

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



**ANALISIS PENEMPATAN KAPASITOR BANK MENGGUNAKAN  
OPERATOR GENETIKA DENGAN *ALLELIC ALPHABET* DAN KONSEP  
JARAK HAMMING PADA SISTEM DISTRIBUSI 20KV  
DI GI PIER PASURUAN PADA PENYULANG SYIAR**

**SKRIPSI**

**Disusun oleh :  
Ario Kunto Adji  
00.12.045**



**Maret 2007**

## LEMBAR PERSETUJUAN



### ANALISIS PENEMPATAN KAPASITOR BANK MENGGUNAKAN OPERATOR GENETIKA DENGAN ALLEGIC ALPHABET DAN KONSEP JARAK HAMMING PADA SISTEM DISTRIBUSI 20KV DI GI PIER PASURUAN PADA PENYULANG SYIAR



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

## KATA PENGANTAR

Alhamdullilah dengan memanjatkan puji syukur atas kehadiran Allah S.W.T. Berkat limpahan rahmat dan hidayahNya, sehingga penyusunan skripsi ini dapat terselesaikan.

Skripsi ini disusun sebagai salah satu persyaratan dalam menyelesaikan studi program strata satu (S-1) jurusan Teknik Elektro, program studi Teknik Energi Listrik, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.

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

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Mochtar Asroni, MSME. selaku Dekan Fakultas Teknik Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. F Yudi Limpraptono, MT. selaku Ketua Jurusan Elektro Fakultas Teknik Industri Institut Teknologi Nasional Malang.
4. Bapak Ir. Yusuf Ismail Nakhoda, MT. selaku Dosen Pembimbing.
5. Bapak Catur. Selaku Kepala bagian pelaksanaan di PLN Bangil Pasuruan.
6. Orang Tuaku yang tersayang.
7. Serta semua pihak yang tidak dapat disebut satu persatu yang turut serta membantu menyelesaikan skripsi ini.

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

Akhirnya, kepada semua pihak yang telah bekerja keras dan bersungguh - sungguh hingga terwujudnya skripsi ini, saya menyampaikan penghargaan terima kasih.

Malang, Maret 2007

Penulis

## DAFTAR ISI

LEMBAR PERSETUJUAN .....	ii
ABSTRAKSI .....	iii
KATA PENGANTAR .....	iv
DAFTAR ISI .....	v
DAFTAR GAMBAR .....	viii
DAFTAR GRAFIK .....	ix
DAFTAR TABEL .....	x

### **BAB I : PENDAHULUAN**

1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	3
1.3. Tujuan .....	3
1.4. Batasan Masalah .....	3
1.5. Metodologi Penelitian .....	4
1.6. Sistematika Penulisan .....	4
1.7. Kontribusi .....	6

### **BAB II : SISTEM DISTRIBUSI TENAGA LISTRIK**

2.1. Sistem Distribusi Tenaga Listrik .....	7
2.2. Struktur Jaringan Distribusi Radial .....	7
2.3. Analisa Aliran Daya dengan Metode <i>Newton Raphson</i> .....	8
2.4. Kapasitor Daya .....	9

2.4.1. Kapasitor Seri dan Kapasitor Shunt.....	10
2.4.2. Faktor – faktor Pemilihan Kapasitor Shunt .....	10
2.4.3. Pengaruh Pemasangan Kapasitor Shunt.....	10
2.4.4. Pengaruh Biaya Dengan Kerugian Daya Kapasitor Bank .....	11

**BAB III : METODE PENYELESAIAN OPERATOR GENETIKA DENGAN  
*ALLELIC ALPHABET DAN KONSEP JARAK HAMMING***

3.1. <i>Implemented GA</i> .....	13
3.1.1. Teori Dasar <i>Genetic Algorithm</i> (GA) .....	13
3.1.2. Operator <i>Genetic</i> .....	15
3.1.3. Kromosom.....	15
3.1.4. Populasi Awal .....	16
3.1.5. <i>Crossover</i> .....	16
3.1.6. Mutasi ( <i>Mutation</i> ) .....	16
3.1.7. Seleksi .....	17
3.1.8. <i>Reinsertion</i> .....	17
3.1.9. Fungsi Fitness .....	17
3.1.10.Kriteria Penghentian .....	18
3.2. Algoritma Dan <i>Flow Chart</i> .....	18
3.2.1. Algoritma <i>Genetic Algorithm</i> .....	18
3.2.2. <i>Flowchart Genetic Algorithm</i> .....	20
3.2.3. Algoritma Program Operator Genetika.....	21
3.2.4. <i>Flowchart Operator Genetika</i> .....	22
3.3. Data Validasi.....	23

**BAB IV : ANALISA PENEMPATAN KAPASITOR MENGGUNAKAN  
METODE OPERATOR GENETIKA JARINGAN DISTRIBUSI  
PADA PENYULANG SYIAR**

4.1. Aplikasi Penempatan Kapasitor Dengan Metode Operator Genetika Dengan <i>Allelic Alphabet</i> Dan Konsep Jarak Hamming .....	24
4.2. Sistem Distribusi Tenaga Listrik 20 kV GI PIER Pasuruan Penyulang SYIAR.....	24
4.3. Data Saluran.....	25
4.4. Pembebatan Sistem 20 kV Penyulang SYIAR .....	28
4.5. Analisa Perhitungan.....	32

**BAB V : KESIMPULAN DAN SARAN**

5.1. Kesimpulan.....	51
5.2. Saran .....	52

**DAFTAR PUSTAKA.....** 53

**LAMPIRAN**

## DAFTAR GAMBAR

2.1. Diagram Satu Garis Penyaluran Energi Listrik ke Pelanggan.....	7
2.2. Jaringan Tegangan Menengah Sistem Distribusi Radial .....	8
3.1. <i>Cromosome dan Allelic Alphabet</i> .....	15
3.2. “a” dan “c” <i>Crossover</i> Individu.....	16
3.3. a) Mutasi Untuk Sebuah Gen Kromosom, b) Aplikasi Menyangkut Mutasi, c) Mutasi Berbagai Kemungkinan Dengan $r = 1$ .....	17
3.4. <i>Flowchart Genetic Algorithm</i> .....	20
3.5. <i>Flowchart Operator Genetika</i> .....	22
4.1. <i>Single Line Diagram</i> Penyalur SYIAR Sebelum Penempatan Kapasitor ...	34
4.2. Hasil Penempatan Kapasitor Dengan Metode Operator Genetika.....	41
4.3. <i>Single Line Diagram</i> Setelah Penempatan Kapasitor .....	42

## **DAFTAR GRAFIK**

4.1. Perbandingan Tiap - tiap Bus Antara Tegangan Sebelum Dan Setelah Kompensasi.....	49
4.2. Rugi-rugi Daya Sebelum Dan Setelah Kompensasi .....	49

## DAFTAR TABEL

3.1. Perbandingan Nilai Tegangan Sebelum dan Setelah Kompensasi Antara Hasil Jurnal Dengan Hasil Program.....	23
4.1. Data Spesifikasi Kabel Saluran.....	25
4.2. Data Hasil Perhitungan Impedansi Saluran Penyulang SYIAR .....	26
4.3. Data Hasil Perhitungan Pembebatan Sistem 20 kV Penyulang SYIAR .....	29
4.4. Profil Tegangan dan Sudut Fasa Tiap Bus Penyulang SYIAR Sebelum Kompensasi.....	35
4.5. Aliran Arus Tiap Saluran Penyulang SYIAR Sebelum Kompensasi .....	37
4.6. Aliran Daya Tiap Saluran Penyulang SYIAR Sebelum Kompensasi.....	39
4.7. Profil Tegangan Dan Sudut Fasa Tiap Bus Penyulang SYIAR Setelah Kompensasi.....	43
4.8. Aliran Arus Tiap Saluran Penyulang SYIAR Setelah Kompensasi.....	45
4.9. Aliran Daya Tiap Saluran Penyulang SYIAR Setelah Kompensasi.....	47

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Permintaan akan energi listrik oleh masyarakat terus meningkat, hal ini sebanding dengan besarnya manfaat yang di rasakan dari energi listrik itu sendiri. Peningkatan tersebut harusnya ditunjang oleh peningkatan mutu kehandalan sistem tenaga listrik yang telah ada, baik itu untuk konsumen perumahan maupun industri yang mengandalkan energi listrik sebagai suplai untuk menjalankan usaha.

Dengan meningkatkan usaha di sektor industri dan meningkatkan taraf hidup masyarakat maka kebutuhan akan energi semakin meningkat pula, sehingga diperlukan penyediaan energi listrik beserta jaringan dan penyaluran yang sangat baik. Beban industri sangat banyak membutuhkan daya reaktif induktif. Dengan meningkatnya beban – beban induktif, maka daya reaktif yang ada di jaringan akan semakin besar yang selanjutnya akan memperbesar komponen rugi – rugi daya, disamping itu dapat memperburuk kondisi tegangan.

Alternatif yang sering dipakai untuk memperbaiki kondisi jaringan akibat rugi-rugi daya tersebut adalah dengan memasang sumber daya reaktif tambahan di sisi beban salah satunya dengan memasang kapasitor. Kapasitor digunakan secara luas dalam sistem distribusi untuk kompensasi daya reaktif guna mencapai pengurangan kerugian energi dan daya, pelepasan kapasitas sistem dan profil tegangan yang dapat diterima. Seberapa besar keuntungannya tergantung pada pemasangan kapasitor

Banyak metode yang di pakai dalam menganalisa masalah penentuan penempatan atau pemasangan kapasitor pada sistem distribusi, diantaranya seperti

*Heuristic*, teknik *Artificial Intelligence*, *Simulated Annealing*, *Tabu Search*, Logika Fuzzy dan Pemrograman evolutif, Logika Fuzzy dan micro-GA, dan *Genetic Algorithm* dan lain – lain. Salah satu metode alternatif yang akan dibahas dalam skripsi ini adalah metode Operator Genetika dengan *Allelic Alphabet* dan konsep jarak *Hamming*.

## **1.2. Rumusan Masalah**

1. Apakah dengan dipasang kapasitor dapat meminimalkan rugi – rugi daya.
2. Apakah dengan dipasang kapasitor dapat memperbaiki profil tegangan.
3. Bagaimana cara menentukan penempatan kapasitor bank menggunakan Operator Genetika dengan *Allelic Alphabet* dan konsep jarak *Hamming*.

## **1.3. Tujuan Pembahasan**

Adapun tujuan penulisan skripsi ini adalah untuk meminimalkan rugi – rugi daya dan memperbaiki profil tegangan dengan penempatan kapasitor bank menggunakan Operator Genetika dengan *Allelic Alphabet* dan konsep jarak *Hamming*

## **1.4. Batasan Masalah**

Untuk menyederhanakan masalah yang akan dibahas, maka diberikan asumsi-asumsi serta batasan-batasan sebagai berikut:

1. Jaringan yang akan di analisis adalah jaringan distribusi primer tipe radial
2. Analisa penentuan pemasangan kapasitor dan pemilihan kapasitas kapasitor menggunakan metode Operator Genetika dengan *Allelic Alphabet* dan konsep jarak *Hamming*.
3. Menganalisa penempatan kapasitor bank pada GI PIER pasuruan pada penyulang Syiar.

## **1.5. Metodologi Penelitian**

Metode yang di gunakan dalam pembahasan makalah skripsi ini adalah :

1. Studi literatur berupa pengumpulan referensi yang berkaitan dengan pokok bahasan makalah skripsi ini, mempelajari dan memahami referensi tersebut.
2. Pengumpulan data lapangan yang dipakai dalam obyek penelitian yakni
  - a. Data Pembebanan Sistem Distribusi .
  - b. Data Saluran Sistem Distribusi .
3. Simulasi dan pembahasan masalah.

Analisa perhitungan aliran daya tiap saluran menggunakan metode *Newton Raphson*, untuk menentukan pemasangan kapasitor yang optimal menggunakan Metode *Genetic Algorithm (GA)* yang disimulasikan dengan program komputer.

4. Menarik kesimpulan apakah metode ini sudah cukup efektif dalam menghasilkan perbaikan – perbaikan pada sistem dan penghematan yang optimal.

## **1.6. Sistematika Penulisan**

Untuk mendapatkan arah yang tepat mengenai hal-hal yang akan dibahas maka skripsi ini disusun sebagai berikut :

### **BAB I : PENDAHULUAN**

Merupakan pendahuluan yang meliputi latar belakang yang melandasi skripsi yang dibahas, rumusan masalah, tujuan yang ingin dicapai, batasan masalah, metodologi penelitian dan sistematika penulisan.

## BAB II : SISTEM DISTRIBUSI TENAGA LISTRIK

Disini akan diuraikan mengenai sistem jaringan distribusi tegangan menengah, klasifikasi sistem distribusi tenaga listrik dan struktur jaringan distribusi tenaga listrik.

## BAB III :METODE PENYELESAIAN OPERATOR GENETIKA DENGAN *ALLELIC ALPHABET DAN KONSEP JARAK HAMMING*

Pada bab ini akan dibahas metode Teori dasar *Genetic Algorithm* (GA), Operator Genetika, Kromosom, Populasi Awal, *Crossover*, Mutasi (*Mutation*), Seleksi, *Reinsertion*, Fungsi Fitness, *Flowchart* dan Kriteria Penghentian.

## BAB IV: ANALISA PENEMPATAN KAPASITOR MENGGUNAKAN METODE OPERATOR GENETIKA JARINGAN DISTRIBUSI PADA PENYULANG SYIAR

Pada bab ini dibahas Aplikasi Penempatan Kapasitor Dengan Metode Operator Genetika Dengan *Allelic Alphabet* Dan Konsep Jarak Hamming, dan Hasil Program.

## BAB V : KESIMPULAN DAN SARAN

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

### **1.7. Kontribusi**

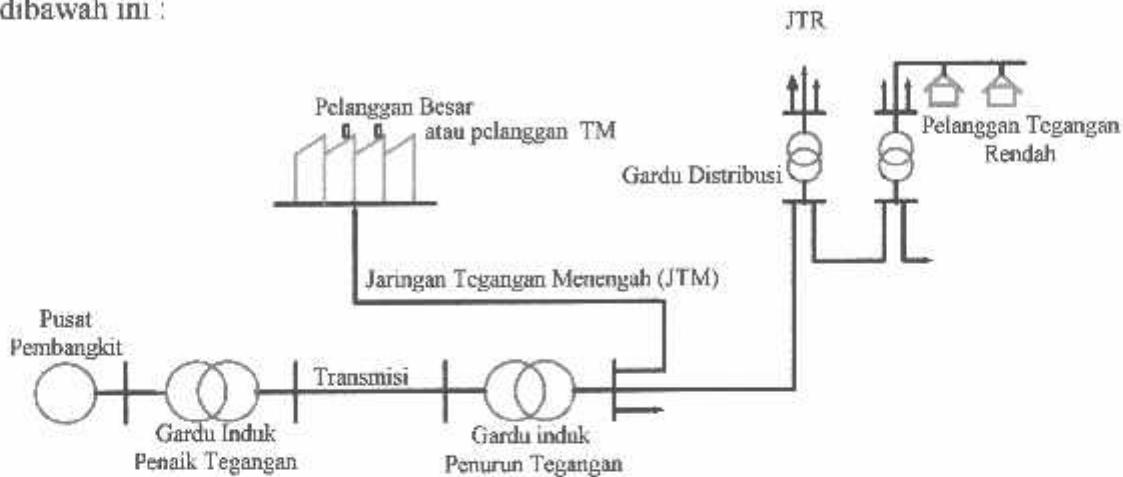
Dengan adanya penelitian penempatan kapasitor bank menggunakan operator genetika dengan *Allelic Alphabet* dan konsep jarak *Hamming* diharapkan dapat menjadi alternative PT.PLN dalam usaha meminimalkan biaya operasi khususnya biaya pembangkitan yang merupakan biaya terbesar pada sistem tenaga listrik namun tetap memperhatikan kualitas pelayanan daya listrik.

## BAB II

### SISTEM DISTRIBUSI TENAGA LISTRIK

#### 2.1. Sistem Distribusi Tenaga Listrik

Diagram satu garis sistem tenaga listrik dapat dijelaskan seperti gambar 1 dibawah ini :



Gambar 2.1.  
Diagram Satu Garis Penyaluran Energi Listrik ke Pelanggan<sup>[4]</sup>

Jaringan setelah keluar dari GI biasa disebut jaringan distribusi. Jaringan distribusi dapat diklasifikasikan menjadi dua bagian sistem, yaitu :

1. Sistem distribusi primer atau sistem distribusi tegangan menengah.
2. Sistem distribusi sekunder atau sistem distribusi tegangan rendah.

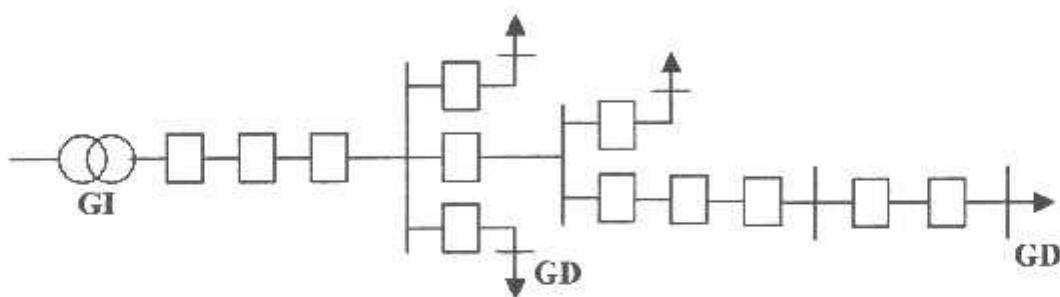
Pengklasifikasi sistem distribusi tenaga listrik menjadi dua ini berdasarkan tingkat tegangan distribusinya.

#### 2.2. Struktur Jaringan Distribusi Radial<sup>[4]</sup>

Jaringan distribusi radial merupakan bentuk paling sederhana yang menghubungkan beban-beban ke titik sumber, biayanya relatif murah. Pada

struktur jaringan ini tidak ada alternatif pasokan tenaga listrik, karena itu tingkat keandalannya relatif rendah.

Bentuk yang paling umum digunakan pada sistem distribusi radial adalah seperti pada gambar 2. Dapat dilihat bahwa sebuah penyulang memasok sejumlah gardu distribusi. Jika terjadi gangguan pada jaringan tegangan menengahnya, maka pemutus beban yang ada di gardu induk akan membuka, hal ini menyebabkan semua gardu distribusinya akan mengalami pemadaman.



Gambar 2.2.  
Jaringan Tegangan Menengah Sistem Distribusi Radial<sup>[4]</sup>

### 2.3. Analisa Aliran Daya dengan Metode *Newton Raphson*

Sebelum melakukan penentuan pemasangan kapasitor pada Sistem Distribusi Radial dengan menggunakan metode *Genetic Algorithm (GA)*, dilakukan suatu proses aliran daya untuk mengetahui kondisi suatu sistem distribusi.

Adapun tujuan mempelajari analisis aliran daya untuk mengetahui nilai tegangan dan untuk mengetahui besarnya daya aktif dan daya reaktif dari saluran. Metode yang digunakan untuk menganalisa aliran daya pada sistem distribusi radial adalah metode *Newton Raphson*.

#### **2.4. Kapasitor Daya**

Secara sederhana kapasitor terdiri dari dua buah plat logam yang dipisahkan oleh suatu bahan dielektrik dan kapasitor ini mempunyai sifat menyimpan muatan listrik. Pada beberapa tahun lalu kebanyakan kapasitor terbuat dari dua buah plat alumunium murni yang dipisahkan oleh tiga atau lebih lapisan kertas yang dilapisi oleh bahan kimia. Kapasitor daya telah mengalami perkembangan begitu cepat selama 30 tahun terahir, karena bahan dielektrik yang digunakan lebih effisien serta teknologi pembuatan kapasitor lebih baik.

#### **2.4.1. Kapasitor Seri dan Kapasitor Shunt**

Kapasitor daya terdiri atas beberapa bagian yaitu kapasitor shunt dalam sistem tenaga membangkitkan daya reaktif untuk memperbaiki faktor daya dan tegangan, sehingga meningkatkan kapasitas sistem dan mengurangi rugi – rugi jaringan. Biayanya pemasangan kapasitor shunt lebih murah dari pada biaya pemasangan kapasitor seri. Hal ini disebabkan karena peralatan pelindung untuk kapasitor seri lebih banyak, biasanya kapasitor seri didisain untuk daya yang lebih besar daripada kapasitor shunt dengan tujuan untuk mengatasi kenaikan beban.

#### **2.4.2. Faktor – faktor Pemilihan Kapasitor Shunt**

Faktor – faktor yang mempengaruhi pemilihan kapasitor shunt adalah sebagai berikut :

- Memperbaiki faktor daya.
- Memperbaiki tingkat tegangan pada sistem saluran udara dengan faktor daya yang tinggi.
- Mengurangi rugi – rugi saluran.

#### **2.4.3. Pengaruh Pemasangan Kapasitor Shunt**

Kapasitor shunt adalah kapasitor yang dihubungkan paralel dengan saluran dan secara intensif digunakan pada sistem distribusi. Kapasitor shunt mencatu daya reaktif atau arus yang menentang komponen arus beban induktif. Dengan dipasangnya kapasitor shunt pada jaringan distribusi akan dapat memperbaiki profil tegangan, memperbaiki faktor tegangan dan menaikkan kapasitor sistem serta dapat mengurangi rugi saluran.

Ada 2 cara pemakaian kapasitor shunt :

- Kapasitor tetap.
- Kapsitor Saklar.

a. Kapasitor Tetap

Adalah kapasitor untuk kompensasi daya reaktif yang kapasitasnya tetap dan selalu terpasang di jaringan. Penggunaan kapasitor ini harus memperhatikan kenaikan tegangan yang terjadi pada saat beban ringan agar tidak melebihi batas tegangan yang ditetapkan.

b. Kapasitor Saklar

Adalah kapasitor untuk kompensasi daya reaktif yang dapat dihubungkan dan dilepaskan dari jaringan dan dapat besar kapasitasnya sesuai dengan kondisi beban.

Proses membuka-menutup dari saklar kapasitor shunt dapat dilakukan dengan cara manual maupun dengan cara otomatis. Pengendalian secara manual ( pada lokasi atau kendali jarak jauh ) dapat dilakukan pada GI. Untuk pengendalian secara otomatis, termasuk didalamnya peralatan pengendali tegangan, arus, tegangan – waktu dan suhu. Tipe yang paling popular adalah pengendali saklar waktu (*Time – Switch Control*), pengendali tegangan dan pengendali tegangan – arus.

#### 2.4.4. Pengaruh Biaya Dengan Kerugian Daya Kapasitor Bank

Biaya kerugian daya puncak mencerminkan biaya yang berhubungan dengan penggunaan kapasitas sistem.

Mengimplikasikan peningkatan kapasitas cabang dan perbaikan profil tegangan

$$C_T = C_1 + C_0 \dots \quad (1)$$

Dimana:

$C_T$  = Total Biaya

$C_1$  = Biaya Investasi

$C_0$  – Biaya Operasional

Biaya diproporsional terhadap injeksi daya reaktif

Dimana

$K_s$  = Biaya Pembelian Kapasitor Bank

$Q_i$  = Daya Reaktif Nominal Kapasitor Bank yang Terpasang

$N_a$  = Total Kapasitor Bank yang Terpasang.

Biaya kerugian daya operasi dari Kapasitor Bank

$$C_0 = K_p \cdot \sum_{i=1}^N R_i \cdot I_i^2 + \frac{K_p}{1000} \cdot \sum_{i=1}^{N_y} 0.5 \cdot Q_i \quad \dots \dots \dots \quad (3)$$

Dimana :

$O_i$  = Nominal Kapasitas kVAR dari Kapasitor Bank ke i yang dipasang

$K_p$  = Biaya Kerugian Daya Puncak

N = Total Jumlah Cabang Dalam Sistem Distribusi

$R_i$  = Resistansi Cabang ke i Dari Sistem Distribusi

$J_i$  = Arus Cabang ke  $i$  Dari Sistem Distribusi

## BAB III

### METODE PENYELESAIAN OPERATOR GENETIKA DENGAN *ALLELIC ALPHABET DAN KONSEP JARAK HAMMING*

#### 3.1. *Implemented GA*

##### 3.1.1. Teori Dasar *Genetic Algorithm (GA)*

*Genetic Algorithm (GA)* adalah suatu algoritma yang meniru prinsip evolusi alam sebagai metode untuk memecahkan optimasi parameter. Teori *Genetic Algorithm (GA)* didasari oleh teori Darwin. Landasan *Genetic Algorithm (GA)* terinspirasi dari mekanisme alam, dimana individu yang lebih kuat memiliki kemungkinan untuk menjadi pemenang dan mempunyai kesempatan hidup yang lebih besar di dalam lingkungan yang kompetitif.

*Genetic Algorithm (GA)* bekerja dengan populasi *string*, dan melakukan proses pencarian nilai optimal secara parallel. Dengan menggunakan operator *Genetic Algorithm (GA)* akan melalui rekombinasi antar individu. Elemen dasar yang diproses *Genetic Algorithm (GA)* adalah *string* (kromosom) dengan panjang tertentu yang tersusun dari rangkaian substring (gen), dan biasanya merupakan kode biner (0,1). Pada substring (gen) dapat diasumsikan suatu nilai biner yang dinamakan *allele*.

*Genetic Algorithm (GA)* merupakan prosedur iterative, bekerja dengan suatu kumpulan string sebagai kandidat solusi dengan jumlah konstan. Populasi ini kemudian berkembang dari generasi ke generasi melalui operator genetika. Setiap langkah interasi disebut generasi, individu dalam populasi saat itu akan dievaluasi dan diseleksi untuk menentukan populasi pada generasi selanjutnya.

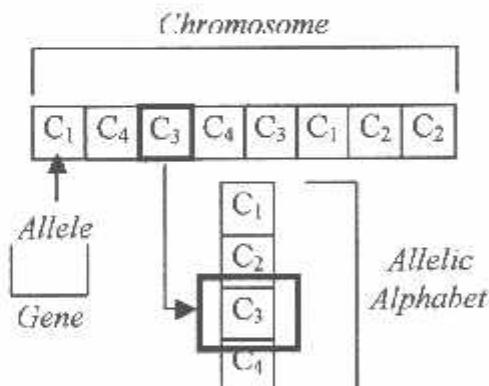
Selama proses evolusi *genetic*, kromosom yang lebih sehat memiliki kecenderungan menghasilkan *off spring* (keturunan) yang sehet pula, dan kromosom yang sehat diharapkan menghasilkan keturunan yang lebih banyak serta mungkin dapat bertahan pada generasi selanjutnya.

### 3.1.2. Operator *Genetic*

*Genetic Algorithm (GA)* merupakan metode yang memperlakukan kromosom atau populasi ( string yang akan menampilkan calon solusi dari suatu masalah ) ke populasi yang baru dengan menggunakan seleksi dan operator. Operator *genetic* yang digunakan adalah reproduksi, crossover, dan mutasi yang masing – masing menggunakan proses probabilitas dalam pemilihan dan pengoperasian.

### 3.1.3. Kromosom

Dalam kasus ini setiap gen berhubungan dengan satu *node* dalam system, oleh karena itu panjang kromosom sama dengan jumlah *node* yang dianalisis. Untuk masing – masing gen di gunakan *Allelic Alphabet* yang terbentuk didasarkan pada ukuran kapasitor yang dipasang.



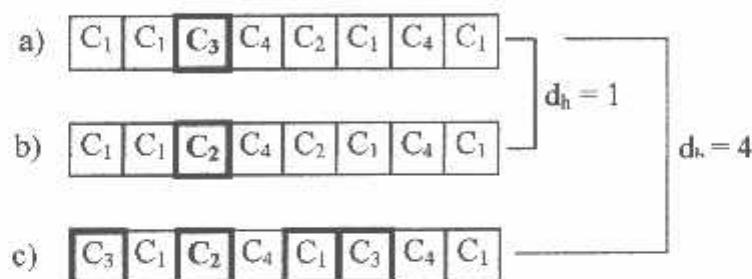
Gambar 3.1  
Chromosome dan Allelic Alphabet<sup>[3]</sup>

### 3.1.4. Populasi Awal

Populasi awal dihasilkan secara acak, seluruh individu adalah cocok

### 3.1.5. Crossover

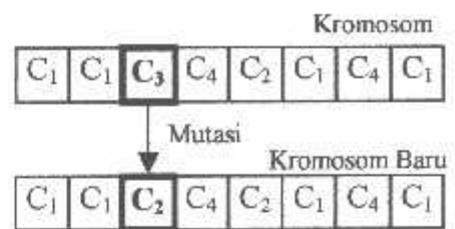
Crossover adalah operator genetika yang utama. Operator ini bekerja dengan mengambil 2 individu dan memotong *string* kromosom mereka pada posisi yang terpilih secara random.



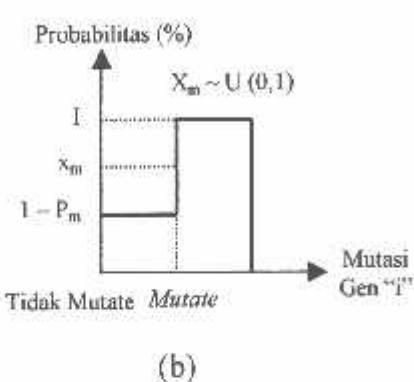
Gambar 3.2  
“a” dan “c” crossover individu [3]

### 3.1.6. Mutasi (Mutation)

Mutasi berfungsi untuk menghasilkan kromosom baru yang mempunyai sifat baru dengan cara melakukan perubahan pada sebuah generator atau lebih dari sebuah kromosom. Tujuan dilakukan mutasi adalah agar kromosom dalam populasi semakin bervariasi pada operasi mutasi ini digunakan operator *Genetic Algorithm (GA)*.

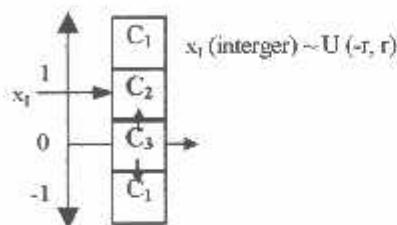


(a)



(b)

Dua berbagai kemungkinan mutasi dengan jarak  $r = 1$ , di dalam *allelic alphabet* untuk gen "i"



(c)

Gambar 3.3

- Mutasi Untuk Sebuah Gen Kromosom,
- aplikasi menyangkut mutasi,
- Mutasi berbagai Kemungkinan Dengan  $r = 1$  [3]

### 3.1.7. Seleksi

Metode seleksi dikenal sebagai seleksi berbasis ranking, digunakan dengan tekanan selektif 2.

### 3.1.8. Reinsertion

Proses pemasukan ulang menjaga individual terbaik di antara induk dengan keturunan.

### 3.1.9. Fungsi Fitness

Fungsi "fitness" meliputi kemampuan untuk membandingkan solusi dari satu generasi ke generasi yang lain.

Nilai fungsi *fitness* tidak dapat langsung dihubungkan dengan nilai tujuannya melainkan harus dirangking terlebih dahulu dengan nilai tujuannya. Dengan cara ini dapat ditentukan kromosom – kromosom mana yang layak digunakan dalam proses selanjutnya sehingga konvergensi awal dapat dihindari dan akan mempercepat penelitian ketika mendekati konvergen.

### **3.1.10. Kriteria Penghentian**

Kriteria untuk menghentikan algoritma berhubungan dengan operasi angka generasi arbitrasi.

## **3.2. Algoritma Dan *Flow Chart***

### **3.2.1. Algoritma *Genetic Algorithm***

1. Memasukan inputan data beban yang meliputi tegangan, sudut phasa tegangan, daya aktif (P), daya reaktif (Q) dan data impedansi saluran
2. Menentukan parameter inputan *Genetic Algorithm* yang meliputi jumlah populasi, maksimum generasi, nilai kemungkinan *crossover*, nilai kemungkinan mutasi dan panjang kromosom tiap-tiap individu
3. Generasi = 0, Populasi = 0
4. Melakukan *fitness* dari kromosom tiap-tiap individu
5. Melakukan proses statistik
6. Melakukan proses seleksi
7. Melakukan proses *crossover*
8. Melakukan proses mutasi
9. Apakah *offspring* sudah mencapai max populasi ?
  - a. Jika “ya” lanjut ke langkah 10

b. Jika “tidak” kembali ke langkah 5 dengan  $\text{Pop} = \text{Pop} + 1$

10. Menghitung *fitness* dari *offspring*

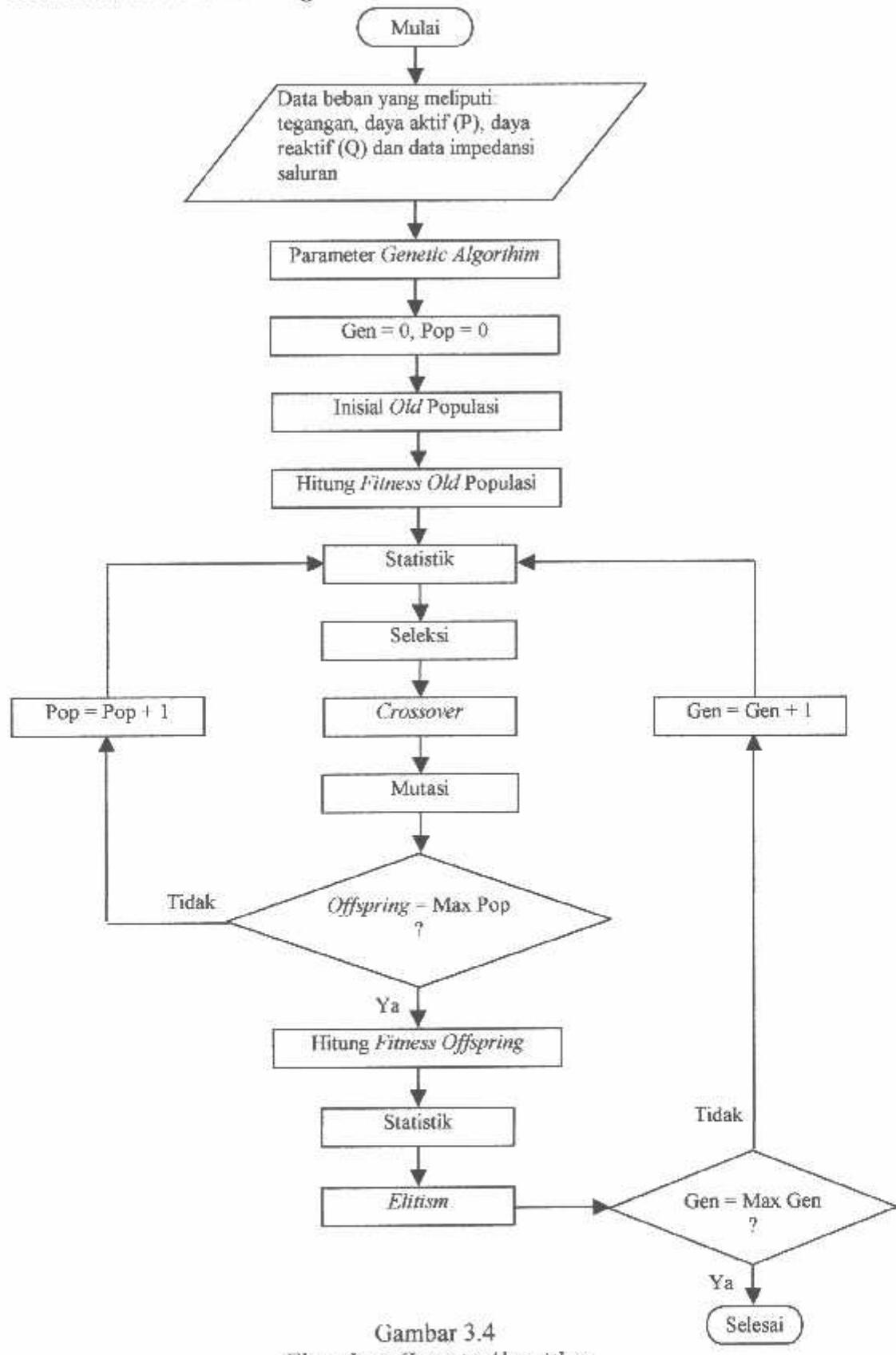
11. Melakukan proses *elitism*

12. Apakah Max Gen sudah terpenuhi ?

a. Jika “ya” selesai

b. Jika “tidak” kembali ke langkah 5 dengan  $\text{Gen} = \text{Gen} + 1$

### 3.2.2. Flowchart Genetic Algorithm

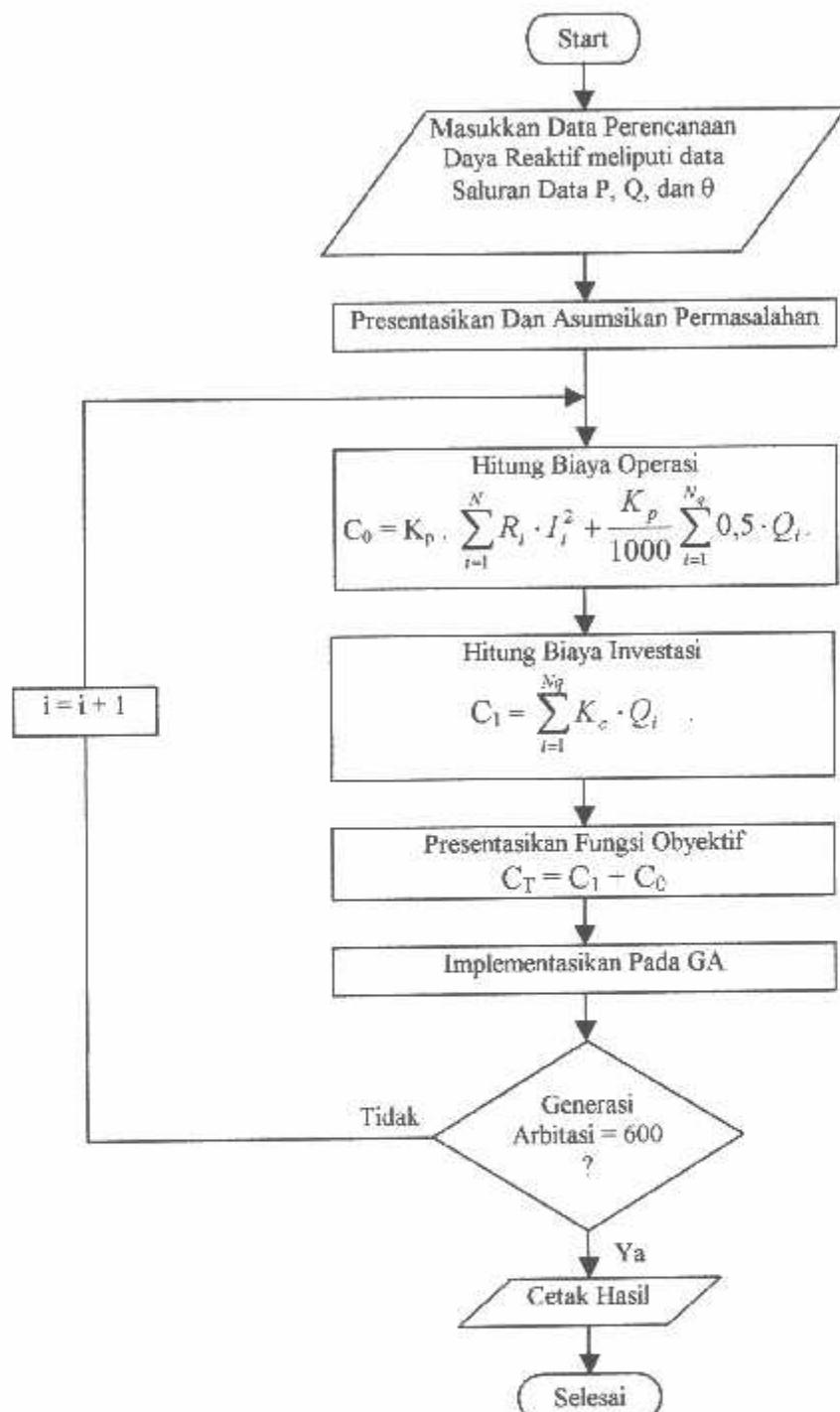


Gambar 3.4  
Flowchart Genetic Algorithm

### 3.2.3. Algoritma Program Operator Genetika

1. Mulai.
2. Masukkan data perencanaan daya reaktif meliputi data saluran, data P (daya nyata), Q (daya reaktif), dan  $\theta$  (sudut fasa).
3. Presentasikan dan asumsikan permasalahan dengan asumsi sebagai berikut :
  1. Asumsikan system distribusi seimbang.
  2. Pengaturan tap trafo dianggap sebagai nilai tetap.
4. Hitung biaya operasi  $C_0 = K_p \cdot \sum_{i=1}^N R_i \cdot I_i^2 + \frac{K_p}{1000} \sum_{i=1}^{Nq} 0,5 \cdot Q_i$ .
5. Hitung biaya investasi  $C_1 = \sum_{i=1}^{Nq} K_c \cdot Q_i$ .
6. Presentasikan fungsi obyektif  $C_T = C_1 + C_0$ .
7. Implementasikan pada GA.
8. Apakah generasi arbitasi sudah mencapai 600.
  1. Jika tidak, kembali ke langkah 4 dimana  $i = i + 1$ .
  2. Jika ya, lanjutkan ke langkah 9 .
9. Cetak hasil.
10. Selesai.

### 3.2.4. Flowchart Operator Genetika



Gambar 3.5  
Flowchart Operator Genetika

**BAB IV**

**ANALISA PENEMPATAN KAPASITOR MENGGUNAKAN**

**METODE OPERATOR GENETIKA JARINGAN DISTRIBUSI**

**PADA PENYULANG SYIAR**

**4.1. Aplikasi Penempatan Kapasitor Dengan Metode Operator Genetika Dengan *Allelic Alphabet* Dan Konsep Jarak Hamming.**

Dalam memilih suatu metode untuk aplikasi praktis sering sulit, pilihan itu memerlukan analisis yang cermat atas kelebihan-kelebihan dan kekurangan-kekurangan dari metode yang ada, seperti *Heuristic*, *Simulated Annealing*, *Tabu Search*, Logika Fuzzy , Pemrograman evolutif dan Logika Fuzzy. Salah satu solusi dari permasalahan diatas maka digunakan suatu alternatif dengan menggunakan metode Operator Genetika Dengan *Allelic Alphabet* Dan Konsep Jarak Hamming yang menyediakan cara yang lebih baik dalam menganalisa masalah penempatan kapasitor.

**4.2. Sistem Distribusi Tenaga Listrik 20 kV GI PIER Pasuruan Penyulang SYIAR.**

Dalam Perhitungan aliran daya data yang diambil dari GI PIER Pasuruan yang melayani 5 ( Lima ) buah penyulang

Namun pada skripsi ini hanya menganalisa satu penyulang saja, yaitu pada penyulang SYIAR. Sistem Distribusi Radial Penyulang SYIAR memakai :

- Tegangan Dasar : 20kV
- Daya Dasar : 30MVA

Selanjutnya bus-bus yang diklarifikasi, yaitu busbar GI PIER Pasuruan diasumsikan sebagai *slack bus*, sedangkan bus-bus yang lain sepanjang saluran Radial sebagai *load bus*. Dalam hal ini tidak ada bus generator karena sepanjang saluran tidak terdapat pembangkitan.

#### 4.3.Data Saluran

Jaringan distribusi Penyulang SYIAR menggunakan saluran udara dengan spesifikasi sebagai berikut :

Tabel 4.1  
Data Spesifikasi Kabel Saluran

Jenis Konduktor	Penampang Nominal (mm <sup>2</sup> )	Impedansi Saluran (Ω/km)
AAAC	150	0,2162 + j 0,3305
AAAC	120	0,2688 + j 0,3376
AAAC	70	0,4608 + j 0,3572
AAAC	50	0,6452 + j 0,3678
AAAC	35	0,9217 + j 0,3795

Dari data spesifikasi saluran pada penyulang SYIAR diketahui bahwa jenis yang digunakan adalah AAAC (*all-aluminium-alloyconductors*) dengan penampang nominal 150 mm<sup>2</sup> dan impedansi saluran 0,2162 + j 0,3305 Ω/km.

Untuk perhitungan Impedansi saluran diberikan contoh pada node 1 ke 2 sebagai berikut :

Jarak antara node dari 1 ke 2 dengan panjang = 581,5 m = 0,5815 km

$$R = 0,5815 \text{ km} \times 0,2162 \Omega/\text{km} = 0,1257 \Omega$$

$$X = 0,5815 \text{ km} \times 0,3305 \Omega/\text{km} = 0,1922 \Omega$$

Tabel 4.2  
Data Hasil Perhitungan Impedansi Saluran Penyulang SYIAR

No Saluran	Node		Impedansi Saluran			
	from	to	R	X	R	X
1	1	2	0,12572	0,19219	0,00943	0,01442
2	2	3	0,30997	0,47384	0,02325	0,03555
3	3	4	0,18721	0,28618	0,01404	0,02147
4	4	5	0,01892	0,02892	0,00142	0,00217
5	5	6	0,14823	0,22659	0,01112	0,01700
6	6	7	0,45054	0,68873	0,03380	0,05167
7	7	8	0,03366	0,05146	0,00253	0,00386
8	8	9	0,09005	0,13765	0,00676	0,01033
9	8	10	0,19408	0,29669	0,01456	0,02226
10	6	11	0,10042	0,15352	0,00753	0,01152
11	11	12	0,09798	0,14978	0,00735	0,01124
12	12	13	0,30800	0,47083	0,02311	0,03532
13	13	14	0,26679	0,40784	0,02001	0,03060
14	11	15	0,02056	0,03143	0,00154	0,00236
15	15	16	0,01090	0,01666	0,00082	0,00125
16	16	17	0,07336	0,11214	0,00550	0,00841
17	17	18	0,03963	0,06058	0,00297	0,00454
18	18	19	0,08732	0,13349	0,00655	0,01001
19	19	20	0,06603	0,10093	0,00495	0,00757
20	20	21	0,06393	0,09773	0,00480	0,00733
21	21	22	0,27209	0,41593	0,02041	0,03120
22	22	23	0,08853	0,13534	0,00664	0,01015
23	23	24	0,03349	0,05119	0,00251	0,00384
24	24	25	0,13675	0,20904	0,01026	0,01568
25	25	26	0,00659	0,01008	0,00049	0,00076
26	26	27	0,04417	0,06752	0,00331	0,00507
27	27	28	0,24949	0,38140	0,01872	0,02861
28	28	29	0,18946	0,28962	0,01421	0,02173
29	29	30	0,03098	0,04736	0,00232	0,00355
30	30	31	0,18375	0,28089	0,01378	0,02107
31	31	32	0,05950	0,09095	0,00446	0,00682
32	32	33	0,24410	0,37315	0,01831	0,02799
33	31	34	0,02281	0,03487	0,00171	0,00262
34	34	35	0,10799	0,16508	0,00810	0,01238
35	35	36	0,06173	0,09436	0,00463	0,00708
36	36	37	0,06473	0,09895	0,00486	0,00742

37	37	38	0,20416	0,31209	0,01532	0,02341
38	20	39	0,06683	0,10216	0,00501	0,00766
39	39	40	0,14036	0,21456	0,01053	0,01610
40	40	41	0,02311	0,03533	0,00173	0,00265
41	41	42	0,01092	0,01669	0,00082	0,00125
42	42	43	0,35751	0,54651	0,02682	0,04100
43	43	44	0,17073	0,26100	0,01281	0,01958
44	44	45	0,07269	0,11111	0,00545	0,00834
45	45	46	0,17192	0,26281	0,01290	0,01972
46	43	47	0,12369	0,18908	0,00928	0,01418
47	47	48	0,21049	0,32177	0,01579	0,02414
48	48	49	0,16224	0,24801	0,01217	0,01861
49	39	50	0,01727	0,02641	0,00130	0,00198
50	50	51	0,02808	0,04293	0,00211	0,00322
51	51	52	0,20574	0,31450	0,01543	0,02359
52	52	53	0,37154	0,56796	0,02787	0,04261
53	53	54	0,27903	0,42654	0,02093	0,03200
54	54	55	0,24329	0,37191	0,01825	0,02790
55	54	56	0,29286	0,44770	0,02197	0,03359
56	56	57	0,23311	0,35635	0,01749	0,02673
57	57	58	0,08639	0,13207	0,00648	0,00991
58	58	59	0,06441	0,09846	0,00483	0,00739
59	57	60	0,23685	0,36206	0,01777	0,02716
60	60	61	0,25202	0,38526	0,01891	0,02890
61	61	62	0,10793	0,16499	0,00810	0,01238
62	62	63	0,12377	0,18921	0,00929	0,01419
63	49	64	0,19629	0,30006	0,01473	0,02251
64	64	65	0,37184	0,56843	0,02790	0,04264
65	65	66	0,02951	0,04511	0,00221	0,00338
66	66	67	0,17787	0,27190	0,01334	0,02040
67	67	68	0,02387	0,03649	0,00179	0,00274
68	64	69	0,19292	0,29491	0,01447	0,02212
69	69	70	0,13091	0,20012	0,00982	0,01501
70	70	71	0,01749	0,02674	0,00131	0,00201
71	71	72	0,05526	0,08448	0,00415	0,00634
72	72	73	0,04313	0,06593	0,00324	0,00495
73	73	74	0,03386	0,05176	0,00254	0,00388
74	74	75	0,08244	0,12602	0,00618	0,00945
75	75	76	0,10226	0,15633	0,00767	0,01173
76	76	77	0,10672	0,16313	0,00801	0,01224
77	77	78	0,15687	0,23981	0,01177	0,01799

78	78	79	0,20334	0,31084	0,01525	0,02332
79	79	80	0,02778	0,04247	0,00208	0,00319
80	74	81	0,18585	0,28410	0,01394	0,02131
81	81	82	0,28571	0,43676	0,02143	0,03276
82	82	83	0,30873	0,47195	0,02316	0,03541
83	83	84	0,25133	0,38421	0,01885	0,02882
84	84	85	0,19231	0,29398	0,01443	0,02205
85	81	86	0,20204	0,30885	0,01516	0,02317
86	86	87	0,20660	0,31583	0,01550	0,02369
87	86	88	0,24493	0,37442	0,01837	0,02809

#### 4.4.Pembebanan Sistem 20 kV Penyalang SYIAR

Pembebanan diperoleh dengan mengambil data dari masing-masing trafo distribusi, dimana besarnya beban pada masing-masing fasa diasumsikan seimbang. Jika besarnya pembebanan adalah nol, maka pada node tidak terdapat trafo distribusi tetapi hanya merupakan simpul. Pada tahap ini rugi-rugi yang terjadi pada trafo distribusi diabaikan, dengan mengasumsikan faktor daya 0,86.

Dibawah ini diberikan contoh perhitungan pembebanan pada *Node 2*.

$$\text{Kapasitas Trafo} = 157,5 \text{ kVA}$$

$$\text{Beban Gardu} = 35 \%$$

$$\text{Faktor Daya} = -\cos \phi = 0,86$$

$$\sin \phi = 0,5102$$

$$\text{Pembebanan} = 157,5 \times 35 \% = 55 \text{ kVA}$$

$$\text{Beban Aktif} = 55 \times \cos \phi = 55 \times 0,86 = 47,3 \text{ kW}$$

$$\text{Beban Reaktif} = 55 \times \sin \phi = 55 \times 0,5102 = 28,07 \text{ kVAR}$$

Untuk hasil perhitungan diatas maka node 3 sampai dengan 10 terlihat pada Tabel 4.2.

Dan selanjutnya untuk merubah P (daya aktif) dan Q (daya reaktif) kedalam bentuk per-unit (pu) :

Dari hasil perhitungan pembebanan diatas, pada node 2 diketahui nilai :

$$P_2 = 47,3 \text{ kW}$$

$$Q_2 = 28,07 \text{ kVAR}$$

Untuk merubah beban aktif (P) dan beban reaktif (Q) dalam pu

$$p = \frac{P(kW)}{P_{dasar}} \quad Q = \frac{Q(kVAR)}{P_{dasar}}$$

Maka :

$$P_2 = \frac{47,3}{30} = 1,577 \text{ pu} \quad Q_2 = \frac{28,07}{30} = 0,936 \text{ pu}$$

Dengan cara yang sama pada contoh diatas maka diperoleh hasil perhitungan seperti Tabel 4.3 dibawah ini.

Tabel 4.3  
Data Hasil Perhitungan Pembebanan Sistem 20 kV Penyulang SYIAR

No	Kode Trafo	Pembebanan				
		S (kVA)	P (kW)	Q (kVAR)	P (pu)	Q (pu)
1	-	0	0	0	0,00	0,00
2	-	0	0	0	0,00	0,00
3	-	0	0	0	0,00	0,00
4	71	55	47,3	28,07	1,58	0,94
5	-	0	0	0	0,00	0,00
6	-	0	0	0	0,00	0,00
7	69	71	61,06	36,23	2,04	1,21
8	-	0	0	0	0,00	0,00
9	203	65	55,90	33,17	1,86	1,11
10	90	144	123,84	73,48	4,13	2,45
11	-	0	0	0	0,00	0,00
12	-	0	0	0	0,00	0,00

13	-	0	0	0	0,00	0,00
14	26	96	81,7	48,48	2,72	1,62
15	-	0	0	0	0,00	0,00
16	130	89	76,54	45,42	2,55	1,51
17	-	0	0	0	0,00	0,00
18	-	0	0	0	0,00	0,00
19	202	139	119,54	70,93	3,98	2,36
20	-	0	0	0	0,00	0,00
21	-	0	0	0	0,00	0,00
22	-	0	0	0	0,00	0,00
23	-	0	0	0	0,00	0,00
24	78	74	63,64	37,76	2,12	1,26
25	-	0	0	0	0,00	0,00
26	-	0	0	0	0,00	0,00
27	-	0	0	0	0,00	0,00
28	199	79	67,94	40,31	2,26	1,34
29	-	0	0	0	0,00	0,00
30	-	0	0	0	0,00	0,00
31	-	0	0	0	0,00	0,00
32	-	0	0	0	0,00	0,00
33	200	92	79,12	46,95	2,64	1,57
34	-	0	0	0	0,00	0,00
35	251	169	145,34	86,24	4,84	2,87
36	-	0	0	0	0,00	0,00
37	-	0	0	0	0,00	0,00
38	252	135	116,1	68,89	3,87	2,30
39	-	0	0	0	0,00	0,00
40	-	0	0	0	0,00	0,00
41	41	122	104,92	62,26	3,50	2,08
42	-	0	0	0	0,00	0,00
43	-	0	0	0	0,00	0,00
44	-	0	0	0	0,00	0,00
45	-	0	0	0	0,00	0,00
46	186	111	95,46	56,64	3,18	1,89
47	-	0	0	0	0,00	0,00
48	-	0	0	0	0,00	0,00
49	134	95	81,7	48,48	2,72	1,62
50	-	0	0	0	0,00	0,00
51	-	0	0	0	0,00	0,00
52	-	0	0	0	0,00	0,00
53	-	0	0	0	0,00	0,00

54	-	0	0	0	0,00	0,00
55	144	85	73,1	43,37	2,44	1,45
56	125	79	67,94	40,31	2,26	1,34
57	-	0	0	0	0,00	0,00
58	-	0	0	0	0,00	0,00
59	186	105	90,3	53,58	3,01	1,79
60	-	0	0	0	0,00	0,00
61	-	0	0	0	0,00	0,00
62	-	0	0	0	0,00	0,00
63	246	88	75,68	44,91	2,52	1,50
64	-	0	0	0	0,00	0,00
65	135	121	104,06	61,75	3,47	2,06
66	-	0	0	0	0,00	0,00
67	-	0	0	0	0,00	0,00
68	25	109	93,74	55,62	3,12	1,85
69	-	0	0	0	0,00	0,00
70	245	85	73,1	43,37	2,44	1,45
71	-	0	0	0	0,00	0,00
72	-	0	0	0	0,00	0,00
73	154	106	91,16	54,09	3,04	1,80
74	94	124	106,64	63,28	3,55	2,11
75	-	0	0	0	0,00	0,00
76	95	79	64,95	40,31	2,17	1,34
77	-	0	0	0	0,00	0,00
78	-	0	0	0	0,00	0,00
79	-	0	0	0	0,00	0,00
80	145	115	98,9	58,68	3,30	1,96
81	-	0	0	0	0,00	0,00
82	117	135	116,1	68,89	3,87	2,30
83	201	132	113,52	57,36	3,78	1,91
84	86	101	86,86	51,54	2,90	1,72
85	85	105	90,3	53,58	3,01	1,79
86	-	0	0	0	0,00	0,00
87	185	99	85,14	50,52	2,84	1,68
88	239	81	69,66	41,33	2,32	1,38

#### 4.5. Analisa Perhitungan

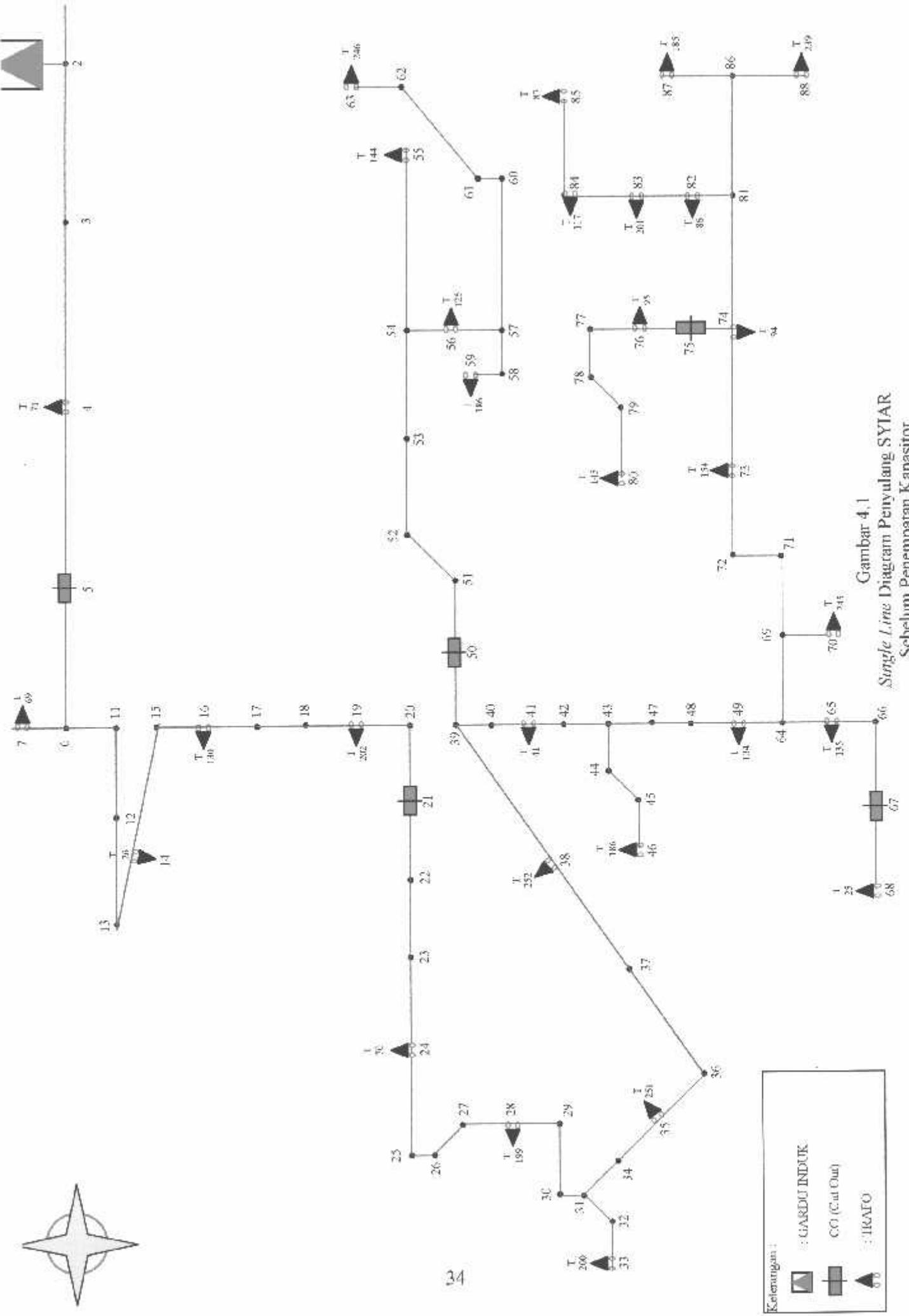
Perhitungan penempatan kapasitor diawali dengan melakukan studi aliran daya dengan menggunakan metode *Newton Raphson*. Studi aliran daya dilakukan untuk mengetahui harga tegangan dan sudut fasa tiap-tiap bus, arus yang mengalir pada saluran, dan aliran daya tiap saluran. Setelah studi aliran daya dilakukan, barulah dilakukan perhitungan rugi-rugi daya pada saluran.

Untuk menentukan lokasi, kapasitor, dan setting kapasitor terlebih dahulu menentukan rugi daya saluran terbesar yang digunakan sebagai input untuk menjalankan metode Operator Genetika agar kapasitor dapat diletakkan pada lokasi yang memberikan profil tegangan yang paling optimum. Sedangkan untuk pencarian grafik yang optimal secara random ditentukan oleh program komputer demikian pula untuk perhitungan besar nilai penghematan.

Untuk memudahkan perhitungan dan analisis pada sistem tenaga, biasanya dipakai harga-harga dalam per-unit. Harga per-satuan adalah harga sebenarnya dibagi dengan harga dasar, dimana harga dasar ini dapat diilih sembarang. Harga yang dipilih pada studi ini adalah 20 kV dan 30 MVA sebagai harga tegangan dasar dan daya dasar. Mengingat bahwa pada jaringan tidak dilakukan pengukuran faktor daya maka pada perhitungan ini diasumsikan nilai faktor daya sebesar 0,86. Pada penyulang ini jumlah bus dan jumlah saluran masing-masing adalah :

- Slack bus                    = 1
- Load bus                    = 87
- Jumlah saluran            = 88

Setelah dilakukan analisis aliran daya dengan menggunakan metode *Newton Raphson* maka diperoleh profil tegangan tiap-tiap bus, arus tiap saluran, dan Tabel Aliran daya tiap saluran.



**Tabel 4.4**  
**Profil Tegangan dan Sudut Fasa Tiap Bus Penyulang SYIAR**  
**Sebelum Kompensasi**

<b>Bus</b>	<b>absV(pu)</b>	<b>sudV(deg)</b>
1	1,00000	0,00000
2	0,99822	-0,04785
3	0,99383	-0,16655
4	0,99118	-0,23875
5	0,99092	-0,24595
6	0,98886	-0,30246
7	0,98833	-0,31733
8	0,98830	-0,31816
9	0,98828	-0,31885
10	0,98819	-0,32146
11	0,98758	-0,33756
12	0,98633	-0,37189
13	0,98233	-0,48807
14	0,97893	-0,58279
15	0,97868	-0,58988
16	0,97854	-0,59364
17	0,97767	-0,61819
18	0,97695	-0,62266
19	0,9759	-0,65197
20	0,97515	-0,67306
21	0,97443	-0,69351
22	0,97134	-0,78089
23	0,97034	-0,80945
24	0,96996	-0,82026
25	0,96845	-0,82027
26	0,96838	-0,86535
27	0,96789	-0,87927
28	0,96514	-0,95819
29	0,96312	-1,01656
30	0,96279	-1,02616
31	0,96083	-1,08301
32	0,96080	-1,08369
33	0,96071	-1,08649
34	0,96059	-1,08982
35	0,95948	-1,12213
36	0,95889	-1,13933
37	0,95828	-1,15738
38	0,95633	-1,21448
39	0,95573	-1,23208
40	0,95469	-1,26279
41	0,95452	-1,26785
42	0,95444	-1,27007
43	0,95198	-1,34308
44	0,95189	-1,34549
45	0,95186	-1,34652

46	0,95178	-1,34894
47	0,95118	-1,36668
48	0,94983	-1,40693
49	0,94879	-1,43803
50	0,95570	-1,23286
51	0,95566	-1,26412
52	0,95534	-1,24339
53	0,95477	-1,26015
54	0,95435	-1,27274
55	0,95426	-1,27536
56	0,95400	-1,28282
57	0,95381	-1,28852
58	0,95377	-1,28967
59	0,95374	-1,29053
60	0,95372	-1,29116
61	0,95362	-1,29397
62	0,95358	-1,29517
63	0,95354	-1,29660
64	0,94761	-1,47336
65	0,94724	-1,48434
66	0,94723	-1,48475
67	0,94715	-1,48724
68	0,94713	-1,48758
69	0,94665	-1,50246
70	0,94660	-1,50390
71	0,94657	-1,50492
72	0,94631	-1,51266
73	0,94611	-1,51871
74	0,94597	-1,52300
75	0,94590	-1,52506
76	0,94582	-1,52761
77	0,94576	-1,52919
78	0,94568	-1,53152
79	0,94558	-1,53453
80	0,94557	-1,53494
81	0,94545	-1,53493
82	0,94488	-1,55682
83	0,94455	-1,56721
84	0,94432	-1,57390
85	0,94424	-1,57651
86	0,94529	-1,54362
87	0,94521	-1,54626
88	0,94521	-1,54618

Tabel 4.5  
Aliran Arus Tiap Saluran Penyulang SYIAR  
Sebelum Kompensasi

Dari	Ke	Arus re (A)	Arus im (A)	Dari	Ke	Arus re (A)	Arus im (A)
1	2	145,691	90,099	2	1	-145,691	-90,099
2	3	145,691	90,099	3	2	-145,691	-90,099
3	4	145,691	90,099	4	3	-145,691	-90,099
4	5	143,31	88,673	5	4	-143,31	-88,673
5	6	143,31	88,673	6	5	-143,31	-88,673
6	7	12,143	7,297	7	6	-12,143	-7,297
7	8	9,064	5,447	8	7	-9,064	-5,447
8	9	2,819	1,694	9	8	-2,819	-1,694
8	10	6,245	3,753	10	8	-6,245	-3,753
6	11	131,168	81,376	11	6	-131,168	-81,376
11	12	131,168	81,376	12	11	-131,168	-81,376
12	13	131,168	81,376	13	12	-131,168	-81,376
13	14	131,168	81,376	14	13	-131,168	-81,376
14	15	127,020	78,858	15	14	-127,020	-78,858
15	16	127,020	78,858	16	15	-127,020	-78,858
16	17	123,134	76,496	17	16	-123,134	-76,496
17	18	123,134	76,496	18	17	-123,134	-76,496
18	19	123,134	76,496	19	18	-123,134	-76,496
19	20	117,051	72,793	20	19	-117,051	-72,793
20	21	117,051	72,793	21	20	-117,051	-72,793
21	22	117,051	72,793	22	21	-117,051	-72,793
22	23	117,051	72,793	23	22	-117,051	-72,793
23	24	117,051	72,793	24	23	-117,051	-72,793
24	25	113,798	70,800	25	24	-113,798	-70,800
25	26	113,798	70,800	26	25	-113,798	-70,800
26	27	113,798	70,800	27	26	-113,798	-70,800
27	28	113,798	70,800	28	27	-113,798	-70,800
28	29	110,314	68,648	29	28	-110,314	-68,648
29	30	110,314	68,648	30	29	-110,314	-68,648
30	31	110,314	68,648	31	30	-110,314	-68,648
31	32	4,071	2,521	32	31	-4,071	-2,521
32	33	4,071	2,521	33	32	-4,071	-2,521
31	34	106,244	66,126	34	31	-106,244	-66,126
34	35	106,244	66,126	35	34	-106,244	-66,126
35	36	98,759	61,485	36	35	-98,759	-61,485
36	37	98,759	61,485	37	36	-98,759	-61,485
37	38	98,759	61,485	38	37	-98,759	-61,485
38	39	92,767	57,755	39	38	-92,767	-57,755
39	40	76,892	47,847	40	39	-76,892	-47,847
40	41	76,892	47,847	41	40	-76,892	-47,847
41	42	71,469	44,465	42	41	-71,469	-44,465
42	43	71,469	44,465	43	42	-71,469	-44,465

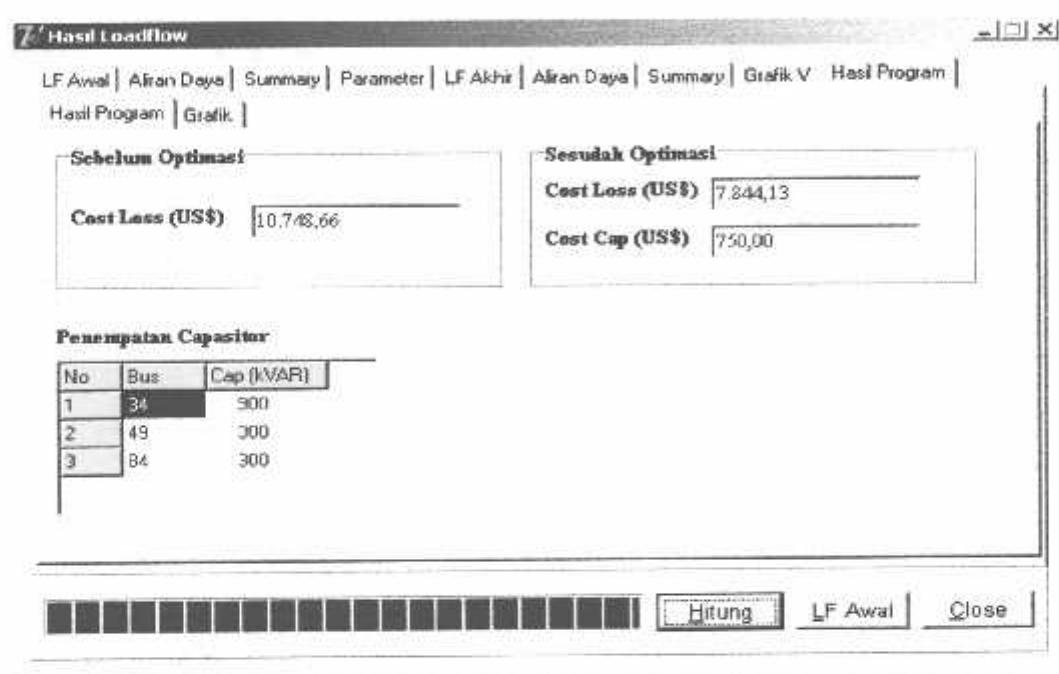
43	44	4,943	3,093	44	43	-4,943	-3,093
44	45	4,943	3,093	45	44	-4,943	-3,093
45	46	4,943	3,093	46	45	-4,943	-3,093
43	47	66,526	41,372	47	43	-66,526	-41,372
47	48	66,526	41,372	48	47	-66,526	-41,372
48	49	66,526	41,372	49	48	-66,526	-41,372
39	50	15,875	9,908	50	39	-15,875	-9,908
50	51	15,875	9,908	51	50	-15,875	-9,908
51	52	15,875	9,908	52	51	-15,875	-9,908
52	53	15,875	9,908	53	52	-15,875	-9,908
53	54	15,875	9,908	54	53	-15,875	-9,908
54	55	3,779	2,357	55	54	-3,779	-2,357
54	56	12,096	7,551	56	54	-12,096	-7,551
56	57	8,584	5,359	57	56	-8,584	-5,359
57	58	4,67	2,915	58	57	-4,67	-2,915
58	59	4,67	2,915	59	58	-4,67	-2,915
57	60	3,914	2,444	60	57	-3,914	-2,444
60	61	3,914	2,444	61	60	-3,914	-2,444
61	62	3,914	2,444	62	61	-3,914	-2,444
62	63	3,914	2,444	63	62	-3,914	-2,444
49	64	62,286	38,710	64	49	-62,286	-38,710
64	65	10,277	6,464	65	64	-10,277	-6,464
65	66	4,871	3,064	66	65	-4,871	-3,064
66	67	4,871	3,064	67	66	-4,871	-3,064
67	68	4,871	3,064	68	67	-4,871	-3,064
64	69	52,009	32,246	69	64	-52,009	-32,246
69	70	3,800	2,391	70	69	-3,800	-2,391
69	71	48,209	29,855	71	69	-48,209	-29,855
71	72	48,209	29,855	72	71	-48,209	-29,855
72	73	48,209	29,855	73	72	-48,209	-29,855
73	74	43,469	26,869	74	73	-43,469	-26,869
74	75	8,678	5,468	75	74	-8,678	-5,468
75	76	8,678	5,468	76	75	-8,678	-5,468
76	77	5,145	3,242	77	76	-5,145	-3,242
77	78	5,145	3,242	78	77	-5,145	-3,242
78	79	5,145	3,242	79	78	-5,145	-3,242
79	80	5,145	3,242	80	79	-5,145	-3,242
74	81	29,245	17,908	81	74	-29,245	-17,908
81	82	21,190	12,83	82	81	-21,190	-12,83
82	83	15,148	9,019	83	82	-15,148	-9,019
83	84	9,224	5,82	84	83	-9,224	-5,82
84	85	4,702	2,966	85	84	-4,702	-2,966
81	86	8,055	5,078	86	81	-8,055	-5,078
86	87	4,430	2,793	87	86	-4,430	-2,793
86	88	3,625	2,285	88	86	-3,625	-2,285

Tabel 4.6  
Aliran Daya Tiap Saluran Penyalang SYIAR  
Sebelum Kompensasi

Dari	Ke	P (kW)	Q (kVAR)	Dari	Ke	P (kW)	Q (kVAR)
1	2	2913,812	1801,978	2	1	-2910,123	-1796,339
2	3	2910,123	1796,339	3	2	-2901,027	-1782,434
3	4	2901,027	1782,434	4	3	-2895,534	-1774,037
4	5	2848,234	1745,967	5	4	-2847,697	-1745,146
5	6	2847,697	1745,146	6	5	-2843,487	-1738,710
6	7	240,905	143,041	7	6	-179,751	-142,903
7	8	179,755	106,673	8	7	-55,900	-106,667
8	9	55,901	33,171	9	8	-123,840	-33,170
8	10	123,85	73,496	10	8	-2600,189	-73,480
6	11	2602,581	1595,669	11	6	-2597,854	-1592,011
11	12	2600,189	1592,011	12	11	-2597,854	-1588,442
12	13	2597,854	1588,442	13	12	-2590,515	-1576,747
13	14	2590,515	1576,747	14	13	-2584,159	-1567,030
14	15	2502,459	1518,55	15	14	-2501,999	-1517,847
15	16	2501,999	1517,847	16	15	-2501,755	-1517,475
16	17	2425,215	1472,055	17	16	-2423,674	-1469,698
17	18	2423,674	1469,698	18	17	-2422,000	-1468,425
18	19	2422,000	1468,425	19	18	-2420,166	-1465,620
19	20	2300,626	1394,690	20	19	-2299,371	-1392,772
20	21	2299,371	1392,772	21	20	-2298,156	-1390,916
21	22	2298,156	1390,916	22	21	-2292,987	-1383,013
22	23	2292,987	1383,013	23	22	-2291,305	-1380,442
23	24	2291,305	1380,442	24	23	-2290,668	-1379,469
24	25	2227,028	1341,709	25	24	-2224,572	-1337,954
25	26	2224,572	1337,954	26	25	-2224,454	-1337,773
26	27	2224,454	1337,773	27	26	2223,660	-1336,560
27	28	2223,660	1336,560	28	27	-2219,179	-1329,709
28	29	2151,235	1289,299	29	28	-2148,040	-1284,410
29	30	2148,040	1284,410	30	29	-2147,517	-1283,610
30	31	2147,517	1283,610	31	30	-2144,415	-1278,868
31	32	79,127	46,961	32	31	-79,126	-46,959
32	33	79,126	46,959	33	32	-79,120	-46,950
31	34	2065,288	1231,908	34	31	-2064,931	-1231,362
34	35	2064,931	1231,362	35	34	-2063,240	-1228,777
35	36	1917,900	1142,537	36	35	-1917,065	-1141,260
36	37	1917,065	1141,260	37	36	-1916,188	-1139,920
37	38	1916,188	1139,920	38	37	-1913,425	-1135,697
38	39	1797,325	1066,807	39	38	-1796,527	-1065,587
39	40	1489,083	882,769	40	39	-1487,932	-881,009
40	41	1487,932	881,009	41	40	-1487,743	-880,719
41	42	1382,823	818,459	42	41	-1382,745	-818,341
42	43	1382,745	818,341	43	42	1380,212	-814,469
43	44	95,474	56,662	44	43	-95,468	-56,653
44	45	95,468	56,653	45	44	-95,466	-56,649
45	46	95,466	56,649	46	45	-95,460	-56,640

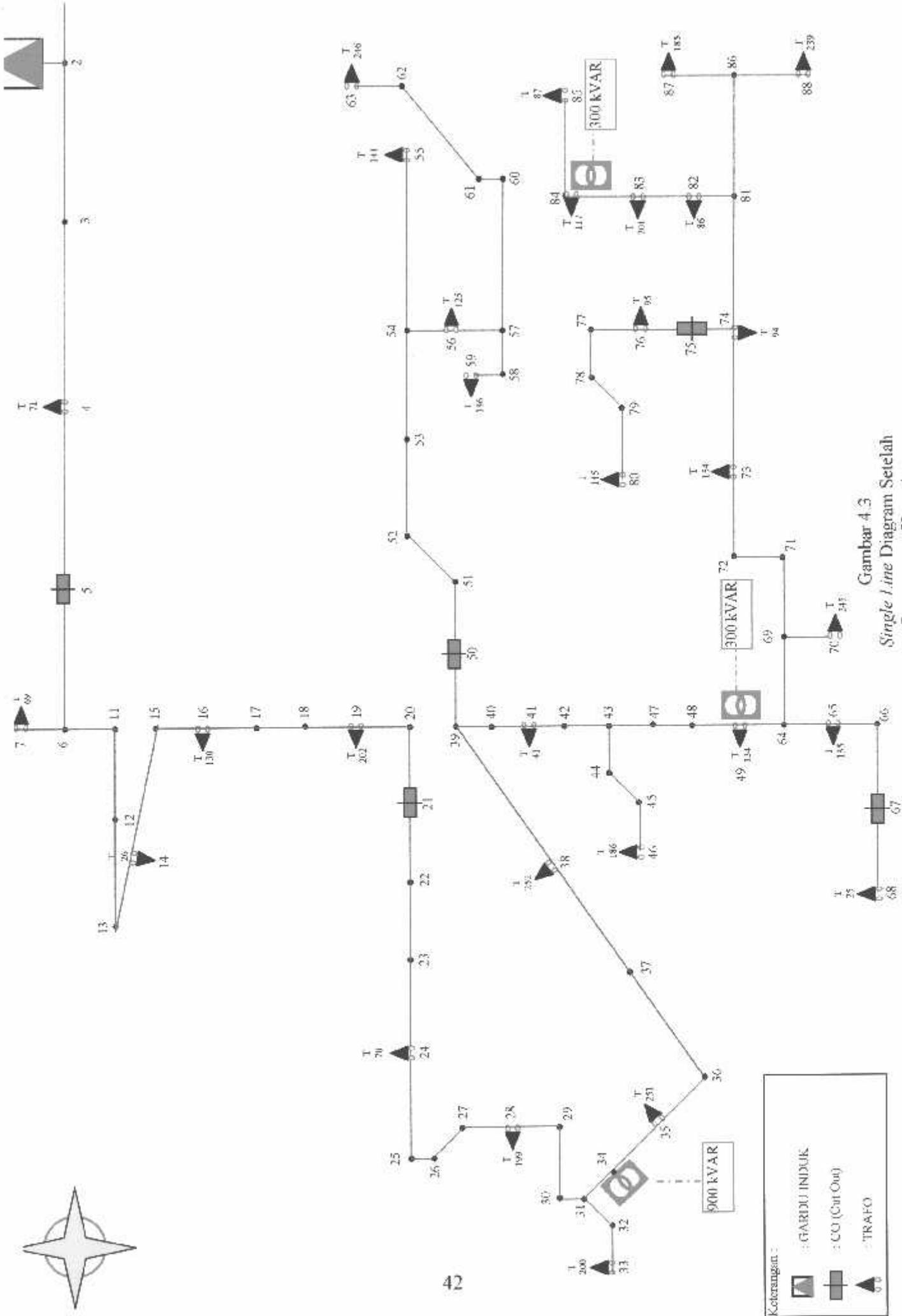
43	47	1284,738	757,807	47	43	-1283,979	-756,647
47	48	1283,979	756,647	48	47	-1282,687	-754,672
48	49	1282,687	754,672	49	48	-1281,692	-753,150
39	50	307,444	182,818	50	39	-307,438	-182,809
50	51	307,438	182,809	51	50	-307,428	-182,794
51	52	307,428	182,794	52	51	-307,356	-182,684
52	53	307,356	182,684	53	52	-307,226	-182,485
53	54	307,226	182,485	54	53	-307,128	-182,335
54	55	73,105	43,377	55	54	-73,100	-43,370
54	56	234,023	138,958	56	54	-233,964	-138,867
56	57	166,024	98,557	57	56	-166,000	-98,521
57	58	166,024	53,587	58	57	-90,302	-53,583
58	59	90,305	53,583	59	58	-90,300	-53,580
57	60	90,302	44,934	60	57	-75,690	-44,926
60	61	75,695	44,926	61	60	-75,685	-44,918
61	62	75,690	44,918	62	61	-75,683	-44,914
62	63	75,685	44,914	63	62	-75,680	-44,910
49	64	75,683	704,670	64	49	-1198,936	-703,056
64	65	1199,992	117,465	65	64	-197,808	-117,382
65	66	197,862	55,632	66	65	-93,747	-55,630
66	67	93,748	55,630	67	66	-93,741	-55,621
67	68	93,747	55,621	68	67	-93,740	-55,620
64	69	93,741	585,591	69	64	-1000,351	-584,486
69	70	1001,073	43,374	70	69	-73,100	-43,370
69	71	73,103	541,112	71	69	-927,192	-541,026
71	72	927,192	541,026	72	71	-927,014	-540,755
72	73	927,014	540,755	73	72	-926,876	-540,543
73	74	835,716	486,453	74	73	-835,627	-486,317
74	75	166,878	99,048	75	74	-166,869	-99,034
75	76	166,869	99,034	76	75	-166,858	-99,018
76	77	98,918	58,708	77	76	-98,914	-58,702
77	78	98,914	58,702	78	77	-98,909	-58,693
78	79	98,909	58,693	79	78	-98,901	-58,682
79	80	98,901	58,682	80	79	-98,900	-58,680
74	81	562,110	323,990	81	74	-561,891	-323,655
81	82	407,063	231,762	82	81	-406,887	-231,494
82	83	290,787	162,604	83	82	-290,716	-162,495
83	84	177,196	105,135	84	83	-177,166	-105,089
84	85	90,306	53,549	85	84	-90,300	-53,540
81	86	154,828	91,894	86	81	-154,810	-91,866
86	87	85,146	50,529	87	86	-85,140	-50,520
86	88	69,664	41,337	88	86	-69,660	-41,330

Dari hasil analisa diatas dapat disimpulkan bahwa penyulang SYIAR perlu dipasang kapasitor untuk menaikan profil tegangan yang dianggap krisis yang beroperasi diluar batas yang diijinkan oleh PT PLN ( 0,95% - 1,05% ) yang terjadi pada bus 48 – 49 dan 64 – 88, megurangi rugi-rugi saluran, dan untuk mengurangi biaya operasional seminim mungkin. Seperti terlihat pada gambar 4.2 tampilan hasil perhitungan dengan metode Operator Genetika.



Gambar 4.2  
Hasil Penempatan Kapasitor Dengan Metode Operator Genetika

Dari gambar 4.2 diatas dapat dilihat bahwa kapasitor dipasang pada bus 34 dengan kapasitas 900 kVAR tipe Fixed, bus 49 dengan kapasitas 300 kVAR tipe Fixed, bus 84 dengan kapasitas 300 kVAR tipe Fixed, seperti yang terlihat pada gambar 4.3 *singleline* Diagram Penempatan Kapasitor Metode operator genetika. Dimana besar biaya yang dikeluarkan adalah sebesar 7.488,13 US\$ atau sebesar Rp. 68.141.983,- (dengan asumsi 1 US\$ adalah Rp9100,-).



**Gambar 4.3**  
Single Line Diagram Setelah  
Penempatan Kapsitor

Setelah dilakukan pemasangan kapasitor diperoleh perbaikan profil tegangan, penurunan batas pembebahan saluran serta pengurangan Aliran daya tiap saluran,

**Tabcl 4.7**  
**Profil Tegangan dan Sudut Fasa Tiap Bus Penyalang SYIAR**  
**Setelah Kompensasi**

<b>Bus</b>	<b>absV (pu)</b>	<b>sudV (deg)</b>
1	1,00000	0,00000
2	0,99897	-0,07485
3	0,99642	-0,26005
4	0,99489	-0,37237
5	0,99474	-0,38361
6	0,99357	-0,47186
7	0,99305	-0,48659
8	0,99302	-0,48741
9	0,99299	-0,48810
10	0,9929	-0,49068
11	0,99289	-0,52847
12	0,99223	-0,58379
13	0,99015	-0,76566
14	0,98837	-0,91729
15	0,98824	-0,92876
16	0,98817	-0,93485
17	0,98773	-0,97506
18	0,98726	-0,99506
19	0,98674	-1,04503
20	0,98639	-1,08023
21	0,98605	-1,11433
22	0,98461	-1,25974
23	0,98414	-1,30714
24	0,98396	-1,32509
25	0,98328	-1,39721
26	0,98325	-1,40069
27	0,98303	-1,42401
28	0,9818	-1,55594
29	0,98093	-1,65454
30	0,98078	-1,67068
31	0,97994	-1,76651
32	0,97992	-1,76716
33	0,97983	-1,76986
34	0,97985	-1,77816
35	0,97902	-1,81888
36	0,97859	-1,84093
37	0,97813	-1,86408
38	0,97671	-1,93721

39	0,97628	-1,96011
40	0,97559	-2,00219
41	0,97548	-2,00912
42	0,97543	-2,01224
43	0,97387	-2,11442
44	0,97379	-2,11674
45	0,97376	-2,11770
46	0,97368	-2,12002
47	0,97339	-2,14817
48	0,97257	-2,20569
49	0,97193	-2,25010
50	0,97626	-1,96085
51	0,97621	-1,96207
52	0,97590	-1,97095
53	0,97535	-1,98700
54	0,97493	-1,99907
55	0,97484	-2,00158
56	0,97459	-2,00873
57	0,97440	-2,01419
58	0,97436	-2,01529
59	0,97434	-2,01611
60	0,97431	-2,01672
61	0,97422	-2,01941
62	0,97418	-2,02057
63	0,97414	-2,02193
64	0,97102	-2,29269
65	0,97066	-2,30315
66	0,97064	-2,30354
67	0,97056	-2,30591
68	0,97055	-2,30623
69	0,9703	-2,32920
70	0,97026	-2,33056
71	0,97025	-2,33233
72	0,97006	-2,34222
73	0,96992	-2,34995
74	0,96982	-2,35558
75	0,96975	-2,35754
76	0,96967	-2,35997
77	0,96962	-2,36147
78	0,96954	-2,36368
79	0,96944	-2,36655
80	0,96943	-2,36694
81	0,96953	-2,37922
82	0,96931	-2,40929
83	0,96926	-2,42967
84	0,96934	-2,44752
85	0,96925	-2,45000
86	0,96938	-2,38368
87	0,96929	-2,38619
88	0,96929	-2,38612

**Tabel 4.8**  
**Aliran Arus Tiap Saluran Penyulang SYIAR**  
**Setelah Kompensasi**

Dari	Ke	Arus re (A)	Arus im (A)	Dari	Ke	Arus re (A)	Arus im (A)
1	2	144,443	13,203	2	1	-144,443	-13,203
2	3	144,443	13,203	3	2	-144,443	-13,203
3	4	144,443	13,203	4	3	-144,443	-13,203
4	5	142,075	11,777	5	4	-142,075	-11,777
5	6	142,075	11,777	6	5	-142,075	-11,777
6	7	12,064	7,298	7	6	-12,064	-7,298
7	8	9,005	5,448	8	7	-9,005	-5,448
8	9	2,800	1,694	9	8	-2,800	-1,694
8	10	6,204	3,754	10	8	-6,204	-3,754
6	11	130,011	4,479	11	6	-130,011	-4,479
11	12	130,011	4,479	12	11	-130,011	-4,479
12	13	130,011	4,479	13	12	-130,011	-4,479
13	14	130,011	4,479	14	13	-130,011	-4,479
14	15	125,918	1,961	15	14	-125,918	-1,961
15	16	125,918	1,961	16	15	-125,918	-1,961
16	17	122,083	-0,400	17	16	-122,083	0,400
17	18	122,083	-0,400	18	17	-122,083	0,400
18	19	122,083	-0,400	19	18	-122,083	0,400
19	20	116,093	-4,104	20	19	-116,093	4,104
20	21	116,093	-4,104	21	20	-116,093	4,104
21	22	116,093	-4,104	22	21	-116,093	4,104
22	23	116,093	-4,104	23	22	-116,093	4,104
23	24	116,093	-4,104	24	23	-116,093	4,104
24	25	112,904	-6,097	25	24	-112,904	6,097
25	26	112,904	-6,097	26	25	-112,904	6,097
26	27	112,904	-6,097	27	26	-112,904	6,097
27	28	112,904	-6,097	28	27	-112,904	6,097
28	29	109,501	-8,248	29	28	-109,501	8,248
29	30	109,501	-8,248	30	29	-109,501	8,248
30	31	109,501	-8,248	31	30	-109,501	8,248
31	32	3,962	2,519	32	31	-3,962	-2,519
32	33	3,962	2,519	33	32	-3,962	-2,519
31	34	105,540	-10,768	34	31	-105,540	10,768
34	35	104,115	35,136	35	34	-104,115	-35,136
35	36	96,835	30,498	36	35	-96,835	-30,498
36	37	96,835	30,498	37	36	-96,835	-30,498
37	38	96,835	30,498	38	37	-96,835	-30,498
38	39	91,015	26,772	39	38	-91,015	-26,772
39	40	75,599	16,878	40	39	-75,599	-16,878
40	41	75,599	16,878	41	40	-75,599	-16,878
41	42	70,337	13,500	42	41	-70,337	-13,500
42	43	70,337	13,500	43	42	-70,337	-13,500
43	44	4,791	3,088	44	43	-4,791	-3,088
44	45	4,791	3,088	45	44	-4,791	-3,088

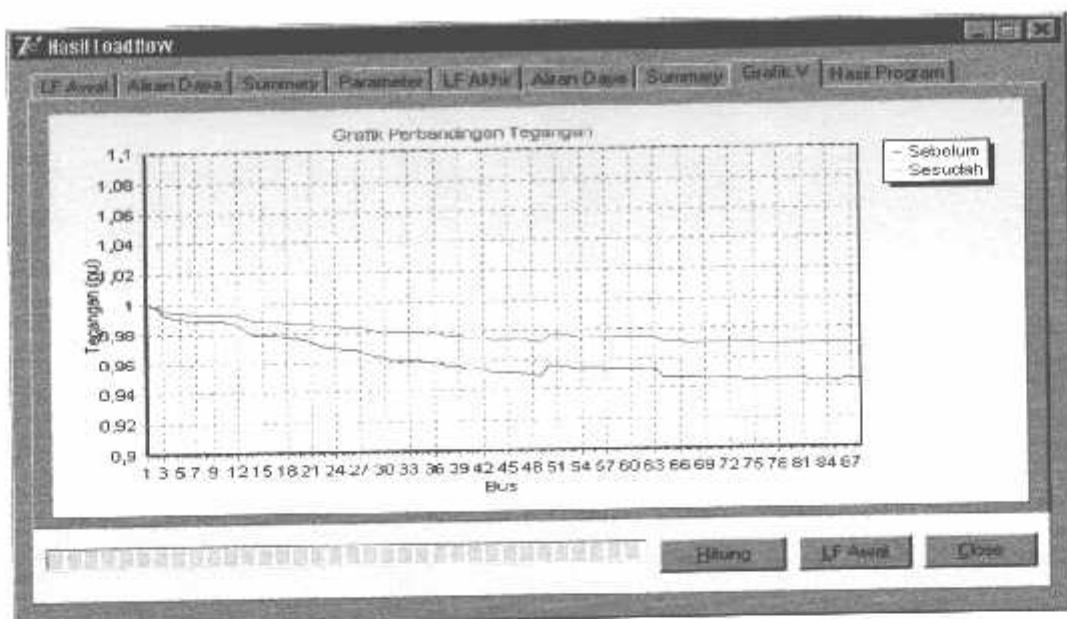
45	46	4,791	3,088	46	45	-4,791	-3,088
43	47	65,545	10,412	47	43	-65,545	-10,412
47	48	65,545	10,412	48	47	-65,545	-10,412
48	49	65,545	10,412	49	48	-65,545	-10,412
39	50	15,415	9,895	50	39	-15,415	-9,895
50	51	15,415	9,895	51	50	-15,415	-9,895
51	52	15,415	9,895	52	51	-15,415	-9,895
52	53	15,415	9,895	53	52	-15,415	-9,895
53	54	15,415	9,895	54	53	-15,415	-9,895
54	55	3,569	2,354	55	54	-3,669	-2,354
54	56	11,746	7,541	56	54	-11,746	-7,541
56	57	8,335	5,352	57	56	-8,335	-5,352
57	58	4,534	2,911	58	57	-4,534	-2,911
58	59	4,534	2,911	59	58	-4,534	-2,911
57	60	3,801	2,441	60	57	-3,801	-2,441
60	61	3,801	2,441	61	60	-3,801	-2,441
61	62	3,801	2,441	62	61	-3,801	-2,441
62	63	3,801	2,441	63	62	-3,801	-2,441
49	64	60,838	23,176	64	49	-60,838	-23,176
64	65	9,938	6,451	65	64	-9,938	-6,451
65	66	4,710	3,057	66	65	-4,710	-3,057
66	67	4,710	3,057	67	66	-4,710	-3,057
67	68	4,710	3,057	68	67	-4,710	-3,057
64	69	50,900	16,725	69	64	-50,900	-16,725
69	70	3,673	2,386	70	69	-3,673	-2,386
69	71	47,227	14,339	71	69	-47,227	-14,339
71	72	47,227	14,339	72	71	-47,227	-14,339
72	73	47,227	14,339	73	72	-47,227	-14,339
73	74	42,645	11,360	74	73	-42,645	-11,360
74	75	8,386	5,456	75	74	-8,386	-5,456
75	76	8,386	5,456	76	75	-8,386	-5,456
76	77	4,972	3,235	77	76	-4,972	-3,235
77	78	4,972	3,235	78	77	-4,972	-3,235
78	79	4,972	3,235	79	78	-4,972	-3,235
79	80	4,972	3,235	80	79	-4,972	-3,235
74	81	28,900	2,419	81	74	-28,900	-2,419
81	82	21,119	-2,648	82	81	-21,119	2,648
82	83	15,285	-6,450	83	82	-15,285	6,450
83	84	9,559	-9,654	84	83	-9,559	9,654
84	85	4,536	2,959	85	84	-4,536	-2,959
81	86	7,781	5,066	86	81	-7,781	-5,066
86	87	4,280	2,787	87	86	-4,280	-2,787
86	88	3,501	2,280	88	86	-3,501	-2,280

Tabel 4.9  
Aliran Daya Tiap Saluran Penyulang SYIAR  
Setelah Kompensasi

Dari	Ke	P (kW)	Q (kVAR)	Dari	Ke	P (kW)	Q (kVAR)
1	2	2888,858	264,066	2	1	-2886,213	-260,023
2	3	2886,213	260,023	3	2	-2879,692	-250,054
3	4	2879,692	250,054	4	3	-2875,753	-244,034
4	5	2828,453	215,964	5	4	-2828,069	-215,376
5	6	2828,069	215,376	6	5	-2825,056	-210,771
6	7	240,904	143,04	7	6	-240,815	-142,903
7	8	179,755	106,673	8	7	-179,751	-106,667
8	9	55,901	33,171	9	8	-55,9	-33,17
8	10	123,850	73,496	10	8	-123,84	-73,48
6	11	2584,151	67,731	11	6	-2582,452	-65,133
11	12	2582,452	65,133	12	11	-2580,794	-62,598
12	13	2580,151	62,598	13	12	-2575,582	-54,292
13	14	2575,582	54,292	14	13	-2571,067	-47,39
14	15	2489,367	-1,09	15	14	-2489,041	1,588
15	16	2489,041	-1,588	16	15	-2488,868	1,853
16	17	2413,328	-47,273	17	16	-2411,234	48,944
17	18	2411,234	-48,944	18	17	-2410,048	49,847
18	19	2410,048	-49,847	19	18	-2408,746	51,836
19	20	2289,206	-122,766	20	19	-2288,315	124,836
20	21	2298,315	-124,128	21	20	-2287,452	125,447
21	22	2287,452	-125,447	22	21	-2283,781	131,06
22	23	2283,781	-131,060	23	22	-2282,586	132,886
23	24	2282,586	-132,886	24	23	-2282,134	133,577
24	25	2218,494	-171,337	25	24	-2216,746	174,010
25	26	2216,746	-174,010	26	25	-2216,662	174,138
26	27	2216,662	-174,138	27	26	-2216,097	175,002
27	28	2216,097	-175,002	28	27	-2216,907	179,878
28	29	214,967	-220,288	29	28	-214,683	223,78
29	30	2142,683	-223,780	30	29	-2142,309	224,351
30	31	2142,309	-224,351	31	30	-2140,093	227,738
31	32	79,127	46,960	32	31	-79,125	-46,958
32	33	79,125	46,958	33	32	-79,120	-46,950
31	34	2060,967	-274,698	34	31	-2060,710	275,091
34	35	2060,710	624,909	35	34	-2059,406	-622,916
35	36	1914,066	536,676	36	35	-1913,430	-535,703
36	37	1913,430	535,703	37	36	-1912,763	-534,683
37	38	1912,768	534,683	38	37	-1910,658	-531,467
38	39	1794,558	462,577	39	38	-1793,957	-461,657
39	40	1486,531	278,866	40	39	-1485,688	-277,579
40	41	1485,688	277,579	41	40	-1485,550	-277,367
41	42	1380,63	215,107	42	41	-1380,574	-215,021
42	43	1380,574	215,021	43	42	-1378,740	-212,218
43	44	95,473	56,661	44	43	-95,468	-56,652
44	45	95,468	56,652	45	44	-95,466	-56,649
45	46	95,466	56,649	46	45	-95,460	-56,640

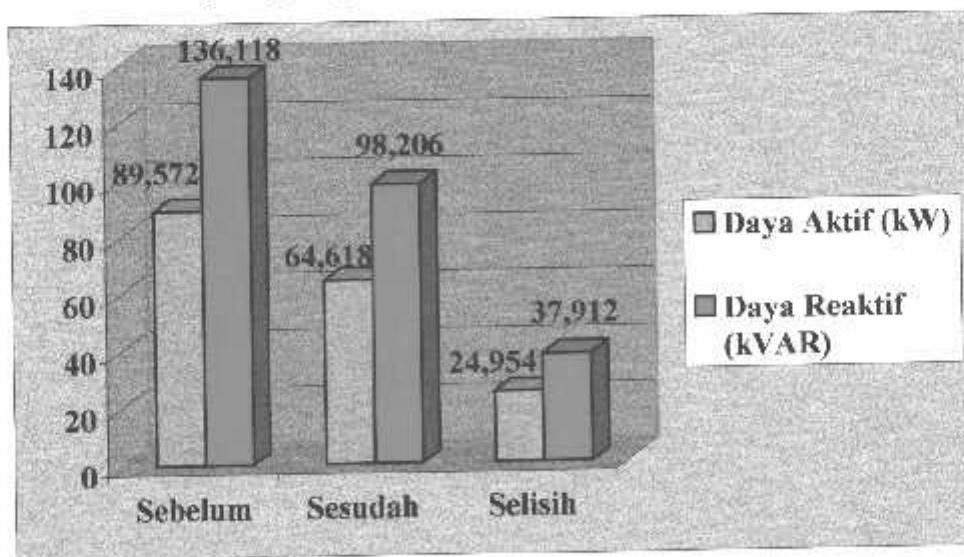
43	47	1283,266	155,557	47	43	-1282,722	-154,724
47	48	1282,722	154,724	48	47	-1281,795	-153,307
48	49	1281,795	153,307	49	48	-1281,080	-152,215
39	50	307,426	182,791	50	39	-307,420	-182,782
50	51	307,420	182,782	51	50	-307,411	-182,768
51	52	307,411	182,768	52	51	-307,342	-182,662
52	53	307,342	182,662	53	52	-307,217	-182,472
53	54	307,217	182,472	54	53	-307,124	-182,329
54	55	73,105	43,377	55	54	-73,100	-43,370
54	56	234,019	138,951	56	54	-233,962	-138,864
56	57	166,022	98,554	57	56	-165,999	-98,519
57	58	90,304	53,587	58	57	-90,302	-53,583
58	59	90,302	53,583	59	58	-90,300	-53,580
57	60	75,695	44,933	60	57	-75,690	-44,925
60	61	75,690	44,925	61	60	-75,685	-44,917
61	62	75,685	44,917	62	61	-75,683	-44,914
62	63	75,683	44,914	63	62	-75,680	-44,910
49	64	1199,38	403,735	64	49	-1198,548	-402,463
64	65	197,859	117,461	65	64	-197,807	-117,381
65	66	93,747	55,631	66	65	-93,746	-55,630
66	67	93,746	55,630	67	66	-93,741	-55,621
67	68	93,741	55,621	68	67	-93,740	-55,620
64	69	1000,688	285,002	69	64	-1000,135	-284,155
69	70	73,103	43,374	70	69	-73,100	-43,370
69	71	927,032	240,782	71	69	-926,990	-240,717
71	72	926,990	240,717	72	71	-926,855	-240,511
72	73	926,855	240,511	73	72	-926,750	-240,350
73	74	835,590	186,26	74	73	-835,524	-186,159
74	75	166,876	99,045	75	74	-166,868	-99,032
75	76	166,868	99,032	76	75	-166,857	-99,017
76	77	98,917	58,707	77	76	-98,914	-58,701
77	78	98,914	58,701	78	77	-98,908	-58,692
78	79	98,908	58,692	79	78	-98,901	-58,681
79	80	98,901	58,681	80	79	-98,900	-58,680
74	81	562,008	23,834	81	74	-561,852	-23,595
81	82	407,025	-68,296	82	81	-406,895	68,494
82	83	290,795	-137,384	83	82	-290,732	137,480
83	84	177,212	-194,840	84	83	-177,166	194,911
84	85	90,306	53,549	85	84	-90,300	-53,540
81	86	154,827	91,891	86	81	-154,810	-91,865
86	87	85,145	50,528	87	86	-85,140	-50,520
86	88	69,664	41,337	88	86	-69,660	-41,330

**Grafik 4.1**  
**Perbandingan Tiap-tiap Bus Antara**  
**Tegangan Sebelum dan Setelah Kompensasi**



Dari grafik 4.1 terlihat bahwa tegangan terendah terjadi pada bus 85 sebesar 0,94424 pu atau 18,8848 kV dan setelah kompensasi naik menjadi 0,96925 pu atau 19,3850 kV. Batas tegangan yang diijinkan oleh PLN adalah sebesar 0,95 pu sampai 1,05 pu. Sedangkan rugi-rugi daya sebelum dan setelah kompensasi dapat ditunjukkan seperti pada grafik 4.2.

**Grafik 4.2**  
**Rugi-Rugi Daya Sebelum dan Setelah Kompensasi**



Dari grafik 4.2 bahwa penurunan rugi daya aktif adalah 24,954 kW dari 89,572 kW menjadi 64,618 kW sehingga terjadi penurunan sebesar 27,86 % sedangkan untuk daya reaktif juga terjadi penurunan sebesar 37,912 kVAR dari 136,118 kVAR menjadi 98,206 kVAR sehingga terjadi penurunan sebesar 27,85 %.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Setelah dilakukan analisa mengenai penentuan letak dan kapasitas kapasitor pada jaringan distribusi radial dengan menggunakan metode operator genetika, maka dapat diambil kesimpulan :

1. Dari hasil perhitungan aliran daya pada penyulang SYIAR sebelum dipasang kapasitor rugi – rugi dayanya sebesar  $89.572 + j136.118$  setelah dipasang kapasitor rugi – rugi dayanya menjadi sebesar  $64.618 + j98.206$  sehingga mengalami penurunan sebesar  $24.954 + j37.912$ .
2. Jika dilihat dari node 85 sebelum dipasang kapasitor nilai tegangannya sebesar 0.94424 pu dibawah batas yang diijinkan, tetapi setelah dipasang kapasitor nilai tegangan menjadi 0.96925 pu. Sehingga setelah dipasang kapasitor nilai tegangan dapat diperbaiki sebesar 2,50 %.
3. Berdasarkan hasil perhitungan dengan metode operator genetika letak kapasitor yang optimal terletak pada node 34, node 49 dan 84. Dengan kompensasi pada Node 34 dengan kapasitas 900 kVAR, bus 49 dengan kapasitas 300 kVAR, bus 84 dengan kapasitas 300 kVAR

## **1.2. Saran**

Penentuan letak dan kapasitas kapasitor pada jaringan primer tipe radial dengan menggunakan metode operator genetika perlu dikembangkan lagi terhadap sistem jaringan yang lebih luas dan dapat digunakan lebih efektif dan efisien dalam pemecahan permasalahan pemasangan kapasitor.

## DAFTAR PUSTAKA

- [1] AS Pabla “ *Sistem Distribusi Daya Listrik*” Punjab State Electricity Board Chandigarh, Penerbit Erlangga Jakarta 10420.
- [2] Djiteng Marsudi,Ir,”*Operasi Sistem Tenaga Listrik*”,Balai Penerbit dan Humas ISTN,1990
- [3] F. Salgado, E. Lopez, H. Rudnick “ *VAR Planning in Distribution Systems via Genetic Operators*” Departement of Electrical Engineering, Universidad de Concepcion, Chile.
- [4] Hasan Basri, Ir,”*Sistem Distribusi Tenaga Listrik*”,Balai Penerbit dan Humas ISTN bumi srengseng Indah PS.Minggu,Jakarta Selatan,1996
- [5] Sri Kusumadewi,”*Artificial Intelligence*”,Balai Penerbit Graha Ilmu,2003
- [6] William D Stevenson,Jr,”*Analisa Sistem Tenaga*”,Penerbit Erlangga edisi keempat,1996



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNIK INDUSTRI  
JURUSAN TEKNIK ELEKTRO

**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : Ario Kunto Adji  
NIM : 0012045  
Jurusan : Teknik Elektro S1  
Konsentrasi : Teknik Energi Listrik  
Judul Skripsi : Analisis Penempatan Kapasitor Bank Menggunakan Operator Genetika Dengan *Allelic Alphabet* Dan Konsep Jarak *Hamming* Pada Sistem Distribusi 20KV Di GI PIER Pasuruan Pada Penyulang Syiar

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

Hari : Jum'at  
Tanggal : 23 Maret 2007  
Dengan Nilai : B+ (78,20) *Bey*

**PANITIA UJIAN SKRIPSI**



Ir. Mechtar Asroni, MSME  
NIP.Y. 101 810 0036

**SEKRETARIS**

Ir. F. Yudi Limpraptono, M.T.  
NIP.Y. 103 950 0274

**ANGGOTA PENGUJI**

**PENGUJI I**

  
Ir. Djojo Priatmono, MT  
NIP. Y 101 850 0107

**PENGUJI II**

  
Bambang Prio Hartono, ST, MT  
NIP. Y 102 840 0082



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNIK INDUSTRI  
JURUSAN TEKNIK ELEKTRO

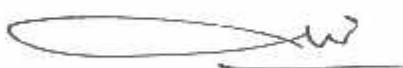
LEMBAR PERBAIKAN SKRIPSI

Nama Mahasiswa : Ario Kunto Adji  
NIM : 0012045  
Jurusan : Teknik Elektro SI  
Konsentrasi : Teknik Energi Listrik  
Judul Skripsi : Analisis Penempatan Kapasitor Bank Menggunakan Operator Genetika Dengan *Allelic Alphabet* Dan Konsep Jarak *Hamming* Pada Sistem Distribusi 20KV Di GI PIER Pasuruan Pada Penyulang Syiar

Hari / Tanggal Ujian Skripsi : Jum'at / 23 Maret 2007

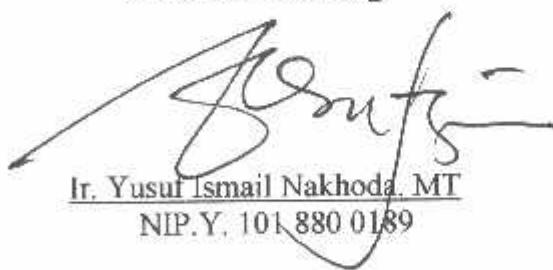
No	Materi Perbaikan	Paraf
1	Revisi lihat di buku skripsi	

Diperiksa/ Disetujui  
Penguji II



Bambang Prio Hartono, ST, MT  
NIP.Y. 102 840 0082

Mengetahui  
Dosen Pembimbing



Ir. Yusuf Ismail Nakhoda, MT  
NIP.Y. 101 880 0189



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### Formulir Perbaikan Ujian Skripsi

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

NAMA : Ario Tumang Adji  
NIM : 00 62 045  
Perbaikan meliputi :

Penulisan tittel di buku skripsi

Malang,

( \_\_\_\_\_ )

**Lampiran**

**Tampilan Program**



**Tampilan Data**

General | Data Bus | Data Saluran | Data Generator |

Jumlah Bus	88	
Jumlah Saluran	87	
Tegangan Dasar	20	kV
Daya Dasar	30	MVA
Parameter Saluran	ohm	

Next | Close

Gambar  
Tampilan Inputan Data ( General )

**Tampilan Data**

General | Data Bus | Data Saluran | Data Generator |

Bus	absV [pu]	sudV (deg)	Pg [kW]	Qg (kVAR)	PL [kW]	QL [kVAR]	Cap [ohm]	Type Bus	▲
1	1	0	0	0	0	0	0	1	—
2	1	0	0	0	0	0	0	3	
3	1	0	0	0	0	0	0	3	
4	1	0	0	0	47,3	28,07	0	3	
5	1	0	0	0	0	0	0	3	
6	1	0	0	0	0	0	0	3	
7	1	0	0	0	61,06	36,23	0	3	
8	1	0	0	0	0	0	0	3	
9	1	0	0	0	55,9	33,17	0	3	
10	1	0	0	0	123,84	73,48	0	3	
11	1	0	0	0	0	0	0	3	
12	1	0	0	0	0	0	0	3	
13	1	0	0	0	0	0	0	3	
14	1	0	0	0	81,7	48,48	0	3	
...	...	...	...	...	...	...	...	...	▼

Next | Close

Gambar  
Tampilan Inputan Data ( Data Pembebatan )

7' Tampilan Data

General		Data Bus		Data Saluran		Data Generator			
No	Dan	Ka	R (ohm)	X (ohm)	Lc (ohm)	Tr	Tu	Su (deg)	Kap (kVA)
1	1	2	0.12572	0.19219	0	0	0	0	10000
2	2	3	0.30997	0.47384	0	0	0	0	10000
3	3	4	0.18721	0.28618	0	0	0	0	10000
4	4	5	0.01892	0.02892	0	0	0	0	10000
5	5	6	0.14923	0.22659	0	0	0	0	10000
6	6	7	0.45054	0.68873	0	0	0	0	10000
7	7	8	0.03366	0.05146	0	0	0	0	10000
8	8	9	0.09005	0.13765	0	0	0	0	10000
9	8	10	0.19408	0.29669	0	0	0	0	10000
10	6	11	0.10042	0.15352	0	0	0	0	10000
11	11	12	0.09798	0.14978	0	0	0	0	10000
12	12	13	0.308	0.49003	0	0	0	0	10000
13	13	14	0.26679	0.40784	0	0	0	0	10000
14	14	15	0.02058	0.03143	0	n	n	n	10000

Next | Close

Gambar  
Tampilan Inputan Data ( Data Saluran )

3. Tekan Tombol **Next** kemudian tekan tombol **LF Awal** untuk melihat hasil perhitungan aliran daya sebelum kompensasi.

7' Hasil Loadflow

LF Awal		Aliran Daya		Summary		Parameter		LF Akhir		Aliran Daya		Summary		Grafik V		Hasil Program	
Bus	absV (pu)	sudV (deg)	Pg (kW)	Dg (kVAR)	Pl (kW)	Ql (kVAR)	Spu (pu)	Type Bus									
74	0.94597	-1.52300	0.000	0.000	106.640	63.280	0.000	3									
75	0.94590	-1.52506	0.000	0.000	0.000	0.000	0.000	3									
76	0.94582	-1.52761	0.000	0.000	67.340	40.310	0.000	3									
77	0.94576	-1.52913	0.000	0.000	0.000	0.000	0.000	3									
78	0.94568	-1.53152	0.000	0.000	0.000	0.000	0.000	3									
79	0.94558	-1.53453	0.000	0.000	0.000	0.000	0.000	3									
80	0.94557	-1.53494	0.000	0.000	98.900	58.680	0.000	3									
81	0.94545	-1.53883	0.000	0.000	0.000	0.000	0.000	3									
82	0.94488	-1.55682	0.000	0.000	116.100	68.890	0.000	3									
83	0.94455	-1.56721	0.000	0.000	113.520	57.360	0.000	3									
84	0.94432	-1.57390	0.000	0.000	86.860	51.540	0.000	3									
85	0.94424	-1.57651	0.000	0.000	90.300	53.540	0.000	3									
86	0.94529	-1.54362	0.000	0.000	0.000	0.000	0.000	3									
87	0.94521	-1.54626	0.000	0.000	85.140	50.520	0.000	3									
88	0.94521	-1.54618	0.000	0.000	69.660	41.330	0.000	3									

Bit ini | LF Awal | Close

Gambar  
Tampilan Hasil Aliran Daya Untuk Mengetahui  
Harga Tegangan Dan Sudut Fasa Tiap-tiap Bus Sebelum Kompensasi

Hasil Loadflow

LF Awal | Aliran Daya | Summary | Parameter | LF Akhir | Aliran Daya | Summary | Grafik V | Hasil Program |

No	Dari	Kepada	P (kW)	Q (kVAr)	Arus rel (A)	Arus im (A)	Dari	Kepada	P (kW)	Q
1	1	2	2313.812	1801.978	145.691	90.099	2	1	2310.123	-1
2	2	3	2310.123	1796.239	145.691	90.099	3	2	2301.027	-1
3	3	4	2301.027	1782.434	145.691	90.099	4	3	2295.534	-1
4	4	5	2348.234	1745.967	143.310	88.673	5	4	2347.697	-1
5	5	6	2347.697	1745.146	143.310	88.673	6	5	2343.487	-1
6	6	7	240.905	143.041	12.143	7.297	7	6	240.815	-1
7	7	8	179.755	106.673	9.064	5.447	8	7	179.751	-1
8	8	9	55.901	33.171	2.819	1.694	9	8	-55.900	-1
9	8	10	123.850	73.496	6.245	3.753	10	8	-123.840	-1
10	6	11	2502.581	1695.669	131.168	81.376	11	6	-2500.189	-1
11	11	12	2500.189	1592.011	131.168	81.376	12	11	-2597.854	-1
12	12	13	2597.854	1588.442	131.168	81.376	13	12	-2590.515	-1
13	13	14	2590.515	1576.747	131.168	81.376	14	13	-2584.159	-1
14	14	15	2502.459	1518.550	127.020	78.858	15	14	-2501.999	-1
15										
16										

Hitung | LF Awal | Close

Gambar

Tampilan Hasil Aliran Daya Untuk Mengetahui Daya  
Tiap Saluran dan Arus yang Mengalir pada Saluran Sebelum Kompensasi

Hasil Loadflow

LF Awal | Aliran Daya | Summary | Parameter | LF Akhir | Aliran Daya | Summary | Grafik V | Hasil Program |

**Summary Leadflow**

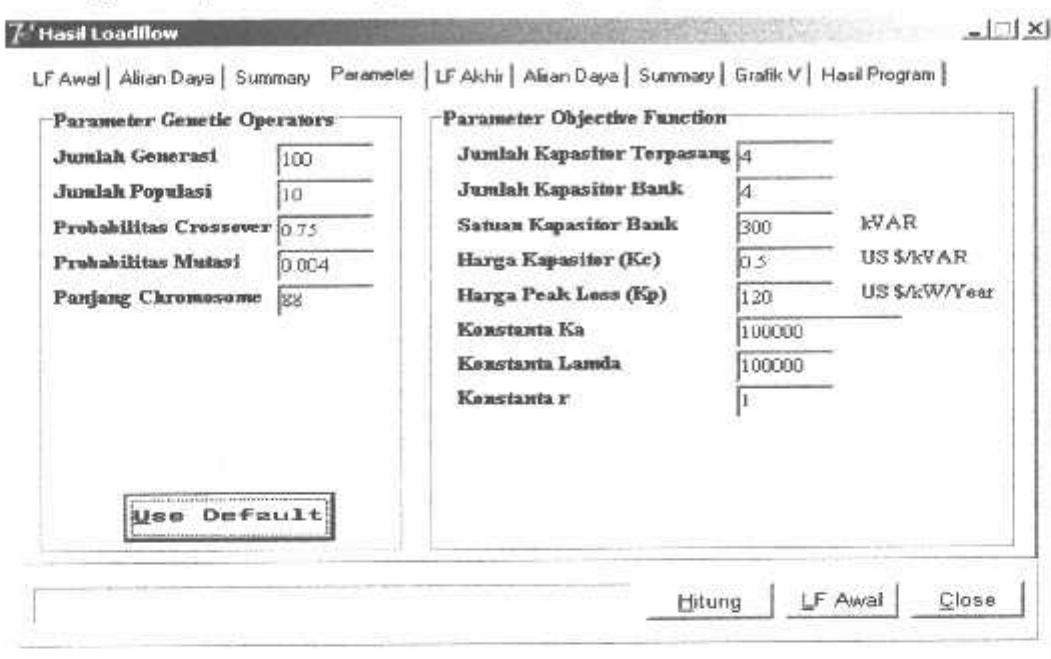
Jumlah Pembangkitan	2913.812+j1801.978	kVA
Jumlah Penggunaan	2324.240+j1663.860	kVA
Jumlah Rugi-Rugi	89.572+j136.118	kVA
Rerata	3	
Waktu Hitung	0:0:125	

Hitung | LF Awal | Close

Gambar

Tampilan Total Pembangkitan, Penggunaan, dan  
Rugi-rugi Sebelum Kompensasi

4. Kemudian pilih tombol Parameter untuk melihat parameter operator genetika, tekan **Hitung** untuk mulai proses perhitungan.



Gambar  
Tampilan Parameter Operator Genetika dan Parameter *Objective Function*

5. Kemudian pilih tombol **Hitung** untuk memperoleh hasil perhitungan Aliran Daya dengan Metode operator genetika setelah kompensasi.

The screenshot shows the 'Hasil Loadflow' window displaying a table of power flow results for 22 buses. The columns include Bus number, absV (pu), sudV (deg), Pg (kW), Qg (kVAR), PL (kW), QL (kVAR), Sups (pu), and Type Bus. The results show compensated voltages and powers for each bus.

Bus	absV (pu)	sudV (deg)	Pg (kW)	Qg (kVAR)	PL (kW)	QL (kVAR)	Sups (pu)	Type Bus
74	0.96982	-2.36558	0.000	0.000	106.640	63.290	0.000	3
75	0.96975	-2.35754	0.000	0.000	0.000	0.000	0.000	3
76	0.96967	-2.35997	0.000	0.000	67.940	40.310	0.000	3
77	0.96962	-2.36147	0.000	0.000	0.000	0.000	0.000	3
78	0.96954	-2.36368	0.000	0.000	0.000	0.000	0.000	3
79	0.96944	-2.36655	0.000	0.000	0.000	0.000	0.000	3
80	0.96943	-2.36694	0.000	0.000	98.900	58.680	0.000	3
81	0.96953	-2.37922	0.000	0.000	0.000	0.000	0.000	3
82	0.96931	-2.40929	0.000	0.000	116.100	68.890	0.000	3
83	0.96926	-2.42967	0.000	0.000	113.520	57.360	0.000	3
84	0.96934	-2.44752	0.000	300.000	86.860	51.540	0.000	3
85	0.96925	-2.45000	0.000	0.000	90.300	53.540	0.000	3
86	0.96938	-2.38368	0.000	0.000	0.000	0.000	0.000	3
87	0.96929	-2.38619	0.000	0.000	85.140	50.520	0.000	3
88	0.96929	-2.38612	0.000	0.000	69.660	41.330	0.000	3

At the bottom, there are 'Hitung', 'LF Awal', and 'Close' buttons.

Gambar  
Tampilan Hasil Aliran Daya Untuk Mengetahui Harga Tegangan  
Dan Sudut Fasa Tiap-tiap Bus Setelah Kompensasi

Hasil Loadflow

LF Awal | Aliran Daya | Summary | Parameter | LF Akhir | Aliran Daya | Summary | Grafik V | Hasil Program |

No	Dari	Ke	P (kW)	Q (kVAR)	Arus re (A)	Arus im (A)	Dari	Ke	P (kW)	Q
1	1	2	2886.050	264.066	144.443	13.203	2	1	2886.213	-2
2	2	3	2886.213	260.023	144.443	13.203	3	2	2879.692	-2
3	3	4	2979.692	250.054	144.443	13.203	4	3	2875.753	-2
4	4	5	2828.453	215.964	142.075	11.777	5	4	2828.069	-2
5	5	6	2828.069	215.376	142.075	11.777	6	5	2825.056	-2
6	6	7	240.904	143.040	12.064	7.298	7	6	240.815	-1
7	7	8	179.755	106.673	9.005	5.448	8	7	179.751	-1
8	8	9	55.901	33.171	2.800	1.694	9	8	55.900	-1
9	8	10	123.850	73.496	6.204	3.754	10	8	123.840	-1
10	6	11	2584.151	67.731	130.011	4.479	11	6	2582.452	-4
11	11	12	2582.452	65.133	130.011	4.479	12	11	2580.794	-4
12	12	13	2580.794	62.598	130.011	4.479	13	12	2575.582	-4
13	13	14	2575.582	54.292	130.011	4.479	14	13	2571.067	-4
14	14	15	2489.367	-1.090	125.918	1.961	15	14	-2489.041	-

Hitung | LF Awal | Close

Gambar  
Tampilan Hasil Aliran Daya Untuk Mengetahui Daya Tiap Saluran  
Dan Arus yang Mengalir pada Saluran Setelah Kompensasi

Hasil Loadflow

LF Awal | Aliran Daya | Summary | Parameter | LF Akhir | Aliran Daya | Summary | Grafik V | Hasil Program |

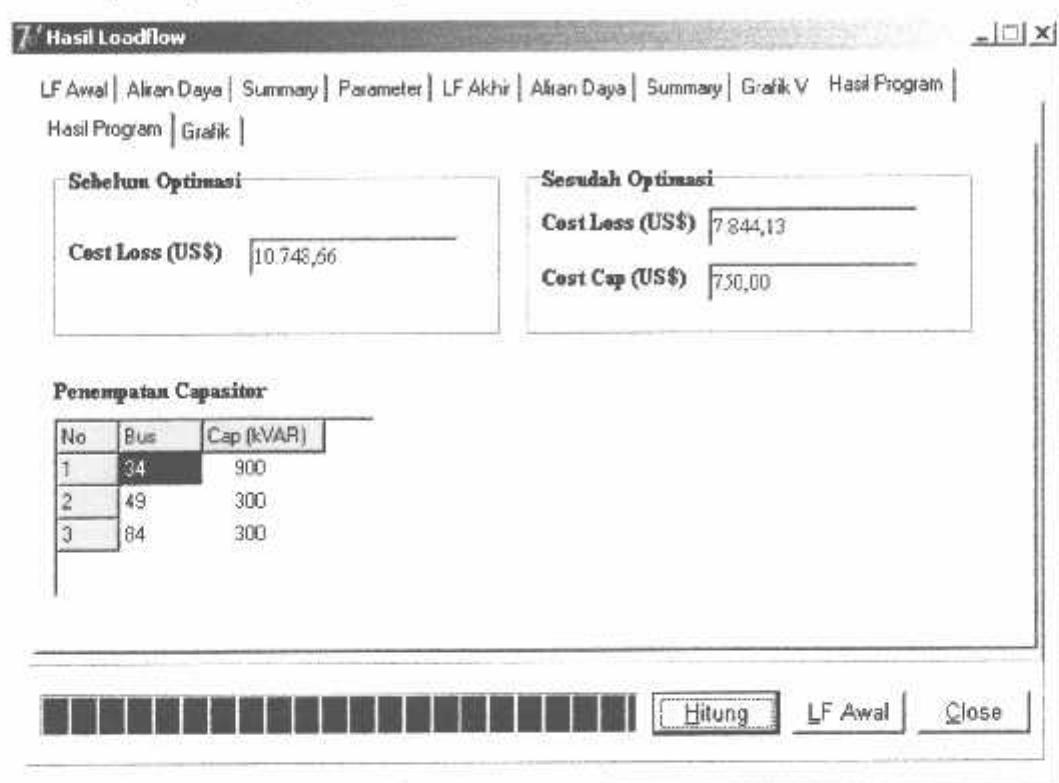
**Summary Loadflow**

Jumlah Pembangkit	2888.858+j1764.066	kVA
Jumlah Pembebanan	2824.240+j1665.860	kVA
Jumlah Rugi-Rugi	64.618+j 98.206	kVA
Iterasi	3	
Waktu Hitung	00:02:01:0	

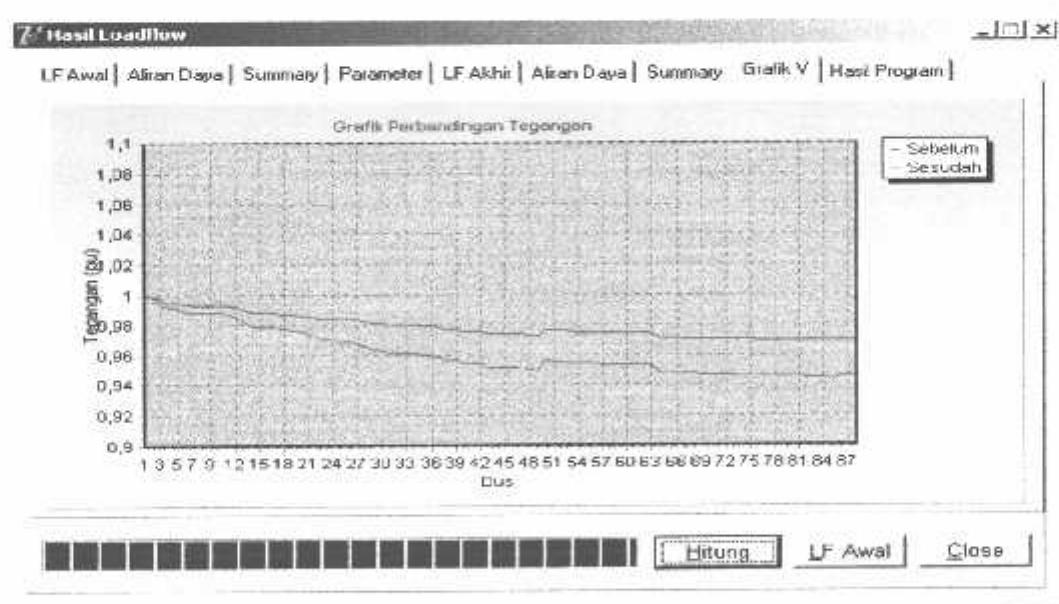
Hitung | LF Awal | Close

Gambar  
Tampilan Total Pembangkit, Pembebanan,  
Dan Rugi-rugi Setelah Kompensasi

6. Kemudian pilih tombol **Hasil Program** untuk mengetahui hasil penempatan kapasitor pada saluran.



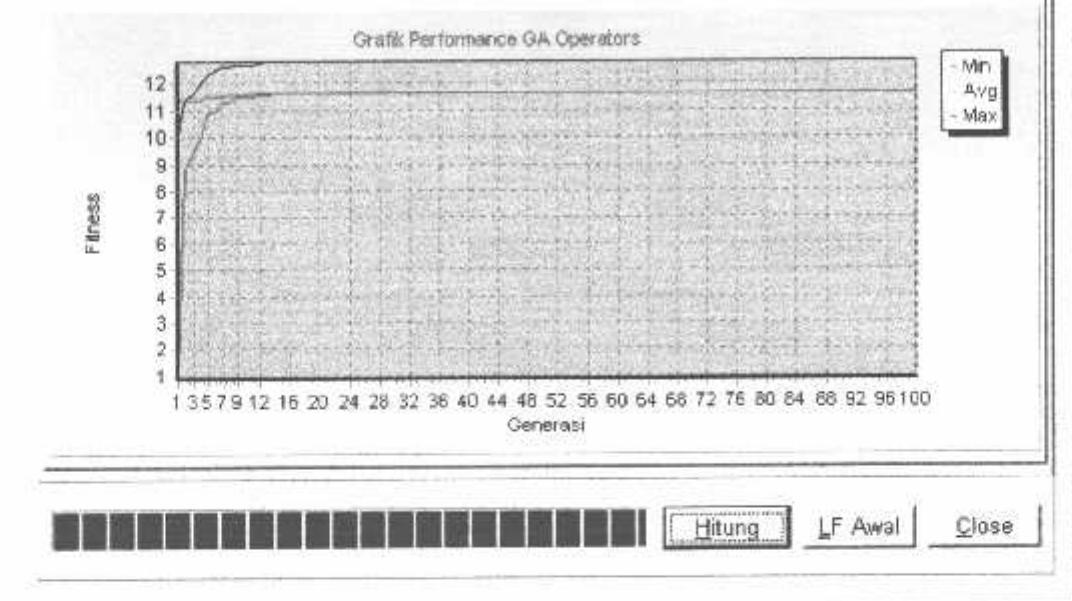
Gambar  
Hasil Penempatan Dengan Metode Operator Genetika



Grafik  
Perbandingan Tegangan Sebelum dan setelah Kompensasi

## Hasil Loadflow

LF Awal | Aliran Daya | Summary | Parameter | LF Akhir | Aliran Daya | Summary | Grafik | Hasil Program |  
Hasil Program | Grafik |



Grafik  
Performance Operator Genetika

```

unit uForwardBack;

interface

uses uUtils,uComplex,uLoadflow,uMatrix,SysUtils,Math;

procedure LoadflowForwardBack(var rBus:TBusArr1;
  var rBranch:TBranchArr1;
  var rParamLF:TParamLF);overload;

procedure LoadflowForwardBack(const rTypeLoad:integer;
  var rBus:TBusArr1;
  var rBranch:TBranchArr1;
  var rParamLF:TParamLF);overload;

implementation

procedure CreateVBusSLoad(var rBus:TBusArr1;
  var rParamLF:TParamLF;
  var rV,rSL:CxArr1);
var i,Nbus:integer;
begin
  Nbus:=high(rBus)+1;
  SetLength(rV,Nbus-1);
  SetLength(rSL,Nbus-1);
  for i:=1 to Nbus-1 do
  begin
    rV[i-1].real:=rBus[i].absV;
    rV[i-1].imag:=rBus[i].sudV;
    rSL[i-1].real:=rBus[i].PL/rParamLF.Pbase;
    rSL[i-1].imag:=rBus[i].QL/rParamLF.Pbase;
  end;
end;

function CreateZa(var rNbus:integer;
  var rBranch:TBranchArr1;
  var rParamLF:TParamLF):CxArr2;
var i,dari,ke:integer;
begin
  SetLength(result,rNbus,rNbus);
  rParamLF.Zbase:=sqr(rParamLF.Vbase*rParamLF.Vkonst)/
    (rParamLF.Pbase*rParamLF.Pkonst);
  rParamLF.Ibase:=(rParamLF.Vbase*rParamLF.Vkonst)/rParamLF.Zbase;
  for i:=0 to high(rBranch) do
  begin
    dari:=rBranch[i].dari-1;
    ke:=rBranch[i].ke-1;
    if rParamLF.ParamBranch=pbPu then
    begin
      result[dari,ke]:=Cmplx(rBranch[i].R,rBranch[i].X);
    end
    else if rParamLF.ParamBranch=pbOhm then
    begin
      result[dari,ke]:=Cmplx(rBranch[i].R/rParamLF.Zbase,
        rBranch[i].X/rParamLF.Zbase);
    end;
  end;
end;

```

---

```

    end;
end;
end;

function CreateMatrixZ(var rA: iArr2;
    var rZ: CxArr2): CxArr2;
var i,j,Nbus:integer;
Za,mA:CxArr2;
begin
Nbus:=high(rZ)+1;
SetLength(Za,Nbus-1,Nbus-1);
for i:=0 to Nbus-1 do
begin
    for j:=0 to Nbus-1 do
        begin
            if rZ[i,j].imag<>0 then
                begin
                    Za[j-1,j-1]:=Cmplx(rZ[i,j]);
                end;
            end;
        end;
SetLength(mA,Nbus-1,Nbus-1);
for i:=0 to Nbus-2 do
begin
    for j:=0 to Nbus-2 do
        begin
            mA[i,j]:=Cmplx(rA[i,j],0);
        end;
    end;
result:=MatrixMul(MatrixMul(MatrixTranspose(mA),Za),mA);
end;

function SaveV0Bus(var rV: CxArr1): CxArr1;
var i,Nbus:integer;
begin
Nbus:=high(rV)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
    result[i]:=Cmplx(rV[i]);
end;
end;

function CalcK(var rV,rSL: CxArr1): CxArr1; overload;
var i,Nbus:integer;
begin
Nbus:=high(rV)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
    result[i]:=Conjugate(Divide(rSL[i],rV[i]));
end;
end;

function CalcK(const rTypeLoad:integer;

```

---

```

    var rV,rSL:CxArr1):CxArr1;overload;
var i,Nbus:integer;
    SL:TCmplx;
    absV:double;
begin
Nbus:=high(rV)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
absV:=sqrt(sqr(rV[i].real)+sqr(rV[i].imag));
if rTypeLoad=0 then
begin
    SL.real:=rSL[i].real;
    SL.imag:=rSL[i].imag;
end
else if rTypeLoad=1 then
begin
    SL.real:=rSL[i].real*absV;
    SL.imag:=rSL[i].imag*absV;
end
else if rTypeLoad=2 then
begin
    SL.real:=rSL[i].real*sqr(absV);
    SL.imag:=rSL[i].imag*sqr(absV);
end
else
begin
    SL.real:=rSL[i].real*Power(absV,1.38);
    SL.imag:=rSL[i].imag*Power(absV,3.22);
end;
result[i]:=Conjugate(Divide(SL,rV[i]));
end;
end;

function CalcdU(var rZa:CxArr2;
    var rK:CxArr1):CxArr1;
begin
result:=MatrixMul(rZa,rK);
end;

function UpdateVBus(var rVs:TCmplx;
    var rdU:CxArr1):CxArr1;
var i,Nbus:integer;
begin
Nbus:=high(rdU)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
result[i]:=Subtract(rVs,rdU[i]);
end;
end;

function isConvergen(var rV0,rV:CxArr1;
    var rParamLF:TParamLF):boolean;
var i,Nbus:integer;

```

---

```

dV:TCmplx;
begin
result:=false;
Nbus:=high(rV)+1;
for i:=0 to Nbus-1 do
begin
dV:=Subtract(rV[i],rV0[i]);
if getAbs(dV)<=rParamLF.Toleransi then
begin
result:=true;
break;
end;
end;
end;

function CalcArus(var rA:TxArr2;
var rK:CxArr1):CxArr1;
var i,j,Nbus:integer;
mA:CxArr2;
begin
Nbus:=high(rK)+1;
SetLength(mA,Nbus,Nbus);
for i:=0 to Nbus-1 do
begin
for j:=0 to Nbus-1 do
begin
mA[i,j]:=Cmplx(rA[i,j],0);
end;
end;
result:=MatrixMul(mA,rK);
end;

function CalcLossBranch(var rZ:CxArr2;
var rI:CxArr1):CxArr1;
var i,j,Nbus:integer;
tmp:double;
begin
Nbus:=high(rZ)+1;
SetLength(result,Nbus-1);
for i:=0 to Nbus-1 do
begin
for j:=0 to Nbus-1 do
begin
if rZ[i,j].imag<>0 then
begin
tmp:=sqr(getAbs(rI[j-1]));
result[j-1]:=Multiply(rZ[i,j],tmp);
end;
end;
end;
end;
end;

function CalcDaya(var rV,rI:CxArr1):CxArr1;
var i,Nbus:integer;
begin

```

---

```

Nbus:=high(rV)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
  result[i]:=Multiply(rV[i],Conjugate(rI[i]));
end;
end;

procedure UpdateBus(var rV,rSa:CxArr1;
  var rParamLF:TParamLF;
  var rBus:TBusArr1);
var i,Nbus:integer;
begin
  Nbus:=high(rBus)+1;
  rParamLF.SumLoad:=Cmplx(0.0,0.0);
  for i:=0 to NBus-1 do
  begin
    if i>0 then
    begin
      rBus[i].absV:=sqrt(sqr(rV[i-1].real)+sqr(rV[i-1].imag));
      rBus[i].sudV:=arctan(rV[i-1].imag/rV[i-1].real);
    end
    else if i=0 then
    begin
      rBus[i].Pgen:=rSa[0].real*rParamLF.Pbase;
      rBus[i].Qgen:=rSa[0].imag*rParamLF.Pbase;
    end;
    rParamLF.SumLoad.real:=rParamLF.SumLoad.real+rBus[i].PL;
    rParamLF.SumLoad.imag:=rParamLF.SumLoad.imag+rBus[i].QL;
  end;
  rParamLF.SumGen:=Multiply(Cmplx(rSa[0]),rParamLF.Pbase);
  rParamLF.SumLoss:=Subtract(rParamLF.SumGen,rParamLF.SumLoad);
end;

procedure UpdateBranch(var rdSa,rSa,rdI:CxArr1;
  var rParamLF:TParamLF;
  var rBranch:TBranchArr1);
var i,Nsal,ke:integer;
  pos:double;
begin
  Nsal:=high(rBranch)+1;
  for i:=0 to Nsal-1 do
  begin
    ke:=rBranch[i].ke-1;
    rBranch[i].Sij:=Multiply(Cmplx(rSa[ke-1]),rParamLF.Pbase);
    rBranch[i].Sji:=Multiply(Negative(Subtract(rSa[ke-1],rdSa[ke-1])),rParamLF.Pbase);
    rBranch[i].Aij:=Multiply(Cmplx(rdI[ke-1]),rParamLF.Ibase);
    pos:=getAbs(rdSa[ke-1])/getAbs(rSa[ke-1]);
    rBranch[i].Aji:=Multiply(Negative(Multiply(rdI[ke-1],(1-pos))),rParamLF.Ibase);
  end;
end;

procedure LoadflowForwardBack(var rBus:TBusArr1;

```

---

```

Nbus:=high(rV)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
  result[i]:=Multiply(rV[i],Conjugate(rI[i]));
end;
end;

procedure UpdateBus(var rV,rSa:CxArr1;
  var rParamLF:TParamLF;
  var rBus:TBusArr1);
var i,Nbus:integer;
begin
  Nbus:=high(rBus)+1;
  rParamLF.SumLoad:=Cmplx(0.0,0.0);
  for i:=0 to NBus-1 do
  begin
    if i>0 then
    begin
      rBus[i].absV:=sqrt(sqr(rV[i-1].real)+sqr(rV[i-1].imag));
      rBus[i].sudV:=arctan(rV[i-1].imag/rV[i-1].real);
    end
    else if i=0 then
    begin
      rBus[i].Pgen:=rSa[0].real*rParamLF.Pbase;
      rBus[i].Qgen:=rSa[0].imag*rParamLF.Pbase;
    end;
    rParamLF.SumLoad.real:=rParamLF.SumLoad.real+rBus[i].PL;
    rParamLF.SumLoad.imag:=rParamLF.SumLoad.imag+rBus[i].QL;
  end;
  rParamLF.SumGen:=Multiply(Cmplx(rSa[0]),rParamLF.Pbase);
  rParamLF.SumLoss:=Subtract(rParamLF.SumGen,rParamLF.SumLoad);
end;

procedure UpdateBranch(var rdSa,rSa,rdI:CxArr1;
  var rParamLF:TParamLF;
  var rBranch:TBranchArr1);
var i,Nsal,ke:integer;
  pos:double;
begin
  Nsal:=high(rBranch)+1;
  for i:=0 to Nsal-1 do
  begin
    ke:=rBranch[i].ke-1;
    rBranch[i].Sij:=Multiply(Cmplx(rSa[ke-1]),rParamLF.Pbase);
    rBranch[i].Sji:=Multiply(Negative(Subtract(rSa[ke-1],rdSa[ke-1])),rParamLF.Pbase);
    rBranch[i].Aij:=Multiply(Cmplx(rdI[ke-1]),rParamLF.Ibase);
    pos:=getAbs(rdSa[ke-1])/getAbs(rSa[ke-1]);
    rBranch[i].Aji:=Multiply(Negative(Multiply(rdI[ke-1],(1-pos))),rParamLF.Ibase);
  end;
end;

procedure LoadflowForwardBack(var rBus:TBusArr1;

```

---

```

var rBranch:TBranchArr1;
var rParamLF:TParamLF);
var i,Nbus,Nsal:integer;
Vs:TCmplx;
A:iArr2;
V,V0,SL,K,dU,dI,dS,Sa:CxArr1;
Z,Za:CxArr2;
Cek:boolean;
begin
Nbus:=high(rBus)+1;
Nsal:=high(rBranch)+1;
if (Nbus-1)>>Nsal then
begin
raise Exception.Create('Bukan jaringan radial');
end;
CreateVBusSLoad(rBus,rParamLF,V,SL);
Z:=CreateZa(Nbus,rBranch,rParamLF);
A:=CreateMatrixA(Z);
Za:=CreateMatrixZ(A,Z);
Cek:=false;
Vs:=Cmplx(1.0,0.0);
rParamLF.Iterasi:=0;
for i:=1 to rParamLF.MaxIterasi do
begin
if Cek=true then
begin
break;
end;
V0:=SaveV0Bus(V);
K:=CalcK(V,SL);
dU:=Calcdu(Za,K);
V:=UpdateVBus(Vs,dU);
Cek:=isConvergen(V0,V,rParamLF);
inc(rParamLF.Iterasi);
end;
dI:=CalcArus(A,K);
dS:=CalcLossBranch(Z,dI);
Sa:=CalcDaya(V,dI);
UpdateBus(V,Sa,rParamLF,rBus);
UpdateBranch(dS,Sa,dI,rParamLF,rBranch);
end;

procedure LoadflowForwardBack(const rTypeLoad:integer;
var rBus:TBusArr1;
var rBranch:TBranchArr1;
var rParamLF:TParamLF);
var i,Nbus,Nsal:integer;
Vs:TCmplx;
A:iArr2;
V,V0,SL,K,dU,dI,dS,Sa:CxArr1;
Z,Za:CxArr2;
Cek:boolean;
begin
Nbus:=high(rBus)+1;
Nsal:=high(rBranch)+1;

```

---

```
if (Nbus-1)<>Nsal then
begin
  raise Exception.Create('Bukan jaringan radial');
end;
CreateVBusSLoad(rBus,rParamLF,V,SL);
Z:=CreateZa(Nbus,rBranch,rParamLF);
A:=CreateMatrixA(Z);
Za:=CreateMatrixZ(A,Z);
Cek:=false;
Vs:=Cmplx(1.0,0.0);
rParamLF.Iterasi:=0;
for i:=1 to rParamLF.MaxIterasi do
begin
  if Cek=true then
  begin
    break;
  end;
  V0:=SaveV0Bus(V);
  K:=CalcK(rTypeLoad,V,SL);
  dU:=CalcdU(Za,K);
  V:=UpdateVBus(Vs,dU);
  Cek:=isConvergen(V0,V,rParamLF);
  inc(rParamLF.Iterasi);
end;
dI:=CalcArus(A,K);
dS:=CalcLossBranch(Z,dI);
Sa:=CalcDaya(V,dI);
UpdateBus(V,Sa,rParamLF,rBus);
UpdateBranch(dS,Sa,dI,rParamLF,rBranch);
end;
end.
```

```
unit uInputLF;

interface

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

type
  TfrmInputLF = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    edtNbus: TEdit;
    edtNsal: TEdit;
    edtVbase: TEdit;
    edtPbase: TEdit;
    cmbParam: TComboBox;
    cmbVKonst: TComboBox;
    cmbPkonst: TComboBox;
    TabSheet2: TTabSheet;
    fgBus: TStringGrid;
    TabSheet3: TTabSheet;
    fgBranch: TStringGrid;
    TabSheet4: TTabSheet;
    fgGen: TStringGrid;
    Panel1: TPanel;
    btnClose: TButton;
    btnNext: TButton;
    SaveDialog1: TSaveDialog;
    procedure btnCloseClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure edtNbusChange(Sender: TObject);
    procedure edtNsalChange(Sender: TObject);
    procedure cmbPkonstChange(Sender: TObject);
    procedure cmbParamChange(Sender: TObject);
    procedure btnNextClick(Sender: TObject);
  private
    { Private declarations }
  protected
    procedure ShowHasil;virtual;abstract;
  public
    { Public declarations }
  end;

var
  frmInputLF: TfrmInputLF;

implementation

{$R *.dfm}
```

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

procedure TfrmInputLF.FormCreate(Sender: TObject);
begin
  edtNbus.Text:='1';
  edtNsal.Text:='1';
  edtVbase.Text:='150';
  edtPbase.Text:='100';
  cmbVkonst.Text:='kV';
  cmbPkonst.Text:='MVA';
  cmbParam.Text:='pu';
  fgBus.Cells[0,0]:='Bus';
  fgBus.Cells[1,0]:='absV (pu)';
  fgBus.Cells[2,0]:='sudV (deg)';
  fgBus.Cells[3,0]:='Pg (MW)';
  fgBus.Cells[4,0]:='Qg (MVAR)';
  fgBus.Cells[5,0]:='PL (MW)';
  fgBus.Cells[6,0]:='QL (MVAR)';
  fgBus.Cells[7,0]:='Cap (pu)';
  fgBus.Cells[8,0]:='Type Bus';
  fgBranch.Cells[0,0] :='No';
  fgBranch.Cells[1,0] :='Dari';
  fgBranch.Cells[2,0] :='Ke';
  fgBranch.Cells[3,0] :='R (pu)';
  fgBranch.Cells[4,0] :='X (pu)';
  fgBranch.Cells[5,0] :='Lc (pu)';
  fgBranch.Cells[6,0] :='Tr';
  fgBranch.Cells[7,0] :='Tu';
  fgBranch.Cells[8,0] :='Su (deg)';
  fgBranch.Cells[9,0] :='Kap (MVA)';
  fgBranch.Cells[10,0]:=Length (m)';
  fgBranch.Cells[11,0]:=Diameter';
  fgGen.Cells[0,0]:="No";
  fgGen.Cells[1,0]:="Bus";
  fgGen.Cells[2,0]:=Qmin (MVAR)';
  fgGen.Cells[3,0]:=Qmax (MVAR)';
  fgGen.Cells[4,0]:=a2';
  fgGen.Cells[5,0]:=a1';
  fgGen.Cells[6,0]:=a0';
  fgGen.Cells[7,0]:=Fix Cost';
  fgGen.Cells[8,0]:=Var Cost';
  fgGen.Cells[9,0]:=Pmin (MW)';
  fgGen.Cells[10,0]:=Pmax (MW)';
  fgGen.Cells[11,0]:=Nama';
end;

procedure TfrmInputLF.edtNbusChange(Sender: TObject);
var i:integer;
begin
  if edtNbus.Text="" then
    begin
```

```

fgBus.RowCount:=2;
end
else
begin
try
fgBus.RowCount:=StrToInt(edtNbus.Text)+1;
for i:=1 to StrToInt(edtNbus.Text) do
begin
fgBus.Cells[0,i]:=IntToStr(i);
fgBus.Cells[1,i]:='1';
fgBus.Cells[2,i]:='0';
fgBus.Cells[3,i]:='0';
fgBus.Cells[4,i]:='0';
fgBus.Cells[5,i]:='0';
fgBus.Cells[6,i]:='0';
fgBus.Cells[7,i]:='0';
fgBus.Cells[8,i]:='3';
end;
fgBus.Cells[8,1]:='1';
except
raise Exception.Create('Tolong Masukkan angka!');
end;
end;
end;

procedure TfrmInputLF.edtNsalChange(Sender: TObject);
var i:integer;
begin
if edtNsal.Text="" then
begin
fgBranch.RowCount:=2;
end
else
begin
try
fgBranch.RowCount:=StrToInt(edtNsal.Text)+1;
for i:=1 to StrToInt(edtNsal.Text) do
begin
fgBranch.Cells[0,i]:=IntToStr(i);
fgBranch.Cells[3,i]:='0';
fgBranch.Cells[4,i]:='0';
fgBranch.Cells[5,i]:='0';
fgBranch.Cells[6,i]:='0';
fgBranch.Cells[7,i]:='0';
fgBranch.Cells[8,i]:='0';
fgBranch.Cells[9,i]:='10000';
fgBranch.Cells[10,i]:='0';
end;
except
raise Exception.Create('Tolong Masukkan angka!');
end;
end;
end;

procedure TfrmInputLF.cmbPkonstChange(Sender: TObject);

```

---

```

begin
if cmbPKonst.Text='VA' then
begin
  fgBus.Cells[3,0]:='Pg (W)';
  fgBus.Cells[4,0]:='Qg (VAR)';
  fgBus.Cells[5,0]:='PL (W)';
  fgBus.Cells[6,0]:='QL (VAR)';
  fgBranch.Cells[9,0] :='Kap (VA)';
  fgGen.Cells[9,0]:=Pmin (W)';
  fgGen.Cells[10,0]:=Pmax (W)';
end
else if cmbPKonst.Text='kVA' then
begin
  fgBus.Cells[3,0]:='Pg (kW)';
  fgBus.Cells[4,0]:='Qg (kVAR)';
  fgBus.Cells[5,0]:='PL (kW)';
  fgBus.Cells[6,0]:='QL (kVAR)';
  fgBranch.Cells[9,0] :='Kap (VA)';
  fgGen.Cells[9,0]:=Pmin (kW)';
  fgGen.Cells[10,0]:=Pmax (kW)';
end
else if cmbPKonst.Text='MVA' then
begin
  fgBus.Cells[3,0]:='Pg (MW)';
  fgBus.Cells[4,0]:='Qg (MVAR)';
  fgBus.Cells[5,0]:='PL (MW)';
  fgBus.Cells[6,0]:='QL (MVAR)';
  fgBranch.Cells[9,0] :='Kap (VA)';
  fgGen.Cells[9,0]:=Pmin (MW)';
  fgGen.Cells[10,0]:=Pmax (MW)';
end;
end;

procedure TfrmInputLF.cmbParamChange(Sender: TObject);
begin
if cmbParam.Text='pu' then
begin
  fgBranch.Cells[3,0]:='R (pu)';
  fgBranch.Cells[4,0]:='X (pu)';
  fgBranch.Cells[5,0]:='Lc (pu)';
  fgBus.Cells[7,0]:='Cap (pu)';
end
else if cmbParam.Text='ohm' then
begin
  fgBranch.Cells[3,0]:='R (ohm)';
  fgBranch.Cells[4,0]:='X (ohm)';
  fgBranch.Cells[5,0]:='Lc (ohm)';
  fgBus.Cells[7,0]:='Cap (ohm)';
end;
end;

procedure TfrmInputLF.btnNextClick(Sender: TObject);
var NamaFile>Nama:string;
  input:Textfile;
  NCable,Nbus,Nsal,Ngen,i,typ,dari,ke,param:integer;

```

---

```

R,X,Lc,Tr,Tu,Su,Cap,CapSal,phi,Pmin,Pmax,Harga,length:double;
VKonst,PKonst,Vbase,Pbase,absV,sudV,Pg,Qg,PL,QL:double;
jam,menit,detik,mdetik:word;
mulai,selesai,selang:TDateTime;
load:string;
begin
if btnNext.Caption='&Save' then
begin
Nbus:=StrToInt(edtNbus.Text);
Nsal:=StrToInt(edtNsal.Text);
Vbase:=StrToFloat(edtVbase.Text);
VKonst:=1;
if cmbVKonst.Text='V' then
begin
VKonst:=1;
end
else if cmbVKonst.Text='kV' then
begin
VKonst:=1000;
end
else if cmbVKonst.Text='MV' then
begin
VKonst:=1000000;
end;
Pbase:=StrToFloat(edtPbase.Text);
PKonst:=1;
if cmbPKonst.Text='VA' then
begin
PKonst:=1;
end
else if cmbPKonst.Text='kVA' then
begin
PKonst:=1000;
end
else if cmbPKonst.Text='MVA' then
begin
PKonst:=1000000;
end;
Param:=1;
if cmbParam.Text='pu' then
begin
Param:=1;
end
else if cmbParam.Text='ohm' then
begin
Param:=2;
end;
try
if SaveDialog1.Execute then
begin
NamaFile:=SaveDialog1.FileName;
AssignFile(input,NamaFile+'.txt');
Rewrite(input);
Writeln(input,Nbus);
Writeln(input,Nsal);

```

---

```

Writeln(input,Vbase:6:2);
Writeln(input,VKonst:7:0);
Writeln(input,Pbase:6:2);
Writeln(input,PKonst:7:0);
Writeln(input,param);
phi:=4*arctan(1);
for i:=1 to Nbus do
begin
absV:=StrToFloat(fgBus.Cells[1,i]);
sudV:=StrToFloat(fgBus.Cells[2,i])*180/phi;
Pg:=StrToFloat(fgBus.Cells[3,i]);
Qg:=StrToFloat(fgBus.Cells[4,i]);
PL:=StrToFloat(fgBus.Cells[5,i]);
QL:=StrToFloat(fgBus.Cells[6,i]);
Cap:=StrToFloat(fgBus.Cells[7,i]);
Typ:=StrToInt(fgBus.Cells[8,i]);
Writeln(input,absV:7:5,'',sudV:7:5,'',Pg:9:3,'',Qg:9:3,
'',PL:9:3,'',QL:9:3,'',Cap:7:5,'',Typ);
end;
for i:=1 to Nsal do
begin
dari:=StrToInt(fgBranch.Cells[1,i]);
ke:=StrToInt(fgBranch.Cells[2,i]);
R:=StrToFloat(fgBranch.Cells[3,i]);
X:=StrToFloat(fgBranch.Cells[4,i]);
Lc:=StrToFloat(fgBranch.Cells[5,i]);
Tr:=StrToFloat(fgBranch.Cells[6,i]);
Tu:=StrToFloat(fgBranch.Cells[7,i]);
Su:=StrToFloat(fgBranch.Cells[8,i]);
CapSal:=StrToFloat(fgBranch.Cells[9,i]);
//Length:=StrToFloat(fgBranch.Cells[21,i]);
Writeln(input,dari,'',ke,'',R:7:5,'',X:7:5,'',
Lc:7:5,'',Tr:7:5,'',Tu:7:5,'',Su:7:5,'',
CapSal:7:2);
end;
Ngen:=0;
load:='3';
for i:=1 to Nbus do
begin
if fgBus.Cells[8,i]<>load then
begin
inc(Ngen);
end;
end;
Writeln(input,Ngen);
for i:=1 to Ngen do
begin
dari:=StrToInt(fgGen.Cells[1,i]);
R:=StrToFloat(fgGen.Cells[2,i]);
X:=StrToFloat(fgGen.Cells[3,i]);
Lc:=StrToFloat(fgGen.Cells[4,i]);
Tr:=StrToFloat(fgGen.Cells[5,i]);
Tu:=StrToFloat(fgGen.Cells[6,i]);
Su:=StrToFloat(fgGen.Cells[7,i]);
CapSal:=StrToFloat(fgGen.Cells[8,i]);

```

---

```
Pmin:=StrToFloat(fgGen.Cells[9,i]);
Pmax:=StrToFloat(fgGen.Cells[10,i]);
Writeln(input,dari,'.R:7:2','.X:7:2','.Lc:7:5','');
Tr:7:5,'.Tu:7:5,'.Su:7:2,'.CapSal:7:2,';
Pmin:7:5,'.Pmax:7:2);
begin;
end;
CloseFile(input);
MessageDlg('File berhasil disimpan!',mtInformation,[mbOK],0);
except
  MessageDlg('Tolong dicek angka-angkanya kembali!',mtWarning,[mbOK],0);
end;
end
else if btnNext.Caption='&Next' then
begin
  ShowHasil;
end;
end;
end.
```

```
unit uLoadflow;

interface

uses uUtils,uComplex;

type
  TParamBranch=(pbOhm,pbPu);

  TParamLF=record
    MaxIterasi,Iterasi:byte;
    Vbase,Vkonst,Pbase,Pkonst,Zbase,Ibase,Toleransi:double;
    ParamBranch:TParamBranch;
    SumGen,SumLoad,SumLoss:TCmplx;
  end;

  TBus=record
    absV,sudV,Pgen,Qgen,PL,QL,Cap:double;
    typeBus:byte;
  end;

  TBusArr1=array of TBus;

  TBranch=record
    dari,ke:integer;
    R,X,Lc,Tr,Tu,Su,KapSal,Length,JenisCable:double;
    Sij,Sji,Aij,Aji:TCmplx;
  end;

  TBranchArr1=array of TBranch;

  TGenLF=record
    nama:string;
    bus:integer;
    Qmin,Qmax,a2,a1,a0,FixCost,VarCost,Pmin,Pmax:double;
  end;

  TGenLFArr1=array of TGenLF;

  TSwith=record
    dari,ke:integer;
  end;

  TSwithArr1=array of TSwith;

function CalcCostGenLF(const rGen:TGenLF;
  const rPgen:double):double;
procedure DecodeCommDataToLFData(const rBus:TBusArr1;
  var rNbus,rNsal:integer;
  var rV,rSg,rSL:CxArr1;
  var rCap:dArr1;
  var rTypBus:iArr1;
  const rBranch:TBranchArr1;
  var rZ,rTp:CxArr2;
  var rLc,rTr:dArr2);
```

```
procedure DecodeData(var rParamLF:TParamLF;
  var rNbus:integer;
  var rV,rSg,rSL:CxArr1;
  var rCap:dArr1;
  var rTypBus:iArr1;
  var rZ,rTp:CxArr2;
  var rLc,rTr:dArr2);
function FindSumGen(const rNbus:integer;
  const rTypBus:iArr1):integer;
procedure Admitansi(const rNbus:integer;
  const rZ,rTp:CxArr2;
  const rLc,rTr:dArr2;
  const rCap:dArr1;
  var rY:CxArr2);
function RecToPolar(const rData:CxArr1):CxArr1;
function PolarToRec(const rData:CxArr1):CxArr1;
function MaxDataArray(const rData:dArr1):double;
procedure AliranDaya(const rNbus:integer;const rV:CxArr1;
  const rY:CxArr2;
  const rLc:dArr2;
  var rAlirS:CxArr2);
procedure DayaGen(const rNbus:integer;
  const rV,rSL:CxArr1;
  const rY:CxArr2;
  const rTyp:iArr1;
  var rSg:CxArr1);
procedure DayaSlack(const rNbus:integer;
  const rAlirS:CxArr2;
  const rTyp:iArr1;
  const rSL:CxArr1;
  var rSg:CxArr1);
procedure ArusBranch(const rNbus:integer;
  const rV:CxArr1;
  const rLc:dArr2;
  const rY:CxArr2;
  var rArus:CxArr2);
procedure UpdateAkhir(const rNbus,rNsal:integer;
  var rParamLF:TParamLF;
  const rV,rSg,rSL:CxArr1;
  const rAlir,rArus:CxArr2;
  var rBus:TBusArr1;
  var rBranch:TBranchArr1);overload;
procedure UpdateAkhir(const rNbus:integer;
  var rParamLF:TParamLF;
  var rSg,rSL:CxArr1;
  var rAlir,rArus:CxArr2);overload;
function FindVarControl(var rBus:TBusArr1):iArr1;overload;
function FindVarControl(var rSg:CxArr1;
  var rTypeBus:iArr1):iArr1;overload;
function FindBatasControl(var rBus:TBusArr1;
  var rParamLF:TParamLF):TBatasArr1;

var gBus:TBusArr1;
  gBranch:TBranchArr1;
  gParamLF:TParamLF;
```

```

gGenLF:TGenLFArr1;

implementation

function CalcCostGenLF(const rGen:TGenLF;
  const rPgen:double):double;
begin
  result:=0;
  if rPgen>0 then
  begin
    result:=rGen.a2*sqr(rPgen)+rGen.a1*rPgen+rGen.a0;
  end;
end;

procedure DecodeCommDataToLFData(const rBus:TBusArr1;
  var rNbus,rNsal:integer;
  var rV,rSg,rSL:CxArr1;
  var rCap:dArr1;
  var rTypBus:iArr1;
  const rBranch:TBranchArr1;
  var rZ,rTp:CxArr2;
  var rLc,rTr:dArr2);
var i,dari,ke:integer;
begin
  rNbus:=high(rBus)+1;
  rNsal:=high(rBranch)+1;
  SetLength(rV,rNbus);
  SetLength(rSg,rNbus);
  SetLength(rSL,rNbus);
  SetLength(rCap,rNbus);
  SetLength(rTypBus,rNbus);
  for i:=0 to rNbus-1 do
  begin
    rV[i].real:=rBus[i].absV;
    rV[i].imag:=rBus[i].sudV;
    rSg[i].real:=rBus[i].Pgen;
    rSg[i].imag:=rBus[i].Qgen;
    rSL[i].real:=rBus[i].PL;
    rSL[i].imag:=rBus[i].QL;
    rCap[i]:=rBus[i].Cap;
    rTypBus[i]:=rBus[i].typeBus;
  end;
  SetLength(rZ,rNbus,rNbus);
  SetLength(rTp,rNbus,rNbus);
  SetLength(rLc,rNbus,rNbus);
  SetLength(rTr,rNbus,rNbus);
  for i:=0 to rNsal-1 do
  begin
    dari:=rBranch[i].dari-1;
    ke:=rBranch[i].ke-1;
    rZ[dari,ke].real:=rBranch[i].R;
    rZ[dari,ke].imag:=rBranch[i].X;
    rLc[dari,ke]:=rBranch[i].Lc;
    rTr[dari,ke]:=rBranch[i].Tr;
    rTp[dari,ke].real:=rBranch[i].Tu;
  end;
end;

```

---

```

rTp[dari,ke].imag:=rBranch[i].Su;
end;
end;

procedure DecodeData(var rParamLF;TParamLF;
var rNbus:integer;
var rV,rSg,rSL:CxArr1;
var rCap:dArr1;
var rTypBus:iArr1;
var rZ,rTp:CxArr2;
var rLc,rTr:dArr2);
var i,j:integer;
begin
if rParamLF.MaxIterasi=0 then rParamLF.MaxIterasi:=15;
if rParamLF.Toleransi=0 then rParamLF.Toleransi:=0.0001;
rParamLF.Zbase:=sqr(rParamLF.Vbase*rParamLF.Vkonst)/
(rParamLF.Pbase*rParamLF.Pkonst);
rParamLF.Ibase:=(rParamLF.Vbase*rParamLF.Vkonst)/rParamLF.Zbase;
for i:=0 to rNbus-1 do
begin
rSg[i].real:=rSg[i].real/rParamLF.Pbase;
rSg[i].imag:=rSg[i].imag/rParamLF.Pbase;
rSL[i].real:=rSL[i].real/rParamLF.Pbase;
rSL[i].imag:=rSL[i].imag/rParamLF.Pbase;
end;
if rParamLF.ParamBranch=pbOhm then
begin
for i:=0 to rNbus-1 do
begin
for j:=0 to rNbus-1 do
begin
if rZ[i,j].imag<>0 then
begin
rZ[i,j].real:=rZ[i,j].real/rParamLF.Zbase;
rZ[i,j].imag:=rZ[i,j].imag/rParamLF.Zbase;
rLc[i,j]:=rLc[i,j]/rParamLF.Zbase;
end;
end;
end;
end;
{for i:=0 to rNbus-1 do
begin
for j:=0 to rNbus-1 do
begin
if rZ[i,j].imag<>0 then
begin
rZ[j,i]:=Cmplx(rZ[i,j]);
rLc[j,i]:=rLc[i,j];
end;
end;
end;
end;}
end;

function FindSumGen(const rNbus:integer;
const rTypBus:iArr1):integer;

```

---

```

var i:integer;
begin
  result:=0;
  for i:=0 to rNbus-1 do
  begin
    if rTypBus[i]=2 then
    begin
      inc(result);
    end;
  end;
end;

procedure Admitansi(const rNbus:integer;
  const rZ,rTp:CxArr2;
  const rLc,rTr:dArr2;
  const rCap:dArr1;
  var rY:CxArr2);
var i,j,k:integer;
  a,b:double;
  Lc:dArr2;
  CY:CxArr2;
  sum,Za,Ca:TCmplx;
begin
  SetLength(CY,rNbus,rNbus);
  SetLength(Lc,rNbus,rNbus);
  sum:=Cmplx(1.0,0.0);
  for i:=0 to rNbus-1 do
  begin
    for j:=0 to rNbus-1 do
    begin
      if rZ[i,j].imag<>0 then
      begin
        if rTr[i,j]<>0 then
        begin
          a:=rTr[i,j];
          CY[i,j]:=Divide(sum,rZ[i,j]);
          Lc[i,j]:=(1/sqr(a)-1/a)*CY[i,j].imag;
          Lc[j,i]:=(1-1/a)*CY[i,j].imag;
          CY[i,j]:=Divide(CY[i,j],a);
          CY[j,i]:=Cmplx(CY[i,j]);
        end
        else
        begin
          CY[i,j]:=Divide(sum,rZ[i,j]);
          CY[j,i]:=Cmplx(CY[i,j]);
          Lc[i,j]:=rLc[i,j];
          Lc[j,i]:=rLc[i,j];
        end;
      end;
    end;
  end;
  SetLength(rY,rNbus,rNbus);
  for i:=0 to rNbus-1 do
  begin
    for j:=0 to rNbus-1 do

```

---

```

begin
if j=i then
begin
  rY[i,j]:=Cmplx(0.0,0.0);
  for k:=0 to rNbus-1 do
  begin
    rY[i,j].real:=rY[i,j].real+CY[i,k].real;
    rY[i,j].imag:=rY[i,j].imag+CY[i,k].imag+Lc[i,k];
  end;
end
else
begin
  rY[i,j].real:=-CY[i,j].real;
  rY[i,j].imag:=-CY[i,j].imag;
end;
end;
end;
for i:=0 to rNbus-1 do
begin
  for j:=0 to rNbus-1 do
  begin
    if rTp[i,j].real<>0 then
    begin
      a:=rTp[i,j].real*cos(rTp[i,j].imag);
      b:=rTp[i,j].real*sin(rTp[i,j].imag);
      rY[i,i].real:=rY[i,i].real-CY[i,j].real;
      rY[i,i].imag:=rY[i,i].imag-CY[i,j].imag-rLc[i,j];
      rY[i,i].real:=rY[i,i].real+CY[i,j].real/(sqr(a)+sqr(b));
      rY[i,i].imag:=rY[i,i].imag+CY[i,j].imag/(sqr(a)+sqr(b))+rLc[i,j];
      Za:=Cmplx(CY[i,j]);
      Za:=Negative(Za);
      Ca:=Cmplx(a,b);
      sum:=Divide(Za,Ca);
      rY[j,i]:=Cmplx(sum);
      Ca:=Conjugate(Ca);
      sum:=Divide(Za,Ca);
      rY[i,j]:=Cmplx(sum);
      {sUpfc:=TComplex.Create(aUpfc[i,j].tap*cos(aUpfc[i,j].sudut),
        aUpfc[i,j].tap*sin(aUpfc[i,j].sudut));
      CLc:=TComplex.Create(0.0,aLc[i,j]);
      result[i,i]:=result[i,i]-Cx[i,j]-CLc;
      result[i,i]:=result[i,i]+Cx[i,j]/sqr(sUpfc.Abs)+CLc;
      result[i,j]:=-Cx[i,j]/conj(sUpfc);
      result[j,i]:=-Cx[i,j]/sUpfc;
      CLc.Free;
      sUpfc.Free;}
      end;
    end;
  end;
end;
for i:=0 to rNbus-1 do
begin
  if rCap[i]<>0 then
  begin
    rY[i,i].imag:=rY[i,i].imag+rCap[i];
  end;

```

---

```

end;
end;

function RecToPolar(const rData:CxArr1):CxArr1;
var i,NData:integer;
    abs,sud:double;
begin
  NData:=high(rData)+1;
  SetLength(result,NData);
  for i:=0 to NData-1 do
  begin
    abs:=getAbs(rData[i]);
    sud:=getAngleRad(rData[i]);
    result[i]:=Cmplx(abs,sud);
  end;
end;

function PolarToRec(const rData:CxArr1):CxArr1;
var i,NData:integer;
    real,imag:double;
begin
  NData:=high(rData)+1;
  SetLength(result,NData);
  for i:=0 to NData-1 do
  begin
    real:=rData[i].real*cos(rData[i].imag);
    imag:=rData[i].real*sin(rData[i].imag);
    result[i]:=Cmplx(real,imag);
  end;
end;

function MaxDataArray(const rData:dArr1):double;
var i,NData:integer;
begin
  NData:=high(rData)+1;
  result:=abs(rData[0]);
  for i:=1 to NData-1 do
  begin
    if result<abs(rData[i]) then
    begin
      result:=abs(rData[i]);
    end;
  end;
end;

procedure AliranDaya(const rNbus:integer;const rV:CxArr1;
                      const rY:CxArr2;
                      const rLc:dArr2;
                      var rAlirS:CxArr2);
var i,j:integer;
    Lca,tmp1,tmp2:TCmplx;
begin
  SetLength(rAlirS,rNbus,rNbus);
  for i:=0 to rNbus-1 do
  begin

```

---

```

for j:=0 to rNbus-1 do
begin
rAlirS[i,j].real:=0.0;
rAlirS[i,j].imag:=0.0;
if j<>i then
begin
  if rY[i,j].imag<>0 then
  begin
    Lca:=Cmplx(0.0,rLc[i,j]);
    tmp1:=Multiply(Multiply(Conjugate(rV[i]),Subtract(rV[i],rV[j])),
      Negative(rY[i,j]));
    tmp2:=Multiply(Multiply(Conjugate(rV[i]),rV[i]),Lca);
    rAlirS[i,j]:=Conjugate(Add(tmp1,tmp2));
    //result[i,j]:=conj(aV[i])*(aV[i]-aV[j])*(-aY[i,j])+*
    //            conj(aV[i])*aV[i]*dLc;
    //result[i,j]:=conj(result[i,j]);
  end;
end;
end;
end;
end;

procedure DayaGen(const rNbus:integer;
  const rV,rSL:CxArr1;
  const rY:CxArr2;
  const rTyp:iArr1;
  var rSg:CxArr1);
var i,j:integer;
  sum:double;
begin
  for i:=0 to rNbus-1 do
  begin
    sum:=0.0;
    if rTyp[i]=2 then
    begin
      for j:=0 to rNbus-1 do
      begin
        //sum:=sum+(Fj*(Ej*Gij+Fj*-Bij)-Ei*(Fj*Gij-Ej*-Bij));
        sum:=sum+(rV[i].imag*(rV[j].real*rY[i,j].real+rV[j].imag*-rY[i,j].imag)-
          rV[i].real*(rV[j].imag*rY[i,j].real-rV[j].real*-rY[i,j].imag));
      end;
      rSg[i].imag:=sum+rSL[i].imag;
    end;
  end;
end;

procedure DayaSlack(const rNbus:integer;
  const rAlirS:CxArr2;
  const rTyp:iArr1;
  const rSL:CxArr1;
  var rSg:CxArr1);
var i,j:integer;
  sumP,sumQ:double;
begin
  for i:=0 to rNbus-1 do

```

---

```

begin
  if rTyp[i]=1 then
    begin
      sumP:=0.0;
      sumQ:=0.0;
      for j:=0 to rNbus-1 do
        begin
          if rAlirS[i,j].imag<>0 then
            begin
              sumP:=sumP+rAlirS[i,j].real;
              sumQ:=sumQ+rAlirS[i,j].imag;
            end;
          end;
          rSg[i].real:=sumP+rSL[i].real;
          rSg[i].imag:=sumQ+rSL[i].imag;
        end;
      end;
    end;
  end;

procedure ArusBranch(const rNbus:integer;
  const rV:CxArr1;
  const rLc:dArr2;
  const rY:CxArr2;
  var rArus:CxArr2);
var i,j:integer;
  xLc,tmp1,tmp2:TCmplx;
begin
  SetLength(rArus,rNbus,rNbus);
  SetLength(rArus,rNbus,rNbus);
  for i:=0 to rNbus-1 do
    begin
      for j:=0 to rNbus-1 do
        begin
          if rY[i,j].imag<>0 then
            begin
              xLc:=Cmplx(0,rLc[i,j]);
              tmp1:=Multiply(Subtract(rV[i],rV[j]),Negative(rY[i,j]));
              tmp2:=Multiply(rV[i],xLc);
              rArus[i,j]:=Conjugate(Add(tmp1,tmp2));
              //result[i,j]:=(aV[i]-aV[j])*(-aY[i,j])+aV[i]*xLc;
              //result[i,j]:=Conj(result[i,j]);
            end
          else
            begin
              rArus[i,j].real:=0.0;
              rArus[i,j].imag:=0.0;
            end;
        end;
    end;
end;

procedure UpdateAkhir(const rNbus,rNsai:integer;
  var rParamLF:TParamLF;
  const rV,rSg,rSL:CxArr1;
  const rAlir,rArus:CxArr2);

```

---

```

var rBus:TBusArr1;
var rBranch:TBranchArr1;
var i,dari,ke:integer;
begin
rParamLF.SumGen:=Cmplx(0.0,0.0);
rParamLF.SumLoad:=Cmplx(0.0,0.0);
for i:=0 to rNbus-1 do
begin
rBus[i].absV:=rV[i].real;
rBus[i].sudV:=rV[i].imag;
rBus[i].Pgen:=rSg[i].real*rParamLF.Pbase;
rBus[i].Qgen:=rSg[i].imag*rParamLF.Pbase;
rBus[i].PL:=rSL[i].real*rParamLF.Pbase;
rBus[i].QL:=rSL[i].imag*rParamLF.Pbase;
rParamLF.SumGen:=Add(rParamLF.SumGen,rSg[i]);
rParamLF.SumLoad:=Add(rParamLF.SumLoad,rSL[i]);
end;
rParamLF.SumGen:=Multiply(rParamLF.SumGen,rParamLF.Pbase);
rParamLF.SumLoad:=Multiply(rParamLF.SumLoad,rParamLF.Pbase);
rParamLF.SumLoss:=Subtract(rParamLF.SumGen,rParamLF.SumLoad);
for i:=0 to rNsal-1 do
begin
dari:=rBranch[i].dari-1;
ke:=rBranch[i].ke-1;
rBranch[i].Sij:=Multiply(Cmplx(rAlir[dari,ke]),rParamLF.Pbase);
rBranch[i].Sji:=Multiply(Cmplx(rAlir[ke,dari]),rParamLF.Pbase);
rBranch[i].Aij:=Multiply(Cmplx(rArus[dari,ke]),rParamLF.Ibase);
rBranch[i].Aji:=Multiply(Cmplx(rArus[ke,dari]),rParamLF.Ibase);
end;
end;

procedure UpdateAkhir(const rNbus:integer;
var rParamLF:TPParamLF;
var rSg,rSL:CxArr1;
var rAlir,rArus:CxArr2);
var i,j:integer;
begin
rParamLF.SumGen:=Cmplx(0.0,0.0);
rParamLF.SumLoad:=Cmplx(0.0,0.0);
for i:=0 to rNbus-1 do
begin
rSg[i].real:=rSg[i].real*rParamLF.Pbase;
rSg[i].imag:=rSg[i].imag*rParamLF.Pbase;
rSL[i].real:=rSL[i].real*rParamLF.Pbase;
rSL[i].imag:=rSL[i].imag*rParamLF.Pbase;
rParamLF.SumGen:=Add(rParamLF.SumGen,rSg[i]);
rParamLF.SumLoad:=Add(rParamLF.SumLoad,rSL[i]);
end;
rParamLF.SumLoss:=Subtract(rParamLF.SumGen,rParamLF.SumLoad);
for i:=0 to rNbus-1 do
begin
for j:=0 to rNbus-1 do
begin
if rAlir[i,j].imag<>0 then
begin

```

```

rAlir[i,j]:=Multiply(rAlir[i,j],rParamLF.Pbase);
rArus[i,j]:=Multiply(rArus[i,j],rParamLF.Ibase);
end;
end;
end;
end;

function FindVarControl(var rBus:TBusArr1):iArr1;
var i,Nbus,sa:integer;
begin
Nbus:=high(rBus)+1;
sa:=0;
for i:=0 to Nbus-1 do
begin
if rBus[i].typeBus<>3 then
begin
inc(sa);
end;
end;
for i:=0 to Nbus-1 do
begin
if rBus[i].typeBus=3 then
begin
if rBus[i].Qgen<>0 then
begin
inc(sa);
end;
end;
end;
SetLength(result,sa);
sa:=0;
for i:=0 to Nbus-1 do
begin
if rBus[i].typeBus<>3 then
begin
result[sa]:=i;
inc(sa);
end;
end;
for i:=0 to Nbus-1 do
begin
if rBus[i].typeBus=3 then
begin
if rBus[i].Qgen<>0 then
begin
result[sa]:=i;
inc(sa);
end;
end;
end;
end;
end;

function FindVarControl(var rSg:CxArr1;
var rTypeBus:iArr1):iArr1;
var i,Nbus,sa:integer;

```

---

```
begin
  Nbus:=high(rSg)+1;
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    if rTypeBus[i]<>3 then
    begin
      inc(sa);
    end;
  end;
  for i:=0 to Nbus-1 do
  begin
    if rTypeBus[i]=3 then
    begin
      if rSg[i].imag<>0 then
      begin
        inc(sa);
      end;
    end;
  end;
  SetLength(result,sa);
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    if rTypeBus[i]<>3 then
    begin
      result[sa]:=i;
      inc(sa);
    end;
  end;
  for i:=0 to Nbus-1 do
  begin
    if rTypeBus[i]=3 then
    begin
      if rSg[i].imag<>0 then
      begin
        result[sa]:=i;
        inc(sa);
      end;
    end;
  end;
  end;

function FindBatasControl(var rBus:TBusArr1;
                           var rParamLF:TParamLF):TBatasArr1;
var i,Nbus,sa:integer;
begin
  Nbus:=high(rBus)+1;
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    if rBus[i].typeBus<>3 then
    begin
      inc(sa);
    end;
  end;
```

---

```
end;
for i:=0 to Nbus-1 do
begin
  if rBus[i].typeBus=3 then
    begin
      if rBus[i].Qgen<>0 then
        begin
          inc(sa);
        end;
      end;
    end;
  SetLength(result,sa);
  sa:=0;
  for i:=0 to Nbus-1 do
  begin
    if rBus[i].typeBus<>3 then
      begin
        result[sa].min:=0.95;
        result[sa].max:=1.05;
        inc(sa);
      end;
    end;
  for i:=0 to Nbus-1 do
  begin
    if rBus[i].Qgen<>0 then
      begin
        result[sa].min:=0;
        result[sa].max:=rBus[i].Qgen/rParamLF.Pbase;
        inc(sa);
      end;
    end;
  end;
end.
```

```

unit uNewtonRaphson;

interface

uses uUtils,uComplex,uLoadflow,uMatrix;

procedure NewtonRaphson(var rBus:TBusArr1;
    var rBranch:TBranchArr1;
    var rParamLF:TParamLF);overload;
procedure NewtonRaphson(var rBus:TBusArr1;
    var rBranch:TBranchArr1;
    var rParamLF:TParamLF;
    var rJaq:dArr2);overload;
procedure NewtonRaphson(var rParamLF:TParamLF;
    var rV,rSg,rSL:CxArr1;
    var rCap:dArr1;
    var rTypBus:iArr1;
    var rZ,rTp,rAlir,rArus:CxArr2;
    var rLc,rTr:dArr2);overload;

implementation

function MismatchDaya(var rNbus,rNgen:integer;
    var rV,rSg,rSL:CxArr1;
    var rTyp:iArr1;
    var rY:CxArr2);dArr1;
var i,j,Ns,Np,Nq:integer;
    sumP,sumQ:double;
begin
  Ns:=rNbus-1+rNbus-rNgen-1;
  SetLength(result,Ns);
  Np:=1;
  Nq:=rNbus-2;
  for i:=0 to rNbus-1 do
  begin
    if rTyp[i]<>1 then
    begin
      inc(Np);
      sumP:=0.0;
      for j:=0 to rNbus-1 do
      begin
        //sumP:=sumP+Ui*Uj*(Gij*cos(di-dj)+Bij*sin(di-dj));
        sumP:=sumP+rV[i].real*rV[j].real*(rY[i,j].real*
          cos(rV[i].imag-rV[j].imag)+rY[i,j].imag*
          sin(rV[i].imag-rV[j].imag));
      end;
      result[Np]:=rSg[i].real-rSL[i].real-sumP;
    end;
    if rTyp[i]=3 then
    begin
      inc(Nq);
      sumQ:=0.0;
      for j:=0 to rNbus-1 do
      begin
        //sumQ:=sumQ+Ui*Uj*(Gij*sin(di-dj)-Bij*cos(di-dj));
      end;
    end;
  end;
end;

```

---

```

sumQ:=sumQ+rV[i].real*rV[j].real*(rY[i,j].real*
    sin(rV[i].imag-rV[j].imag)-rY[i,j].imag*
    cos(rV[i].imag-rV[j].imag));
end;
result[Nq]:=rSg[i].imag-rSL[i].imag-sumQ;
end;
end;
end;

function Jaqobian(var rNbus,rNgen:integer;
  var rV:CxArr1;
  var rTyp:iArr1;
  var rY:CxArr2):dArr2;
var i,j,k,row,col:integer;
  sum,Pj,Qj:double;
begin
  row:=rNbus-1+rNbus-rNgen-1;
  SetLength(result,row,row);
  //Pembentukan Jaqobian H dP/d0
  row:=-1;
  for i:=0 to rNbus-1 do
  begin
    if rTyp[i]<>1 then
    begin
      inc(row);
      col:=-1;
      for j:=0 to rNbus-1 do
      begin
        if rTyp[j]<>1 then
        begin
          inc(col);
          if j=i then
          begin
            sum:=0.0;
            for k:=0 to rNbus-1 do
            begin
              //sum:=sum+((Gjk*sin(dj-dk)-Bjk*cos(dj-dk))*Uk);
              sum:=sum+((rY[j,k].real*sin(rV[j].imag-rV[k].imag)-
                rY[j,k].imag*cos(rV[j].imag-rV[k].imag))*
                rV[k].real);
            end;
            //Qj:=sum*Uj;
            Qj:=sum*rV[j].real;
            //result[row,col]:=-Qj-Bij*sqr(Ui);
            result[row,col]:=-Qj-rY[i,j].imag*sqr(rV[i].real);
          end
          else
          begin
            //result[row,col]:=Ui*Uj*(Gij*sin(di-dj)-Bij*cos(di-dj));
            result[row,col]:=rV[i].real*rV[j].real*(rY[i,j].real*
              sin(rV[i].imag-rV[j].imag)-rY[i,j].imag*
              cos(rV[i].imag-rV[j].imag));
          end;
        end;
      end;
    end;
  end;

```

---

```

    end;
end;
//Pembentukan Matrik N dP/dV
row:=-1;
for i:=0 to rNbus-1 do
begin
if rTyp[i]<>1 then
begin
inc(row);
col:=rNbus-2;
for j:=0 to rNbus-1 do
begin
if rTyp[j]=3 then
begin
inc(col);
if j=i then
begin
sum:=0.0;
for k:=0 to rNbus-1 do
begin
//sum:=sum+((Gjk*cos(dj-dk)+Bjk*sin(dj-dk))*Uk);
sum:=sum+((rY[j,k].real*cos(rV[j].imag-rV[k].imag)-
rY[j,k].imag*sin(rV[j].imag-rV[k].imag))*rV[k].real);
end;
//Pj:=sum*Uj;
Pj:=sum*rV[j].real;
//result[row,col]:=Pj+Gij*Ui;
result[row,col]:=Pj+rY[i,j].real*sqr(rV[i].real);
end
else
begin
//result[row,col]:=Uj*(Gij*cos(di-dj)+Bij*sin(di-dj));
result[row,col]:=rV[i].real*rV[j].real*(rY[i,j].real*
cos(rV[i].imag-rV[j].imag)+rY[i,j].imag*
sin(rV[i].imag-rV[j].imag));
end;
end;
end;
end;
end;
//Pembentukan Jaqobian M dQ/d0
row:=rNbus-2;
for i:=0 to rNbus-1 do
begin
if rTyp[i]=3 then
begin
inc(row);
col:=-1;
for j:=0 to rNbus-1 do
begin
if rTyp[j]<>1 then
begin
inc(col);
if j=i then

```

---

```

begin
  sum:=0;
  for k:=0 to rNbus-1 do
    begin
      //sum:=sum+((Gjk*cos(dj-dk)+Bjk*sin(dj-dk))*Uk);
      sum:=sum+((rY[j,k].real*cos(rV[j].imag-rV[k].imag)-
                  rY[j,k].imag*sin(rV[j].imag-rV[k].imag))*rV[k].real);
    end;
    //Pj:=sum*Ui;
    Pj:=sum*rV[i].real;
    //result[row,col]:=Pj-Gij*sqr(Ui);
    result[row,col]:=Pj-rY[i,j].real*sqr(rV[i].real);
  end
  else
    begin
      //result[row,col]:=-Ui*Uj*(Gij*cos(di-dj)+Bij*sin(di-dj));
      result[row,col]:=-rV[i].real*rV[j].real*(rY[i,j].real*
          cos(rV[i].imag-rV[j].imag)+rY[i,j].imag*
          sin(rV[i].imag-rV[j].imag));
    end;
  end;
end;
//Pembentukan Jaqobian L dQ/dV
row:=rNbus-2;
for i:=0 to rNbus-1 do
begin
  if rTyp[i]=3 then
    begin
      inc(row);
      col:=rNbus-2;
      for j:=0 to rNbus-1 do
        begin
          if rTyp[j]=3 then
            begin
              inc(col);
              if j=i then
                begin
                  sum:=0.0;
                  for k:=0 to rNbus-1 do
                    begin
                      //sum:=sum+((Gjk*sin(dj-dk)-Bjk*cos(dj-dk))*Uk);
                      sum:=sum+((rY[j,k].real*sin(rV[j].imag-rV[k].imag)-
                                  rY[j,k].imag*cos(rV[j].imag-rV[k].imag))*rV[k].real);
                    end;
                    //Qj:=sum*Ui;
                    Qj:=sum*rV[i].real;
                    //result[row,col]:=Qj-Bij*Ui;
                    result[row,col]:=Qj-rY[i,j].imag*sqr(rV[i].real);
                  end
                end
              begin

```

---

```

//result[row,col]:=Uj*(Gij*sin(di-dj)-Bij*cos(di-dj));
result[row,col]:=rV[i].real*rV[j].real*(rY[i,j].real*
sin(rV[i].imag-rV[j].imag)-rY[i,j].imag*
cos(rV[i].imag-rV[j].imag));
end;
end;
end;
end;
end;
end;

procedure UpdateTegangan(var rNbus:integer;
var rdS:dArr1;
var rJaq:dArr2;
var rTyp:iArr1;
var rV:CxArr1);
var i,Np,Nq:integer;
dV:double;
YE:dArr1;
begin
YE:=ElGauss(rJaq,rdS);
Np:=-1;
Nq:=rNbus-2;
for i:=0 to rNbus-1 do
begin
if rTyp[i]<>1 then
begin
inc(Np);
rV[i].imag:=rV[i].imag+YE[Np];
end;
if rTyp[i]=3 then
begin
inc(Nq);
dV:=YE[Nq]*rV[i].real;
rV[i].real:=rV[i].real+dV;
//rV[i].real:=rV[i].real+YE[Nq];
end;
end;
end;
end;

procedure NewtonRaphson(var rBus:TBusArr1;
var rBranch:TBranchArr1;
var rParamLF:TParamLF);
var i,Nbus,Nsal,Ngen:integer;
max:double;
V,Sg,SL,CxArr1;
Cap,dS:dArr1;
TypBus:iArr1;
Z,Tp,Y,Alir,Arus:CxArr2;
Lc,Tr,mJaq:dArr2;
begin
DecodeCommDataToLFData(rBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,
rBranch,Z,Tp,Lc,Tr);
DecodeData(rParamLF,Nbus,V,Sg,SL,Cap,TypBus,Z,Tp,Lc,Tr);
Ngen:=FindSumGen(Nbus,TypBus);

```

---

```

Admitansi(Nbus,Z,Tp,Lc,Tr,Cap,Y);
rParamLF.Iterasi:=0;
for i:=0 to rParamLF.MaxIterasi-1 do
begin
  dS:=MismatchDaya(Nbus,Ngen,V,Sg,SL,TypBus,Y);
  max:=MaxDataArray(dS);
  if max<=rParamLF.Toleransi then break;
  rJaq:=Jaqobian(Nbus,Ngen,V,TypBus,Y);
  UpdateTegangan(Nbus,dS,rJaq,TypBus,V);
  inc(rParamLF.Iterasi);
end;
V:=PolarToRec(V);
AliranDaya(Nbus,V,Y,Lc,Alir);
ArusBranch(Nbus,V,Lc,Y,Arus);
DayaGen(Nbus,V,SL,Y,TypBus,Sg);
DayaSlack(Nbus,Alir,TypBus,SL,Sg);
V:=RecToPolar(V);
UpdateAkhir(Nbus,Nsal,rParamLF,V,Sg,SL,Alir,Arus,rBus,rBranch);
end;

procedure NewtonRaphson(var rBus:TBusArr1;
  var rBranch:TBranchArr1;
  var rParamLF:TParamLF;
  var rJaq:dArr2);
var i,Nbus,Nsal,Ngen:integer;
  max:double;
  V,Sg,SL:CxArr1;
  Cap,dS:dArr1;
  TypBus:iArr1;
  Z,Tp,Y,Alir,Arus:CxArr2;
  Lc,Tr:dArr2;
begin
  DecodeCommDataToLFData(rBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,
  rBranch,Z,Tp,Lc,Tr);
  DecodeData(rParamLF,Nbus,V,Sg,SL,Cap,TypBus,Z,Tp,Lc,Tr);
  Ngen:=FindSumGen(Nbus,TypBus);
  Admitansi(Nbus,Z,Tp,Lc,Tr,Cap,Y);
  rParamLF.Iterasi:=0;
  for i:=0 to rParamLF.MaxIterasi-1 do
  begin
    dS:=MismatchDaya(Nbus,Ngen,V,Sg,SL,TypBus,Y);
    max:=MaxDataArray(dS);
    if max<=rParamLF.Toleransi then break;
    rJaq:=Jaqobian(Nbus,Ngen,V,TypBus,Y);
    UpdateTegangan(Nbus,dS,rJaq,TypBus,V);
    inc(rParamLF.Iterasi);
  end;
  V:=PolarToRec(V);
  AliranDaya(Nbus,V,Y,Lc,Alir);
  ArusBranch(Nbus,V,Lc,Y,Arus);
  DayaGen(Nbus,V,SL,Y,TypBus,Sg);
  DayaSlack(Nbus,Alir,TypBus,SL,Sg);
  V:=RecToPolar(V);
  UpdateAkhir(Nbus,Nsal,rParamLF,V,Sg,SL,Alir,Arus,rBus,rBranch);
end;

```

---

```

procedure NewtonRaphson(var rParamLF:TParamLF;
  var rV,rSg,rSL:CxArr1;
  var rCap:dArr1;
  var rTypBus:iArr1;
  var rZ,rTp,rAlir,rArus:CxArr2;
  var rLc,rTr:dArr2);overload;
var i,Nbus,Ngen:integer;
max:double;
dS:dArr1;
Y:CxArr2;
mJaq:dArr2;
begin
Nbus:=high(rV)+1;
DecodeData(rParamLF,Nbus,rV,rSg,rSL,rCap,rTypBus,rZ,rTp,rLc,rTr);
Ngen:=FindSumGen(Nbus,rTypBus);
Admitansi(Nbus,rZ,rTp,rLc,rTr,rCap,Y);
rParamLF.Iterasi:=0;
for i:=0 to rParamLF.MaxIterasi-1 do
begin
  dS:=MismatchDaya(Nbus,Ngen,rV,rSg,rSL,rTypBus,Y);
  max:=MaxDataArray(dS);
  if max<=rParamLF.Toleransi then break;
  mJaq:=Jaqobian(Nbus,Ngen,rV,rTypBus,Y);
  UpdateTegangan(Nbus,dS,mJaq,rTypBus,rV);
  inc(rParamLF.Iterasi);
end;
rV:=PolarToRec(rV);
AliranDaya(Nbus,rV,Y,rLc,rAlir);
ArusBranch(Nbus,rV,rLc,Y,rArus);
DayaGen(Nbus,rV,rSL,Y,rTypBus,rSg);
DayaSlack(Nbus,rAlir,rTypBus,rSL,rSg);
rV:=RecToPolar(rV);
UpdateAkhir(Nbus,rParamLF,rSg,rSL,rAlir,rArus);
end;
end.

```

---

```

unit uSensitivitas;

interface

uses uUtils,uComplex,uMatrix,uLoadflow,SysUtils;

procedure Bentuk_Sensitivitas(const rKz,rTyp:iArr1;
  const rgKz:TBatasArr1;
  const rCap:dArr1;
  const rV:CxArr1;
  const rZ,rTp:CxArr2;
  const rLc,rTr:dArr2;
  var rSu:dArr2);

implementation

Procedure Bentuk_Y_Polar(var rY:CxArr2);
var i,j,Nbus:integer;
  absY,sudY,phi:double;
begin
  phi:=4*arctan(1);
  Nbus:=high(rY)+1;
  for i:=0 to Nbus-1 do
  begin
    for j:=0 to Nbus-1 do
    begin
      if rY[i,j].imag<>0 then
      begin
        absY:=sqrt(sqrt(rY[i,j].real)+sqrt(rY[i,j].imag));
        sudY:=arctan(abs(rY[i,j].imag/rY[i,j].real));
        if (rY[i,j].real>0) and (rY[i,j].imag>0) then
        begin
          sudY:=sudY;
        end
        else if (rY[i,j].real<0) and (rY[i,j].imag>0) then
        begin
          sudY:=phi-sudY;
        end
        else if (rY[i,j].real<0) and (rY[i,j].imag<0) then
        begin
          sudY:=phi+sudY;
        end
        else if (rY[i,j].real>0) and (rY[i,j].imag<0) then
        begin
          sudY:=2*phi-sudY;
        end;
        rY[i,j]:=Cmplx(absY,sudY);
      end;
    end;
  end;
end;

Procedure Jaqobian_H(const rNbus,i,j,Ka,La:integer;
  const rV:CxArr1;
  const rY:CxArr2;

```

---

```

    var rFXA:dArr2);
var k:integer;
  sum:double;
begin
  sum:=0;
  if j<>i then
  begin
    rFXA[Ka,La]:=rV[i].real*rV[j].real*rY[i,j].real*
      Sin(rV[i].imag-rV[j].imag-rY[i,j].imag);
  end
  else
  begin
    for k:=0 to rNbus-1 do
    begin
      if k<>i then
      begin
        sum:=sum+rV[i].real*rV[k].real*rY[i,k].real*
          Sin(rV[i].imag-rV[k].imag-rY[i,k].imag);
      end
    end;
    rFXA[Ka,La]:=-sum;
  end;
end;

Procedure Jaqobian_N(const rNbus,i,j,Ka,La:integer;
  const rV:CxArr1;
  const rY:CxArr2;
  var rFXA:dArr2);
var k:integer;
  sum:double;
begin
  sum:=0;
  if j<>i then
  begin
    rFXA[Ka,La]:=rV[i].real*rY[i,j].real*
      Cos(rV[i].imag-rV[j].imag-rY[i,j].imag);
  end
  else
  begin
    for k:=0 to rNbus-1 do
    begin
      if k<>i then
      begin
        sum:=sum+rV[k].real*rY[i,k].real*
          Cos(rV[i].imag-rV[k].imag-rY[i,k].imag);
      end;
    end;
    rFXA[Ka,La]:=2*rV[i].real*rY[i,i].real*Cos(-rY[i,i].imag)+Sum;
  end;
end;

Procedure Jaqobian_J(const rNbus,i,j,Ka,La:integer;
  const rV:CxArr1;
  const rY:CxArr2;
  var rFXA:dArr2);

```

---

```

var k:integer;
  sum:double;
begin
  sum:=0;
  if j<>i then
  begin
    rFXA[Ka,La]:=(-1)*rV[i].real*rV[j].real*rY[i,j].real*
      Cos(rV[i].imag-rV[j].imag-rY[i,j].imag);
  end
  else
  begin
    for k:=0 to rNbus-1 do
    begin
      if k<>i then
      begin
        sum:=sum+rV[i].real*rV[k].real*rY[i,k].real*
          Cos(rV[i].imag-rV[k].imag-rY[i,k].imag);
      end;
    end;
    rFXA[Ka,La]:=sum;
  end;
end;

```

```

Procedure Jaqobian_L(const rNbus,i,j,Ka,La:integer;
  const rV:CxArr1;
  const rY:CxArr2;
  var rFXA:dArr2);
var k:integer;
  sum:double;
begin
  sum:=0;
  if j<>i then
  begin
    rFXA[Ka,La]:=rV[i].real*rY[i,j].real*
      Sin(rV[i].imag-rV[j].imag-rY[i,j].imag);
  end
  else
  begin
    for k:=0 to rNbus-1 do
    begin
      if k<>i then
      begin
        sum:=sum+rV[k].real*rY[i,k].real*
          Sin(rV[i].imag-rV[k].imag-rY[i,k].imag);
      end;
    end;
    rFXA[Ka,La]:=2*rV[i].real*rY[i,i].real*Sin(-rY[i,i].imag)+sum;
  end;
end;

```

---

```

Procedure Bentuk_FX(const rNbus:integer;
  const rV:CxArr1;
  const rY:CxArr2;
  const rTyp:iArr1;
  var rFXS:dArr2);

```

```
var i,j,Ka,La:integer;
begin
  Ka:=-1;
  for i:=0 to rNbus-1 do
  begin
    Ka:=Ka+1;
    La:=-1;
    for j:=0 to rNbus-1 do
    begin
      if rTyp[j]=1 then
      begin
        La:=La+1;
        if j=i then
        begin
          rFXS[Ka,La]:=-1;
        end
        else
        begin
          rFXS[Ka,La]:=0;
        end;
        La:=La+1;
        rFXS[Ka,La]:=0;
      end
      else if rTyp[j]=2 then
      begin
        La:=La+1;
        Jaqobian_H(rNbus,i,j,Ka,La,rV,rY,rFXS);
        La:=La+1;
        rFXS[Ka,La]:=0;
      end
      else if rTyp[j]=3 then
      begin
        La:=La+1;
        Jaqobian_H(rNbus,i,j,Ka,La,rV,rY,rFXS);
        La:=La+1;
        Jaqobian_N(rNbus,i,j,Ka,La,rV,rY,rFXS);
      end
      end;
      Ka:=Ka+1;
      La:=-1;
      for j:=0 to rNbus-1 do
      begin
        if rTyp[j]=1 then
        begin
          La:=La+1;
          rFXS[Ka,La]:=0;
          La:=La+1;
          if j=i then
          begin
            rFXS[Ka,La]:=-1;
          end
          else
          begin
            rFXS[Ka,La]:=0;
          end;
        end;
      end;
    end;
  end;
```

```

end
else if rTyp[j]=2 then
begin
  La:=La+1;
  Jaqobian_J(rNbus,i,j,Ka,La,rV,rY,rFXS);
  La:=La+1;
  if j=i then
  begin
    rFXS[Ka,La]:=-1;
  end
  else
  begin
    rFXS[Ka,La]:=0;
  end;
  end;
else If rTyp[j]=3 then
begin
  La:=La+1;
  Jaqobian_J(rNbus,i,j,Ka,La,rV,rY,rFXS);
  La:=La+1;
  Jaqobian_L(rNbus,i,j,Ka,La,rV,rY,rFXS);
end;
end;
end;

```

```

Procedure Bentuk_FU(const rNbus,rNu:integer;
  const rV:CxArr1;
  const rY:CxArr2;
  const rTyp,rKz:Arr1;
  var rFUS:Arr2);
var i,j,k,Ka,La:integer;
begin
  Ka:=-1;
  for i:=0 to rNbus-1 do
  begin
    Ka:=Ka+1;
    La:=-1;
    for k:=0 to rNu-1 do
    begin
      j:=rKz[k];
      if rTyp[i]=1 then
      begin
        La:=La+1;
        Jaqobian_N(rNbus,i,j,Ka,La,rV,rY,rFUS);
      end
      else if rTyp[i]=2 then
      begin
        La:=La+1;
        Jaqobian_N(rNbus,i,j,Ka,La,rV,rY,rFUS);
      end
      else if rTyp[i]=3 then
      begin
        La:=La+1;
        rFUS[Ka,La]:=0;
      end;
    end;
  end;
end;

```

---

```

    end;
end;
Ka:=Ka+1;
La:=-1;
for k:=0 to rNu-1 do
begin
  j:=rKz[k];
  if rTyp[j]=1 then
  begin
    La:=La+1;
    Jaqobian_L(rNbus,i,j,Ka,La,rV,rY,rFUS);
  end
  else if rTyp[j]=2 then
  begin
    La:=La+1;
    Jaqobian_L(rNbus,i,j,Ka,La,rV,rY,rFUS);
  end
  else if rTyp[j]=3 then
  begin
    La:=La+1;
    If j=i then
    begin
      rFUS[Ka,La]:=-1;
    end
    else
    begin
      rFUS[Ka,La]:=0;
    end;
  end;
  end;
end;

function GetAdmitansi(const rY:CxArr2):dArr2;
var i,j,rows,cols:integer;
begin
  rows:=high(rY)+1;
  cols:=high(rY[0])+1;
  SetLength(result,rows,cols);
  for i:=0 to rows-1 do
  begin
    for j:=0 to cols-1 do
    begin
      result[i,j]:=rY[i,j].imag;
    end;
  end;
end;

procedure Bentuk_Sensitivitas(const rKz,rTyp:iArr1;
  const rgKz:TBatasArr1;
  const rCap:dArr1;
  const rV:CxArr1;
  const rZ,rTp:CxArr2;
  const rLc,rTr:dArr2;
  var rSu:dArr2);

```

---

```
var Nbus,Nu,Na:integer;
  FXS,FUS:dArr2;
  Y:CxArr2;
begin
  Nbus:=high(rV)+1;
  Nu:=high(rKz)+1;
  Na:=2*Nbus;
  Admitansi(Nbus,rZ,rTp,rLc,rTr,rCap,Y);
  Bentuk_Y_Polar(Y);
  SetLength(FXS,Na,Na);
  Bentuk_FX(Nbus,rV,Y,rTyp,FXS);
  SetLength(FUS,Na,Nu);
  Bentuk_FU(Nbus,Nu,rV,Y,rTyp,rKz,FUS);
  FXS:=MatrixInvers(FXS);
  FXS:=MatrixNegative(FXS);
  rSU:=MatrixMul(FXS,FUS);
end;

end.
```

---

```

unit uTopology;

interface

uses uUtils,uComplex,uLoadflow,uMatrix,SysUtils;

procedure LoadflowTopology(var rBus:TBusArr1;
                           var rBranch:TBranchArr1;
                           var rParamLF:TParamLF);

implementation

procedure CreateVBusSLoad(var rBus:TBusArr1;
                           var rParamLF:TParamLF;
                           var rV,rSL:CxArr1);
var i,Nbus:integer;
begin
  Nbus:=high(rBus)+1;
  SetLength(rV,Nbus-1);
  SetLength(rSL,Nbus-1);
  for i:=1 to Nbus-1 do
    begin
      rV[i-1].real:=rBus[i].absV;
      rV[i-1].imag:=rBus[i].sudV;
      rSL[i-1].real:=rBus[i].PL/rParamLF.Pbase;
      rSL[i-1].imag:=rBus[i].QL/rParamLF.Pbase;
    end;
end;

function CreateZa(var rNbus:integer;
                  var rBranch:TBranchArr1;
                  var rParamLF:TParamLF):CxArr2;
var i,dari,ke:integer;
begin
  SetLength(result,rNbus,rNbus);
  rParamLF.Zbase:=sqr(rParamLF.Vbase*rParamLF.Vkonst)/
    (rParamLF.Pbase*rParamLF.Pkonst);
  rParamLF.Ibase:=(rParamLF.Vbase*rParamLF.Vkonst)/rParamLF.Zbase;
  for i:=0 to high(rBranch) do
    begin
      dari:=rBranch[i].dari-1;
      ke:=rBranch[i].ke-1;
      if rParamLF.ParamBranch=pbPu then
        begin
          result[dari,ke]:=Cmplx(rBranch[i].R,rBranch[i].X);
        end
      else if rParamLF.ParamBranch=pbOhm then
        begin
          result[dari,ke]:=Cmplx(rBranch[i].R/rParamLF.Zbase,
                                 rBranch[i].X/rParamLF.Zbase);
        end;
    end;
end;

function CreateMatrixZ(var rA:iArr2;

```

---

```

var rZ:CxArr2):CxArr2;
var i,j,Nbus:integer;
Za,mA:CxArr2;
begin
Nbus:=high(rZ)+1;
SetLength(Za,Nbus-1,Nbus-1);
for i:=0 to Nbus-1 do
begin
  for j:=0 to Nbus-1 do
  begin
    if rZ[i,j].imag<>0 then
    begin
      Za[j-1,j-1]:=Cmplx(rZ[i,j]);
    end;
  end;
end;
SetLength(mA,Nbus-1,Nbus-1);
for i:=0 to Nbus-2 do
begin
  for j:=0 to Nbus-2 do
  begin
    mA[i,j]:=Cmplx(rA[i,j],0);
  end;
end;
result:=MatrixMul(MatrixMul(MatrixTranspose(mA),Za),mA);
end;

function SaveV0Bus(var rV:CxArr1):CxArr1;
var i,Nbus:integer;
begin
Nbus:=high(rV)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
  result[i]:=Cmplx(rV[i]);
end;
end;

function CalcK(var rV,rSL:CxArr1):CxArr1;
var i,Nbus:integer;
begin
Nbus:=high(rV)+1;
SetLength(result,Nbus);
for i:=0 to Nbus-1 do
begin
  result[i]:=Conjugate(Divide(rSL[i],rV[i]));
end;
end;

function CalcdU(var rZa:CxArr2;
  var rK:CxArr1):CxArr1;
begin
  result:=MatrixMul(rZa,rK);
end;

```

---

```

function UpdateVBus(var rVs:TCmplx;
    var rdU:CxArr1):CxArr1;
var i,Nbus:integer;
begin
  Nbus:=high(rdU)+1;
  SetLength(result,Nbus);
  for i:=0 to Nbus-1 do
  begin
    result[i]:=Subtract(rVs,rdU[i]);
  end;
end;

function isConvergen(var rV0,rV:CxArr1;
    var rParamLF:TParamLF):boolean;
var i,Nbus:integer;
  dV:TCmplx;
begin
  result:=false;
  Nbus:=high(rV)+1;
  for i:=0 to Nbus-1 do
  begin
    dV:=Subtract(rV[i],rV0[i]);
    if getAbs(dV)<=rParamLF.Toleransi then
    begin
      result:=true;
      break;
    end;
  end;
end;

function CalcArus(var rA:iArr2;
    var rK:CxArr1):CxArr1;
var i,j,Nbus:integer;
  mA:CxArr2;
begin
  Nbus:=high(rK)+1;
  SetLength(mA,Nbus,Nbus);
  for i:=0 to Nbus-1 do
  begin
    for j:=0 to Nbus-1 do
    begin
      mA[i,j]:=Cmplx(rA[i,j],0);
    end;
  end;
  result:=MatrixMul(mA,rK);
end;

function CalcLossBranch(var rZ:CxArr2;
    var rI:CxArr1):CxArr1;
var i,j,Nbus:integer;
  tmp:double;
begin
  Nbus:=high(rZ)+1;
  SetLength(result,Nbus-1);
  for i:=0 to Nbus-1 do

```

---

```

begin
  for j:=0 to Nbus-1 do
    begin
      if rZ[i,j].imag<>0 then
        begin
          tmp:=sqr(getAbs(rI[j-1]));
          result[j-1]:=Multiply(rZ[i,j],tmp);
        end;
      end;
    end;
  end;

function CalcDaya(var rV,rI:CxArr1):CxArr1;
var i,Nbus:integer;
begin
  Nbus:=high(rV)+1;
  SetLength(result,Nbus);
  for i:=0 to Nbus-1 do
    begin
      result[i]:=Multiply(rV[i],Conjugate(rI[i]));
    end;
  end;

procedure UpdateBus(var rV,rSa:CxArr1;
  var rParamLF:TParamLF;
  var rBus:TBusArr1);
var i,Nbus:integer;
begin
  Nbus:=high(rBus)+1;
  rParamLF.SumLoad:=Cmplx(0.0,0.0);
  for i:=0 to NBus-1 do
    begin
      if i>0 then
        begin
          rBus[i].absV:=sqrt(sqr(rV[i-1].real)+sqr(rV[i-1].imag));
          rBus[i].sudV:=arctan(rV[i-1].imag/rV[i-1].real);
        end
      else if i=0 then
        begin
          rBus[i].Pgen:=rSa[0].real*rParamLF.Pbase;
          rBus[i].Qgen:=rSa[0].imag*rParamLF.Pbase;
        end;
      rParamLF.SumLoad.real:=rParamLF.SumLoad.real+rBus[i].PL;
      rParamLF.SumLoad.imag:=rParamLF.SumLoad.imag+rBus[i].QL;
    end;
  rParamLF.SumGen:=Multiply(Cmplx(rSa[0]),rParamLF.Pbase);
  rParamLF.SumLoss:=Subtract(rParamLF.SumGen,rParamLF.SumLoad);
end;

procedure UpdateBranch(var rdSa,rSa,rdI:CxArr1;
  var rParamLF:TParamLF;
  var rBranch:TBranchArr1);
var i,Nsa,kc:integer;
  pos:double;
begin

```

---

```

Nsal:=high(rBranch)+1;
for i:=0 to Nsal-1 do
begin
  ke:=rBranch[i].ke-1;
  rBranch[i].Sij:=Multiply(Cmplx(rSa[ke-1]),rParamLF.Pbase);
  rBranch[i].Sji:=Multiply(Negative(Subtract(rSa[ke-1],rdSa[ke-1])),
                           rParamLF.Pbase);
  rBranch[i].Aij:=Multiply(Cmplx(rdl[ke-1]),rParamLF.Ibase);
  pos:=getAbs(rdSa[ke-1])/getAbs(rSa[ke-1]);
  rBranch[i].Aji:=Multiply(Negative(Multiply(rdl[ke-1],(1-pos))),
                           rParamLF.Ibase);
end;
end;

procedure LoadflowTopology(var rBus:TBusArr1;
                           var rBranch:TBranchArr1;
                           var rParamLF:TParamLF);
var i,Nbus,Nsal:integer;
Vs:TCmplx;
A:iArr2;
V,V0,SL,K,dU,dI,dS,Sa:CxArr1;
Z,Za:CxArr2;
Cek:boolean;
begin
Nbus:=high(rBus)+1;
Nsal:=high(rBranch)+1;
if (Nbus-1)<>Nsal then
begin
  raise Exception.Create('Bukan jaringan radial');
end;
CreateVBusSLoad(rBus,rParamLF,V,SL);
Z:=CreateZa(Nbus,rBranch,rParamLF);
A:=CreateMatrixA(Z);
Za:=CreateMatrixZ(A,Z);
Cek:=false;
Vs:=Cmplx(1.0,0.0);
rParamLF.Iterasi:=0;
for i:=1 to rParamLF.MaxIterasi do
begin
  if Cek=true then
  begin
    break;
  end;
  V0:=SaveV0Bus(V);
  K:=CalcK(V,SL);
  dU:=CalcdU(Za,K);
  V:=UpdateVBus(Vs,dU);
  Cek:=isConvergen(V0,V,rParamLF);
  inc(rParamLF.Iterasi);
end;
dI:=CalcArus(A,K);
dS:=CalcLossBranch(Z,dI);
Sa:=CalcDaya(V,dI);
UpdateBus(V,Sa,rParamLF,rBus);
UpdateBranch(dS,Sa,dI,rParamLF,rBranch);

```

---

end;

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 uHasil;

interface

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

type
  TfrmHasil = class(TForm)
    TabSheet2: TTabSheet;
    fgBus: TStringGrid;
    TabSheet3: TTabSheet;
    fgBranch: TStringGrid;
    TabSheet4: 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;
    TabSheet9: TTabSheet;
    Panel1: TPanel;
    btnClose: TButton;
    btnLFAwal: TButton;
    TabSheet1: TTabSheet;
    TabSheet5: TTabSheet;
    TabSheet6: TTabSheet;
    TabSheet7: TTabSheet;
    Chart1: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    fgBus2: 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;
```

---

```
edIterasi2: TEdit;
edtTime2: TEdit;
pbGen: TProgressBar;
GroupBox2: TGroupBox;
Label6: TLabel;
Label7: TLabel;
Label13: TLabel;
Label14: TLabel;
Label18: TLabel;
edtMaxGen: TEdit;
edtPopSize: TEdit;
edtPCross: TEdit;
edtPMutasi: TEdit;
edtLength: TEdit;
GroupBox3: TGroupBox;
TabSheet8: TTabSheet;
PageControl2: TPageControl;
TabSheet10: TTabSheet;
Label28: TLabel;
fgHasilSPEA: TStringGrid;
TabSheet11: TTabSheet;
Chart2: TChart;
Series3: TLineSeries;
Series4: TLineSeries;
Series5: TLineSeries;
btnHitung: TButton;
btnUseDefault: TButton;
Label15: TLabel;
edtCountCap: TEdit;
Label19: TLabel;
edtJmlCapBank: TEdit;
Label20: TLabel;
edtKap: TEdit;
Label21: TLabel;
Label22: TLabel;
edtKc: TEdit;
Label23: TLabel;
Label24: TLabel;
edtKa: TEdit;
Label25: TLabel;
edtLamda: TEdit;
GroupBox4: TGroupBox;
GroupBox5: TGroupBox;
Label16: TLabel;
edtKp: TEdit;
Label17: TLabel;
Label26: TLabel;
edtR: TEdit;
edtCostLoss1: TEdit;
Label27: TLabel;
Label29: TLabel;
edtCostLoss2: TEdit;
Label30: TLabel;
edtCostCap: TEdit;
procedure btnCloseClick(Sender: TObject);
```

---

```
procedure btnLFAwalClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnHitungClick(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmHasil: TfrmHasil;

implementation

uses uUtils, uLoadflow, uNewtonRaphson, uComplex, uTopology, uFitness,
  uGenetic, uVarGenetic, uObjFunc, uSPEA, uObjFuncOps;

{$R *.dfm}

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

procedure TfrmHasil.btnLFAwalClick(Sender: TObject);
var i:integer;
  mulai,selesai,selang:TDateTime;
  jam,menit,detik,mdetik:word;
begin
  mulai:=time;
  NewtonRaphson(gBus,gBranch,gParamLF);
  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(gBus[i].absV,5);
    fgBus.Cells[2,i+1]:=RealToStr(gBus[i].sudV*
      ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
    fgBus.Cells[3,i+1]:=RealToStr(gBus[i].Pgen,3);
    fgBus.Cells[4,i+1]:=RealToStr(gBus[i].Qgen,3);
    fgBus.Cells[5,i+1]:=RealToStr(gBus[i].PL,3);
    fgBus.Cells[6,i+1]:=RealToStr(gBus[i].QL,3);
    fgBus.Cells[7,i+1]:=RealToStr(gBus[i].Cap,3);
    fgBus.Cells[8,i+1]:=IntToStr(gBus[i].typeBus);
    Series1.Add(gBus[i].absV,IntToStr(i+1));
  end;
  for i:=0 to high(gBranch) do
  begin
    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(gBranch[i].Sij.real,3);
fgBranch.Cells[4,i+1]:=RealToStr(gBranch[i].Sij.imag,3);
fgBranch.Cells[5,i+1]:=RealToStr(gBranch[i].Aij.real,3);
fgBranch.Cells[6,i+1]:=RealToStr(gBranch[i].Aij.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(gBranch[i].Sji.real,3);
fgBranch.Cells[10,i+1]:=RealToStr(gBranch[i].Sji.imag,3);
fgBranch.Cells[11,i+1]:=RealToStr(gBranch[i].Aji.real,3);
fgBranch.Cells[12,i+1]:=RealToStr(gBranch[i].Aji.imag,3);
end;
edtSumGen.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi.Text:=IntToStr(gParamLF.Iterasi);
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
  IntToStr(detik)+':'+IntToStr(mdetik);
btnUseDefault.Enabled:=true;
end;

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

procedure TfrmHasil.btnHitungClick(Sender: TObject);
var BatasCap:TBus;
  Ncap,sa,r,i,j,param,length,maxgen,popsize,CountCap:integer;
  pcross,pmutasi,Kp,Kc,Ka,CapBank,Lamda,CostCap:double;
  Min,Avg,Max,Cap:dArr1;
  BestChrom:iArr1;
  Nbus,Nsal,ia,ja:integer;
  V,Sg,SL:CxArr1;
  Lc,Tr,HasilCap:dArr2;
  TypBus:iArr1;
  Z,Tp,Alir,Arus:CxArr2;
  mulai,selesai,selang:TDateTime;
  jam,menit,detik,mdetik:word;
  ELCSebelum,ELCSesudah,CostLoss1,CostLoss2:double;
  ga:TGenInt1;
  LBus:TBusArr1;
  newparent:TNewParent;
  typcross:TTypeCross;
begin
  Kp:=StrToFloat(edtKp.Text);
  Kc:=StrToFloat(edtKc.Text);
  Ka:=StrToFloat(edtKa.Text);
  r:=StrToInt(edtR.Text);
  Lamda:=StrToFloat(edtLamda.Text);
  CountCap:=StrToInt(edtCountCap.Text);
  Ncap:=StrToInt(edtJmlCapBank.Text);
  CapBank:=StrToFloat(edtKap.Text);

```

---

```

SetLength(Cap,Ncap);
for i:=0 to Ncap-1 do
begin
  Cap[i]:=(i+1)*CapBank;
end;
gObjFuncOps:=TObjFuncOps.Create(Kp,Kc,Ka,Lamda,r,CountCap,Cap);
CostLoss1:=gObjFuncOps.HitungObjFuncAwal;
edtCostLoss1.Text:=FormatFloat('#,##0.00',CostLoss1);
PopSize:=StrToInt(edtPopSize.Text);
MaxGen:=StrToInt(edtMaxGen.Text);
pbGen.Max:=MaxGen;
Length:=StrToInt(edtLength.Text);
Pcross:=StrToFloat(edtPcross.Text);
Pmutasi:=StrToFloat(edtPmutasi.Text);
newparent:=npReplikasi;
typcross:=crTwo;
ga:=TGenInt1.Create(MaxGen,PopSize,Length,Pcross,Pmutasi,0.5,Ka,newparent,
typcross);
BestChrom:=ga.BestChrom;
Min:=ga.Min;
Avg:=ga.Avg;
Max:=ga.Max;
gObjFuncOps.HitungObjFuncAkhir(BestChrom,LBus,CostCap,CostLoss2);
edtCostLoss2.Text:=FormatFloat('#,##0.00',CostLoss2);
edtCostCap.Text:=FormatFloat('#,##0.00',CostCap);
sa:=0;
for i:=0 to high(gBus) do
begin
  if BestChrom[i]<>0 then
  begin
    inc(sa);
  end;
end;
fgHasilSPEA.RowCount:=sa+1;
sa:=0;
for i:=0 to high(gBus) do
begin
  if BestChrom[i]<>0 then
  begin
    fgHasilSPEA.Cells[0,sa+1]:=IntToStr(sa+1);
    fgHasilSPEA.Cells[1,sa+1]:=IntToStr(i+1);
    fgHasilSPEA.Cells[2,sa+1]:=RealToStr(BestChrom[i]*CapBank,0);
    inc(sa);
  end;
end;
Series3.Clear;
Series4.Clear;
Series5.Clear;
for i:=0 to high(Min) do
begin
  Series3.Add(Min[i],IntToStr(i+1));
  Series4.Add(Avg[i],IntToStr(i+1));
  Series5.Add(Max[i],IntToStr(i+1));
end;
//
```

---

```

DecodeCommDataToLFData(LBus,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;
Series2.Clear;
for i:=0 to high(gBus) do
begin
  fgBus2.Cells[0,i+1]:=IntToStr(i+1);
  fgBus2.Cells[1,i+1]:=RealToStr(V[i].real,5);
  fgBus2.Cells[2,i+1]:=RealToStr(V[i].imag*
    ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
  fgBus2.Cells[3,i+1]:=RealToStr(Sg[i].real,3);
  fgBus2.Cells[4,i+1]:=RealToStr(Sg[i].imag,3);
  fgBus2.Cells[5,i+1]:=RealToStr(SL[i].real,3);
  fgBus2.Cells[6,i+1]:=RealToStr(SL[i].imag,3);
  fgBus2.Cells[7,i+1]:=RealToStr(Cap[i],3);
  fgBus2.Cells[8,i+1]:=IntToStr(TypBus[i]);
  Series2.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;
  fgBranch2.Cells[0,i+1]:=IntToStr(i+1);
  fgBranch2.Cells[1,i+1]:=IntToStr(gBranch[i].dari);
  fgBranch2.Cells[2,i+1]:=IntToStr(gBranch[i].ke);
  fgBranch2.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
  fgBranch2.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
  fgBranch2.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
  fgBranch2.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
  fgBranch2.Cells[7,i+1]:=IntToStr(gBranch[i].ke);
  fgBranch2.Cells[8,i+1]:=IntToStr(gBranch[i].dari);
  fgBranch2.Cells[9,i+1]:=RealToStr(Alir[ja,ia].real,3);
  fgBranch2.Cells[10,i+1]:=RealToStr(Alir[ja,ia].imag,3);
  fgBranch2.Cells[11,i+1]:=RealToStr(Arus[ja,ia].real,3);
  fgBranch2.Cells[12,i+1]:=RealToStr(Arus[ja,ia].imag,3);
end;
edtSumGen2.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad2.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss2.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi2.Text:='3';//IntToStr(gParamLF.Iterasi);
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime2.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
  IntToStr(detik)+':'+IntToStr(mdetik);
//
ga.Free;
gObjFuncOps.Free;
end;

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
  //input parameter Genetic Algorithm

```

```
edtMaxGen.Text:='100';
edtPopSize.Text:='10';
edtPCross.Text:='0.75';
edtPMutasi.Text:='0.004';
edtLength.Text:=IntToStr(high(gBus)+1);
//input parameter objective function
edtCountCap.Text:='4';
edtJmlCapBank.Text:='4';
edtKap.Text:='300';
edtKc.Text:='0.5';
edtKp.Text:='120';
edtKa.Text:='100000';
edtLamda.Text:='100000';
edtR.Text:='1';
btnHitung.Enabled:=true;
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)';
    //
  end;
end;
```

```

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';
  //
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';
  //
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';
  //
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, jpeg;

type
  TfrmMenu = class(TForm)
    Panel1: TPanel;
    btnNew: TButton;
    btnOpen: TButton;
    btnExit: TButton;
    StatusBar1: TStatusBar;
    OpenDialog1: TOpenDialog;
    Image1: TImage;
    StaticText1: TStaticText;
    StaticText3: TStaticText;
    StaticText2: TStaticText;
    StaticText4: TStaticText;
    StaticText5: TStaticText;
    StaticText6: TStaticText;
    StaticText7: TStaticText;
    procedure btnExitClick(Sender: TObject);
    procedure btnNewClick(Sender: TObject);
    procedure btnOpenClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmMenu: TfrmMenu;

implementation

uses uInputLFChild,uComplex, uUtils, uLoadflow;

{$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';
  end;

```

```

frmInput.btnExit.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: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
    end;
  end;

```

---

```

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)';
  frmInput.fgBranch.Cells[10,0]:='P (W)';
  frmInput.fgBranch.Cells[11,0]:='Q (VAR)';
  frmInput.fgBranch.Cells[16,0]:='P (W)';
  frmInput.fgBranch.Cells[17,0]:='Q (VAR)';
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)';
  frmInput.fgBranch.Cells[10,0]:='P (kW)';
  frmInput.fgBranch.Cells[11,0]:='Q (kVAR)';
  frmInput.fgBranch.Cells[16,0]:='P (kW)';
  frmInput.fgBranch.Cells[17,0]:='Q (kVAR)';
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)';
  frmInput.fgBranch.Cells[10,0]:='P (MW)';
  frmInput.fgBranch.Cells[11,0]:='Q (MVAR)';
  frmInput.fgBranch.Cells[16,0]:='P (MW)';
  frmInput.fgBranch.Cells[17,0]:='Q (MVAR)';
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].CapSal:=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.btnExit.Caption:='&Next';
frmInput.ShowModal;
finally
  frmInput.Free;
end;
end;
except
  MessageDlg('File Corrupt atau Error Program!',mtWarning,[mbOK],0);
end;
end;
end.

```

---