin
      result[sa].min:=FBatasV.min;
      result[sa].max:=FBatasV.max;
      inc(sa);
    end;
  end;
  for i:=0 to high(gBus) do
  begin
    if gBus[i].typeBus=3 then
    begin
      if gBus[i].Qgen>0 then
      begin
        result[sa].min:=0;
        result[sa].max:=gBus[i].Qgen;
        inc(sa);
```

```
      end;
    end;
  end;
  for i:=0 to high(gBranch) do
  begin
    if gBranch[i].Tr<>0 then
    begin
      result[sa].min:=FBatasTrafo.min;
      result[sa].max:=FBatasTrafo.max;
      inc(sa);
    end;
  end;
end;

procedure TObjFunc.DecodeChromToLFData(const rChrom:dArr1;
        var rLBus:TBusArr1;
        var rLBranch:TBranchArr1);
var i,Nbus,Nsal,sa:integer;
begin
  Nbus:=high(gBus)+1;
  SetLength(rLBus,Nbus);
  Nsal:=high(gBranch)+1;
  SetLength(rLBranch,Nsal);
  for i:=0 to Nbus-1 do
  begin
    rLBus[i].absV:=gBus[i].absV;
    rLBus[i].sudV:=gBus[i].sudV;
    rLBus[i].Pgen:=gBus[i].Pgen;
    rLBus[i].Qgen:=gBus[i].Qgen;
    rLBus[i].PL:=gBus[i].PL;
    rLBus[i].QL:=gBus[i].QL;
    rLBus[i].Cap:=gBus[i].Cap;
    rLBus[i].typeBus:=gBus[i].typeBus;
  end;
  for i:=0 to Nsal-1 do
  begin
    rLBranch[i].dari:=gBranch[i].dari;
    rLBranch[i].ke:=gBranch[i].ke;
    rLBranch[i].R:=gBranch[i].R;
    rLBranch[i].X:=gBranch[i].X;
    rLBranch[i].Lc:=gBranch[i].Lc;
    rLBranch[i].Tr:=gBranch[i].Tr;
    rLBranch[i].Tu:=gBranch[i].Tu;
    rLBranch[i].Su:=gBranch[i].Su;
    rLBranch[i].KapSal:=gBranch[i].KapSal;
    rLBranch[i].Length:=gBranch[i].Length;
    rLBranch[i].JenisCable:=gBranch[i].JenisCable;
  end;
  sa:=0;
  for i:=0 to Nbus-1 do
```

```
  begin
    if gBus[i].typeBus=2 then
    begin
      rLBus[i].Pgen:=rChrom[sa];
      inc(sa);
    end;
  end;
  for i:=0 to Nbus-1 do
  begin
    if gBus[i].typeBus<>3 then
    begin
      rLBus[i].absV:=rChrom[sa];
      inc(sa);
    end;
  end;
  for i:=0 to Nbus-1 do
  begin
    if gBus[i].typeBus=3 then
    begin
      if gBus[i].Qgen>0 then
      begin
        rLBus[i].Qgen:=rChrom[sa];
        inc(sa);
      end;
    end;
  end;
  for i:=0 to Nsal-1 do
  begin
    if gBranch[i].Tr<>0 then
    begin
      rLBranch[i].Tr:=rChrom[sa];
      inc(sa);
    end;
  end;
end;

function TObjFunc.FindPinV(const rLBus:TBusArr1):double;
var i:integer;
begin
  result:=0;
  for i:=0 to high(rLBus) do
  begin
    if rLBus[i].absV>FBatasV.max then
    begin
      result:=result+1;
      //result:=result+sqr(rLBus[i].absV-FBatasV.max);
    end;
    if rLBus[i].absV<FBatasV.min then
    begin
      result:=result+1;
```

```
      //result:=result-sqr(FBatasV.min-rLBus[i].absV);
    end;
  end;
end;

function TObjFunc.FindPinPslack(const rLBus:TBusArr1):double;
var i,sa:integer;
begin
  result:=0;
  sa:=0;
  for i:=0 to high(gBus) do
  begin
    if gBus[i].typeBus=1 then
    begin
      if rLBus[i].Pgen>gGenLF[sa].Pmax then
      begin
        result:=result+sqr(rLBus[i].Pgen-gGenLF[sa].Pmax);
      end;
      if rLBus[i].Pgen<gGenLF[sa].Pmin then
      begin
        result:=result+sqr(gGenLF[sa].Pmin-rLBus[i].Pgen);
      end;
      inc(sa);
    end;
  end;
end;

function TObjFunc.FindPinKapBranch(const rLBranch:TBranchArr1):double;
var i,Nsal:integer;
    absDaya:double;
begin
  Nsal:=high(rLBranch)+1;
  result:=0;
  for i:=0 to Nsal-1 do
  begin
    if rLBranch[i].Sij.real>0 then
    begin
      absDaya:=getAbs(rLBranch[i].Sij);
    end
    else
    begin
      absDaya:=getAbs(rLBranch[i].Sji);
    end;
    if absDaya>rLBranch[i].KapSal then
    begin
      result:=result+sqr(absDaya-rLBranch[i].KapSal);
    end;
  end;
end;
```

```
function TObjFunc.FindPinKapQgen(const rLBus:TBusArr1):double;
var i,sa:integer;
begin
  result:=0;
  sa:=0;
  for i:=0 to high(rLBus) do
  begin
    if rLBus[i].typeBus<>3 then
    begin
      if rLBus[i].Qgen>gGenLF[sa].Qmax then
      begin
        result:=result-sqr(rLBus[i].Qgen-gGenLF[sa].Qmax);
      end;
      if rLBus[i].Qgen<gGenLF[sa].Qmin then
      begin
        result:=result-sqr(abs(gGenLF[sa].Qmin)-abs(rLBus[i].Qgen));
      end;
      inc(sa);
    end;
  end;
end;

function TObjFunc.FindCostGen(const rNo:integer;
        const rP:double):double;
begin
  result:=0;
  if rP<>0 then
  begin
    result:=gGenLF[rNo].a2*sqr(rP)+gGenLF[rNo].a1*rP+gGenLF[rNo].a0;
  end;
end;

function TObjFunc.FindTotalCostGen(const rLBus:TBusArr1):double;
var i,sa,Nbus:integer;
begin
  Nbus:=high(rLBus)+1;
  result:=0;
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    if rLBus[i].typeBus<>3 then
    begin
      result:=result+FindCostGen(sa,rLBus[i].Pgen);
      inc(sa);
    end;
  end;
end;

function TObjFunc.doHitung(const rChrom:dArr1):double;
var LBus:TBusArr1;
```

```
    LBranch:TBranchArr1;
    pinV,pinP,pinS,PinQ,CostTotal:double;
begin
  DecodeChromToLFData(rChrom,LBus,LBranch);
  NewtonRaphson(LBus,LBranch,gParamLF);
  pinV :=FindPinV(LBus);
  pinP:=FindPinPslack(LBus);
  pinS:=FindPinKapBranch(gBranch);
  pinQ:=FindPinKapQgen(LBus);
  CostTotal:=gParamLF.Sum1.oss.real;
  result:=CostTotal+FKv*pinV+FKp*pinP+FKs*pinS+FKq*pinQ;
end;

procedure TObjFunc.doHitungAkhir(const rChrom:dArr1;
        var rLBus:TBusArr1;
        var rLBranch:TBranchArr1;
        var rCostTotal:double);
begin
  DecodeChromToLFData(rChrom,rLBus,rLBranch);
  NewtonRaphson(rLBus,rLBranch,gParamLF);
  rCostTotal:=FindTotalCostGen(rLBus);
end;

end.

unit uPSO;

interface

uses uUtils,uRandom,uObjFunc,uHasil;

type
  TIndiDouble1=record
    chrom:dArr1;
    fitness:double;
  end;

  TParticle=record
    curr:TIndiDouble1;
    prev:TIndiDouble1;
    velo:dArr1;
  end;

  TParticles=array of TParticle;


  THPSOM=class
  private
    FIterasi,FPopSize,Fc1,Fc2,FLength:integer;
    FdV:double;
```

```
    FW:TBatas;
    FV,FBatasChrom:TBatasArr1;
    FParticles:TParticles;
    FGbest:TIndiDouble1;
    FRandom:TRandomu;
    FGlobal:dArr1;
    function getIndividu(const rIndi:TIndiDouble1):TIndiDouble1;
    function getMutasi(const rChrom:dArr1):dArr1;
    procedure InitParticles;
    procedure UpdateSwarm(const rIterasi:integer);
    procedure doHitung;
    function getBestSwarm:dArr1;
    function getGlobal:dArr1;
  public
    constructor Create(const rIterasi,rPopSize,rc1,rc2,rLength:integer;
            const rdV:double;
            const rW:TBatas;
            const rBatasChrom:TBatasArr1);
    destructor Destroy;override;
    property Iterasi:integer read FIterasi write FIterasi;
    property PopSize:integer read FPopSize write FPopSize;
    property c1:integer read Fc1 write Fc1;
    property c2:integer read Fc2 write Fc2;
    property w:TBatas read Fw write Fw;
    property dV:double read FdV write FdV;
    property BestSwarm:dArr1 read getBestSwarm;
    property Global:dArr1 read getGlobal;
  end;

implementation

//constructor
constructor THPSOM.Create(const rIterasi,rPopSize,rc1,rc2,rLength:integer;
        const rdV:double;
        const rW:TBatas;
        const rBatasChrom:TBatasArr1);
var i:integer;
begin
  inherited Create;
  FIterasi:=rIterasi;
  FPopSize:=rPopSize;
  Fc1:=rc1;
  Fc2:=rc2;
  FLength:=rLength;
  FW.min:=rW.min;
  FW.max:=rW.max;
  FdV:=rdV;
  SetLength(FV,FLength);
  SetLength(FBatasChrom,FLength);
  for i:=0 to FLength-1 do
```

```
  begin
    FBatasChrom[i].min:=rBatasChrom[i].min;
    FBatasChrom[i].max:=rBatasChrom[i].max;
    FV[i].min:=-FdV*rBatasChrom[i].min;
    FV[i].max:=FdV*rBatasChrom[i].max;
  end;
  FRandom:=TRandomu.Create;
end;

//destructor
destructor THPSOM.Destroy;
begin
  try
    FRandom.Free;
  finally
    inherited Destroy;
  end;
end;

//data accessing
function THPSOM.getIndividu(const rIndi:TIndiDouble1):TIndiDouble1;
var i:integer;
begin
  SetLength(result.chrom,FLength);
  for i:=0 to FLength-1 do
  begin
    result.chrom[i]:=rIndi.chrom[i];
  end;
  result.fitness:=rIndi.fitness;
end;

function THPSOM.getMutasi(const rChrom:dArr1):dArr1;
var i:integer;
    wmax,w,rand:double;
begin
  SetLength(result,FLength);
  for i:=0 to FLength-1 do
  begin
    rand:=random;
    if rand<=0.03 then
    begin
      wmax:=0.1*(FBatasChrom[i].max-FBatasChrom[i].min);
      w:=0+random*(wmax-0);
      rand:=random;
      if rand<=0.5 then
      begin
        result[i]:=rChrom[i]+w;
        if result[i]>FBatasChrom[i].max then
        begin
          result[i]:=FBatasChrom[i].max;
```

```
      end;
    end
    else
    begin
      result[i]:=rChrom[i]-w;
      if result[i]<FBatasChrom[i].min then
      begin
        result[i]:=FBatasChrom[i].min;
      end;
    end;
  end
  else
  begin
    result[i]:=rChrom[i];
  end;
 end;
end;

procedure THPSOM.InitParticles;
var i,j,pos:integer;
  min:double;
begin
 SetLength(FParticles,FPopSize);
 for i:=0 to FPopSize-1 do
 begin
  SetLength(FParticles[i].curr.chrom,FLength);
  SetLength(FParticles[i].prev.chrom,FLength);
  SetLength(FParticles[i].velo,FLength);
  for j:=0 to FLength-1 do
  begin
   FParticles[i].curr.chrom[j]:=FRandom.NextDouble(FBatasChrom[j].min,
                          FBatasChrom[j].max);
   FParticles[i].prev.chrom[j]:=0;
   FParticles[i].velo[j]:=0;
  end;
  FParticles[i].curr.fitness:=gObjFunc.doHitung(FParticles[i].curr.chrom);
 end;
 min:=FParticles[0].curr.fitness;
 pos:=0;
 for i:=1 to FPopSize-1 do
 begin
  if min>FParticles[i].curr.fitness then
  begin
   min:=FParticles[i].curr.fitness;
   pos:=i;
  end;
 end;
 FGBest:=getIndividu(FParticles[pos].curr);
end;
```

```
procedure THPSOM.UpdateSwarm(const rIterasi:integer);
var i,j:integer;
   dv,w,rand:double;
   tmpIndi:TIndiDouble1;
begin
  w:=FW.max-(FW.max-FW.min)/FIterasi*rIterasi;
  for i:=0 to FPopSize-1 do
  begin
    for j:=0 to FLength-1 do
    begin
      dv:=w*FParticles[i].velo[j]+
         Fc1*FRandom.NextDouble*
         (FParticles[i].prev.chrom[j]-FParticles[i].curr.chrom[j])+
         Fc2*FRandom.NextDouble*
         (FGBest.chrom[j]-FParticles[i].curr.chrom[j]);
      if dv>FV[i].max then
      begin
        dV:=FV[i].max;
      end;
      if dv<FV[i].min then
      begin
        dV:=FV[i].min;
      end;
      FParticles[i].curr.chrom[j]:=FParticles[i].curr.chrom[j]+dv;
      if FParticles[i].curr.chrom[j]>FBatasChrom[j].max then
      begin
        FParticles[i].curr.chrom[j]:=FBatasChrom[j].max;
      end;
      if FParticles[i].curr.chrom[j]<FBatasChrom[j].min then
      begin
        FParticles[i].curr.chrom[j]:=FBatasChrom[j].min;
      end;
      FParticles[i].velo[j]:=dv;
    end;
    FParticles[i].curr.fitness:=gObjFunc.doHitung(FParticles[i].curr.chrom);
    if (FParticles[i].curr.fitness<FParticles[i].prev.fitness) or
       (FParticles[i].prev.fitness=0) then
    begin
      FParticles[i].prev:=getIndividu(FParticles[i].curr);
    end;
    if FParticles[i].curr.fitness<FGBest.fitness then
    begin
      FGBest:=getIndividu(FParticles[i].curr);
    end;
  end;
  for i:=0 to FPopSize-1 do
  begin
    rand:=random;
    if rand<=0.3 then
    begin
```