

# **SKRIPSI**

**PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK  
MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA  
JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI  
PENYULANG TEJAKULA BALI**



**Disusun Oleh:  
REFI MULANDA  
01.12.001**

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

**SEPTEMBER 2007**

---

**LEMBAR PERSETUJUAN**

**PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK  
MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA  
JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI  
PENYULANG TEJAKULA BALI**

**SKRIPSI**

*Disusun Guna Melengkapi dan Memenuhi Syarat-Syarat  
Mencapai Gelar Sarjana Teknik Elektro  
Konsentrasi Teknik Energi listrik S-1*

**Disusun Oleh :  
REFI MULANDA  
NIM. 01.12.001**

**Diperiksa dan disetujui,**

**Dosen Pembimbing I,**



**Ir. H. Taufik Hidayat, MT**  
NIP.P. 101 870 0151

**Dosen pembimbing II,**



**Irrine Budi S., ST, MT**  
NIP. 132 314 400

**Mengetahui,**

**Ketua Jurusan Teknik Elektro S-1**  
  
**Ir. Yudi Limpraptono, MT**  
NIP.Y. 103 950 0274

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



**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**

**Nama** : Refi Mulanda  
**Nim** : 01.12.001  
**Jurusan** : Teknik Elektro S-1  
**Konsentrasi** : Teknik Energi Listrik  
**Judul Skripsi** : PENGGUNAAN METODE KRUSKAL EVOLUTIONARY  
UNTUK MENENTUKAN LETAK DAN KAPASITAS  
KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER  
TIPE RADIAL DI PENYULANG TEJAKULA BALI

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

**Hari** : Selasa  
**Tanggal** : 04 September 2007  
**Dengan Nilai** : 72,4 ( B+ ) *By*

**Panitia Ujian Skripsi,**

**Ketua,**

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

**Sekretaris,**

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

**Anggota Penguji**

**Penguji I,**

Ir. M Abdul Hamid, MT  
NIP.Y. 101 880 0188

**Penguji II,**

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

## ABSTRAKSI

**Penggunaan Metode *Kruskal Evolutionary* untuk Menentukan Letak dan Kapasitas Kapasitor pada Jaringan Distribusi Primer Tipe Radial di Penyulang Tejakula Bali**, Refi Mulanda/ 01.12.082, Teknik Elektro S-1, Konsentrasi Teknik Energi Listrik, Dosen Pembimbing: I. Ir. H. Taufik Hidayat, ST, MT, Dosen Pembimbing II. Irine Budi S.ST, MT

Bertambahnya industri-industri menyebabkan peran penggunaan alat-alat listrik akan semakin luas. 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. Salah satu cara pemasangan sumber daya reaktif tambahan di sisi beban adalah dengan pemasangan kapasitor shunt pada jaringan distribusi primer, sehingga dapat mengurangi rugi-rugi saluran. Yang menjadi permasalahan pada penelitian ini adalah bagaimana cara untuk menentukan letak dan kapasitas kapasitor yang optimal pada jaringan distribusi primer tipe radial khususnya di Penyulang Tejakula Bali, bagaimana memperbaiki kondisi tegangan dan mengurangi rugi-rugi daya setelah pemasangan Kapasitor Shunt sehingga didapat keuntungan dari segi teknis maupun ekonomis. Sedangkan yang menjadi tujuan dalam penelitian ini adalah untuk mengetahui bagaimana cara menentukan letak dan kapasitas kapasitor yang optimal pada jaringan distribusi primer tipe radial di Penyulang Tejakula Bali, memperbaiki kondisi tegangan dan mengurangi rugi-rugi daya setelah pemasangan Kapasitor Shunt sehingga didapat keuntungan dari segi teknis maupun ekonomis.

Dalam penelitian ini menggunakan Metode *Kruskal Evolutionary* dimana metode ini memiliki kelebihan yaitu dapat menentukan letak, jumlah dan kapasitas kapasitor berdasarkan dua fungsi sasaran yang akan dicapai yaitu untuk menekan rugi-rugi saluran dan menekan biaya operasional kapasitor tersebut. Sedangkan untuk analisa aliran daya disini menggunakan Metode *Newton Raphson*.

Dengan menggunakan data jaringan sistem distribusi 20 kv pada Penyulang Tejakula Bali berdasarkan perhitungan diperoleh hasil penempatan dan kapasitas kapasitor dengan menggunakan *Kruskal Evolutionary* yaitu pada bus 27 dengan kapasitas 150 kVAR, dan rugi-rugi daya mengalami penurunan yaitu untuk daya aktif sebesar 13,937 kW dari 59.590 menjadi 45,653 kW. Sedangkan untuk daya reaktif sebesar 21,306 kVAR dari 91,094 kVAR menjadi 69,788 kVAR serta dapat memperbaiki kondisi tegangan dari nilai terendah 0,952 pu menjadi 0,978 pu, dengan kata lain terjadi perbaikan tegangan dengan faktor kenaikan sebesar 2,73%. Penentuan letak dan kapasitas kapasitor pada jaringan distribusi primer tipe radial dengan menggunakan metode *Kruskal Evolutionary* perlu dikembangkan dan diaplikasikan dalam menganalisa saluran-saluran yang telah ada maupun untuk perencanaan perluasan jaringan.

Kata kunci : Penempatan kapasitor, Sistem Distribusi Radial, *Metode Kruskal Evolutionary*

---

## KATA PENGANTAR

Dengan segala kerendahan hati, penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena hanya dengan lindungan, rahmat dan karuniaNya-lah penulis dapat menyelesaikan penyusunan skripsi ini, sebagai syarat untuk melengkapi dan memenuhi syarat mencapai gelar sarjana.

Skripsi yang berjudul **“PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG TEJAKULA BALI”**

ini tersusun juga atas bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. I Made Wartana, MT selaku Pembantu Dekan I Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro Energi Listrik S-1 Institut Teknologi Nasional Malang.
4. Ir. Yusuf Ismail Nahkoda, MT selaku Sekretaris Jurusan Teknik Elektro (S1) ITN Malang.
5. Bapak Ir. H. Taufik Hidayat, MT, selaku Dosen Pembimbing I yang telah meluangkan waktunya untuk membimbing penulis.
6. Ibu Irrine Budi S ., ST . MT, selaku Pembimbing II yang telah meluangkan waktunya untuk membimbing penulis.

7. Bapak Ir. M Abdul Hamid, MT, selaku penguji skripsi I Elektro Energi Listrik S-1 Institut Teknologi Nasional Malang.
8. Bapak Bambang Prio Hartono., ST. MT, selaku penguji skripsi II Elektro Energi Listrik S-1 Institut Teknologi Nasional Malang.
9. Seluruh Dosen dan Karyawan yang secara langsung maupun tidak langsung telah membantu terselesaikannya penyusunan skripsi ini.
10. Ayah dan Ibu yang selalu memberikan dukungan lahir maupun batin kepada penulis.
11. Rosita widyastutik sebagai istri, yang selalu memberikan dukungan dan suport.
12. Semua pihak yang tidak dapat saya sebutkan satu per satu, yang telah membantu dalam penyelesaian skripsi ini.

Akhirnya, sebagai puncak dari tujuan penulisan skripsi ini adalah semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, September 2007

Penulis

## DAFTAR ISI

<b>BERITA ACARA</b> .....	i
<b>ABSTRAKSI</b> .....	ii
<b>KATA PENGHANTAR</b> .....	iii
<b>DAFTAR ISI</b> .....	v
<b>DAFTAR GAMBAR</b> .....	vii
<b>DAFTAR TABEL</b> .....	viii
<b>DAFTAR GRAFIK</b> .....	ix
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Pembahasan .....	2
1.4 Batasan Masalah .....	3
1.5 Metodologi Pembahasan .....	3
1.6 Sistematika Penulisan .....	4
1.7 Relevansi .....	5
<b>BAB II LANDASAN TEORI</b> .....	6
2.1 Sistem Distribusi Tenaga Listrik .....	6
2.2 Struktur Jaringan Distribusi Tenaga Listrik .....	7
2.3 Sistem Jaringan Distribusi Radial .....	7
2.3.1. Sistem Radial Pohon .....	8
2.3.2. Sistem Radial dengan Tie dan Switch Pemisah .....	9
2.3.3. Sistem Radial dengan Pembagian Phase Area .....	10
2.3.4. Sistem Radial dengan Beban Terpusat .....	10
2.4 Faktor Daya .....	11
2.5 Kapasitor Daya .....	13
2.5.1. Kapasitor Seri dan Kapasitor Shunt .....	13
2.6 Pengaruh Pemasangan Kapasitor Shunt .....	13
2.6.1. Pengurangan Rugi Daya dengan Kapasitor Shunt .....	14
2.6.2. Perbaikan Tegangan .....	16
2.6.3. Perbaikan Faktor Daya dan Kenaikan Kapasitas Sistem .....	17
2.6.3.1. Perhitungan Pengaruh Perbaikan Faktor Daya .....	18

2.6.3.2. Penentuan Rating Kapasitor untuk Perbaikan Faktor	
Daya Beban .....	19
2.7 Sistem Per-unit .....	20
<b>BAB III IMPLEMENTASI METODE KRUSKAL EVOLUTIONARY</b>	
<b>UNTUK PENENTUAN LETAK KAPASITOR PADA JARINGAN</b>	
<b>DISTRIBUSI 20 KV.....</b>	22
3.1 Analisa Aliran Daya Jaringan.....	22
3.1.1. Metode Newton Raphson.....	22
3.1.2. Algoritma Aliran Daya Newton Raphson.....	26
3.2 .Metode Kruskal Evolutionary .....	27
3.2.1. Algoritma Kruskal Evolutionary Programming .....	27
3.2.2.Metode Evolutionary Programming.....	30
3.2.3 Objective Function Kruskal Evolutionary Programming.....	32
3.2.4. Flow chart Penyelesaian Masalah.....	33
3.3. Validasi Perogram .....	34
3.4. Total Biaya Sebelum Dan Setelah Pemasangan Kapasitor.....	36
3.4.1. Biaya Daya ( $K_p$ ).....	36
<b>BAB IV ANALISA KAPASITAS DAN PENEMPATAN KAPASITOR</b>	
<b>PADA JARINGAN DISTRIBUSI PRIMER 20 KV PENYULANG</b>	
<b>TEJAKULA, BALI .....</b>	37
4.1. Analisis Penentuan Letak Kapasitor Dengan Dengan Menggunakan	
Metode Kruskal Evolutionary .....	37
4.2. Konsep Analisa Perhitugn .....	38
4.3. Data Saluran .....	39
4.4. Hasil Perhitungan Program Kruskal Evolutionary .....	40
4.5. Hasil Analisa Program Dengan Menggunakan Metode Kruskal	
Evolurtionary .....	44
<b>BAB V KESIMPULAN DAN SARAN .....</b>	61
5.1 Kesimpulan .....	61
5.2 Saran.....	61
<b>DAFTAR PUSTAKA.....</b>	x
<b>LAMPIRAN</b>	



## DAFTAR GAMBAR

Gambar :	Halaman :
2-1 Jaringan Distribusi Tegangan Menengah (JTM), Jaringan Tegangan Rendah (JTR) dan Sambungan Rumah ke Pelanggan .....	6
2-2 Jaringan Distribusi Radial .....	8
2-3 Jaringan Distribusi Radial dengan Tie dan Switch Pemisah .....	9
2-4 Jaringan Distribusi Radial dengan Phase Area .....	10
2-5 Jaringan Distribusi dengan Beban Terpusat .....	11
2-6 Segitiga Daya .....	11
2-7 Segitiga Arus .....	12
2-8 Saluran Primer dengan Beban Terpusat .....	15
2-9 Bagian Satu Garis dan Fasor Diagram Suatu Penyulang dengan Faktor Daya Tertinggal .....	17
2-10 Perbaikan Faktor Daya .....	17
2-11 Diagram Fasor dan Sudut Daya Beban Distribusi .....	18
3-1 Flow Chart Kruskal Evolutionary .....	29
3-2 Flow Chart Pemecahan Masalah.....	33
4-1 Form Utama .....	40
4-2 Form Data Bus dan Saluran .....	41
4-3 Form Data Kapasitor dan Set Parameter.....	41
4-4 Form Load Flow Awal .....	42
4-5 Form Load Flow Akhir .....	43
4-6 Penempatan Kapasitas Kapasitor: Bus ke 27, 150 kVAR .....	45
4-7 Single Diagram Penyulang Tejakula Sebelum Dipasang Kapasitor. 59	
4-8 Single Diagram Penyulang Tejakula Sesudah Dipasang Kapasitor.. 60	

## DAFTAR TABEL

Tabel :	Halaman :
3-1 Data Impedansi Saluran .....	34
3-2 Data Pembebanan.....	34
3-3 Perbandingan Hasil Program Dengan Hasil Pada Jurnal .....	35
4-1 Data saluran Sistem 20 kV Penyulang Tejakula.....	39
4-2 Data Saluran Penyulang Tejakula.....	39
4-3 Load Flow Awal Sebelum Pemasangan Kapasitor .....	46
4-4 Load Flow Akhir Setelah Pemasangan Kapasitor.....	48
4-5 Rugi Daya Tiap Saluran Sebelum Kompensasi .....	50
4-6 Rugi Daya Tiap Saluran Sesudah Kompensasi.....	51
4-7 Aliran Daya Sebelum Pemasangan Kapasitor .....	53
4-8 Aliran Daya Sesudah Pemasangan Kapasitor.....	56

## DAFTAR GRAFIK

Gambar :	Halaman :
4-1 Tegangan Tiap Tiap Bus Terhadap Asumsi Tegangan Awal, Tegangan Sebelum Optimasi .....	47
4-2 Tegangan Tiap Tiap Bus Terhadap Asumsi Tegangan Awal, Tegangan Sesudah Optimasi.....	49
4-3 Rugi-rugi Saluran Sebelum dan Sesudah Kompensasi.....	52
4-4 Aliran Daya Sebelum Pemasangan Kapasitor .....	55
4-5 Aliran Daya Sesudah Pemasangan Kapasitor.....	58

# BAB 1

## PENDAHULUAN

### 1.1. LATAR BELAKANG

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.

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.

Banyak metode yang dipakai dalam menganalisa masalah penentuan penempatan dan kapasitas kapasitor pada jaringan distribusi, pada umumnya banyak teknik untuk membahas masalah alokasi kapasitor dengan menggunakan metode-metode konvensional tetapi pembahasan-pembahasan yang dilakukan masih kurang dan masih perlu dilakukan analisa yang lebih mendalam lagi.

Hal ini mendorong penulis untuk mengangkat permasalahan dengan menggunakan metode *Kruskal Evolutionary* agar dapat menjadi salah satu alternatif yang efektif untuk penempatan dan kapasitas kapasitor dalam jaringan distribusi primer.

Dalam skripsi ini mengemukakan suatu pendekatan terhadap penempatan *shunt capacitor* dalam sistem distribusi radial dengan menggunakan teknik pemecahan berdasarkan metode Kruskal Evolutionary. Algoritma tersebut memiliki kelebihan yaitu dapat menentukan letak, jumlah dan kapasitas kapasitor berdasarkan dua fungsi sasaran yang akan dicapai yaitu untuk menekan rugi-rugi saluran dan menekan biaya operasional kapasitor tersebut.[4]

## 1.2. RUMUSAN MASALAH

Adapun yang menjadi permasalahan dalam penelitian ini adalah:

1. Bagaimana cara untuk menentukan letak dan kapasitas kapasitor yang optimal pada jaringan distribusi primer tipe radial, penyulang Tejakula Bali dengan menggunakan *Metode Kruskal Evolutionary*?
2. Bagaimana pemasangan Kapasitor Shunt pada jaringan distribusi primer untuk memperbaiki kondisi tegangan?
3. Bagaimana pemasangan Kapasitor Shunt untuk mengurangi rugi-rugi daya sehingga didapat keuntungan dari segi teknis maupun ekonomis?

## 1.3. TUJUAN PEMBAHASAN

Menentukan letak kapasitas kapasitor yang optimal pada jaringan distribusi primer tipe radial penyulang Tejakula Bali dengan menggunakan *Metode Kruskal Evolutionary*.

---

#### 1.4. BATASAN MASALAH

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

1. Algoritma yang digunakan adalah metode Kruskal Evolutionary untuk mengevaluasi lokasi penempatan kapasitor-kapasitor shunt.
2. Jaringan yang akan dianalisa adalah jaringan distribusi primer type radial 20 kV Penyulang Tejekula.
3. Biaya operasi dan pemeliharaan tidak diperhitungkan.
4. Tidak membahas secara detail metode aliran daya yang digunakan yaitu metode Newton Raphson.
5. Implementasi dari analisa menggunakan perangkat lunak Borland Delphi 7.0.
6. Tidak membahas segala sesuatu yang menyangkut sintaksis dari perancangan perangkat lunak.

#### 1.5. METODOLOGI PEMBAHASAN

Metode yang digunakan dalam pembahasan skripsi ini dilakukan dengan langkah-langkah :

1. Studi Literatur

Yaitu suatu kajian pustaka dengan mempelajari teori-teori yang terkait melalui literatur yang ada, yang berhubungan dengan permasalahan.

2. Pengumpulan data lapangan yang dipakai dalam objek penelitian yaitu:

- a. Data impedansi saluran distribusi penyulang Tejakula.
-

- b. Data pembebanan tiap transformator distribusi.
3. Melakukan analisa dengan bantuan perangkat lunak komputer dengan bahasa pemrograman Delphi Versi 6.0.
    - a. Analisa aliran daya menggunakan metode Newton Raphson.
    - b. Analisa penentuan lokasi dan kapasitas kapasitor shunt menggunakan Metode Kruskal Evolutionary.
    - c. Analisa pengaruh pemasangan kapasitor terhadap sistem diantaranya perbaikan profil tegangan, pengurangan rugi-rugi saluran dan keuntungan yang diperoleh.
    - d. Validasi Softwer
    - e. Kesimpulan

## **1.6. SISTEMATIKA PENULISAN**

Adapun sistematika penulisan pada skripsi ini adalah sebagai berikut :

### **BAB 1 : PENDAHULUAN**

Pada bab ini berisi latar belakang, rumusan masalah,tujuan pembahasan, batasan masalah, metodologi penulisan, sistematika penulisan dan relevansi.

---

- BAB II : LANDASAN TEORI**  
Di sini akan diuraikan mengenai sistem jaringan distribusi, faktordaya, penjelasan teori tentang kapasitor, perbaikan tegangan dan sistem per unit dalam jaringan
- BAB III : ANALISIS PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI 20 KV MENGGUNAKAN KRUSKAL EVOLUTIONARY**  
Menguraikan studi aliran daya Newton Raphson, Algoritma Kruskal Evolutionary sebagai Algoritma Program Pemecahan Masalah.
- BAB IV : SIMULASI DAN ANALISA HASIL**  
Bab ini akan dibahas mengenai penentuan letak dan kapasitas kapasitor, analisa perhitungan dengan menggunakan Metode Kruskal Evolutionary, profil tegangan dan aliran daya sebelum dan setelah kompensasi.
- BAB V : KESIMPULAN**  
Pada bab ini terdapat intisari dari hasil pembahasan, yang berisi kesimpulan yang dapat dipergunakan sebagai bahan pertimbangan untuk pengembangan dan penulisan selanjutnya.

### 1.7. Relevansi

Dengan diperkenalkannya Metode Kruskal Evolutionary Programming, maka hal ini akan menambah alternatif pilihan terhadap metode yang dapat digunakan dalam penyelesaian masalah penempatan kapasitor. Dimana metode tersebut merupakan metode dengan proses penyelesaian yang sangat baik.

---





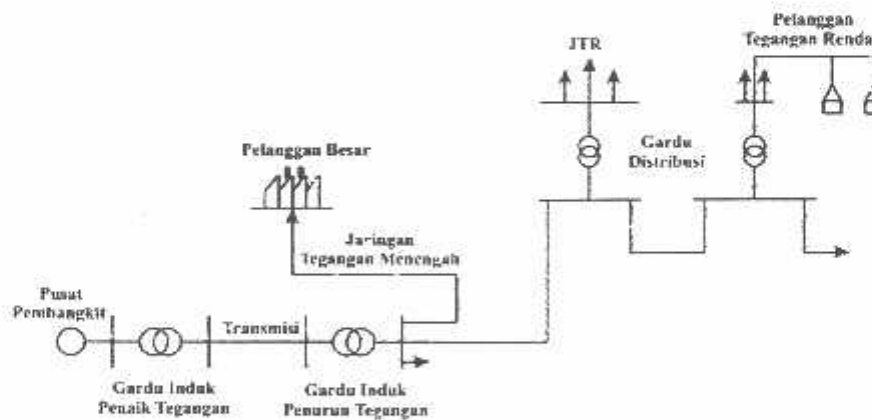
## BAB II

### LANDASAN TEORI

#### 2.1. Sistem Distribusi Tenaga Listrik

Di Indonesia, tenaga listrik dibangkitkan di pusat-pusat pembangkit tenaga listrik, seperti PLTA, PLTU, PLTG, PLTGU, PLTP dan PLID yang kemudian disalurkan melalui saluran transmisi setelah terlebih dahulu dinaikkan tegangannya oleh transformator penaik tegangan (*step up transformer*) yang ada pada pusat listrik. Saluran transmisi tegangan tinggi di PLN mempunyai tegangan 150 kV dan 500 kV, selanjutnya tegangan tinggi tersebut diturunkan menjadi tegangan menengah sebesar 20 kV di Gardu Induk, dengan menggunakan transformator penurun tegangan (*step down transformer*), setelah melalui jaringan distribusi, tegangan diturunkan lagi pada gardu-gardu distribusi menjadi tegangan rendah sebesar 380/220 V untuk selanjutnya disalurkan ke pelanggan-pelanggan PLN.

Sebagai gambaran, diagram satu garis sistem tenaga listrik dapat diperjelas seperti pada gambar 2.1 di bawah ini :



Gambar 2-1

Diagram Satu Garis Sistem Penyaluran Energi Listrik Ke Pelanggan<sup>[1]</sup>

Jaringan setelah keluar dari gardu induk 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.

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

Sistem jaringan yang digunakan untuk menyalurkan dan mendistribusikan tenaga listrik tersebut dapat menggunakan sistem satu fasa dengan dua kawat maupun sistem tiga fasa dengan empat kawat.

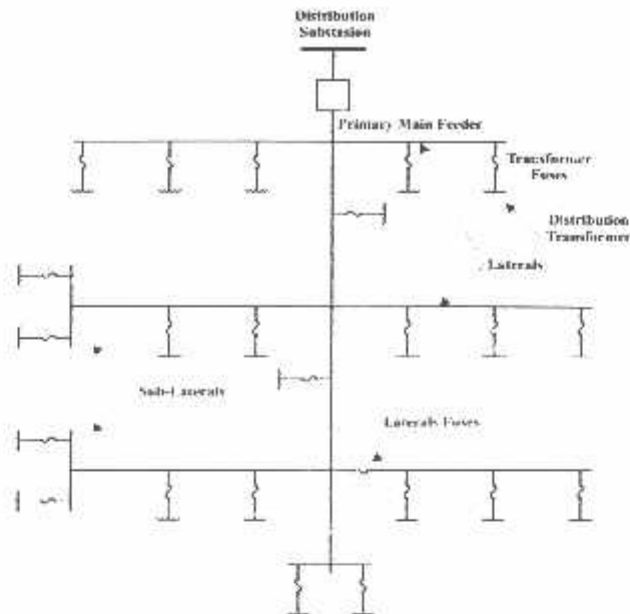
## **2.2. Struktur Jaringan Distribusi Tenaga Listrik**

Ada beberapa bentuk jaringan yang umum dipergunakan untuk menyalurkan dan mendistribusikan tenaga listrik yaitu :

1. Sistem jaringan distribusi radial.
3. Sistem jaringan distribusi rangkaian tertutup (*loop*)
4. Sistem jaringan distribusi *mesh*.

## **2.3. Sistem Jaringan Distribusi Radial**

Bentuk jaringan ini merupakan bentuk dasar yang paling banyak digunakan dan yang paling sederhana. Sistem ini dikatakan radial karena dari kenyataan bahwa jaringan ini ditarik secara radial dari gardu induk ke pusat-pusat beban atau konsumen yang dilayaninya. Sistem ini terdiri dari saluran utama (*main feeder*) dan saluran cabang (*lateral*) seperti pada gambar 2-2.



Gambar 2-2

### Contoh Sistem Jaringan Distribusi Radial <sup>[2]</sup>

Pelayanan tenaga listrik untuk suatu daerah beban tertentu dilaksanakan dengan memasang transformator disebarkan titik pada jaringan yang sedekat mungkin dengan daerah beban yang dilayani. Untuk daerah beban yang menyimpang jauh dari saluran utama maupun saluran cabang, maka akan ditarik lagi saluran tambahan yang dicabangkan pada saluran tersebut.

Kelemahan yang dimiliki oleh sistem radial adalah jatuh tegangan yang cukup besar dan bila terjadi gangguan pada sistem akan mengakibatkan jatuhnya sebagian atau bahkan keseluruhan beban sistem.

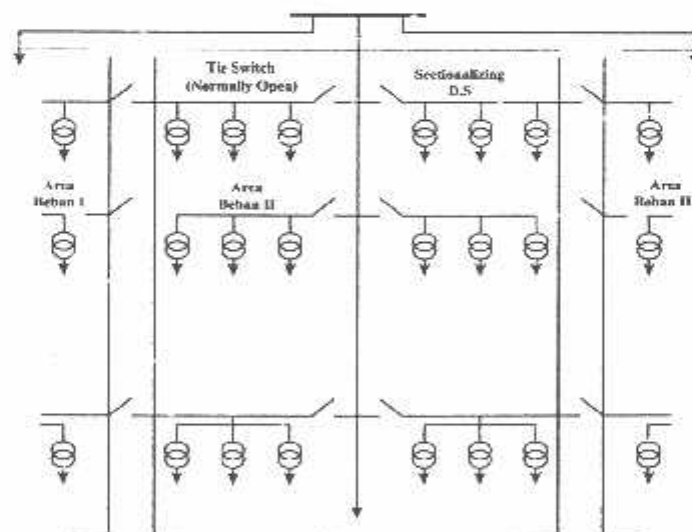
#### 2.3.1. Sistem Radial Pohon

Sistem radial jaringan pohon ini merupakan bentuk yang paling dasar dari sistem jaringan radial. Saluran utama (*main feeder*) ditarik dari suatu gardu induk sesuai dengan kebutuhan kemudian dicabangkan melalui saluran cabang (*lateral*

*feeder* ), selanjutnya dicabangkan lagi melalui saluran anak cabang ( *sub-lateral feeder* ). Ukuran dari masing-masing saluran tergantung dari kerapatan arus yang ada pada sistem. Dari gambar 2-2, *main feeder* merupakan saluran yang dialiri arus terbesar, selanjutnya arus mengecil pada tiap cabang tergantung dari besarnya beban.

### 2.3.2. Sistem Radial dengan *Tie* dan *Switch* Pemisah

Sistem ini merupakan pengembangan dari sistem radial pohon, untuk meningkatkan keandalan sistem saat terjadinya gangguan maka *feeder* yang terganggu akan dilokalisir sedangkan area yang semula dilayani oleh *feeder* tersebut pelayanannya dialihkan pada *feeder* yang sehat atau yang tidak terganggu. Sistem radial dengan *Tie* dan *Switch* Pemisah dapat dilihat pada gambar 2-3.

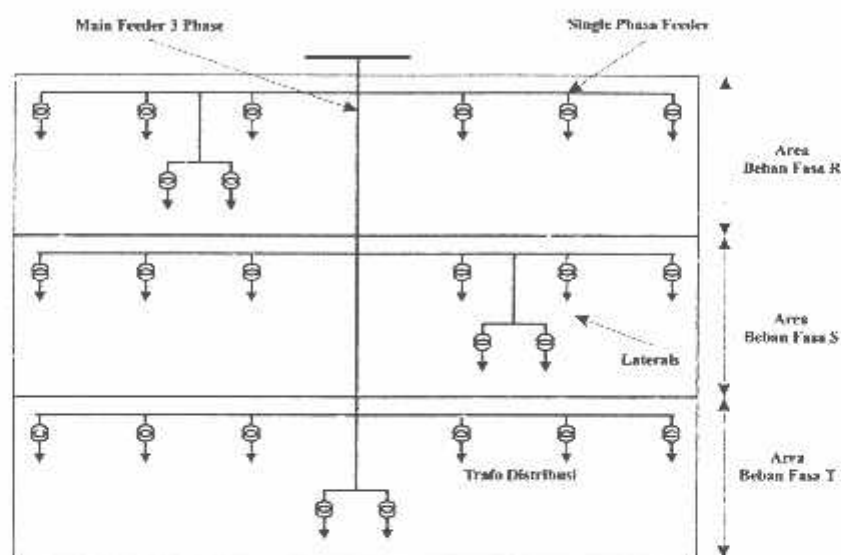


Gambar 2-3

Contoh Sistem Jaringan Distribusi Radial  
Dengan *Tie* dan *Switch* Pemisah <sup>[2]</sup>

### 2.3.3. Sistem Radial dengan Pembagian Fasa Area

Pada bentuk ini masing-masing fasa dari jaringan bertugas untuk melayani daerah beban yang berlainan. Bentuk ini akan dapat menimbulkan kondisi sistem tiga fasa yang tidak seimbang (simetris), bila digunakan pada daerah beban yang baru dan belum mantap pembagian bebannya. Contoh dari sistem jaringan ini dapat dilihat pada gambar 2-4.

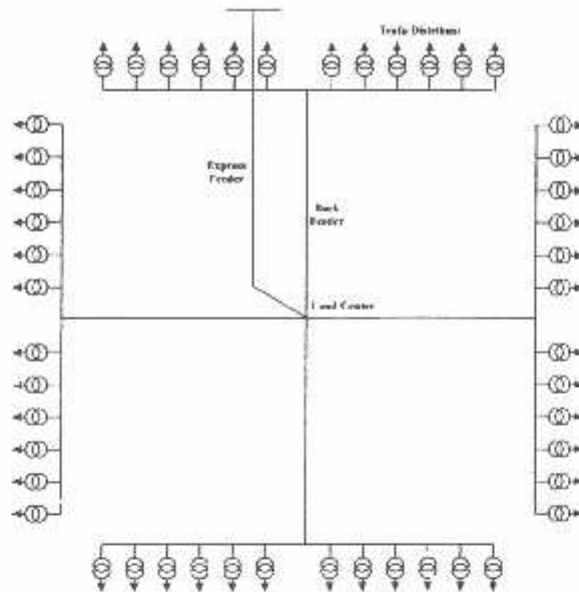


Gambar 2-4

Contoh Jaringan Distribusi Radial Dengan Fasa Area [2]

### 2.3.4. Sistem Radial Dengan Beban Terpusat

Bentuk dari sistem ini mensuplai daya dengan menggunakan *main feeder* yang disebut *express feeder* langsung ke pusat beban, dan dari titik pusat beban ini dikirim ke beban menggunakan *back feeder* secara radial seperti terlihat pada gambar 2-5.



Gambar 2-5

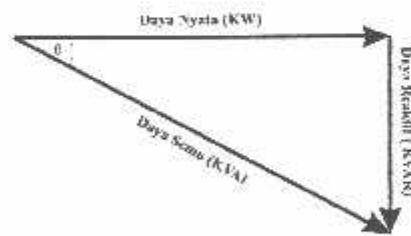
Contoh Jaringan Distribusi Radial Dengan Beban Terpusat <sup>[2]</sup>

#### 2.4. Faktor Daya

Faktor daya pada dasarnya didefinisikan sebagai perbandingan antara daya nyata dan daya semu, dan dinyatakan oleh persamaan :

$$\text{Faktor daya} = \frac{\text{Daya Nyata (kW)}}{\text{Daya Semu (kVA)}} \dots\dots\dots (2.1)$$

Untuk daya semu sendiri dibentuk oleh dua komponen daya nyata (kW) dan komponen daya reaktif (kVAR). Hubungan ini dapat digambarkan sebagai berikut :



Gambar 2-6 Segi Tiga Daya <sup>[1]</sup>

Dengan faktor daya =  $\text{Cos } \theta = \frac{P}{S}$

Dimana :

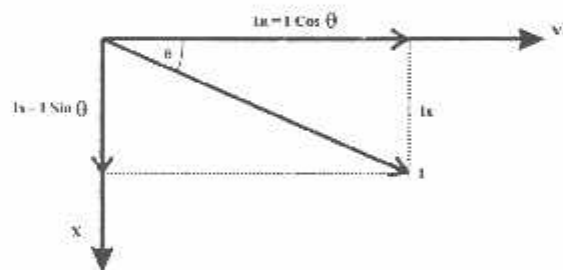
$P$  = Daya Nyata (kW)

$S$  = Daya Semu (kVA)

$Q$  = Daya Reaktif (kVAR)

$\theta$  = Sudut Phasa

Faktor daya dapat pula dilihat hubungannya dengan arus nyata dan arus total. Arus nyata ( $I_R$ ) adalah arus yang mengalir pada beban yang diubah menjadi tenaga. Sedangkan arus total ( $I$ ) adalah arus yang mengalir pada jaringan dan merupakan jumlah vektor antara arus magnetisasi dan arus nyata. Arus magnetisasi ( $I_x$ ) merupakan arus yang mengalir pada beban dan menimbulkan medan magnet. Arus yang terbaca dalam ampere meter adalah arus total ( $I$ ). Hubungan ketiga arus tersebut dapat dilihat pada gambar 2-7 sebagai berikut :



**Gambar 2-7**  
**Segi Tiga Arus**



## 2.5. Kapasitor Daya

Secara sederhana kapasitor terdiri dari dua 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 aluminium murni yang dipisahkan oleh tiga atau lebih lapisan kertas yang dilapisi oleh bahan kimia. Kapasitor daya telah mengalami perkembangan yang begitu cepat selama 30 tahun terakhir. Karena bahan dielektrik yang digunakan lebih efisien serta teknologi pembuatan kapasitor lebih baik.

### 2.5.1. Kapasitor Seri dan Kapasitor Shunt

Fungsi utama dari pemakaian kapasitor seri atau kapasitor shunt, dalam sistem tenaga adalah untuk membangkitkan daya reaktif, untuk memperbaiki faktor daya dan tegangan, sehingga meningkatkan kapasitas sistem dan mengurangi rugi daya jaringan.

Ada beberapa aspek yang tidak menyenangkan dalam kapasitor seri, secara umum biaya pemasangan kapasitor seri lebih tinggi dari biaya pemasangan kapasitor shunt, hal ini disebabkan karena peralatan pelindung untuk kapasitor seri lebih banyak, biasanya kapasitor seri didesain untuk daya yang lebih besar dari kapasitor shunt dengan tujuan untuk mengatasi kenaikan beban.

## 2.6. Pengaruh Pemasangan Kapasitor Shunt <sup>[1]</sup>

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

dipasanginya kapasitor shunt pada jaringan distribusi akan dapat memperbaiki profil tegangan, memperbaiki faktor daya, dan menaikkan kapasitas sistem serta dapat mengurangi rugi saluran.

Ada dua cara dalam pemakaian kapasitor shunt:

- Kapasitor tetap
- Kapasitor saklar

#### a. Kapasitor Tetap

Adalah kapasitor untuk kompensasi daya reaktif yang kapasitasnya tetap dan selalu terpasang di jaringan. Penggunaan kapasitor jenis 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 di hubungkan dan dilepaskan dari jaringan dan dapat diatur besar kapasitasnya sesuai dengan kondisi beban.

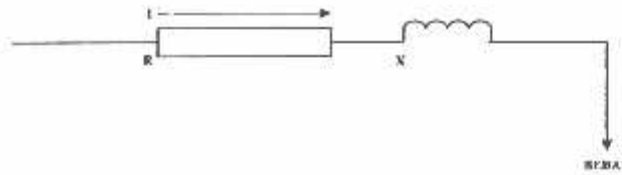
### 2.6.1. Pengurangan Rugi Daya Dengan Kapasitor Shunt

Rugi-rugi saluran perfasa dari saluran 3 fasa scimbang dengan beban terpusat seperti pada gambar 2-5 adalah  $I^2 (R-j X)$  atau dapat dibedakan menjadi:

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

Dimana:  $I_R$  adalah komponen arus aktif

$I_X$  adalah komponen arus reaktif



Gambar 2-8

Saluran Primer Dengan Beban Terpusat <sup>[1]</sup>

Rugi daya ( $I^2R$ ) dapat dibagi menjadi dua komponen yaitu komponen arus aktif dan komponen arus reaktif.

Rugi daya karena komponen arus aktif tidak akan mempengaruhi penempatan kapasitor shunt pada saluran, hal ini dapat dijelaskan sebagai berikut:

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

$$I^2R = (I \cos \theta)^2R + (I \sin \theta)^2R \dots \dots \dots (2.2)$$

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

$$I^2R = (I \cos \theta)^2R + (I \sin \theta - I_c)^2R \dots \dots \dots (2.3)$$

Sehingga pengurangan rugi daya sebagai akibat pemasangan kapasitor didapat:

$$\begin{aligned} \Delta P_{ls} &= I^2R - I_1^2R \\ &= (I \cos \theta)^2R + (I \sin \theta)^2R - (I \cos \theta)^2R + (I \sin \theta - I_c)^2R \\ &= 2 ( (I \sin \theta) I_c R - I_c^2 R ) \dots \dots \dots (2.4) \end{aligned}$$

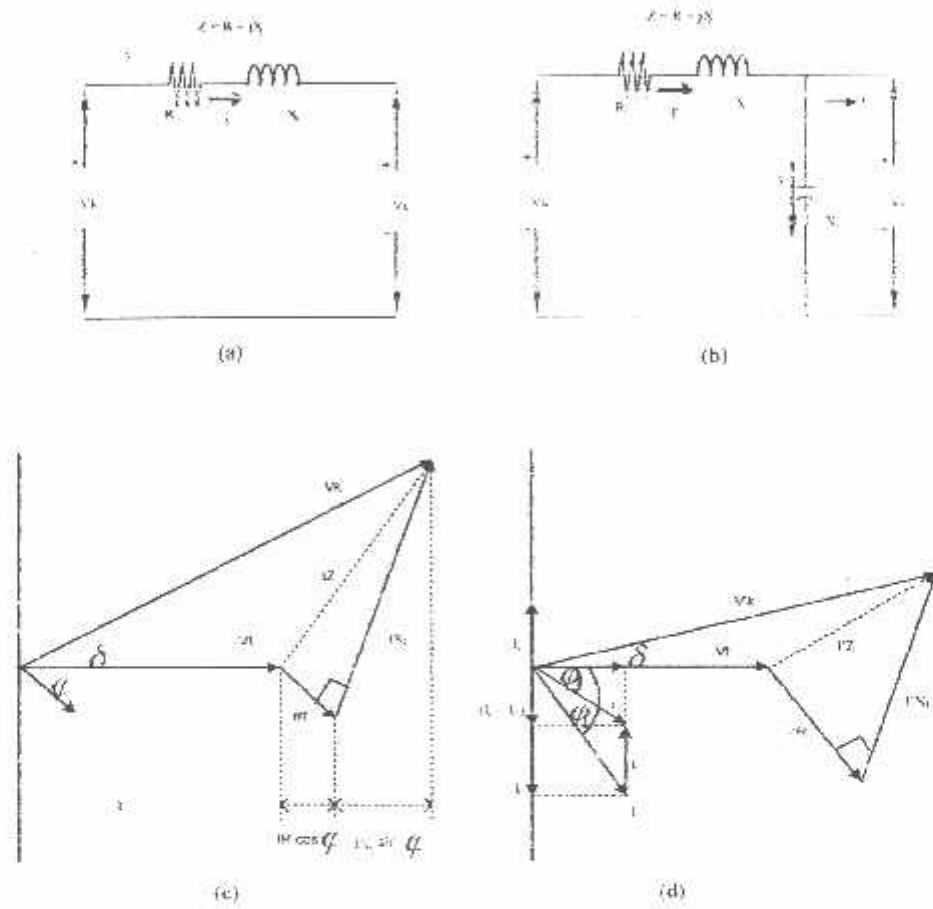
maka hanya komponen arus reaktif ( $I \sin \theta$ ) saja yang berpengaruh terhadap pengurangan rugi daya  $I^2R$  akibat pemasangan kapasitor shunt pada saluran distribusi.

Pengurangan rugi daya saluran 3 fasa adalah:

$$\Delta P_{ls} = 3R (2 (I \sin \theta) I_c - I_c^2) \text{ Watt} \dots \dots \dots (2.5)$$

2.6.2. Perbaikan Tegangan

Pemakaian kapasitor shunt dalam sistem tenaga listrik selain untuk perbaikan faktor daya juga bertujuan menaikkan tegangan. Dan secara vektoris dapat digambarkan sebagai berikut:



Gambar 2-9

Vektor Diagram Sebelum (a) dan Sesudah (b) Pemasangan Kapasitor Shunt Pada Jaringan<sup>[1]</sup>

Jatuh tegangan yang disebabkan arus beban I sebelum kapasitor dipasang:

$$\Delta V = I_R R + I_X X_L \dots \dots \dots (2.6)$$

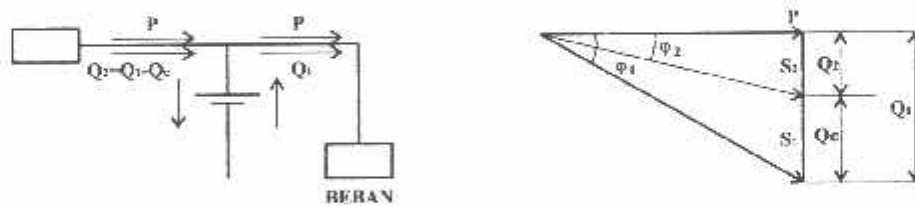
Jatuh tegangan setelah kapasitor dipasang:

$$\Delta V = I_R R + I_f X - I_c X \dots \dots \dots (2.7)$$

### 2.6.3. Perbaikan Faktor Daya dan Kapasitas Sistem

Manfaat terbesar yang diperoleh dari perbaikan faktor daya berasal dari pengurangan daya reaktif dalam sistem. Hal ini menghasilkan pengurangan biaya pemakaian daya yang lebih rendah, kenaikan kapasitas sistem, perbaikan tegangan dan pengurangan rugi daya dalam sistem. Satu-satunya cara untuk memperbaiki faktor daya adalah mengurangi daya reaktif pada jaringan. Jika komponen daya reaktif dapat dikurangi, maka total arus akan berkurang, sedang komponen daya aktif tidak berubah, maka faktor daya akan lebih besar sebagai akibat berkurangnya daya reaktif. Faktor daya akan mencapai 100% jika komponen daya reaktif sama dengan nol (0).

Dengan menambahkan kapasitor, daya reaktif komponen Q akan berkurang, gambar 2-10 menunjukkan perbaikan faktor daya pada sistem, kapasitor mensuplai daya reaktif ke beban.



Gambar 2-10  
Perbaikan Faktor Daya [1]

Diasumsikan bahwa beban disuplai oleh daya nyata P, daya reaktif (*lagging*) Q, dan daya semu S, pada faktor daya tertinggal  $\cos \theta_1$

$$\cos \theta_1 = \frac{P}{S_1} \dots \dots \dots (2.8)$$

Bila suatu kapasitor  $Q_c$  kVAR dipasang pada beban, faktor daya dapat diperbaiki dari  $\cos \theta_1$  menjadi  $\cos \theta_2$  dimana:

$$\cos \theta_2 = \frac{P}{S_2} = \frac{P}{\sqrt{P^2 + Q^2}}$$

$$= \frac{P}{\sqrt{P^2 + (Q_1 - Q_c)^2}} \dots \dots \dots (2.9)$$

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

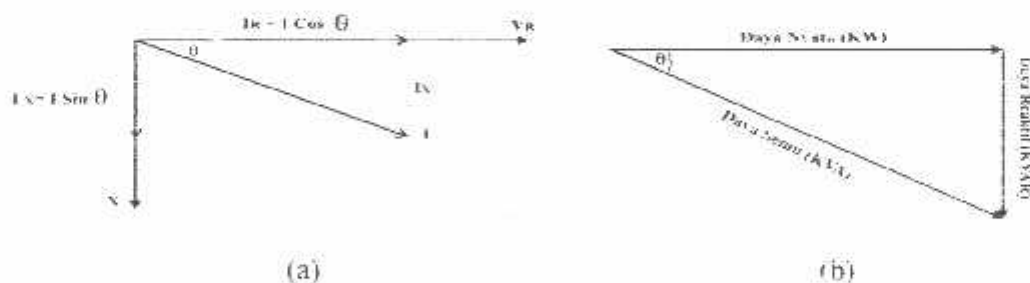
$$\% \text{ Rugi daya} = 100\% \left( \frac{\text{Faktor daya mula - mula}(\cos \theta_1)}{\text{Faktor daya baru}(\cos \theta_2)} \right) \dots \dots \dots (2.10)$$

% Pengurangan rugi daya

$$= 100\% \left( 1 - \left( \frac{\text{Faktor daya mula - mula}(\cos \theta_1)}{\text{Faktor daya baru}(\cos \theta_2)} \right)^2 \right) \dots \dots \dots (2.11)$$

### 2.6.3.1. Perhitungan Pengaruh Perbaikan Faktor Daya

Diagram fasor dari dua komponen arus nyata, arus aktif dan arus reaktif dapat dilihat pada gambar 2-11 berikut:



**Gambar 2-11**  
**Diagram Fasor dan Sudut Daya Beban pada**  
**Jaringan Distribusi [1]**

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

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

$$= \sqrt{(I \cos\theta)^2 + (I \sin\theta)^2}$$

Pada suatu tegangan  $V$ , daya aktif, daya reaktif dan daya semu adalah sebanding dengan arus, dimana hubungannya dapat dinyatakan sebagai berikut:

$$\text{Daya semu (kVA)} = \sqrt{(\text{Daya aktif})^2 + (\text{Daya reaktif})^2}$$

$$(VI) = \sqrt{(VI \cos\theta)^2 + (VI \sin\theta)^2}$$

$$\text{Faktor daya} = \frac{\text{Daya aktif}}{\text{Daya semu}} = \frac{KW}{KVA}$$

$$\text{Daya aktif} = \text{Daya semu} \times \text{Faktor daya}$$

$$kW = kVA \times \text{Faktor daya}$$

$$kW = kVA \cos \theta$$

### 2.6.3.2. Penentuan Rating Kapasitor Untuk Perbaikan Faktor Daya Beban

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

$$\cos \theta = \frac{\text{Daya aktif}}{\text{Daya semu}} = \frac{(kW)}{(kVA)}$$

$$\sin \theta = \frac{\text{Daya reaktif}}{\text{Daya semu}} = \frac{(kVAR)}{(kVA)}$$

$$\tan \theta = \frac{\text{Daya reaktif}}{\text{Daya aktif}} = \frac{(kVAR)}{(kW)}$$

Karena komponen daya aktif biasanya konstan, dan daya semu serta komponen daya reaktif berubah sesuai dengan faktor daya, maka persamaan yang

dinyatakan dalam komponen daya aktif yang paling tepat digunakan. Persamaan ini dapat ditulis sebagai berikut:

$$\begin{aligned} \text{Daya reaktif pada faktor daya mula-mula} &= \text{Daya aktif} \times \tan \theta_1 \\ &= (\text{kW}) \times \tan \theta_1 \end{aligned}$$

$$\begin{aligned} \text{Daya reaktif pada faktor daya baru} &= \text{Daya aktif} \times \tan \theta_2 \\ &= (\text{kW}) \times \tan \theta_2 \end{aligned}$$

Dengan  $\theta_1$  = sudut dari faktor daya mula-mula

$\theta_2$  – sudut dari faktor daya yang telah diperbaiki

Rating kapasitor yang dibutuhkan perbaikan faktor daya:

$$\begin{aligned} \text{Daya reaktif (kVAR)} &= \text{Daya aktif} \times (\tan \theta_1 - \theta_2) \\ &= (\text{kW}) \times (\tan \theta_1 - \theta_2) \end{aligned}$$

Untuk penyederhanaan  $(\tan \theta_1 - \tan \theta_2)$  sering ditulis  $\Delta \tan$ , yang merupakan suatu faktor pengali untuk menentukan daya reaktif.

$$\text{Daya reaktif (kVAR)} = \text{Daya aktif} \times \Delta \tan = (\text{kW}) \times \Delta \tan$$

## 2.7. Sistem Per-Unit <sup>[3]</sup>

Untuk memudahkan perhitungan-perhitungan dalam sistem tenaga listrik digunakan sistem p.u (*per-unit*) yang didefinisikan sebagai perbandingan harga yang sebenarnya dengan harga dasar (*base value*), sehingga dapat dirumuskan sebagai berikut:

$$\text{Besaran per-unit} = \frac{\text{Besaran sebenarnya}}{\text{Besaran dasar dengan ukuran yang sama}} \dots\dots(2.12)$$

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

➤ Untuk data 1 fasa

Arus dasar



$$I_d = \frac{kVA \text{ dasar } 1 \text{ fasa}}{kVA \text{ dasar } L - N} \dots\dots\dots(2.13)$$

Impedansi dasar

$$Z_d = \frac{(kV \text{ dasar } L - N)^2 \times 1000}{kVA \text{ dasar } 1 \text{ fasa}}$$

$$= \frac{(kV \text{ dasar } L - N)^2}{MVA \text{ dasar } 1 \text{ fasa}} \dots\dots\dots(2.14)$$

Dalam persamaan di atas nilai-nilai besaran diberikan untuk rangkaian satu fasa. Jadi tegangannya adalah tegangan antara fasa dengan tanah dan daya setiap fasa. Setelah besaran-besaran dasar ditentukan maka besaran-besaran itu dinormalisasikan terhadap besaran dasar. Dengan demikian impedansi per-satuan didefinisikan sebagai berikut:

$$Z = \frac{\text{Im pedansi sebenarnya } Z (\Omega)}{\text{Im pedansi dasar } Z_d} \dots\dots\dots(2.15)$$

### BAB III

## IMPLEMENTASI METODE KRUSKAL EVOLUTIONARY UNTUK PENENTUAN LETAK KAPASITOR PADA JARINGAN DISTRIBUSI 20 KV

### 3.1. Analisa Aliran Daya Jaringan

Dalam penerapannya analisa suatu jaringan distribusi didasarkan pada performa jaringan tersebut. Performa atau unjuk kerja suatu jaringan dapat diketahui dengan mengimplementasikan suatu analisa aliran daya untuk mengetahui kondisi awal suatu saluran dan profile tegangan pada setiap bus, salah satu metode untuk menganalisa aliran daya awal pada suatu jaringan adalah metode aliran daya *Newton Raphson*. Metode aliran daya awal *Newton Raphson* memiliki orientasi perhitungan pencarian nilai P,Q yang mendekati (*konvergen*) nilai pembebanan pada bus load, dan mencari tegangan serta sudut fasa pada setiap bus, serta menghitung rugi-rugi daya pada setiap saluran.

#### 3.1.1. Metode Newton Raphson

Secara matematis persamaan aliran daya *Newton Raphson* dapat diselesaikan dengan menggunakan koordinat rectangular, koordinat polar atau bentuk hibrid (gabungan antara bentuk kompleks dengan bentuk polar). Dalam pembahasan skripsi ini menggunakan bentuk polar.

Hubungan antara arus simpul  $I_p$  dengan tegangan simpul  $V_Q$  pada suatu jaringan dengan n simpul dapat dituliskan :

$$I_p = \sum_{q=1}^n Y_{pq} V_{pq} \dots\dots\dots(3.1)$$

Injeksi daya pada simpul p adalah :

$$S_p = P_p - jQ_p = V_p^* \cdot I_p \dots\dots\dots(3.2)$$

$$= V_p^* \sum_{q=1}^n Y_{pq} V_{pq} \dots\dots\dots(3.3)$$

Dalam penyelesaian aliran daya dengan *Newton Raphson* bentuk persamaan aliran daya yang dipilih adalah bentuk polar, dimana tegangan dinyatakan dalam bentuk polar, yaitu :

$$V_p^* = |V_p| e^{-j\delta_p}$$

$$V_q = |V_q| e^{j\delta_q}$$

$$V_{pq}^* = |V_{pq}| e^{-j\theta_{pq}}$$

Maka persamaan (3.1) dapat ditulis :

$$P_p - jQ_p = \sum_{q=1}^n |V_p V_q Y_{pq}| e^{-j(\delta_p - \delta_q + \theta_{pq})} \dots\dots\dots(3.4)$$

Dengan memisahkan bagian riil dan bagian imajiner maka diperoleh :

$$P_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \cos(\delta_p - \delta_q + \theta_{pq}) \dots\dots\dots(3.5)$$

$$Q_p = \sum_{q=1}^n |V_p V_q Y_{pq}| \sin((\delta_p - \delta_q + \theta_{pq})) \dots\dots\dots(3.6)$$

Kedua persamaan di atas akan menghasilkan suatu kumpulan persamaan serempak ( simultan ) yang tidak linier untuk setiap simpul sistem tenaga listrik. Untuk mengetahui magnitude tegangan (V) dan sudut fasa ( $\delta$ ) di setiap simpul dapat diselesaikan dengan menggabungkan persamaan (3.5) dan (3.6) yang dilinierkan dengan metode *Newton Raphson* yang dapat dilihat pada persamaan di bawah ini :

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix} \dots\dots\dots(3.7)$$

Dimana :

$\Delta P$  = selisih injeksi bersih daya nyata dengan penjumlahan aliran daya nyata tiap saluran yang menghubungkan simpul dengan  $V$  yang diperoleh dari perhitungan iterasi ke-  $k$ .

$\Delta Q$  = selisih injeksi bersih daya reaktif dengan penjumlahan aliran daya reaktif tiap saluran yang menghubungkan simpul dengan  $V$  yang didapat dari perhitungan iterasi ke-  $k$ .

$\Delta\delta$  = vektor koreksi sudut fasa tegangan.

$\Delta|V|$  = vektor koreksi magnitude tegangan.

H, L, M, N merupakan elemen-elemen off diagonal dan diagonal dari sub matriks

Jacobian yang dibentuk dengan mendefinisikan persamaan (3.5) dan (3.6), dimana:

$$H_{pq} = \frac{\partial P_p}{\partial \delta_q}$$

$$N_{pq} = \frac{\partial P_p}{\partial |V_q|}$$

$$M_{pq} = \frac{\partial Q_p}{\partial \delta_q}$$

$$L_{pq} = \frac{\partial Q_p}{\partial |V_q|}$$

Adapun rumus dari elemen matriks Jacobian adalah:

Untuk H:

$$\frac{\partial P_p}{\partial \delta_q} = |V_p V_q Y_{pq}| \sin(\delta_p - \delta_q + \theta_{pq}) \quad q \neq p$$

$$\frac{\partial P_p}{\partial \delta_p} = \sum_{q \neq p} |V_p V_q Y_{pq}| \sin(\delta_p - \delta_q + \theta_{pq})$$

Untuk N:

$$\frac{\partial P_p}{\partial |V_p|} = |V_p Y_{pp}| \cos(\delta_p - \delta_q + \theta_{pq}) \quad q \neq p$$

$$\frac{\partial P_p}{\partial |V_q|} = 2 |V_p Y_{pp}| \cos \theta_{pp} + \sum_{q=1}^n |V_p Y_{pq}| \cos (\delta_p - \delta_q + \theta_{pq})$$

$q \neq p$

Untuk M:

$$\frac{\partial Q_p}{\partial \delta_q} = - |V_p V_q Y_{pq}| \cos (\delta_p - \delta_q + \theta_{pq}) \quad q \neq p$$

$$\frac{\partial Q_p}{\partial \delta_p} = \sum_{q=1}^n |V_p V_q Y_{pq}| \cos (\delta_p - \delta_q + \theta_{pq})$$

$q \neq p$

Untuk L:

$$\frac{\partial Q_p}{\partial |V_q|} = |V_p Y_{pq}| \sin (\delta_p - \delta_q + \theta_{pq}) \quad q \neq p$$

$$\frac{\partial Q_p}{\partial |V_p|} = 2 |V_p Y_{pp}| \sin \theta_{pp} + \sum_{q=1}^n |V_p Y_{pq}| \sin (\delta_p - \delta_q + \theta_{pq})$$

$q \neq p$

Untuk menghitung selisih daya, maka mula-mula ditentukan nilai awal tegangan simpul dan sudut fasanya. Kemudian daya nyata dan daya reaktif dihitung dengan menggunakan persamaan (3.5) dan (3.6). Selisih antara daya yang telah ditentukan dengan daya hasil perhitungan ini merupakan perubahan daya yang terjadi pada simpul.

$$\Delta P = P_{\text{pembangkitan}} - P_{\text{beban}} - P_{\text{perhitungan}} \dots \dots \dots (3.8)$$

$$\Delta Q = Q_{\text{pembangkitan}} - Q_{\text{beban}} - Q_{\text{perhitungan}} \dots \dots \dots (3.9)$$

Magnitude tegangan  $|V|$  dan sudut fasa  $\delta_p$  yang diasumsikan serta selisih daya yang dihitung ( $\Delta P_p$  dan  $\Delta Q_p$ ) digunakan untuk memperoleh elemen-elemen matriks Jacobian.

Persamaan (3.7) diselesaikan untuk menghitung vektor koreksi magnitude tegangan  $\Delta(|V|)$  dan sudut fasa tegangan ( $\Delta\delta$ ) yang baru. Sehingga diperoleh harga magnitude tegangan dan sudut fasa yang baru, yaitu:

$$|V|^{k+1} = |V|^k + \Delta |V|^k \dots\dots\dots(3.10)$$

$$\delta^{k+1} = \delta^k + \Delta\delta^k \dots\dots\dots(3.11)$$

Proses perhitungan akan berulang sampai selisih daya nyata dan daya reaktif antara yang dijadwalkan dengan yang dihitung, yaitu  $\Delta P$  dan  $\Delta Q$  untuk semua simpul mendekati nilai toleransi atau proses perhitungan iterasi mencapai konvergen.

### 3.1.2. Algoritma Aliran Daya *Newton Raphson*

1. Tentukan nilai  $P_p$  (ditetapkan) dan  $Q_p$  (ditetapkan) yang mengalir ke dalam sistem pada setiap rel untuk nilai yang ditentukan atau perkiraan dari besar dan sudut tegangan untuk iterasi pertama atau tegangan yang ditentukan paling akhir untuk iterasi berikutnya.
  2. Hitung  $\Delta P$  dan  $\Delta Q$  pada setiap rel.
  3. Hitung nilai-nilai matriks Jacobian dengan menggunakan nilai-nilai perkiraan atau yang ditentukan dari besar sudut tegangan dalam persamaan untuk turunan parsial yang ditentukan dengan differensiasi persamaan (3.5) dan (3.6).
  4. Balikanlah Jacobian itu dan hitung koreksi-koreksi tegangan  $\Delta\delta_p$  dan  $\Delta|V_p|$  pada nilai sebelumnya.
-

5. Hitung nilai baru dari  $\delta_p$  dan  $|V_p|$  dengan menambahkan  $\Delta\delta_p$  dan  $\Delta|V_p|$  pada nilai sebelumnya.
6. Kembali ke langkah pertama dan ulangi proses itu dengan menggunakan nilai untuk besar dan sudut tegangan yang ditentukan paling akhir sehingga semua nilai  $\Delta P$  dan  $\Delta Q$  atau semua nilai  $\Delta\delta$  dan  $\Delta|V|$  lebih kecil dari suatu indeks ketetapan yang telah dipilih.

### 3.2. Metode Kruskal Evolutionary

*KEP* dalam hal ini ada dua fase, yang pertama yaitu fase evaluasi Kruskal yang merupakan sebuah tahap pencarian kemungkinan kombinasi pohon rentang minimum yang membawa pada lingkungan solusi didefinisikan dengan mengubah nilai kompensasi satu bus dari sistem listrik.

Yang kedua adalah fase optimasi. Fase optimasi berusaha menghasilkan solusi yang optimal dari hasil yang ditemukan sebelumnya, mengendalikan pencarian pada area baru. Cara ini menggunakan memori yang datanya telah dikumpulkan dari awal fase pertama.

#### 3.2.1. Algoritma Kruskal Evolutionary Programming

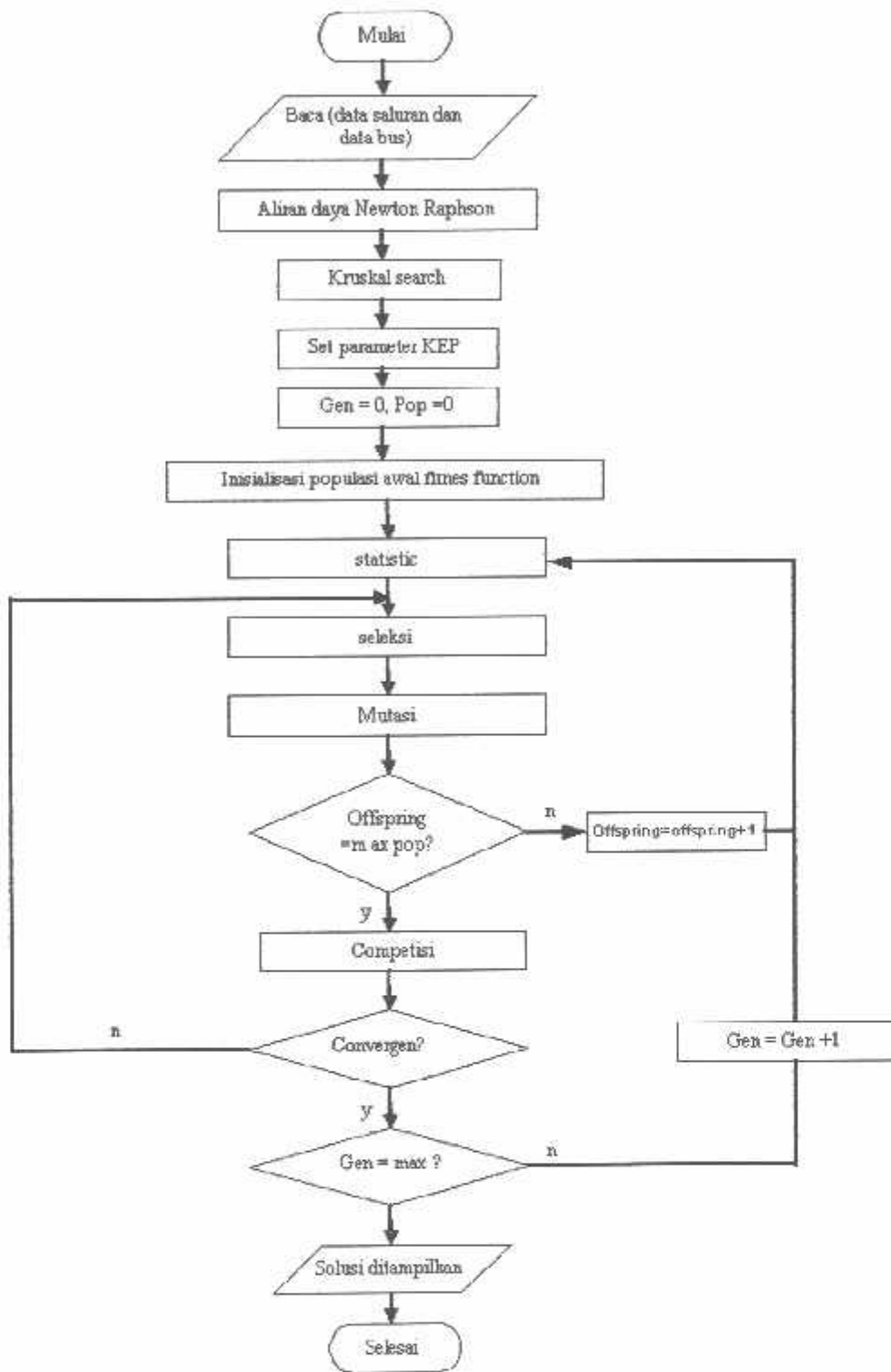
Urutan langkah-langkah yang digunakan pada implementasi pada

Analisa perangkat lunak adalah sebagai berikut :

1. Baca data bus dan saluran pada penyulang.
  2. Analisa aliran daya dengan Newton raphson.
  3. Kruskal searching
  4. Set parameter *KEP*
  5. Tetapkan populasi awal, dan total generasi.
-

6. Mencari nilai Fitness kromosom.
  7. Pencarian nilai minimum, maximum dan mean dari nilai fitness kromosom.
  8. Seleksi tiap-tiap kromosom.
  9. Terapkan operator mutasi.
  10. Apakah jumlah offspring = max populasi  
Jika ya lakukan ,offspring=offspring+1,langkah 9  
Jika tidak lakukan langkah 5
  11. Kompetisikan setiap parent dan child untuk mengganti nilai kromosom pada iterasi selanjutnya.
  12. Apakah solusi sudah konvergen/optimal  
Jika ya lakukan langkah 11  
Jika tidak lakukan langkah 6
  13. Apakah gen = maximum generasi  
Jika ya lakukan langkah 13  
Jika tidak lakukan, gen=gen+1,langkah 5
  14. Solusi ditampilkan
  15. Akhiri perhitungan
-





Gambar 3-1. FlowChart Kruskal Evolutionary

### 3.2.2 Metode Evolutionary Programming

Komponen prinsip Evolutionary Programming adalah sebagai berikut:

#### A. Representasi Populasi

Langkah pertama kali yang dalam penggunaan EP adalah melakukan pengkodean atau representasi terhadap masalah yang akan dilakukan. Seperti GA, Metode EP didasari oleh teori evolusi Charles Darwin. Dengan meniru prinsip evolusi yang ada dalam mahluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam.

EP dibentuk oleh serangkaian kromosom yang ditandai dengan  $x_i$  ( $i=1,2,\dots,N$ ) setiap  $x$  adalah suatu individu dari populasi yang akan dikembangkan. Dalam kromosom terdapat variable string yang disebut gen yang berisi nilai-nilai atau allele. Variabel ini dinyatakan dalam bentuk bilangan-bilangan real. Selanjutnya beberapa kromosom dibentuk dan dikumpulkan menjadi populasi, populasi inilah populasi awal bagi EP untuk melakukan pencarian.

#### B. Inisialisasi

Suatu populasi awal  $P_0$  adalah dibentuk sebagai  $P_0 = \{x_1, x_2, \dots, x_N\}$  oleh pembangkitan  $N$  individu secara random, dimana  $N$  adalah ukuran populasi.

Setiap  $x_i$ ,  $i = 1, 2, \dots, N$  diberikan suatu nilai fitness

$$g(x_i) = G(f(x_i), v_i), \dots \dots \dots (3.12)$$

dimana  $f(x_i)$  merupakan fitness sejati dari gen  $x_i$ ,  $v_i$  diuraikan sebagai alternative acak dalam  $x_i$ . Jadi  $G(f(x_i), v_i)$ , dituliskan sebagai nilai fitness (fitness score).

$$(X^0, Y^0) = (X_{min}, Y_{min}) - pi(X_{max} - X_{min}) + round(pi(Y_{max} - Y_{min})), \dots \dots \dots (3.13)$$

Dimana  $p_i \in [0,1]$  adalah satu bilangan acak, dan  $\text{round}(b)$  adalah bilangan bulat terdekat dari  $\text{rill}(b)$ .

**C. Fungsi Evaluasi (Fungsi Fitness)**

Fungsi evaluasi atau fungsi *fitness* dirancang untuk masing masing permasalahan yang akan diselesaikan. Dengan menggunakan kromosom tertentu, fungsi objektif atau fungsi evaluasi akan mengevaluasi status masing masing kromosom dalam populasi. Dengan kata lain fungsi evaluasi ini digunakan untuk menghitung derajat kesesuaian atau kelayakan pada calon-calon penyelesaian (candidate solution).

$$(X_i^{G+1}, Y_i^{G+1}) = \text{argmin}\{f(X_i^G, Y_i^G), f(U_i^{G-1}, V_i^{G-1})\} \dots \dots \dots (3.14)$$

$$(X_b^{G+1}, Y_b^{G+1}) = \text{argmin}\{f(X_i^G, Y_i^G) \dots \dots \dots (3.15)$$

dimana  $\text{argmin}$  berarti argumen minimum.

**D. Pembentukan Offspring**

N offspring baru dibangkitkan dari N parent oleh operator mutasi. Dengan distribusi Gaussian random yang diterapkan pada setiap  $x_i$ , sehingga  $x_{i+N}$ ,  $i = 1,2,\dots,N$  yang berarti ukuran populasi menjadi 2N atau 2 kali dari ukuran populasi semula dan ini kemudian disebut kombinasi populasi yaitu populasi lama (parent) dan mutasi populasi lama (offspring). Setiap  $x_{i+N}$ ,  $i = 1,2,\dots,N$  diberikan nilai fitness baru,

$$f(x_{i+N}) = G(f(x_{i+N}), v_{i+N}) \dots \dots \dots (3.16)$$

**E. Kompetisi dan Seleksi**

Berdasarkan populasi baru yang dihasilkan yakni kombinasi populasi dengan ukuran 2N, setiap individu dalam populasi harus dikompetisi secara stochastic dengan anggota yang lain berdasarkan nilai fitnessnya. Setiap  $x_i$ ,

$i=1,2,\dots,N$ . suatu nilai  $w_i$  diberikan untuk bobot individu menurut persamaan berikut:

$$w_i = \sum_{r=1}^x w_{ir} \dots\dots\dots(3.17)$$

$$\text{dengan : } w_{ir} = \begin{cases} 1, & \text{jika } \mathcal{G}(x_i) < \mathcal{G}(x_r) \\ 0, & \text{sebaliknya} \end{cases}$$

- a. N individu pertama dituliskan bersama nilai fitnessnya untuk menjadi awal dari generasi berikutnya. Proses akan melakukan pengulangan sampai memberikan kondisi yang paling konvergen.

**3.2.3 Objective Function Kruskal Evolutionary Programming**

Pada metode ini, ada dua fungsi sasaran akan dicapai. Yaitu untuk menekan rugi-rugi saluran/teknis dan untuk meminimalkan biaya/ekonomis. Persamaan tersebut adalah sebagai berikut :

$$\text{Min } v = k_c \sum_{i=0}^m T_i P_i(x^i) + \sum_{k=1}^{n_c} f(u_k^o) \dots\dots\dots(3.18)$$

$$\begin{aligned} \text{Dimana : } G^i(x^i, u^i) &= 0; & i &= 0, 1, \dots, n_i \\ H^i(x^i) &\leq 0; & i &= 0, 1, \dots, n_i \\ 0 \leq u_k^i &\leq u_k^o; & k &\in C_1 \text{ atau} \\ & & k &\in C_2 \end{aligned}$$

Masalah 1 adalah suatu penggabungan non linear integer dengan fungsi tujuan yang tidak dapat dibedakan dimana  $n_i$  menyajikan jumlah level beban pada kurva durasi beban linear. Variasi profil beban  $Q_L$  dapat dirumuskan;

$$Q_L(\tau) = Q_L^o(\tau) s(\tau) \dots\dots\dots(3.19)$$

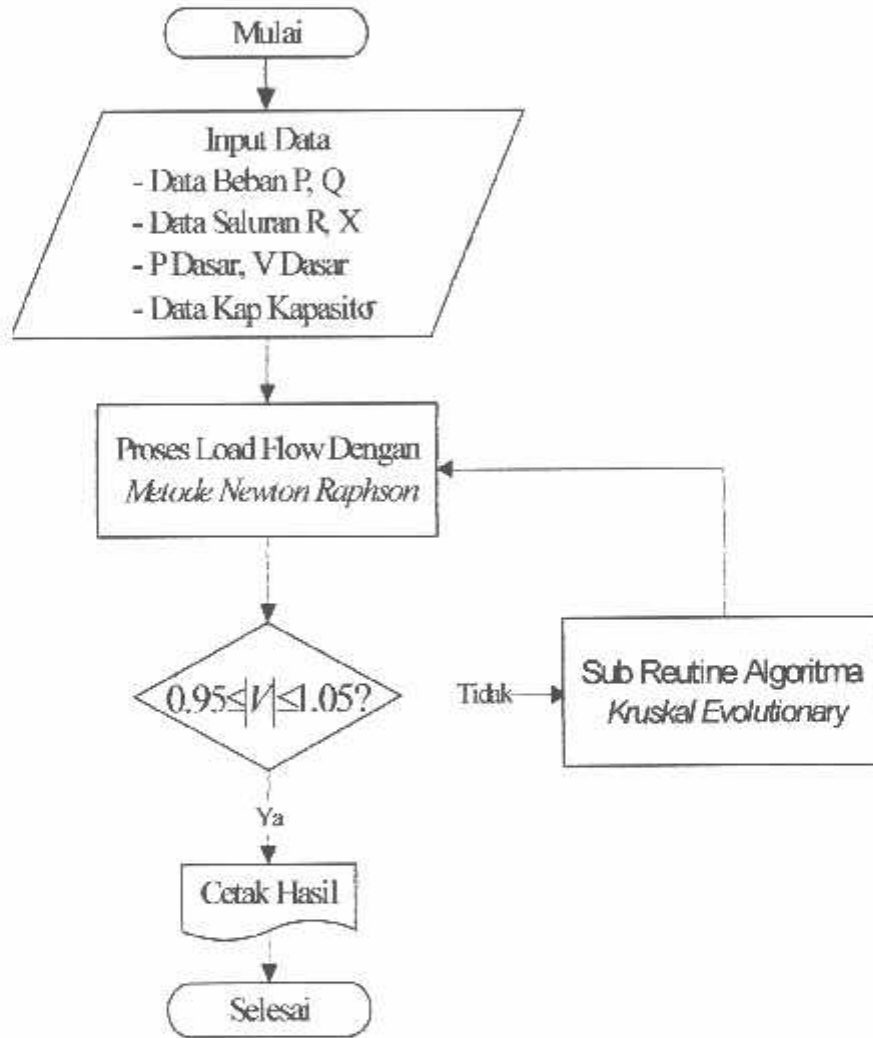
Keterangan :  $Q_L^o$  = menyajikan beban puncak,  
 $n_c$  = jumlah bus-bus.  
 $G^i(x^i, u^i) = 0$  = aliran daya untuk beban ke-i.

$H^i(x^i) \leq 0$  = batasan operasi untuk beban ke- $i$ .

$u_k^a$  = ukuran kapasitor untuk bus  $k$ .

$u_k^i$  = level operasi kapasitor bus  $k$  untuk level beban  $i$ .

### 3.2.4 Flow Chart Penyelesaian Masalah



Gambar 3-2. FlowChart Pemecahan Masalah

### 3.3. Validasi Program

Pada tulisan ini data validasi yang digunakan adalah data yang diperoleh dari jurnal *IEEE TRANSACTION ON POWER SYSTEM, VOL. 10 NO.1 . FEBRUARY 2005.*

Berikut adalah data saluran dan pembebanan yang digunakan.

- Jumlah Saluran = 8
- Jumlah Bus = 9
- Tegangan Dasar = 23 kV

**Tabel 3 – 1**  
**Data Impedansi Saluran**

No Saluran	Dari Node	Ke Node	Impedansi Saluran	
			R (Ohm)	X (Ohm)
1	1	2	0.014	0.6051
2	2	3	0.7463	1.205
3	3	4	0.6984	0.6084
4	4	5	1.9831	1.7276
5	5	6	0.9053	0.7886
6	6	7	2.0552	1.164
7	7	8	4.7953	2.716
8	8	9	5.3434	3.0264

Sumber: IEEE Transactions on Power Systems, Vol. 10, No. 1, February 2005

**Tabel 3 – 2**  
**Data Pembebanan**

No Node	Pembebanan		Type Bus
	P(kW)	Q(kVAR)	
1	1840	460	Load
2	980	340	Load
3	1790	446	Load
4	1598	1840	Load
5	1610	600	Load
6	780	110	Load
7	1150	60	Load
8	580	130	Load
9	1640	200	Load

Sumber: IEEE Transactions on Power Systems, Vol. 10, No. 1, February 2005

Hasil perhitungan program:

Rugi sebelum  $783.700 + j880.417$  kVA

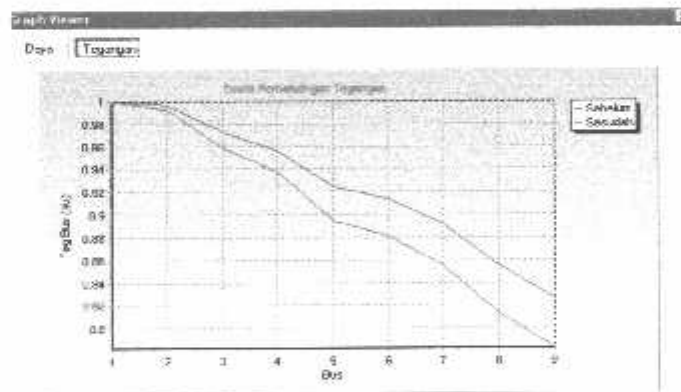
Penempatan kapasitor

- Bus 6, kapasitor 600 kVAR
- Bus 9, kapasitor 600 kVAR

Rugi Sesudah  $675.380 + j 780.632$  kVA

**Tabel 3 – 3**  
**Perbandingan Hasil Program Dengan Hasil Pada Jurnal.**

Result	FE	Jurnal	Error(%)
Kapasitor pada Bus	6, 9	4, 5, 9	-
Ploss Sebelum (kW)	783.7	783.778	0.0099
Ploss Sesudah (kW)	675.380	670.099	0.78



**Grafik 3 – 1**  
**Hasil Perhitungan Kruskal Evolutionary**

### 3.4. Total Biaya Sebelum dan Setelah Pemasangan Kapasitor

Total biaya sebelum pemasangan kapasitor merupakan hubungan antara rugi daya yang ada di saluran sebelum dipasang kapasitor dan biaya daya per tahun. Hubungan tersebut seperti persamaan dibawah ini:

$$S = K_p \cdot P_{\text{loss},T} \dots\dots\dots(3.20)$$

Sehingga total biaya setelah pemasangan kapasitor adalah hubungan antara rugi-rugi daya pada jaringan setelah dipasangnya kapasitor dengan biaya instalasi kapasitor tersebut. Hubungan tersebut akan dijelaskan pada persamaan di bawah ini:

$$S_K = K_p \cdot P_{\text{loss},T} + \sum_{j=1}^k K_j^c Q_j^c \dots\dots\dots(3.21)$$

Dimana:

- S = Total biaya tahunan setelah pemasangan kapasitor(\$/tahun)
- $K_p$  = Biaya daya (\$/kW/tahun)
- $P_{\text{loss},T}$  = Total rugi daya saluran
- $K^c$  = Biaya instalasi kapasitor (\$/kVar/tahun)
- $Q^c$  = Kapasitas terpasang

#### 3.4.1. Biaya Daya ( $K_p$ )

Biaya daya ( $K_p$ ) adalah biaya investasi per-kW ditambah biaya operasi dan pemeliharaan. Dalam skripsi ini berdasarkan referensi maka biaya daya ( $K_p$ ) diset yaitu sebesar U.S \$ 168/kW/Tahun



## BAB IV

### ANALISIS PENENTUAN LETAK KAPASITOR PADA JARINGAN DISTRIBUSI 20 kV TEJAKULA BALI MENGGUNAKAN ALGORITMA KRUSKAL EVOLUTIONARY

Pembahasan pada bab ini adalah mengenai simulasi dan analisa penempatan kapasitor menggunakan metode *Kruskal Evolutionary* sebagai fungsi optimasi untuk mencari solusi terbaik dari lingkungan solusi yang didapat. Analisis pada bab ini menyangkut analisa aliran awal dengan menggunakan metode *Newton Raphson*, dan analisis penentuan letak kapasitor menggunakan metode *Kruskal Evolutionary* untuk memperbaiki profil tegangan untuk mengurangi rugi-rugi daya pada saluran agar kondisi tegangan sesuai dengan batas normal yang diharapkan.

#### 4.1. Analisa Penentuan Letak Kapasitor Dengan Menggunakan Metode Kruskal Evolutionary

Pada aplikasi ini data analisis yang digunakan adalah data pada jaringan distribusi 20 kV penyulang Tejakula Bali, untuk menyelesaikan perhitungan aliran daya ditetapkan single line gambar 4.1 dengan nilai besaran dasar yang digunakan pada jaringan distribusi penyulang Tejakula Bali adalah sebagai berikut:

- Tegangan dasar : 20 kV
- Daya dasar : 100 kVA

Pada proses awal analisa bus-bus yang diklasifikasikan, dengan aturan masing-masing, Busbar diasumsikan sebagai slack bus, sedangkan bus-bus yang

lain sepanjang saluran radial dianggap sebagai load bus. Dalam hal ini tidak ada bus generator karena sepanjang saluran tidak terdapat pembangkitan.

#### 4.2. Konsep 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 di tiap-tiap bus. Setelah studi aliran daya dilakukan, barulah dilakukan perhitungan aliran daya dan rugi-rugi daya pada saluran.

Untuk menentukan lokasi dan kapasitas kapasitor terlebih dahulu menentukan rugi daya saluran terbesar yang digunakan sebagai inputan untuk menjalankan metode Kruskal Evolutionary agar kapasitor dapat diletakan pada lokasi yang memberikan profil tegangan yang paling optimum. Sedangkan pencarian grafik yang optimal secara random di tentukan oleh program komputer.

Untuk memudahkan perhitungan dan analisa pada sistem tenaga, biasanya dipakai harga-harga dalam per-satuan. Harga per-satuan adalah harga yang sebenarnya dibagi dengan harga dasar, dimana harga dasar ini dapat dipilih sembarang. Harga yang dipilih pada studi ini adalah 20 kV dan 100 kVA sebagai harga tegangan dasar dan daya dasar. Mengingat bahwa pada jaringan tidak dilakukan pengukuran faktor daya, maka pada perhitungan ini diambil harga faktor daya sebesar 0,86.

Perhitungan diawali dengan menampilkan single line diagram dari penyulang yang mewakili keadaan sistem yang sesungguhnya. Pada penyulang ini jumlah bus dan jumlah saluran masing-masing adalah :

---

- Slack bus = 1
- Load bus = 70
- Jumlah saluran = 70

#### 4.3. Data Saluran

Jaringan distribusi ini menggunakan saluran kabel udara dengan spesifikasi seperti pada table 4-1.

**Tabel 4-1**  
**Data Saluran Sistem 20 kV Penyulang Tejakula**

Jenis Konduktor	Penampang Nominal (mm <sup>2</sup> )	Resistansi Konduktor (r) (Ω/km)	GMR (mm)	Kuat Hantar Arus (A)
AAAC	35	0,9217	2,4227	170
AAAC	50	0,6452	2,8957	210
AAAC	70	0,4608	3,4262	255
AAAC	120	0,2688	4,6837	365
AAAC	150	0,2162	5,2365	425

Adapun data saluran penyulang Tejakula seperti pada tabel 4-2.

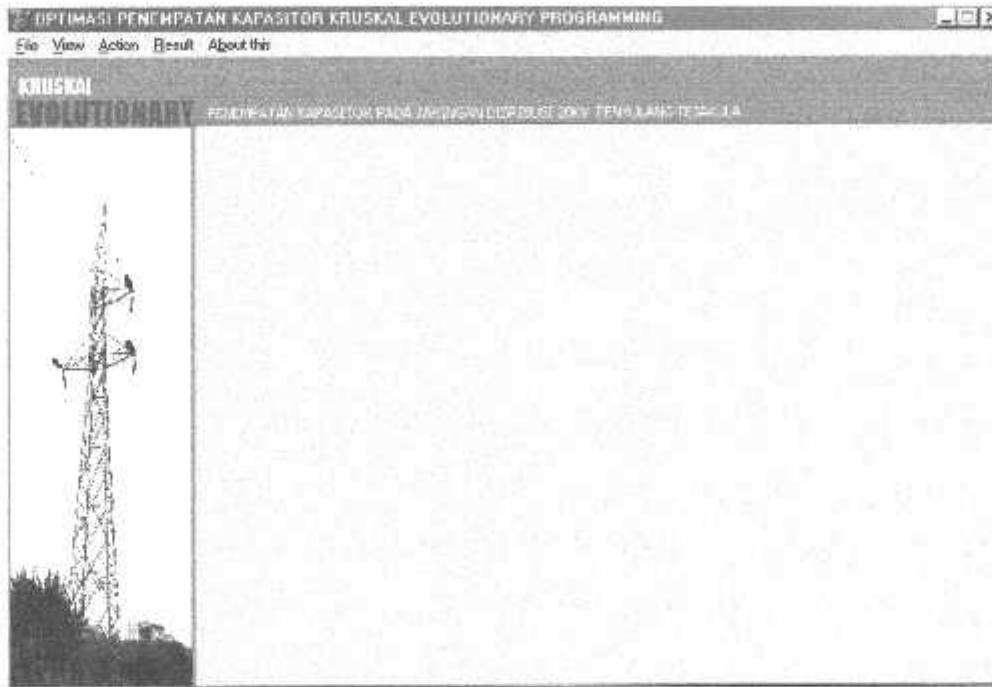
**Tabel 4-2**  
**Data Saluran Penyulang Tejakula**

No Saluran	Dari Node	Ke Node	Panjang (m)	P (kW)	Q (kVAR)
1	1	2	8,776.50	0	0
2	2	3	650.1111	0	0
3	2	4	7,765.22	55.9000	33.1691
4	4	5	2,311.51	0	0
5	4	20	794.5802	178.45	105.886

Tabel 4-2 selengkapnya dapat dilihat pada lampiran

#### 4.4. Hasil Perhitungan Program Kruskal Evolutionary

Hasil perhitungan program Kruskal Evolutionary dengan menggunakan bahasa program *Borland Delphi 7.0* sebagai berikut:



Gambar 4-1 Form Utama

Pada form ini terdapat beberapa fitur yaitu sebagai berikut:

- ✓ File → New File, adalah menu untuk membuat suatu data baru yang disesuaikan dengan coding format data pada aplikasi ini.
- ✓ File → Load Data, adalah menu untuk membuka data yang telah disimpan pada aplikasi ini sebelumnya..
- ✓ View → Data Bus + Saluran, adalah menu untuk menampilkan data pembebanan dan saluran, berikut adalah screen shot dari form ini.

No	Dari	Ke	R (pu)	X (pu)	Lc (pu)	Tr

**Gambar 4-2 Form Data Bus dan Saluran**

- ✓ View → Data Kapasitor + Set Parameter, adalah menu untuk menampilkan data kapasitor dan setting parameter pada algoritma KEP, berikut adalah screen shot dari form ini.

No	Cap (kVAR)	Harga (USD)
1	150	0.50000
2	300	0.35000
3	450	0.25300
4	600	0.22000

**Gambar 4-3 Form Data kapasitor + Set parameter**

- ✓ Action → Aliran Daya Newton Raphson, adalah menu untuk melakukan aliran daya awal dengan menggunakan metode Newton Raphson Hasil perhitungan dapat dilihat pada Table 4 - 8
- ✓ Action → Kruskal Evolutionary Analist, adalah menu untuk melakukan pencarian penempatan kapasitor yang optimal dan untuk melakukan pencarian aliran daya sesudah pemasangan kapasitor Tabel 4 - 9 .
- ✓ dengan menggunakan metode Kruskal Evolutionary.
- ✓ Result → Load Flow Awal, adalah menu untuk menampilkan hasil aliran daya dengan Newton Raphson. Berikut adalah tampilannya.

Bus	Abs V (pu)	Sud V (deg)	Pgen (kW)	Qgen (kW)	Pload (kVAR)	Qload (kVAR)
1	1.00000	0.00000	2028.330	1258.719	0.000	0.000
2	0.98129	-0.51018	0.000	0.000	0.000	0.000
3	0.98125	-0.51127	0.000	0.000	55.900	33.169
4	0.96526	-0.96455	0.000	0.000	0.000	0.000
5	0.96354	-1.01427	0.000	0.000	178.450	105.886
6	0.96283	-1.03518	0.000	0.000	64.672	38.374
7	0.96266	-1.04003	0.000	0.000	0.000	0.000
8	0.96248	-1.04532	0.000	0.000	16.490	10.971
9	0.96246	-1.04600	0.000	0.000	18.920	11.226
10	0.96244	-1.04639	0.000	0.000	11.180	6.634
11	0.96243	-1.04666	0.000	0.000	8.600	5.103
12	0.96130	-1.02974	0.000	0.000	140.500	89.302

Untuk data load flow awal selengkapnya dapat dilihat pada lampiran

**Gambar 4-4 Form Load Flow Awal**

- ✓ Result → Kruskal Evolutionary Search, adalah menu untuk menampilkan hasil pencarian Kruskal Evolutionary. Berikut adalah penjelasan dalam bentuk visualisasi gambar.

Form Load Flow Akhir						
Bus	Abs V (pu)	Sud Y (deg)	P <sub>gen</sub> (kW)	Q <sub>gen</sub> (kW)	P <sub>load</sub> (KVAR)	Q <sub>load</sub> (KVAR)
1	1.00000	0.00000	2014.353	37.412	0.000	0.000
2	0.99028	-0.93493	0.000	0.000	0.000	0.000
3	0.99024	-0.93500	0.000	0.000	55.900	33.169
4	0.98230	-1.57403	0.000	0.000	0.000	0.000
5	0.98062	-1.52203	0.000	0.000	178.451	105.886
6	0.97992	-1.64222	0.000	0.000	64.672	38.374
7	0.97975	-1.64590	0.000	0.000	0.000	0.000
8	0.97957	-1.65201	0.000	0.000	18.490	10.571
9	0.97955	-1.65267	0.000	0.000	18.920	11.226
10	0.97954	-1.65305	0.000	0.000	11.180	6.634
11	0.97953	-1.65331	0.000	0.000	8.600	5.103
12	0.97842	-1.68524	0.000	0.000	190.500	89.302
13	0.97828	-1.68923	0.000	0.000	53.664	31.642

Untuk data load flow Akhir selengkapnya dapat dilihat pada lampiran

**Gambar 4-5 Form Load Flow Akhir**

- ✓ Result → Graph Viewer, adalah menu untuk menampilkan grafik tegangan dan rugi daya saluran dari kondisi sebelum dan sesudah optimasi.
- ✓ About this, adalah menu untuk kotak dialog about yang berisi keterangan dari author pada tugas akhir ini. Berikut adalah screen shot form about this.

Rincian penggunaan program pada aplikasi ini adalah sebagai berikut:

1. Pada tahap awal user diminta untuk memasukan data analisis yang telah disimpan dengan memilih menu File → Load Data.
2. Setelah data telah dimuat maka proses selanjutnya adalah melakukan analisa aliran daya awal Newton Raphson dengan memilih menu Action → Aliran Daya Newton Raphson.
3. Setelah load flow awal dilakukan maka user harus memasukan parameter KEP dengan memilih menu View → Data Kapasitor + Set Parameter.

4. Analisa KEP dapat diterapkan dengan memilih menu Action → Kruskal Evolutionary Analist.
5. Setelah proses semua telah dilakukan user dapat melihat hasil dari setiap proses dengan memilih menu Result → Graph Viewer, Result → Load Flow Awal, atau Result → Kruskal Evolutionary Search.

#### **4.5. Hasil Analisa Program dengan Menggunakan Metode Kruskal Evolutionary**

Setelah dilakukan analisa dengan menggunakan metode Kruskal *Evolutionary*, akan diperoleh hasil yaitu jumlah, letak, kapasitas setelah penempatan kapasitor tersebut. Seperti terlihat pada gambar 4-8 tampilan hasil perhitungan dengan metode *KEP*.

- Banyaknya Saluran = 70
- Banyaknya Bus = 71
- Banyaknya Iterasi = 3

Rugi Total Saluran :  $59.590 + j 91.094$  KVA


Pada aplikasi ini nilai parameter *KEP* yang digunakan adalah sebagai berikut:

- Jumlah Generasi = 4
  - Jumlah Populasi = 4
  - Probabilitas CrossOver = 0.5
  - Konstanta penalty = 1000
-



Setelah dilakukan analisa dengan menggunakan metode *Kruskal Evolutionary*, akan diperoleh hasil yaitu jumlah, letak, kapasitas setelah penempatan kapasitor tersebut. Seperti terlihat pada gambar 4-5 tampilan hasil perhitungan dengan metode *KEP*.

Rugi Total Saluran :  $45.653 + j 69.788$  KVA



No	Node	Kap (KVAR)	Biaya (US\$)
27	150	0.50	

Biaya	
Biaya Sebelum	22.094.733 US \$
Biaya Sesudah	16.710.859 US \$
Selisih Biaya	5.383.873 US \$

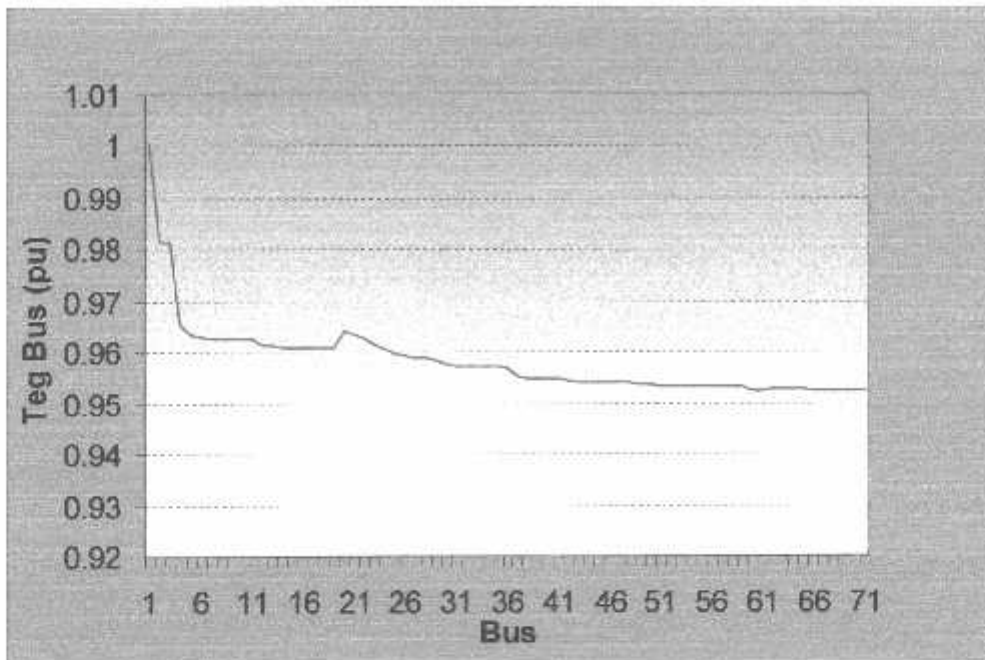
**Gambar 4-6**  
**Penempatan Kapasitas kapasitor : Bus ke 27, 150 kVAR**

Dari gambar diatas dapat dilihat bahwa solusi penempatan kapasitor dipasang pada bus 27, dengan kapasitas yaitu sebesar 150 kVAR. Hal ini dikarenakan rugi-rugi saluran terkecil adalah pada penempatan 150 kVar pada bus 27.

**TABEL 4-3 LOAD FLOW AWAL  
SEBELUM PEMASANGAN KAPASITOR**

Bus	Abs V (pu)	Sud V
1	1	0
2	0.98129	-0.51018
3	0.98125	-0.51127
4	0.96526	-0.96455
5	0.96354	-1.01427
6	0.96283	-1.03518
7	0.96266	-1.04003
8	0.96248	-1.04532
9	0.96246	-1.046
10	0.96244	-1.04639
11	0.96243	-1.04666
12	0.9613	-1.07974
13	0.96116	-1.08387
14	0.96109	-1.08594
15	0.96107	-1.08653
16	0.96105	-1.08706
17	0.96098	-1.08913
18	0.96093	-1.09042
19	0.96091	-1.09109
20	0.96421	-0.99479
21	0.96348	-1.01569
22	0.96275	-1.03705
23	0.96137	-1.07693
24	0.96056	-1.10035
25	0.95976	-1.1237
26	0.95911	-1.14262
27	0.95872	-1.15401
28	0.95872	-1.15407
29	0.95841	-1.1631
30	0.95764	-1.18561
31	0.9574	-1.19277
32	0.95725	-1.1972
33	0.95704	-1.20335
34	0.95728	-1.19619
35	0.95709	-1.2019
36	0.95701	-1.20397
37	0.95521	-1.25675
38	0.95496	-1.26419
39	0.95487	-1.26687
40	0.95484	-1.26767
41	0.95474	-1.27072
42	0.95419	-1.28704

43	0.95412	-1.28891
44	0.95402	-1.29201
45	0.95396	-1.2937
46	0.95394	-1.29417
47	0.95389	-1.29574
48	0.95382	-1.29786
49	0.9537	-1.30153
50	0.95368	-1.30185
51	0.9533	-1.31325
52	0.95314	-1.31796
53	0.95311	-1.31889
54	0.953	-1.32213
55	0.95298	-1.32253
56	0.95296	-1.32325
57	0.95297	-1.32298
58	0.95295	-1.32341
59	0.95297	-1.32304
60	0.95248	-1.33734
61	0.95248	-1.33751
62	0.95283	-1.32718
63	0.95281	-1.32761
64	0.95273	-1.33008
65	0.95257	-1.33475
66	0.95251	-1.3364
67	0.9525	-1.3368
68	0.95252	-1.33637
69	0.9525	-1.33687
70	0.95248	-1.33744
71	0.95246	-1.33795



**Grafik 4 – 1**

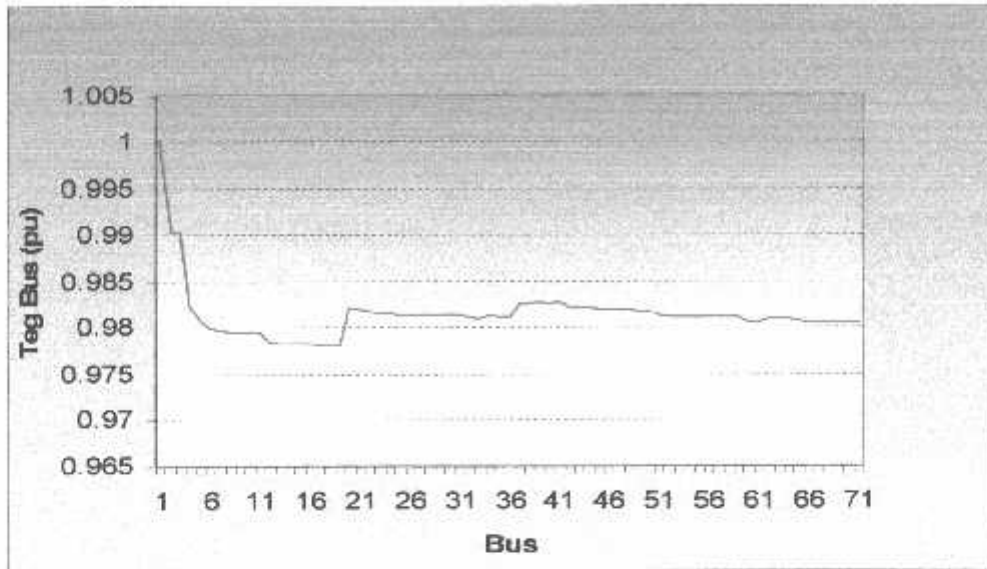
**Tegangan Tiap – Tiap Bus Terhadap Asumsi Tegangan Awal  
Sebelum Optimasi**

Setelah dilakukan analisa aliran daya dengan menggunakan metode *Newton Raphson* maka diperoleh besarnya tegangan tiap-tiap bus seperti pada tabel 4-4 Dan Grafik 4 - 1 terlihat bahwa tegangan tiap bus sebelum dilakukan kompensasi rata-rata dibawah 0,965 pu dengan nilai ekstrim minimum mencapai 0.952 pu.

**TABEL 4 - 4 LOAD FLOW AKHIR  
SESUDAH PEMASANGAN KAPASITOR**

Bus	Abs V (pu)	Sud V
1	1	0
2	0.99028	-0.83493
3	0.99024	-0.836
4	0.9823	-1.57403
5	0.98062	-1.82203
6	0.97992	-1.64222
7	0.97975	-1.6469
8	0.97957	-1.65201
9	0.97955	-1.65267
10	0.97954	-1.65305
11	0.97953	-1.65331
12	0.97842	-1.68524
13	0.97828	-1.68923
14	0.97821	-1.69122
15	0.97819	-1.6918
16	0.97817	-1.6923
17	0.9781	-1.6943
18	0.97806	-1.69555
19	0.97803	-1.69619
20	0.98208	-1.83381
21	0.98193	-1.67566
22	0.9818	-1.71877
23	0.98157	-1.79996
24	0.98146	-1.84833
25	0.98137	-1.89715
26	0.98132	-1.93754
27	0.9813	-1.96235
28	0.9813	-1.96241
29	0.98129	-1.98218
30	0.98126	-2.03151
31	0.98128	-2.04809
32	0.98113	-2.0523
33	0.98093	-2.05816
34	0.98131	-2.05691
35	0.98112	-2.06235
36	0.98105	-2.06432
37	0.98247	-2.23547
38	0.98262	-2.25779
39	0.98268	-2.26589
40	0.98265	-2.26664
41	0.98277	-2.27785
42	0.98223	-2.29326

43	0.98217	-2.29503
44	0.98207	-2.29795
45	0.98201	-2.29955
46	0.982	-2.29999
47	0.98195	-2.30147
48	0.98188	-2.30347
49	0.98176	-2.30693
50	0.98175	-2.30724
51	0.98137	-2.318
52	0.98122	-2.32244
53	0.98119	-2.32331
54	0.98108	-2.32637
55	0.98107	-2.32675
56	0.98104	-2.32743
57	0.98105	-2.32717
58	0.98104	-2.32758
59	0.98105	-2.32723
60	0.98058	-2.34072
61	0.98057	-2.34089
62	0.98091	-2.33114
63	0.9809	-2.33154
64	0.98082	-2.33388
65	0.98066	-2.33828
66	0.98061	-2.33984
67	0.9806	-2.34021
68	0.98061	-2.3398
69	0.98059	-2.34028
70	0.98058	-2.34081
71	0.98056	-2.3413



**Grafik 4-2**

**Tegangan Tiap – Tiap Bus Terhadap Asumsi Tegangan Akhir  
Sesudah Optimasi**

setelah dilakukan kompensasi dengan pemasangan kapasitor di dapat data pada tabel 4 – 4 dan terjadi perbaikan tegangan, rata-rata besar tegangan pada tiap-tiap bus menjadi diatas 0,975 pu.dapat terlihat pada grafik 4 2

**TABEL 4 - 5 RUGI DAYA  
TIAP SALURAN SEBELUM KOMPENSASI**

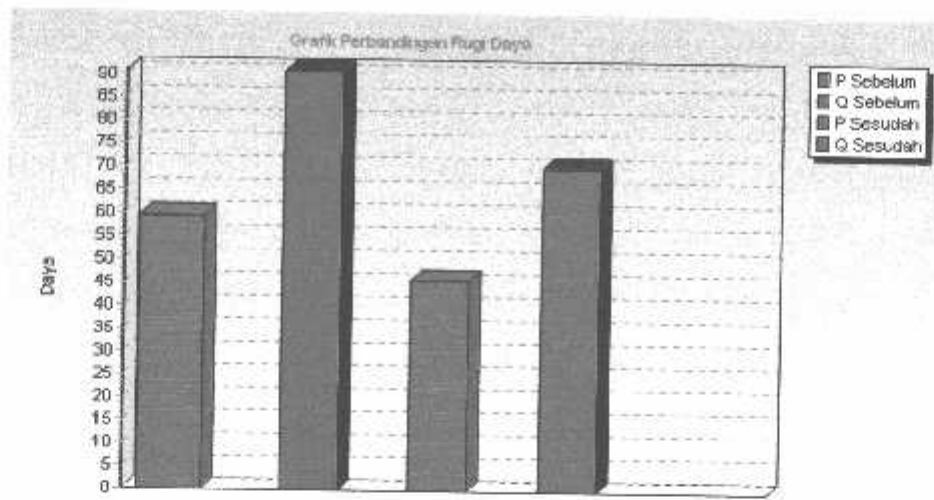
No	Darl	Ke	P (kW)	Q (kVAR)
1	1	2	27.032	41.323
2	2	3	0.002	0.002
3	2	4	22.608	34.561
4	4	5	0.871	1.331
5	4	20	0.948	1.45
6	5	6	0.271	0.415
7	6	7	0.055	0.084
8	7	8	0.008	0.012
9	7	12	0.392	0.599
10	8	9	0.001	0.001
11	9	10	0	0
12	10	11	0	0
13	12	13	0.025	0.038
14	13	14	0.01	0.015
15	14	15	0	0
16	14	16	0.005	0.007
17	16	17	0.006	0.009
18	17	18	0.002	0.004
19	18	19	0.001	0.001
20	20	21	0.637	0.974
21	21	22	0.639	0.977
22	22	23	1.166	1.783
23	23	24	0.664	1.015
24	24	25	0.644	0.984
25	25	28	0.5	0.765
26	26	27	0.289	0.442
27	27	28	0	0
28	27	29	0.231	0.352
29	29	30	0.564	0.862
30	30	31	0.163	0.249
31	31	32	0.017	0.025
32	31	34	0.065	0.099
33	32	33	0.021	0.032
34	34	35	0.02	0.031
35	34	37	0.933	1.426
36	35	36	0.003	0.004
37	37	38	0.111	0.169
38	38	39	0.04	0.061
39	39	40	0.001	0.001
40	39	41	0.054	0.083
41	41	42	0.225	0.344
42	42	43	0.004	0.007
43	42	44	0.018	0.027

44	42	49	0.115	0.176
45	44	45	0.005	0.008
46	45	46	0	0
47	45	47	0.005	0.008
48	47	48	0.003	0.005
49	49	50	0	0
50	49	51	0.088	0.135
51	51	52	0.035	0.053
52	52	53	0.001	0.001
53	52	54	0.027	0.041
54	54	55	0	0.001
55	54	62	0.027	0.042
56	55	56	0	0.001
57	55	57	0	0
58	57	58	0	0
59	57	59	0	0
60	60	61	0	0
61	60	69	0	0.001
62	62	63	0	0
63	62	64	0.014	0.022
64	64	65	0.017	0.026
65	65	66	0.002	0.004
66	65	88	0.004	0.005
67	66	67	0	0
68	68	69	0.001	0.001
69	68	70	0.001	0.002
70	70	71	0	0

**TABEL 4 - 6 RUGI DAYA  
TIAP SALURAN SESUDAH KOMPENSASI**

No	Dari	Ke	P (kW)	Q (kVAR)
1	1	2	19.256	29.435
2	2	3	16.098	24.609
3	2	4	0.002	0.002
4	4	5	0.769	1.176
5	4	20	0.262	0.4
6	5	6	0.053	0.081
7	6	7	0.378	0.578
8	7	8	0.001	0.001
9	7	12	0	0
10	8	9	0	0
11	9	10	0	0
12	10	11	0.024	0.037
13	12	13	0.009	0.014
14	13	14	0.005	0.007
15	14	15	0	0
16	14	16	0.006	0.009
17	16	17	0.002	0.004
18	17	18	0.001	0.001
19	18	19	0.001	0.001
20	20	21	0.526	0.805
21	21	22	0.534	0.817
22	22	23	0.99	1.513
23	23	24	0.577	0.882
24	24	25	0.572	0.875
25	25	26	0.462	0.706
26	26	27	0.278	0.424
27	27	28	0.222	0.339
28	27	29	0	0
29	29	30	0.548	0.838
30	30	31	0.177	0.271
31	31	32	0.092	0.14
32	31	34	0.02	0.03
33	32	33	0.02	0.03
34	34	35	1.889	2.887
35	34	37	0.003	0.004
36	35	36	0.003	0.004
37	37	38	0.238	0.364
38	38	39	0.086	0.132
39	39	40	0.129	0.198
40	39	41	0.001	0.001
41	41	42	0.212	0.324
42	42	43	0.108	0.166

43	42	44	0.004	0.006
44	42	49	0.005	0.008
45	44	45	0.005	0.008
46	45	46	0	0
47	45	47	0.003	0.004
48	47	48	0.003	0.004
49	49	50	0.083	0.127
50	49	51	0	0
51	51	52	0.033	0.05
52	52	53	0.025	0.039
53	52	54	0.001	0.001
54	54	55	0.026	0.039
55	54	62	0	0
56	55	56	0	0.001
57	55	57	0	0
58	57	58	0	0
59	57	59	0	0
60	60	61	0	0.001
61	60	69	0	0
62	62	63	0.013	0.02
63	62	64	0	0
64	64	65	0.016	0.025
65	65	66	0.003	0.005
66	65	68	0	0
67	66	67	0	0
68	68	69	0.001	0.002
69	68	70	0.001	0.001
70	70	71	0	0



**Grafik 4-3**

**Rugi-Rugi Saluran Sebelum Dan Sesudah Kompensasi**

Dari Tabel 4 – 6 dan 4 – 7 dapat disimpulkan pada grafik 4 - terlihat bahwa terjadi pengurangan rugi daya saluran, baik daya aktif maupun daya reaktif mengalami penurunan setelah kapasitor dipasang. Dan untuk biaya sebelum pemasangan kapasitor sebesar 22,094.733 US\$, setelah pemasangan 16,710.859US\$ berarti memiliki nilai efisiensi biaya sebesar 5,383.873 US\$.

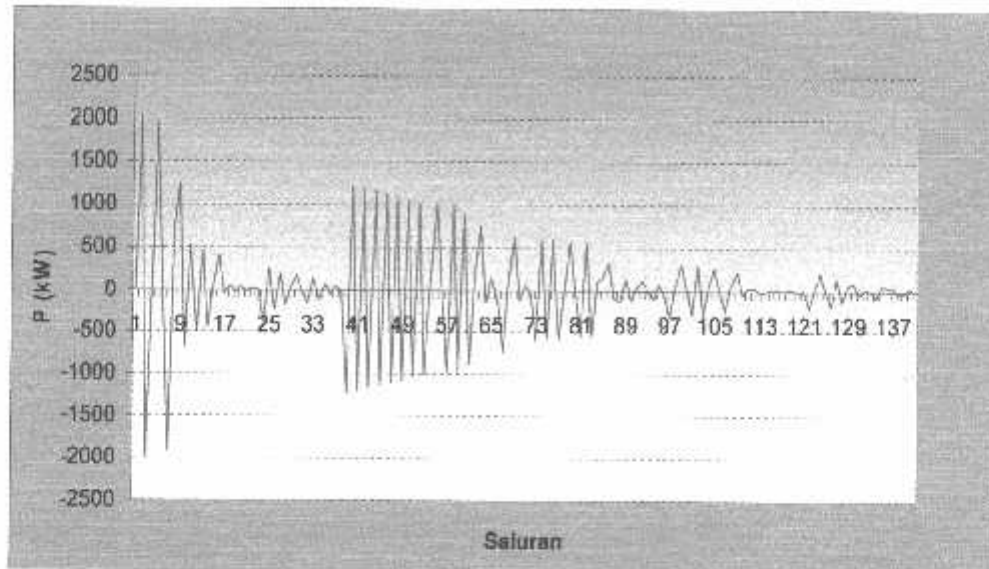


**TABEL 4 – 7 ALIRAN DAYA  
SEBELUM PEMASANGAN KAPASITOR**

No	Dari	Ke	P (kWatt)	Q (kVAR)					
1	1	2	2028.33	1258.718	44	23	22	-1149.69	-686.219
2	2	1	-2001.3	-1217.395	45	23	24	1116.146	666.318
3	2	3	55.902	33.171	46	24	23	-1115.48	-665.303
4	2	4	1945.397	1184.223	47	24	25	1086.672	648.208
5	3	2	-55.9	-33.169	48	25	24	-1086.03	-647.224
6	4	2	-1922.79	-1149.663	49	25	26	1043.372	621.913
7	4	5	692.312	412.333	50	26	25	-1042.87	-621.149
8	4	20	1230.477	737.329	51	26	27	1003.312	597.675
9	5	4	-691.441	-411.002	52	27	26	-1003.02	-597.233
10	5	6	512.991	305.116	53	27	28	2.15	1.276
11	6	5	-512.72	-304.702	54	27	29	1000.872	595.957
12	6	7	448.048	266.328	55	28	27	-2.15	-1.276
13	7	6	-447.993	-266.244	56	29	27	-1000.64	-595.605
14	7	8	57.199	33.948	57	29	30	989.462	588.971
15	7	12	390.795	232.296	58	30	29	-988.898	-588.109
16	8	7	-57.191	-33.936	59	30	31	898.082	534.222
17	8	9	38.701	22.965	60	31	30	-897.92	-533.974
18	9	8	-38.7	-22.964	61	31	32	149.247	88.593
19	9	10	19.78	11.737	62	31	34	748.672	445.38
20	10	9	-19.78	-11.737	63	32	31	-149.231	-88.568
21	10	11	8.6	5.103	64	32	33	133.321	79.127
22	11	10	-8.6	-5.103	65	33	32	-133.3	-79.096
23	12	7	-390.403	-231.697	66	34	31	-748.608	-445.282
24	12	13	239.903	142.396	67	34	35	138.999	82.499
25	13	12	-239.878	-142.357	68	34	37	609.609	362.783
26	13	14	186.214	110.515	69	35	34	-138.979	-82.468
27	14	13	-186.204	-110.5	70	35	36	55.043	32.663
28	14	15	21.07	12.503	71	36	35	-55.04	-32.659
29	14	16	165.134	97.998	72	37	34	-608.676	-361.357
30	15	14	-21.07	-12.502	73	37	38	589.756	350.131
31	16	14	-165.129	-97.99	74	38	37	-589.645	-349.962
32	16	17	118.689	70.435	75	38	39	584.967	347.186
33	17	16	-118.683	-70.425	76	39	38	-584.927	-347.125
34	17	18	73.103	43.38	77	39	40	26.661	15.82
35	18	17	-73.101	-43.376	78	39	41	558.267	331.305
36	18	19	33.541	19.902	79	40	39	-26.66	-15.819
37	19	18	-33.54	-19.901	80	41	39	-558.213	-331.222
38	20	4	-1229.53	-735.879	81	41	42	546.603	324.333
39	20	21	1196.848	716.488	82	42	41	-546.378	-323.99
40	21	20	-1196.21	-715.514	83	42	43	90.82	53.894
41	21	22	1175.141	703.012	84	42	44	140.641	83.481
42	22	21	-1174.5	-702.035	85	42	49	314.916	186.615
43	22	23	1150.852	688.002	86	43	42	-90.816	-53.887
					87	44	42	-140.624	-83.454

88	44	45	122.564	72.738
89	45	44	-122.559	-72.73
90	45	46	18.49	10.972
91	45	47	104.068	61.758
92	46	45	-18.49	-10.971
93	47	45	-104.063	-61.75
94	47	48	55.043	32.663
95	48	47	-55.04	-32.659
96	49	42	-314.801	-186.439
97	49	50	15.48	9.185
98	49	51	299.321	177.254
99	50	49	-15.48	-9.185
100	51	49	-299.233	-177.119
101	51	52	291.493	172.526
102	52	51	-291.458	-172.473
103	52	53	33.541	19.903
104	52	54	257.917	152.571
105	53	52	-33.54	-19.901
106	54	52	-257.89	-152.529
107	54	55	44.801	26.027
108	54	62	213.089	126.502
109	55	54	-44.801	-26.026
110	55	56	23.3	13.268
111	55	57	21.5	12.758
112	56	55	-23.3	-13.268
113	57	55	-21.5	-12.758
114	57	58	12.47	7.4
115	57	59	9.03	5.358
116	58	57	-12.47	-7.399
117	59	57	-9.03	-5.358
118	60	61	10.75	6.379
119	60	69	-29.67	-17.605
120	61	60	-10.75	-6.379
121	62	54	-213.062	-126.461
122	62	63	19.35	11.482
123	62	64	193.712	114.979
124	63	62	-19.35	-11.482
125	64	62	-193.697	-114.957
126	64	65	144.677	85.87
127	65	64	-144.66	-85.844
128	65	66	57.193	33.938
129	65	68	87.468	51.906
130	66	65	-57.19	-33.935
131	66	67	12.47	7.399
132	67	66	-12.47	-7.399
133	68	65	-87.464	-51.901

134	68	69	48.161	28.578
135	68	70	39.303	23.322
136	69	60	29.67	17.606
137	69	68	-48.16	-28.577
138	70	68	-39.302	-23.321
139	70	71	23.392	13.88
140	71	70	-23.392	-13.88



**Grafik 4 – 4**  
**Aliran Daya Sebelum Pemasangan Kapassitor**

Tabel dan grafik aliran daya ini Hasil dari Load flow Awal dan untuk mengetahui Arus yang mengalir pada saluran dan daya tiap saluran sebelum optimasi, dimana arah aliran arus pada tiap saluran tergantung pada suplay tiap bus seperti contoh:

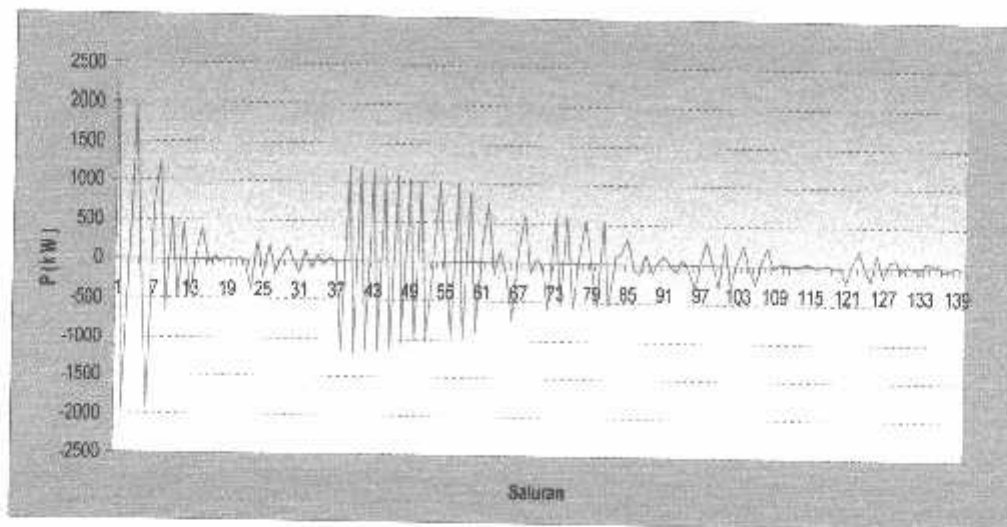
Pada saluran No.1 di suplay dari 2 bus,yakni bus No.1 dan bus No.2 dimana urutan dari No saluran adalah dari bus 1 ke bus 2 ( saluran 1 = bus1 – bus2, atau saluran 1 = - bus2 + bus1) sehingga apabila saluran berharga minus(-) maka suplay daya pada saluran No.1 adalah dari bus2 ke bus1

**TABEL 4 – 8 ALIRAN DAYA  
SESUDAH PEMASANGAN KAPASITOR**

No	Dari	Ke	P (kWatt)	Q (kVAR)
1	1	2	2014.393	37.412
2	2	1	-1995.14	-7.977
3	2	3	55.902	33.171
4	2	4	1939.236	-25.195
5	3	2	-55.9	-33.169
6	4	2	-1923.14	49.803
7	4	5	892.255	412.246
8	4	20	1230.883	-462.049
9	5	4	-691.414	-410.961
10	5	6	512.964	305.075
11	6	5	-512.703	-304.675
12	6	7	448.031	266.301
13	7	6	-447.978	-266.22
14	7	8	57.198	33.947
15	7	12	390.779	232.273
16	8	7	-57.191	-33.936
17	8	9	38.701	22.965
18	9	8	-38.7	-22.964
19	9	10	19.78	11.737
20	10	9	-19.78	-11.737
21	10	11	8.6	5.103
22	11	10	-8.6	-5.103
23	12	7	-390.401	-231.695
24	12	13	239.901	142.393
25	13	12	-239.877	-142.356
26	13	14	186.213	110.514
27	14	13	-186.204	-110.5
28	14	15	21.07	12.503
29	14	16	165.133	97.997
30	15	14	-21.07	-12.502
31	16	14	-165.129	-97.99
32	16	17	118.689	70.434
33	17	16	-118.683	-70.425
34	17	18	73.103	43.379
35	18	17	-73.101	-43.376
36	18	19	33.541	19.902
37	19	18	-33.54	-19.901
38	20	4	-1230.11	463.225
39	20	21	1197.434	-482.616
40	21	20	-1196.91	483.421
41	21	22	1175.838	-495.923
42	22	21	-1175.3	496.74
43	22	23	1151.653	-510.773
44	23	22	-1150.66	512.286
45	23	24	1117.123	-532.188
46	24	23	-1116.55	533.07
47	24	25	1087.736	-550.165
48	25	24	-1087.16	551.039
49	25	26	1044.508	-576.35
50	26	25	-1044.05	577.055
51	26	27	1004.487	-600.529
52	27	26	-1004.21	600.953
53	27	28	2.15	1.276
54	27	29	1002.059	-602.229
55	28	27	-2.15	-1.276
56	29	27	-1001.84	602.568
57	29	30	990.657	-609.202
58	30	29	-990.109	610.04
59	30	31	899.293	-663.927
60	31	30	-899.116	664.198
61	31	32	149.246	88.59
62	31	34	749.87	-752.789
63	32	31	-149.23	-88.566
64	32	33	133.32	79.126
65	33	32	-133.3	-79.096
66	34	31	-749.778	752.929
67	34	35	138.998	82.497
68	34	37	610.781	-835.425
69	35	34	-138.979	-82.468
70	35	36	55.043	32.663
71	36	35	-55.04	-32.659
72	37	34	-608.892	838.313
73	37	38	589.972	-849.539
74	38	37	-589.734	849.903
75	38	39	585.055	-852.679
76	39	38	-584.969	852.811
77	39	40	26.661	15.82
78	39	41	558.308	-868.631
79	40	39	-26.66	-15.819
80	41	39	-558.179	868.829
81	41	42	546.569	324.282
82	42	41	-546.357	-323.958
83	42	43	90.82	53.893
84	42	44	140.64	83.478
85	42	49	314.897	186.586
86	43	42	-90.816	-53.887
87	44	42	-140.623	-83.453

88	44	45	122.563	72.737
89	45	44	-122.558	-72.729
90	45	46	18.49	10.972
91	45	47	104.068	61.758
92	46	45	-18.49	-10.971
93	47	45	-104.063	-61.75
94	47	48	55.043	32.663
95	48	47	-55.04	-32.659
96	49	42	-314.789	-186.42
97	49	50	15.48	9.185
98	49	51	299.309	177.235
99	50	49	-15.48	-9.185
100	51	49	-299.225	-177.107
101	51	52	291.485	172.515
102	52	51	-291.453	-172.465
103	52	53	33.541	19.903
104	52	54	257.912	152.562
105	53	52	-33.54	-19.901
106	54	52	-257.886	-152.523
107	54	55	44.801	26.027
108	54	62	213.085	126.497
109	55	54	-44.801	-26.026
110	55	56	23.3	13.268
111	55	57	21.5	12.758
112	56	55	-23.3	-13.268
113	57	55	-21.5	-12.758
114	57	58	12.47	7.399
115	57	59	9.03	5.358
116	58	57	-12.47	-7.399
117	59	57	-9.03	-5.358
118	60	61	10.75	6.379
119	60	69	-29.67	-17.605
120	61	60	-10.75	-6.379
121	62	54	-213.06	-126.457
122	62	63	19.35	11.482
123	62	64	193.709	114.976
124	63	62	-19.35	-11.482
125	64	62	-193.696	-114.955
126	64	65	144.676	85.868
127	65	64	-144.66	-85.844
128	65	66	57.192	33.938
129	65	68	87.468	51.905
130	66	65	-57.19	-33.935
131	66	67	12.47	7.399
132	67	66	-12.47	-7.399
133	68	65	-87.464	-51.9

134	68	69	48.161	28.578
135	68	70	39.303	23.322
136	69	60	29.67	17.606
137	69	68	-48.16	-28.577
138	70	68	-39.302	-23.321
139	70	71	23.392	13.88
140	71	70	-23.392	-13.88

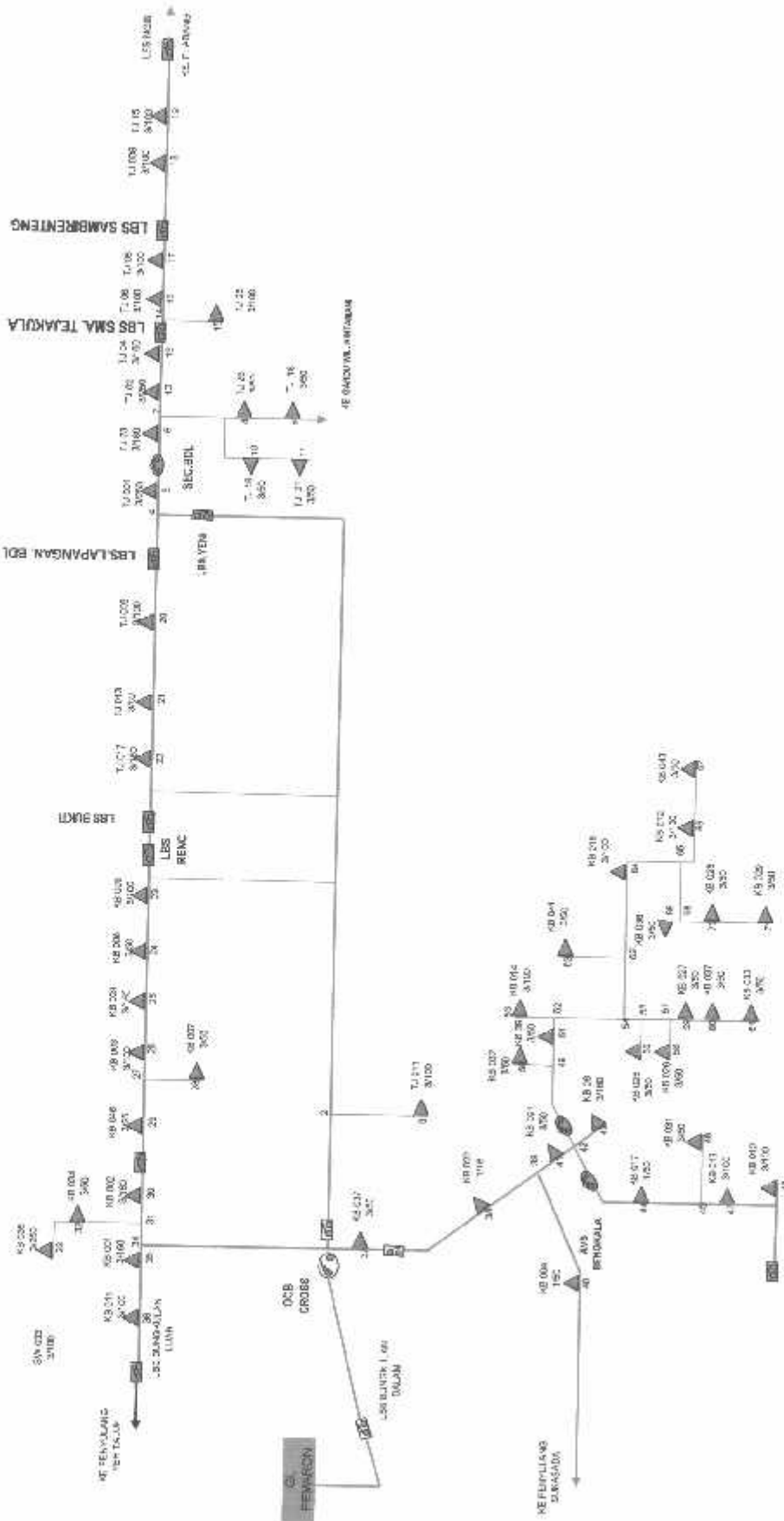


**Grafik 4 – 5**

**Aliran Daya Sesudah Pemasangan Kapasitor**

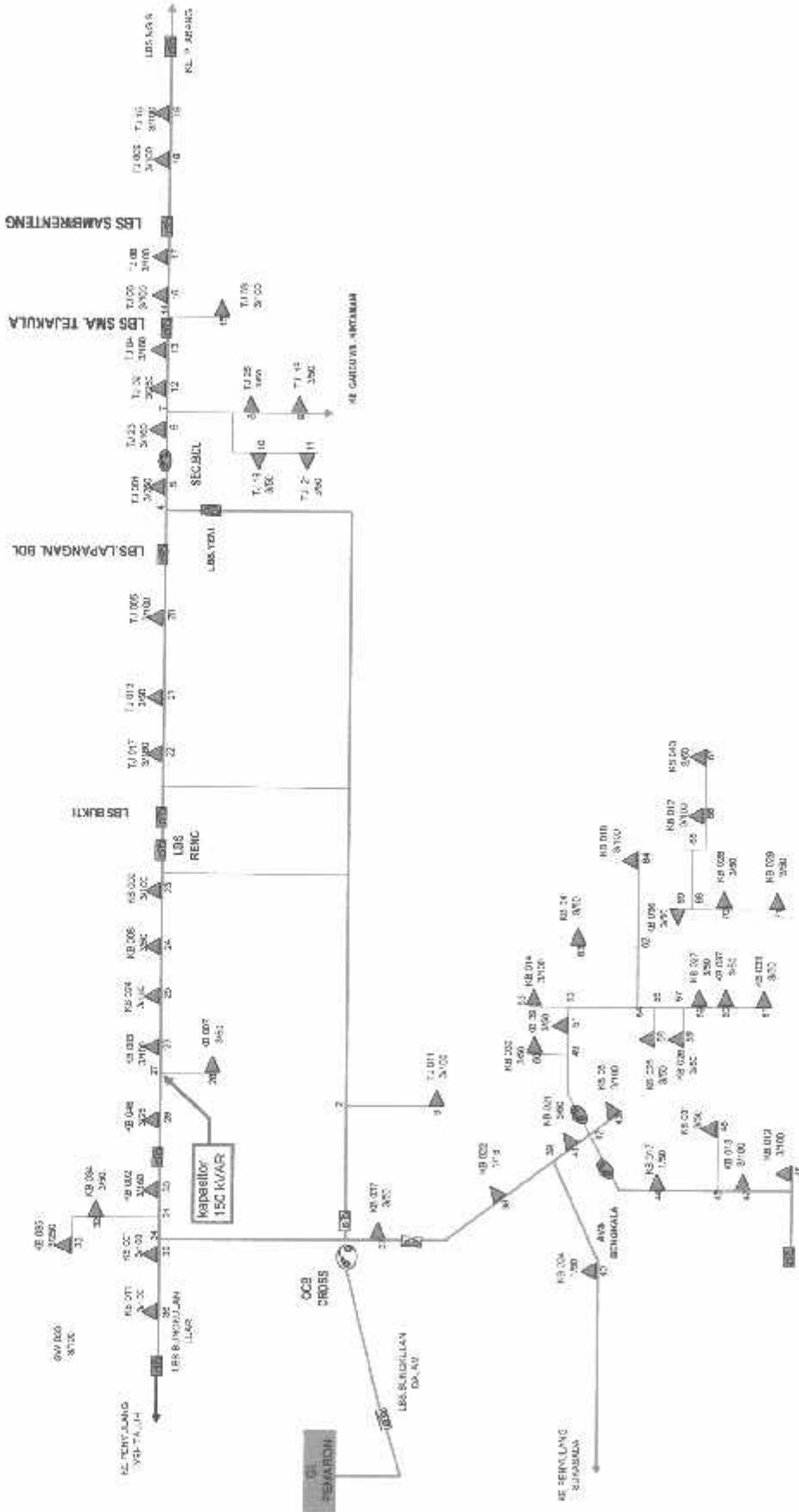
Tabel dan grafik aliran daya sesudah pemasangan kapasitor ini Hasil dari Load flow Akhir dan untuk mengetahui Arus yang mengalir pada saluran dan daya tiap saluran Sesudah optimasi, dimana pembacaan saluran seperti halnya pada pembacaan saluran sebelum di optimasi, hanya yang membedakan adalah nilai praoptimasi dan pascaoptimasi terdapat selisih, dimana pascaoptimasi terjadi penurunan rugi - rugi daya.

PENYULANG TEJAKULA BALI



Gambar 4-7  
Single Diagram Penyulang Tejakula Sebelum Dipasang Kapasitor

PENYULANG TEJAKULA BALI



Gambar 4 - 8  
Single Diagram Penyulang Tejakula Sesudah Dipasang Kapasitor



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Setelah dilakukan analisa penentuan letak dan kapasitas kapasitor dengan metode Algoritma Kruskal Evolutionary pada jaringan distribusi dapat diambil beberapa kesimpulan, antara lain :

1. Kruskal Evolutionary dapat memperbaiki kondisi tegangan dari nilai terendah 0,952 pu menjadi 0,978 pu, dengan kata lain terjadi perbaikan tegangan dengan faktor kenaikan sebesar 2,73%.
2. Rugi-rugi daya mengalami penurunan yaitu untuk daya aktif sebesar 13,937 kW dari 59.590 menjadi 45,653 kW. Sedangkan untuk daya reaktif sebesar 21,306 kVAR dari 91,094 kVAR menjadi 69,788 Kvar, dengan kata lain rugi-rugi daya untuk daya aktif dan daya reaktif terjadi penurunan yaitu sebesar 30,53%.
3. Berdasarkan hasil perhitungan dan analisa dengan metode Algoritma Kruskal Evolutionary, letak dan kapastat kapasitor yang optimal pada *node* 27 dengan kapasitas sebesar 150 kVAR.

#### 5.2. Saran

Untuk pengembangan lebih lanjut diharapkan penggunaan metode Kruskal Evolutionary dapat dikawinkan dengan metode komputasi yang lainnya sehingga output dari optimasi yang dilakukan dapat lebih meningkat.



## DAFTAR PUSTAKA


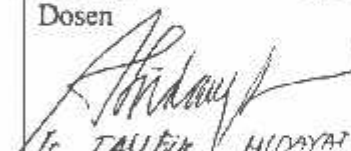
- [1] Hasan Basri, "*Sistem Distribusi Tenaga Listrik*".
- [2] Djiteng Maesudi "*Operasi Sistem Tenaga Listrik*", Balai Penerbit dan Humas ISTN 1990.
- [3] William D. Stevenson, JR, "*Analisis sistem tenaga listrik*", Edisi Keempat, Erlangga, Jakarta 1996.
- [4] Iisiao-Dong Chiang, Jin-Cheng Wang, Jianzong Tong School of Electrical, "*A New Formulation Kruskal Evolutionary Programming Method For Optimal Capacitor Placement In Radial Distribution System*" IEEE Transactions on Power Systems, Vol. 10, No. 1, February 2005

# LAMPIRAN

---



**LEMBAR PERSETUJUAN JUDUL SKRIPSI**  
**JURUSAN TEKNIK ELEKTRO/T. ENERGI LISTRIK S-1**

1.	Nama Mahasiswa : <u>REFI MULANDA</u>	Nim : <u>0112001</u>						
2.	Waktu Pengajuan :	<table border="1"> <tr> <td>Tanggal</td> <td>Bulan</td> <td>Tahun</td> </tr> <tr> <td></td> <td><u>02</u></td> <td><u>2007</u></td> </tr> </table>	Tanggal	Bulan	Tahun		<u>02</u>	<u>2007</u>
Tanggal	Bulan	Tahun						
	<u>02</u>	<u>2007</u>						
Spesifikasi Judul (berilah tanda silang)**)								
3.	<input checked="" type="checkbox"/> a. Sistem Tenaga Elektrik <input type="checkbox"/> b. Energi & Konversi Energi <input type="checkbox"/> c. Tegangan Tinggi & Pengukuran <input type="checkbox"/> d. Sistem Kendali Industri	<input type="checkbox"/> e. Elektronika & Komponen <input type="checkbox"/> f. Elektro Digital & Komputer <input type="checkbox"/> g. Elektronika Komunikasi <input type="checkbox"/> h. Lainnya .....						
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen ) :  <u>Ir. Taufik Hidayat, MT</u>	Ketua Jurusan  <u>Ir. Yudi Limpraptono, MT</u> NIP. P. 1039500274						
5.	Judul yang diajukan Mahasiswa :	<u>PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENJULANG TEBAKULA BALI.</u>						
6.	Perubahan Judul yang Disetujui Dosen materi bidang ilmu	..... ..... .....						
7.	Catatan : ..... ..... .....							
8.	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui Dosen - - 200  <u>Ir. TAUFIK HIDAYAT, MT</u>						

**Perhatian :**

1. Formulir Pengajuan ini harap dikembalikan ke Jurusan paling lambat *satu minggu* setelah disetujui Kelompok Dosen Keahlian dengan dilampirkan Proposal Skripsi beserta persyaratan Skripsi sesuai Form. S-1.
2. keterangan : \*) Coret yang tidak perlu  
 \*) dilingkari a, b, c, ....atau f, sesuai bidang Keahlian.

Form.S-2



**INSTITUT TEKNOLOGI NASIONAL**  
Jl. Bendungan Sigura-gura No. 2  
**MALANG**

Lampiran : 1 (satu) berkas

Perihal : **Kesediaan Sebagai**  
**Dosen Pembimbing I**

Kepada : Yth. Bapak Ir. H. Taufik Hidayat, MT.  
Dosen Institut Teknologi Nasional  
**MALANG**

Yang bertanda tangan di bawah ini :

**Nama** : Refi Mulanda  
**Nim** : 01.12.001  
**Jurusan** : Teknik Elektro S-1  
**Konsentrasi** : Teknik Energi Listrik

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama untuk penyusunan Skripsi dengan judul:

**PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK  
MENENTUKAN PENEMPATAN LETAK KAPASITAS  
KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE  
RADIAL DI PENYULANG TEJAKULA BALI**

seperti proposal terlampir.

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

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

Malang, Februari 2007



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

**Ir. F. Yudi Limpraptono, MT**  
Nip.P/103 950 0274

Hormat kami,

**Refi Mulanda**  
Nim 01.12.001

Form S-3a



**INSTITUT TEKNOLOGI NASIONAL**  
Jl. Bendungan Sigura-gura No. 2  
**MALANG**

Lampiran : 1 (satu) berkas

Perihal : **Kesediaan Sebagai**  
**Dosen Pembimbing II**

Kepada : Yth. Ibu Irrine Budi S.,ST, MT.  
Dosen Institut Teknologi Nasional  
**MALANG**

Yang bertanda tangan di bawah ini :

**Nama** : Refi Mulanda  
**Nim** : 01.12.001  
**Jurusan** : Teknik Elektro S-1  
**Konsentrasi** : Teknik Energi Listrik

Dengan ini mengajukan permohonan, kiranya Ibu bersedia menjadi Dosen Pembimbing pendamping untuk penyusunan Skripsi dengan judul:

**PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN PENEMPATAN LETAK KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG TEJAKULA BALI**

seperti proposal terlampir.

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

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

Malang, Februari 2007



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

**F. Yudi Limpraptono, MT**  
Nip.P. 103 950 0274

Hormat kami,

  
**Refi Mulanda**  
Nim 01.12.001

Form S-3a



**PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI**

Sesuai permohonan dari mahasiswa :

Nama : Refi Mulanda  
Nim : 01.12.001  
Semester : XII  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Energi Listrik

Dengan ini Menyatakan bersedia Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN PENEMPATAN LETAK KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG TEJAKULA BALI**

Demikian surat pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, Februari 2007

**Kami yang Membuat pernyataan**

**Ir. H. Taufik Hidayat, MT.**  
Nip. P. 101 870 0015

**Catatan :**

Setelah disetujui agar formulir ini  
Diserahkan mahasiswa yang  
bersangkutan  
Kepada jurusan untuk diproses  
lebih lanjut

Form S-3b





**PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI**

Sesuai permohonan dari mahasiswa :

Nama : Refi Mulanda  
Nim : 01.12.001  
Semester : XII  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Energi Listrik

Dengan ini Menyatakan bersedia Membimbing Skripsi dari mahasiswa tersebut,  
dengan judul :

**PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK  
MENENTUKAN PENEMPATAN LETAK KAPASITAS KAPASITOR PADA  
JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG  
TEJAKULA BALI**

Demikian surat pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, Februari 2007

**Kami yang Membuat pernyataan**

**Irrine Budi S.,ST, MT.**  
NIP. 132 314 400

**Catatan :**

Setelah disetujui agar formulir ini  
Diserahkan mahasiswa yang  
bersangkutan  
Kepada jurusan untuk diproses  
lebih lanjut

Form S-3b



## BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika\*)

1.	Nama Mahasiswa: <u>REFI MULANDA</u>	Nim: <u>0112001</u>
2.	Keterangan	Tanggal
	Pelaksanaan	<u>21-02-2007</u>
Waktu		
Tempat		
Ruang:		
Spesifikasi Judul (berilah tanda silang)**)		
3.	<input checked="" type="checkbox"/> a. Sistem Tenaga Elektrik	<input type="checkbox"/> e. Elektronika & Komponen
	<input type="checkbox"/> b. Energi & Konversi Energi	<input type="checkbox"/> f. Elektronika Digital & Komputer
	<input type="checkbox"/> c. Tegangan Tinggi & Pengukuran	<input type="checkbox"/> g. Elektronika Komunikasi
	<input type="checkbox"/> d. Sistem Kendali Industri	<input type="checkbox"/> h. lainnya .....
4.	Judul Proposal yang diseminarkan Mahasiswa	<u>PENGEUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG TEJAKULA BALI.</u>
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	..... ..... .....
6.	Catatan: .....	
Catatan: .....		
Persetujuan Judul Skripsi		
7.	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II
	 <u>Ir. Djogo Priatmoko, IUT</u>	 <u>Ir. H. Tawfik Hidayat, MT</u>
	Mengetahui, Ketua Jurusan	Disetujui, Calon Dosen Pembimbing ybs
	 <u>Ir. F. Yudi Limpraptono, MT</u> NIP. B. 1039500274	 <u>Ir. H. Tawfik Hidayat, MT</u>

Perhatian:

1. Keterangan: \*) Coret yang tidak perlu

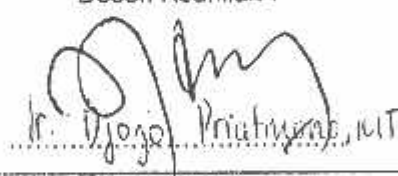



\*\*) dilingkari a, b, c, .....atau g sesuai bidang keahlian

Form S-3c



## BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika\*)

1.	Nama Mahasiswa: <u>REFI MULANDA</u>	Nim: <u>0112001</u>
2.	Keterangan	Tanggal
	Pelaksanaan	<u>21-02-2007</u>
Waktu		
Tempat		
Ruang:		
Spesifikasi Judul (berilah tanda silang)**)		
3.	<input checked="" type="checkbox"/> a. Sistem Tenaga Elektrik	<input type="checkbox"/> e. Elektronika & Komponen
	<input type="checkbox"/> b. Energi & Konversi Energi	<input type="checkbox"/> f. Elektronika Digital & Komputer
	<input type="checkbox"/> c. Tegangan Tinggi & Pengukuran	<input type="checkbox"/> g. Elektronika Komunikasi
	<input type="checkbox"/> d. Sistem Kendali Industri	<input type="checkbox"/> h. lainnya .....
4.	Judul Proposal yang diseminarkan Mahasiswa	<u>PENGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG TEJAKULA BALI</u>
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	..... ..... .....
6.	Catatan: .....	
	.....	
Catatan: .....		
Persetujuan Judul Skripsi		
7.	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II
	 <u>Ir. Djono Priatnyana, MT</u>	 .....
	Mengetahui, Ketua Jurusan.	Disetujui, Calon Dosen Pembimbing ybs
	 <u>Ir. F. Yudi Impraptono, MT</u> NIP. P. 1039500274	 <u>Irtine Budi S. ST, MT</u>

Perhatian:

1. Keterangan: \*) Coret yang tidak perlu  
\*\*) dilingkari a, b, c, ..... atau g sesuai bidang keahlian

Form S-3c



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

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

PT (PERSERO) MALANG  
BANK NIAGA MALANG

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

Nomor : ITN-101/ITA/2/07  
Lampiran : -  
Perihal : **BIMBINGAN SKRIPSI**  
Malang, 26 Februari 2007  
Kepada : Yth. Sdt. Ir. **H. TAUFIK HIDAYAT, MT**  
Dosen Pembimbing  
Jurusan T. Elektro S-1  
di  
Malang

Dengan hormat  
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : REFI MULANDA  
Nim : 0112001  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro S-1  
Konsentrasi : T. Energi Listrik

Maka dengan ini pembimbing tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Februari 2007 s/d 21 Agustus 2007

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik,  
Jurusan Teknik Elektro S-1  
Demikian atas perhatian serta kerjasamanya yang baik kami sampaikan terima kasih.



Ketua Jurusan  
Teknik Elektro S-1

Ir. F. Yudi Limbaptono, MT  
NIP. Y. 1939500274

Tembusan Kepada Yth:

1. Mahasiswa Yang Bersangkutan
2. Arsip

Form S-4a



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

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

NI (PERSERO) MALANG  
BANK NIAGA MALANG

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

Nomor : ITN-102/LTA/2/07  
Lampiran : -  
Perihal : **BIMBINGAN SKRIPSI**  
Kepada : Yth. Sdr. **IRRINE BUDI S, ST, MT**  
Dosen Pembimbing  
Jurusan T. Elektro S-1  
di  
Malang

Malang, 26 Februari 2007

Dengan hormat  
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : REFI MULANDA  
Nim : 0112001  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro S-1  
Konsentrasi : T. Energi Listrik

Maka dengan ini pembimbing tersebut kami serahkan sepenuhnya kepada Saudara/I selama masa waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Februari 2007 s/d 21 Agustus 2007

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik,  
Jurusan Teknik Elektro S-1  
Demikian atas perhatian serta kerjasannya yang baik kami sampaikan terima kasih.



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

- Tembusan Kepada Yth:
1. Mahasiswa Yang Bersangkutan
  2. Arsip

Form S-4a



### FORMULIR BIMBINGAN SKRIPSI

Nama : REFI MULANDA  
Nim : 01.12.001  
Masa Bimbingan : 21 Februari 2007 s/d 21 Agustus 2007  
Judul Skripsi : PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG TEJAKULA BALI

No.	Tanggal	Uraian	Paraf Pembimbing
1.	28/02/07	Sediarhan bab IV	Al
2.	20/04/07	Revisi bab IV Analisa perhitungan busya dipuplus	Al
3.	29/04/07	Lampirkan ke bab V kesimpulan	Al
4.	10/05/07	Lampirkan ke bab III Teori model	Al
5.	25/05/07	Revisi bab III parameter : model dipupulas	Al
6.	29/05/07	Lampirkan ke bab II Teori Kapasitor yg digunakan	Al
7.	05/06/07	Tambah diagram favor penyambungan pemasangan Kapasitor	Al
8.		Revisi lagi :	
9.	18/06/07	ke mengkritik sumbuai	Al
10.	20/08/07	ke mengkritik ulang	Al

Malang,  
Dosen Pembimbing I

  
**H. Taufik Hidayat, MT.**  
Nin.Y. 101 870 0015

Form.S-4b



### FORMULIR BIMBINGAN SKRIPSI

Nama : REFI MULANDA  
Nim : 01.12.001  
Masa Bimbingan : 21 Februari 2007 s/d 21 Agustus 2007  
Judul Skripsi : PENGGUNAAN METODE KRUSKAL EVOLUTIONARY UNTUK MENENTUKAN LETAK DAN KAPASITAS KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER TIPE RADIAL DI PENYULANG TEJAKULA BALI

No.	Tanggal	Uraian	Paraf Pembimbing
1.	4 April 07	Pendahuluan bab II. Uraian rumus table sumber gbr ! Daftar pustaka	
2.	16 Juli 07	Bab I revisi pendahuluan, kenapa sist distribusi radial & KG yg di pakai	
3.	17 Juli 07	Ace Bab I no pembetulan & screenshot nya. Mengetik ke typen	
4.	23 Juli 07	Bab II no perbaikan gbr faktor, bab III no sist. Distribusi di Tejakula Bali	
5.	26 Juli 07	Bab III gbr faktor blue white	
6.	28 Juli 07	Faktor di perbaikan ! bisa bab III di lengkapi !	
7.	20 Juli 07	Bab III data di susun dlm tabel. Susun bab IV. Bab I ole Ace Bab II	
8.	14 Agust 07	Data di pindah di Bab III Aliran daya coba di Chap ! listing dr metode pengerjaan ditampikan	
9.			
10.			

Malang,  
Dosen Pembimbing II

**Irrine Budi S., ST, MT.**  
NIP. 132 314 400



## PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi Jurusan Teknik Elektro jenjang stars satu (S-1)

Yang diselenggarakan pada:

Hari : Selasa

Tanggal : 04 September 2007

Telah dilakukan perbaikan skripsi oleh :



Nama : REFI MULANDA

NIM : 01.12.001

Jurusan : Teknik Elektro S-1


Konsentrasi : Teknik Energi Listrik

Judul Skripsi : PENGGUNAAN METODE KRUSKAL EVOLUTIONARY  
UNTUK MENENTUKAN LETAK DAN KAPASITAS  
KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER  
TIPE RADIAL DI PENYULANG TEJAKULA BALI

No.	Materi Perbaikan	Paraf
1.	Abstrak, Rumusan Masalah, Tujuan	
2.	Tabel 52 – 59 Buat Grafik dan Uraian (Analisa)	

Telah Diperiksa dan Disetujui :

**Dosen Penguji II,**



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

Mengetahui,

**Dosen Pembimbing I,**



**Ir.H Taufik Hidayat, MT**  
NIP.P. 101 870 0151

**Dosen Pembimbing II,**



**Irrine Budi S. ST, MT**  
NIP. 132 314 400





DATA PEMBEBANAN

No Node	Tegangan Awal		Pembangkitan		Pembelian		Type bus
	V (pu)	$\theta$ (°)	P (kW)	Q (kVAR)	P (kW)	Q (kVAR)	
1	1	0	-	-	0	0	Slack
2	1	0	-	-	0	0	Load
3	1	0	-	-	55.9	33.1691	Load
4	1	0	-	-	0	0	Load
5	1	0	-	-	178.45	105.888	Load
6	1	0	-	-	64.872	38.5741	Load
7	1	0	-	-	0	0	Load
8	1	0	-	-	18.49	10.9713	Load
9	1	0	-	-	18.92	11.2265	Load
10	1	0	-	-	11.18	6.6338	Load
11	1	0	-	-	8.6	5.1029	Load
12	1	0	-	-	150.5	89.3015	Load
13	1	0	-	-	53.664	31.8423	Load
14	1	0	-	-	0	0	Load
15	1	0	-	-	21.07	12.5022	Load
16	1	0	-	-	48.44	27.5559	Load
17	1	0	-	-	45.58	27.0458	Load
18	1	0	-	-	39.56	23.4735	Load
19	1	0	-	-	33.54	19.9015	Load
20	1	0	-	-	32.86	19.3912	Load
21	1	0	-	-	21.07	12.5022	Load
22	1	0	-	-	23.65	14.0331	Load
23	1	0	-	-	33.54	19.9015	Load
24	1	0	-	-	28.81	17.0949	Load
25	1	0	-	-	42.656	25.3106	Load
26	1	0	-	-	39.56	23.4735	Load
27	1	0	-	-	0	0	Load
28	1	0	-	-	2.15	1.2757	Load
29	1	0	-	-	11.18	6.6338	Load
30	1	0	-	-	50.616	30.867	Load
31	1	0	-	-	0	0	Load
32	1	0	-	-	15.91	9.4404	Load
33	1	0	-	-	133.3	79.0956	Load
34	1	0	-	-	0	0	Load
35	1	0	-	-	83.936	49.8047	Load
36	1	0	-	-	55.04	32.6588	Load
37	1	0	-	-	18.92	11.2265	Load
38	1	0	-	-	4.6764	2.776	Load
39	1	0	-	-	0	0	Load
40	1	0	-	-	26.66	15.8191	Load
41	1	0	-	-	11.61	6.889	Load
42	1	0	-	-	0	0	Load
43	1	0	-	-	80.816	50.887	Load
44	1	0	-	-	18.08	10.7162	Load
45	1	0	-	-	0	0	Load
46	1	0	-	-	18.49	10.9713	Load
47	1	0	-	-	49.02	29.0868	Load
48	1	0	-	-	55.04	32.6588	Load
49	1	0	-	-	0	0	Load
50	1	0	-	-	15.48	9.1853	Load
51	1	0	-	-	7.74	4.5926	Load
52	1	0	-	-	0	0	Load
53	1	0	-	-	33.54	19.9015	Load
54	1	0	-	-	0	0	Load
55	1	0	-	-	0	0	Load
56	1	0	-	-	23.3	13.2676	Load
57	1	0	-	-	0	0	Load
58	1	0	-	-	12.47	7.3993	Load
59	1	0	-	-	9.03	5.3581	Load
60	1	0	-	-	18.92	11.2265	Load
61	1	0	-	-	10.75	6.3787	Load
62	1	0	-	-	0	0	Load
63	1	0	-	-	19.35	11.4616	Load
64	1	0	-	-	49.02	29.0868	Load
65	1	0	-	-	0	0	Load
66	1	0	-	-	44.72	26.5353	Load
67	1	0	-	-	12.47	7.3993	Load
68	1	0	-	-	0	0	Load
69	1	0	-	-	18.49	10.9713	Load
70	1	0	-	-	15.91	9.4404	Load
71	1	0	-	-	23.392	13.88	Load

DATA SALURAN TEJAKULA

No Saluran	Dari Node	Ke Node	Panjang (m)	Impedansi Saluran			
				R (Ohm)	X (Ohm)	R (pu)	X (pu)
1	1	2	8,776.50	1.8974793	2.90053325	0.000332059	0.000507811
2	2	3	650.1111	0.140554022	0.214861722	0.000024597	0.000037601
3	2	4	7,765.22	1.87883971	2.566403904	0.000293797	0.000449121
4	4	5	2,311.51	0.499747635	0.76395279	0.000087456	0.000133692
5	4	20	794.5802	0.171788249	0.262608772	0.000030063	0.000046957
6	5	6	1,307.45	0.282689756	0.432110797	0.000049487	0.000075615
7	6	7	346.7259	0.074962145	0.114592919	0.000013118	0.000020054
8	7	8	2,961.62	0.640301657	0.978814512	0.000112053	0.000171293
9	7	12	3,250.56	0.70277011	1.074308611	0.000122985	0.000188004
10	8	9	563.4296	0.121813486	0.186213493	0.000021317	0.000032587
11	9	10	628.4407	0.135868888	0.207699665	0.000023777	0.000038347
12	10	11	1,011.28	0.21863958	0.334229346	0.000036262	0.000058490
13	12	13	548.9827	0.118890063	0.161438788	0.000020771	0.000031752
14	13	14	353.9494	0.076523567	0.110980271	0.000019392	0.000029472
15	14	15	902.9321	0.19521392	0.298419059	0.000034162	0.000052223
16	14	16	216.7037	0.046851341	0.071620574	0.000008199	0.000012534
17	16	17	556.2062	0.120251775	0.18382614	0.000021044	0.000032170
18	17	18	583.4296	0.121813486	0.186213493	0.000021317	0.000032587
19	18	19	635.6642	0.1374306	0.210087017	0.000024050	0.000036765
20	20	21	563.4296	0.121813486	0.186213493	0.000021317	0.000032587
21	21	22	585.1	0.12649802	0.19337555	0.000022137	0.000033841
22	22	23	1,112.41	0.240503549	0.36765228	0.000042088	0.000064339
23	23	24	671.7815	0.145239156	0.22202378	0.000025417	0.000038854
24	24	25	686.2284	0.148362579	0.226798465	0.000025863	0.000039690
25	25	26	577.8765	0.124938909	0.190988198	0.000021864	0.000033423
26	26	27	361.1728	0.078085568	0.119367823	0.000013665	0.000020889
27	27	28	975.1667	0.210831033	0.322292583	0.000036895	0.000056401
28	27	29	288.9383	0.062468454	0.095494099	0.000010632	0.000016711
29	29	30	722.3457	0.156171136	0.238735247	0.000027330	0.000041779
30	30	31	252.821	0.054659898	0.083567336	0.000009565	0.000014623
31	31	32	939.0494	0.203022477	0.310355821	0.000035529	0.000054312
32	31	34	144.4881	0.031234227	0.047747049	0.000005468	0.000008346
33	32	33	1,459.74	0.315465694	0.482245119	0.000055205	0.000084393
34	34	35	1,300.2222	0.281108044	0.429723444	0.000049194	0.000075202
35	34	37	3,142.20	0.879344441	1.038498324	0.000118685	0.000181737
36	35	36	1,191.87	0.257682374	0.393913157	0.000045094	0.000068935
37	37	38	397.2901	0.065894125	0.131304388	0.000015031	0.000022978
38	38	39	144.4681	0.031234227	0.047747049	0.000005468	0.000008356
39	39	40	939.0494	0.203022477	0.310355821	0.000035529	0.000054312
40	39	41	216.7037	0.046851341	0.071520574	0.000008199	0.000012534
41	41	42	939.0494	0.203022477	0.310355821	0.000035529	0.000054312
42	42	43	650.1111	0.140554022	0.214861722	0.000024597	0.000037601
43	42	44	1,112.41	0.240503549	0.36765228	0.000042088	0.000064339
44	42	49	1,444.59	0.312342272	0.477470494	0.000054680	0.000083557
45	44	45	433.4074	0.093702681	0.143241148	0.000016398	0.000025067
46	45	48	794.5802	0.171788249	0.262608772	0.000030063	0.000046957
47	45	47	613.9938	0.132745465	0.20292496	0.000023230	0.000035512
48	47	48	1,213.54	0.262367508	0.401075215	0.000045914	0.000070188
49	49	50	650.1111	0.140554022	0.214861722	0.000024597	0.000037601
50	49	51	1,227.99	0.285490831	0.40584992	0.000046461	0.000071024
51	51	52	505.842	0.109319795	0.167114673	0.000019131	0.000029245
52	52	53	865.8148	0.187405365	0.286482296	0.000032796	0.000050134
53	52	54	505.842	0.109319795	0.167114673	0.000019131	0.000029245
54	54	55	281.7148	0.080906743	0.093106740	0.000010655	0.000016254
55	54	62	744.016	0.16085627	0.245897304	0.000028150	0.000043032
56	55	56	939.0494	0.203022477	0.310355821	0.000035529	0.000054312
57	55	57	650.1111	0.140554022	0.214861722	0.000021597	0.000037601
58	57	58	1,083.52	0.234250704	0.35810287	0.000040985	0.000062668
59	57	59	216.7037	0.046851341	0.071620574	0.000008199	0.000012534
60	59	60	498.4185	0.107768084	0.16472732	0.000016858	0.000026827
61	60	61	505.842	0.109319795	0.167114673	0.000019131	0.000029245
62	62	63	883.4519	0.14992429	0.229185837	0.000026237	0.000040108
63	62	64	469.5247	0.101511238	0.15517791	0.000017784	0.000027156
64	64	65	1,011.28	0.21863958	0.334229346	0.000036262	0.000058490
65	65	66	902.9321	0.19521392	0.298419059	0.000034162	0.000052223
66	65	68	577.8765	0.124938909	0.190988198	0.000021864	0.000033243
67	66	67	998.837	0.216616187	0.329454641	0.000037715	0.000057655
68	68	69	325.0555	0.070277011	0.107430861	0.000012298	0.000018800
69	68	70	852.3679	0.18428194	0.281707591	0.000032249	0.000049299
70	70	71	686.2284	0.148362579	0.226798465	0.000025863	0.000039690

DATA GARDU DAN PEMBEBANAN

No Node	GARDU	KAPASITAS	BEBAN GARDU	BEBAN AKTIF (KW)	BEBAN REAKTIF (KVAR)
1	-	0	0	0.0000	0.0000
2	-	0	0	0.0000	0.0000
3	TJ11	100	65	55.9000	33.1691
4	-	0	0	0.0000	0.0000
5	TJ1	250	83	178.4500	105.8860
6	TJ23	160	47	64.6720	38.3741
7	-	0	0	0.0000	0.0000
8	TJ25	50	43	18.4900	10.9713
9	TJ18	50	44	18.9200	11.2263
10	TJ19	50	28	11.1800	6.6338
11	TJ21	50	20	8.9000	5.1029
12	TJ2	250	70	150.5000	89.3015
13	TJ4	160	30	63.6840	31.8423
14	-	0	0	0.0000	0.0000
15	TJ03	50	49	21.0700	12.5022
16	TJ6	100	54	46.4400	27.5559
17	TJ8	100	53	45.5800	27.0456
18	TJ9	100	46	39.5600	23.4735
19	TJ95	100	39	33.5400	19.9015
20	TJ5	100	38	32.0800	19.3512
21	TJ13	50	49	21.0700	12.5022
22	TJ17	50	55	23.6500	14.0331
23	KB9	100	39	33.5400	19.9015
24	KB8	50	67	28.3100	17.0949
25	KB24	150	31	42.6660	25.3106
26	KB3	100	46	39.5600	23.4735
27	-	0	0	0.0000	0.0000
28	KB7	50	5	2.1500	1.2757
29	KB46	25	62	11.1800	6.6338
30	KB2	160	66	90.8180	53.8670
31	-	0	0	0.0000	0.0000
32	KB34	50	37	15.9100	9.4404
33	KB35	250	62	133.3000	79.0956
34	-	0	0	0.0000	0.0000
35	KB1	160	61	83.9360	49.6047
36	KB11	100	64	55.0400	32.6568
37	KB37	50	44	18.9200	11.2265
38	KB22	16	34	4.6784	2.7750
39	-	0	0	0.0000	0.0000
40	KB4	50	62	26.6900	15.8191
41	KB21	50	27	11.6100	6.8890
42	-	0	0	0.0000	0.0000
43	KB6	160	66	90.8160	53.8670
44	KB17	50	42	18.0300	10.7162
45	-	0	0	0.0000	0.0000
46	KB31	50	43	18.4900	10.9713
47	KB13	100	67	49.0200	29.0868
48	KB12	100	64	55.0400	32.6568
49	-	0	0	0.0000	0.0000
50	KB32	50	36	15.4800	9.1853
51	KB39	50	18	7.7400	4.5926
52	-	0	0	0.0000	0.0000
53	KB14	100	39	33.5400	19.9015
54	-	0	0	0.0000	0.0000
55	-	0	0	0.0000	0.0000
56	KB25	50	52	22.3600	13.2676
57	-	0	0	0.0000	0.0000
58	KB26	50	29	12.4700	7.3993
59	KB27	50	21	9.0300	5.3581
60	KB37	50	44	18.9200	11.2265
61	KB33	50	25	10.7500	6.3787
62	-	0	0	0.0000	0.0000
63	KB41	50	45	19.3500	11.4815
64	KB16	100	67	49.0200	29.0868
65	-	0	0	0.0000	0.0000
66	KB12	100	52	44.7200	26.5353
67	KB43	50	29	12.4700	7.3993
68	-	0	0	0.0000	0.0000
69	KB36	50	43	18.4900	10.9713
70	KB28	50	37	15.9100	9.4404
71	KB29	100	17	23.3820	13.8900



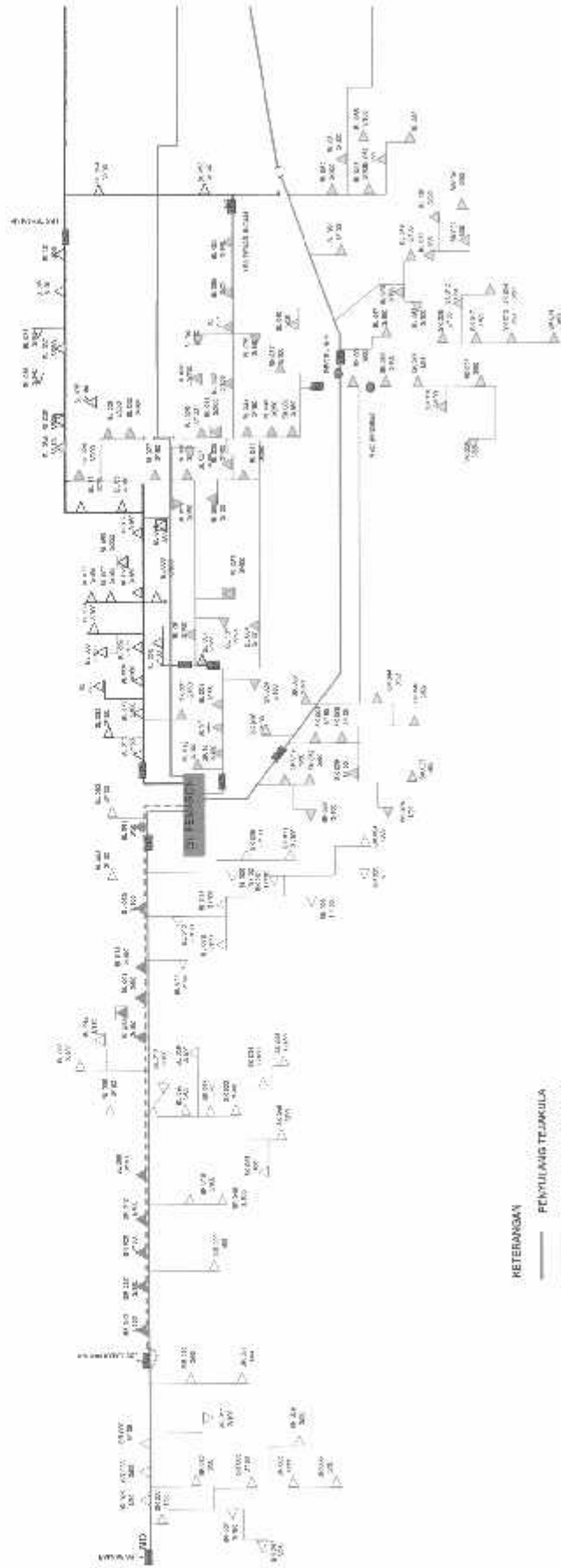
PT. PLN (PERSERO)  
DISTRIBUSI BALI  
AJI BALI UTARA

### DATA GARDU PENYULANG TEJAKULA

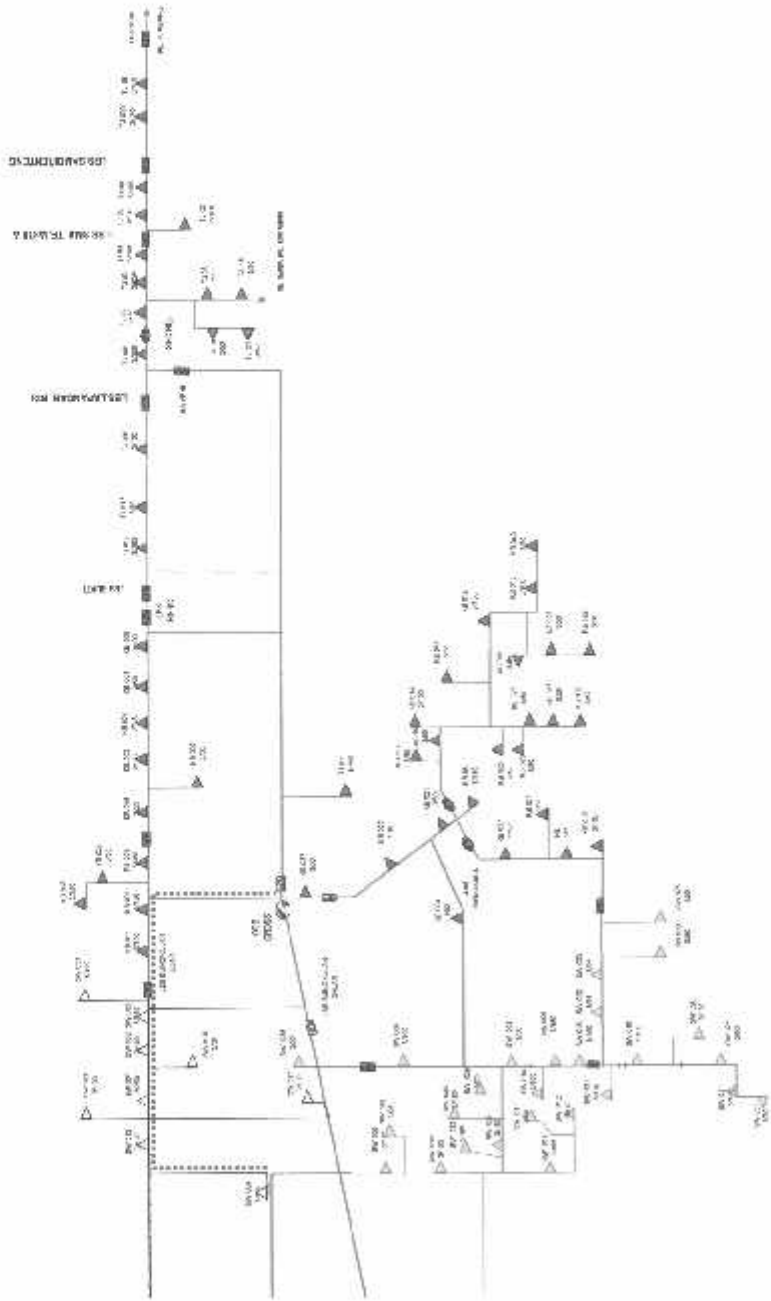
No	GARDU	PENYULANG	GARDU INDUK	LOKASI	KAPASITAS (Kva)	MERK	N	BEBAN GARDU (%)
1	KB011	104	1	DSN. KUBUTAMBAHAN BARAT	100	HICO	T	84
2	KB001	104	1	DS. KUBUTUMBAHAN	160	STARLITE	T	81
3	KB034	104	1	DSN. POJOK TENGAH KBT	50	STARLIT	T	37
4	KB035	104	1	T. UDANG PT. TERUS KBT	250	UNINDO	T	82
5	KB037	104	1	SPBU KUBUTAMBAHAN	50	STARLITE	T	44
6	KB022	104	1	DS. BENGKALA LUAR	16	STARLITE	T	34
7	KB004	104	1	DS. BENGKALA	50	HICO	T	82
8	KB021	104	1	DS. BENGKALA LUAR	50	STARLITE	T	27
9	KB006	104	1	DS. TAMBLANG	160	UNINDO	T	86
10	KB017	104	1	DSN. KAWANAN DS. BILA	50	HICO	T	42
11	KB031	104	1	DSN. BONTIHING KANGINAN	50	SENYANG	T	43
12	KB013	104	1	DSN. BONTIHING	100	UNINDO	T	57
13	KB012	104	1	DS. PAKISAN	100	UNINDO	T	64
14	KB032	104	1	DN. BABAKAN DS. TAMBLANG	50	SENYANG	T	36
15	KB039	104	1	DN. DAUH PURA DN. DEPEHA	50	STARLITE	T	18
16	KB014	104	1	DS. BULIAN TAMBLANG	100	UNINDO	T	39
17	KB025	104	1	DSN. SINGKUNG TAMBLANG	50	STARLITE	T	52
18	KB026	104	1	DSN. TANGKID TAMBLANG	50	STARLIT	T	29
19	KB027	104	1	DSN. KELAMPUAK TAMBLANG	50	STARLITE	T	21
20	KB033	104	1	DSN. TEGAL DS. KB. TAMBHN	50	B. JAYA	T	35
21	KB041	104	1	DSN. BINGIN DS. DEPEHA	50	STARLITE	T	45
22	KB018	104	1	DS. DEPEHA TUNJUNG	100	STARLITE	T	57
23	KB036	104	1	DN. PUDEH TAJUN	50	STARLITE	T	43
24	KB028	104	1	DSN. PUDEH TAJUN	50	UNINDO	T	37
25	KB029	104	1	DSN. BAKUNGAN TAJUN	160	STARLIT	T	17
26	KB019	104	1	DN. PASEK DS. TAJUN	100	STARLIT	T	52
27	KB043	104	1	DSN. BAYAD DS. TAJUN	50	STARLITE	T	29
28	KB002	104	1	DSN. CEMARA KUBU TAMB	160	HICO	T	66
29	KB046	104	1	DSN. CEMARA KUBUTAMBAHAN	25	UNINDO	T	52
30	KB007	104	1	SPMA. KUBUTAMBAHAN	50	UNINDO	T	5
31	KB003	104	1	AIRSANIH	100	UNINDO	T	48
32	KB024	104	1	PDAM AIRSANIH	160	STARLITE	T	31
33	KB008	104	1	DSN. SANIH DS. BUKTI	50	STARLITE	T	67
34	KB009	104	1	DSN. BUKTI DESA BUKTI	100	UNINDO	T	39
35	TJ017	104	1	DS. PAJUNG/ALASARI T.KL	50	B. JAYA	T	55
36	TJ013	104	1	DS. PACUNG TEJAKULA	50	SINYANG	T	49
37	TJ011	104	1	DS. SEMBIRANTEJAKULA	100	HICO	T	65
38	TJ005	104	1	DS. PACUNG TEJAKULA	100	UNINDO	T	38
39	TJ001	104	1	DS. BONDALEM TEJAKULA	250	BB. JAYA	T	83
40	TJ023	104	1	DS. BONDALEM TEJAKULA	160	STARLITE	T	47
41	TJ025	104	1	DSN. KELODAN MADENAN	50	STARLITE	T	43
42	TJ018	104	1	DSN. KEDURAN MADENAN	50	STARLITE	T	44
43	TJ019	104	1	DN SANGAMBU DS. MADENAN	50	STARLITE	T	26
44	TJ021	104	1	DSN. GENTUH MADENAN	50	STARLITE	T	20
45	TJ002	104	1	DS. TEJAKULA	250	BB. JAYA	T	70
46	TJ004	104	1	DS. TEJAKULA	160	UNINDO	T	89
47	TJ006	104	1	DS. LES TEJAKULA	100	TRAFINDO	T	54
48	TJ013	104	1	DS. PACUNG TEJAKULA	50	SINYANG	T	49
49	TJ008	104	1	DS. LES TEJAKULA	100	TRAFINDO	T	54
50	TJ008	104	1	DN/DS. SAMBIRENTENG	100	TRAFINDO	T	53
51	TJ009	104	1	DS. JULAH TEJAKULA	100	TRAFINDO	T	46
52	TJ015	104	1	DS. TEMBOK TEJAKULA	100	TRAFINDO	T	39



PT. PLN ( PERSERO )  
DISTRIBUSI BALI  
AJ BALI UTARA



- KETERANGAN
- PENTILANG TEJAKULA
  - PENTILANG BUKASADA
  - PENTILANG LILIGURDI
  - PENTILANG YEN TALUH
  - - - PENTILANG LOYNA
  - PENTILANG SELURAH BAWANG



---

unit Input;

interface

uses

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

type

```
TfrmInput = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  Panel1: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  edtNbus: TEdit;
  edtNsal: TEdit;
  fgBus: TStringGrid;
  fgSaluran: TStringGrid;
  btnClose: TButton;
  btnSave: TButton;
  label3: TLabel;
  Label4: TLabel;
  edtVbase: TEdit;
  Label5: TLabel;
  cmbKonstV: TComboBox;
  cmbKonstP: TComboBox;
  cmbKonstParam: TComboBox;
  edtPbase: TEdit;
  SaveDialog1: TSaveDialog;
  TabSheet4: TTabSheet;
  Label6: TLabel;
  Label7: TLabel;
  edtNCap: TEdit;
  edtKp: TEdit;
  fgCap: TStringGrid;
  StatusBar1: TStatusBar;
  procedure btnCloseClick(Sender: TObject);
  procedure edtNbusChange(Sender: TObject);
  procedure edtNsalChange(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure cmbKonstPChange(Sender: TObject);
  procedure cmbKonstParamChange(Sender: TObject);
  procedure btnSaveClick(Sender: TObject);
```



```

    procedure edtNCapChange(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmInput: TfrmInput;

implementation

uses Hasil, TypeData;

{SR *.dfm}

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

procedure TfrmInput.cdtNbusChange(Sender: TObject);
var i:integer;
begin
    if cdtNbus.Text="" then
        begin
            _fgBus.RowCount:=2;
        end
    else
        begin
            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]:=IntToStr(1);
                    fgBus.Cells[2,i]:=IntToStr(0);
                end;
            end;
        end;
end;

procedure TfrmInput.edtNsalChange(Sender: TObject);
var i:integer;
begin
    if edtNsal.Text="" then
        begin
            fgSaluran.RowCount:=2;

```

---

```

end
else
begin
  fgSaluran.RowCount:=StrToInt(edtNsal.Text)+1;
  for i:=1 to StrToInt(edtNsal.Text) do
  begin
    fgSaluran.Cells[0,i]:=IntToStr(i);
  end;
end;
end;
end;

```

```

procedure TfrmInput.FormCreate(Sender: TObject);
begin
  fgBus.Cells[0,0]:='No';
  fgBus.Cells[1,0]:='Abs V (pu)';
  fgBus.Cells[2,0]:='Sud V (deg)';
  fgBus.Cells[3,0]:='Pgen (MW)';
  fgBus.Cells[4,0]:='Qgen (MVAR)';
  fgBus.Cells[5,0]:='Pload (MW)';
  fgBus.Cells[6,0]:='Qload (MVAR)';
  fgBus.Cells[7,0]:='Cap';
  fgBus.Cells[8,0]:='Typ Bus';
  fgSaluran.Cells[0,0]:='No';
  fgSaluran.Cells[1,0]:='Dari';
  fgSaluran.Cells[2,0]:='Ke';
  fgSaluran.Cells[3,0]:='R (pu)';
  fgSaluran.Cells[4,0]:='X (pu)';
  fgSaluran.Cells[5,0]:='Lc (pu)';
  fgSaluran.Cells[6,0]:='Tr';
  fgCap.Cells[0,0]:='No';
  fgCap.Cells[1,0]:='Cap (kVAR)';
  fgCap.Cells[2,0]:='Harga (US$)';
end;

```

```

procedure TfrmInput.cmbKonstPChange(Sender: TObject);
begin
  if cmbKonstP.Text='VA' then
  begin
    fgBus.Cells[3,0]:='Pgen (W)';
    fgBus.Cells[4,0]:='Qgen (VAR)';
    fgBus.Cells[5,0]:='Pload (W)';
    fgBus.Cells[6,0]:='Qload (VAR)';
  end
  else if cmbKonstP.Text='kVA' then
  begin
    fgBus.Cells[3,0]:='Pgen (kW)';

```

```

    fgBus.Cells[4,0]:='Qgen (kVAR)';
    fgBus.Cells[5,0]:='Pload (kW)';
    fgBus.Cells[6,0]:='Qload (kVAR)';
end
else if cmbKonstP.Text='MVA' then
begin
    fgBus.Cells[3,0]:='Pgen (MW)';
    fgBus.Cells[4,0]:='Qgen (MVAR)';
    fgBus.Cells[5,0]:='Pload (MW)';
    fgBus.Cells[6,0]:='Qload (MVAR)';
end;
end;

procedure TfrmInput.cmbKonstParamChange(Sender: TObject);
begin
    if cmbKonstParam.Text='pu' then
    begin
        fgSaluran.Cells[3,0]:='R (pu)';
        fgSaluran.Cells[4,0]:='X (pu)';
        fgSaluran.Cells[5,0]:='Lc (pu)';
    end
    else if cmbKonstParam.Text='ohm' then
    begin
        fgSaluran.Cells[3,0]:='R (ohm)';
        fgSaluran.Cells[4,0]:='X (ohm)';
        fgSaluran.Cells[5,0]:='Lc (ohm)';
    end;
end;

procedure TfrmInput.btnSaveClick(Sender: TObject);
var NamaFile:string;
    input:TextFile;
    i,typ,dari,ke:integer;
    R,X,Lc,Tr:double;
    absV,sudV,Pgen,Qgen,Pload,Qload,Cap:double;
begin
    KonstP:=1000000;
    KonstV:=1000;
    param:=1;
    if cmbKonstP.Text='VA' then
    begin
        KonstP:=1;
    end
    else if cmbKonstP.Text='kVA' then
    begin
        KonstP:=1000;
    end;
end;

```

---

```
end
else if cmbKonstP.Text='MVA' then
begin
  KonstP:=1000000;
end;
if cmbKonstV.Text='V' then
begin
  KonstV:=1;
end
else if cmbKonstV.Text='kV' then
begin
  KonstV:=1000;
end
else if cmbKonstV.Text='MV' then
begin
  KonstV:=1000000;
end;
if cmbKonstParam.Text='pu' then
begin
  param:=1;
end
else if cmbKonstParam.Text='ohm' then
begin
  param:=2;
end;
if btnSave.Caption='&Save' then
begin
  try
    if SaveDialog1.Execute then
    begin
      Nbus:=StrToInt(edtNbus.Text);
      Nsal:=StrToInt(edtNsal.Text);
      Ncap:=StrToInt(edtNcap.Text);
      Vbase:=StrToFloat(edtVbase.Text);
      Pbase:=StrToFloat(edtPbase.Text);
      Kp:=StrToFloat(edtKp.Text);
      NamaFile:=SaveDialog1.FileName;
      AssignFile(input,NamaFile+'.txt');
      Rewrite(input);
      Writeln(input,Nbus);
      Writeln(input,Nsal);
      Writeln(input,Vbase:6:2);
      Writeln(input,KonstV:7:0);
      Writeln(input,Pbase:6:2);
      Writeln(input,KonstP:7:0);
      Writeln(input,param);
```

```

for i:=1 to Nbus do
begin
  absV:=StrToFloat(fgBus.Cells[1,i]);
  sudV:=-StrToFloat(fgBus.Cells[2,i]);
  Pgen:=StrToFloat(fgBus.Cells[3,i]);
  Qgen:=StrToFloat(fgBus.Cells[4,i]);
  Pload:=StrToFloat(fgBus.Cells[5,i]);
  Qload:=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,' ',Pgen:9:3,' ',Qgen:9:3,
  ' ',Pload:9:3,' ',Qload:9:3,' ',Cap:7:5,' ',Typ);
end;
for i:=1 to Nsal do
begin
  dari:=StrToInt(fgSaluran.Cells[1,i]);
  ke:=StrToInt(fgSaluran.Cells[2,i]);
  R:=-StrToFloat(fgSaluran.Cells[3,i]);
  X:=StrToFloat(fgSaluran.Cells[4,i]);
  Lc:=StrToFloat(fgSaluran.Cells[5,i]);
  Tr:=StrToFloat(fgSaluran.Cells[6,i]);
  Writeln(input,dari,' ',ke,' ',R:7:5,' ',X:7:5,' ',
  Lc:7:5,' ',Tr:7:5);
end;
Writeln(input,Ncap);
Writeln(input,Kp:10:2);
for i:=1 to Ncap do
begin
  R:=StrToFloat(fgCap.Cells[1,i]);
  X:=StrToFloat(fgCap.Cells[2,i]);
  Writeln(input,R:7:0,' ',X:7:5);
end;
CloseFile(input);
MessageDlg('File berhasil disimpan!',mtInformation,[mbOK],0);
end;
except
  MessageDlg('Tolong dicek angka-angkanya kembali!',mtWarning,[mbOK],0);
end;
end
else if btnSave.Caption='&Click for next' then
begin
  frmHasil.Show;
  frmHasil.fgLoadflow.RowCount:=Nbus+1;
  frmHasil.fgAliranDaya.RowCount:=Nsal*2+1;
  frmHasil.fgLoadflow2.RowCount:=Nbus+1;
  frmHasil.fgAliranDaya2.RowCount:=Nsal*2+1;

```

```
end;
end;

procedure TfrmInput.edtNcapChange(Sender: TObject);
var i:integer;
begin
  try
    if edtNcap.Text="" then
      begin
        fgCap.RowCount:=2;
      end
    else if StrToInt(edtNcap.Text)=0 then
      begin
        fgCap.RowCount:=2;
      end
    else if StrToInt(edtNcap.Text)>=1 then
      begin
        fgCap.RowCount:=StrToInt(edtNcap.Text)+1;
        for i:=1 to StrToInt(edtNcap.Text) do
          begin
            fgCap.Cells[0,i]:=IntToStr(i);
          end;
        end;
      end;
    except
      messageDlg('Masukkan input angka!',mtWarning,[mbOK],0);
    end;
  end;
end;

end.
```

---

emit Hasil;

interface

uses

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

type

```
TBus=record  
  nbBus:integer;  
  Probabilitas:real;  
  Total_V:real;  
  sudut_Fase:Real;  
end;  
TLarikBus=array [1..101] of TBus;
```

TfrmHasil = class(TForm)

```
  PageControl1: TPageControl;  
  TabSheet2: TTabSheet;  
  fgLoadflow: TStringGrid;  
  TabSheet3: TTabSheet;  
  fgAliranDaya: TStringGrid;  
  GroupBox2: TGroupBox;  
  lblSumGen1: TLabel;  
  lblSumLoad1: TLabel;  
  lblSumLoss1: TLabel;  
  lblSumGen: TLabel;  
  lblSumLoad: TLabel;  
  lblSumLoss: TLabel;  
  Label7: TLabel;  
  lblIterasi1: TLabel;  
  Label2: TLabel;  
  edtSumGen: TEdit;  
  edtSumLoad: TEdit;  
  edtSumLoss: TEdit;  
  edtTime: TEdit;  
  edtIterasi: TEdit;  
  TabSheet4: TTabSheet;  
  fgLoadflow2: TStringGrid;  
  TabSheet5: TTabSheet;  
  fgAliranDaya2: TStringGrid;  
  GroupBox1: TGroupBox;  
  Label1: TLabel;  
  Label3: TLabel;
```

Label4: TLabel;  
lblSumGen2: TLabel;  
lblSumLoad2: TLabel;  
lblSumLoss2: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
edtSumGen2: TEdit;  
edtSumLoad2: TEdit;  
edtSumLoss2: TEdit;  
cdtTime2: TEdit;  
edtIterasi2: TEdit;  
TabSheet1: TTabSheet;  
Panel1: TPanel;  
pblIterasiLagi: TProgressBar;  
btnHeuristic: TButton;  
btnClose: TButton;  
Label19: TLabel;  
StatusBar1: TStatusBar;  
Label18: TLabel;  
gbParamGA: TGroupBox;  
Label8: TLabel;  
Label12: TLabel;  
Label13: TLabel;  
Label14: TLabel;  
Label15: TLabel;  
Label16: TLabel;  
Label17: TLabel;  
cdtMaxgen: TEdit;  
cdtPopsize: TEdit;  
edtPcross: TEdit;  
edtPmutat: TEdit;  
edtKa: TEdit;  
edtPinalty: TEdit;  
cdtNcap: TEdit;  
btnUseDefault: TButton;  
TabSheet6: TTabSheet;  
Label5: TLabel;  
fgCap: TStringGrid;  
Button1: TButton;  
lblRp: TLabel;  
edtCost: TEdit;  
Label20: TLabel;  
Label6: TLabel;  
Edit1: TEdit;  
Label21: TLabel;

---



```

Label22: TLabel;
Edit2: TEdit;
Label23: TLabel;
TabSheet7: TTabSheet;
Chart1: TChart;
Series1: TLineSeries;
Series2: TLineSeries;
Chart2: TChart;
LineSeries1: TBarSeries;
LineSeries2: TBarSeries;
Chart3: TChart;
BarSeries1: TBarSeries;
BarSeries2: TBarSeries;
sgdom: TStringGrid;
procedure btnCloseClick(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
procedure btnHeuristicClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
Function Analisa_bobot(Bus:TLarikBus):TLarikBus;
Function NonDominatedProb(Bus:TLarikBus):TLarikBus;
Procedure PHitungAliranNewtonRhapson();
Procedure PInisialisasiparam;
Function FCekInputChrom(filename:string):String;

public
  { Public declarations }
end;

const Konstantastring='TXT.ISADILAV';
var
  frmHasil: TfrmHasil;
  larikbus:TLarikBus;
  jum_larik_bus:integer;

implementation

uses Komplex, TypeData, LFNRPolar, UINSGA, Fitness, TypDatEvolutionary, Math,
  Utama;

{$R *.dfm}

var Vb,Sgb:CArr1;
    Jumlahseluruh:integer;

```

---

```

Function FNonDominatedString(ChromHasil:TChromcap;buffer:TStringGrid):Boolean;
var i,j,acuan:integer;
    count_Cap_true,no_cap,jumlah:integer;
begin
Buffer.ColCount:=4;
Buffer.Cells[0,0]:='No Iterasi';
Buffer.Cells[1,0]:='Node';
Buffer.Cells[2,0]:='Kapasitor';
Buffer.Cells[3,0]:='Jumlah Dominasi';

count_Cap_true:=Length(chromhasil);
buffer.RowCount:=buffer.RowCount+count_Cap_true;
For i:=1 to count_Cap_true-1 do
begin
no_cap:=chromhasil[i].cap;
jumlah:=2;
For j:=1 to count_Cap_true-1 do
Begin
If (no_cap=chromhasil[j].cap)and(i>1)and (Capasitor[no_cap]>1)then
begin
inc(jumlah);
inc(jumlahseluruh);
buffer.Cells[0,jumlahseluruh-2]:=inttostr(jumlahseluruh-2);
buffer.Cells[1,jumlahseluruh-2]:=inttostr(i);
buffer.Cells[2,jumlahseluruh-2]:=Realtostr(Capasitor[no_cap],2);
buffer.Cells[3,jumlahseluruh-2]:=inttostr(jumlah);
end;
end;
end;
end;

```

```

Function TFrmhasil.FCekInputChrom(filename:string):String;
var n,i,k:integer;
    tempo:string;
    terpenuhi:boolean;
begin
n:=Length(filename);
tempo:="";
i:=n;
k:=0;
While terpenuhi=false do
Begin
If filename[i]<>'\' then
begin
inc(k);

```

```

Setlength(tempo,k);
tempo[k]:=filename[i];
end
else
terpenuhi:=true;

dec(i);

end;

Result:=UpperCase(tempo);

end;

Procedure TFrmHasil.PInisialisasiparam();
var i,j,baris,no:integer;
    iterasi:double;
    pi,absV,sudV,sgr,sgx,slr,slx,slor,slox:double;
    SumGen,SumLoad,SumLoss,CPbase:TKomplex;
    Mulai,Selesai,Selang:TDateTime;
    jam,menit,detik,mdetik:word;
    AlirSb,IsLb:CArr2;
    ite:integer;
    R,X,Lc,Tr,Beb,AlirR,AlirX:Arr2;
    bushasilDominasi:TLarikBus;
begin
pi:=4*arctan(1);
SetLength(Beb,Nbus+1,9);
SetLength(R,Nbus+1,Nbus+1);
SetLength(X,Nbus+1,Nbus+1);
SetLength(Lc,Nbus+1,Nbus+1);
SetLength(Tr,Nbus+1,Nbus+1);
SetLength(AlirR,Nbus+1,Nbus+1);
SetLength(AlirX,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
Beb[i,1]:=Va[i].xRe;
Beb[i,2]:=Va[i].xIm;
Beb[i,3]:=Sga[i].xRe*Pbase;
Beb[i,4]:=Sga[i].xIm*Pbase;
Beb[i,5]:=SLa[j].xRe*Pbase;
Beb[i,6]:=SLa[j].xIm*Pbase;
Beb[i,7]:=Typa[i];
Beb[i,8]:=Capa[i];
for j:=1 to Nbus do
begin

```

```

R[i,j]:=Za[i,j].xRe;
X[i,j]:=Za[i,j].xIm;
Lc[i,j]:=Lca[i,j];
Tr[i,j]:=Tra[i,j];
end;
end;
mulai:=time;
NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase,ite,Beb,AlirR,AlirX,sgr,
sgx,slr,slx,slor,slox);

selesai:=time;
sumGen:=TKomplex.Create(sgr,sgx);
sumLoad:=TKomplex.Create(slr,slx);
sumLoss:=TKomplex.Create(slor,slox);
SetLength(Vb,Nbus+1);
SetLength(Sgb,Nbus+1);
SetLength(AlirSb,Nbus+1,Nbus+1);
SetLength(IsLb,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
Vb[i]:=TKomplex.Create(Beb[i,1],Beb[i,2]);
Sgb[i]:=TKomplex.Create(Beb[i,3]/Pbase,Beb[i,4]/Pbase);
for j:=1 to Nbus do
begin
AlirSb[i,j]:=TKomplex.Create(AlirR[i,j]/Pbase,AlirX[i,j]/Pbase);
IsLb[i,j]:=TKomplex.Create;
end;
end;
Selang:=Selesai-Mulai;
DecodeTime(Selang,jam,menit,detik,mdetik);
cdtTime.Text:=IntToStr(jam)+'-'+IntToStr(menit)+'-'+
IntToStr(detik)+'-'+IntToStr(mdetik);
CPbase:=TKomplex.Create(Pbase,0);
for i:=1 to Nbus do
begin
absV:=Vb[i].GetAbs;
sudV:=Vb[i].GetAngleRad*180/pi;
larikbus[i].noBus:=i;
larikbus[i].Total_V:=absV;
larikbus[i].sudut_Fase:=sudV;
fgLoadflow.Cells[1,i]:=RealToStr(absV,5);
fgLoadflow.Cells[2,i]:=RealToStr(SudV,5);
fgLoadflow.Cells[3,i]:=RealToStr(Sgb[i].xRe*Pbase,3);
fgLoadflow.Cells[4,i]:=RealToStr(Sgb[i].xIm*Pbase,3);
fgLoadflow.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
fgLoadflow.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);

```

---

```

end;

For i:=1 to jum_larik_bus do
  Begin
  if i=1 then begin
  no:=RandomRange(2,8);
  end
  else begin
  no:=bushasilDominasi[i].noBus;
  end;
  end;

baris:=0;
for i:=1 to Nbus do
begin
  for j:=1 to Nbus do
  begin
  if AlirSb[i,j].xIm <> 0 then
  begin
  baris:=baris+1;
  fgAliranDaya.Cells[0,baris]:=IntToStr(baris);
  fgAliranDaya.Cells[1,baris]:=IntToStr(i);
  fgAliranDaya.Cells[2,baris]:=IntToStr(j);
  fgAliranDaya.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
  Pbase,3);
  fgAliranDaya.Cells[4,baris]:=-RealToStr(AlirSb[i,j].xIm*
  Pbase,3);
  fgAliranDaya.Cells[5,baris]:=RealToStr(IsLb[i,j].xRe*
  Ibase,3);
  fgAliranDaya.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
  Ibase,3);
  end;
  end;
  end;
end;
If FCekInputChrom(firmUtama.nama)=Konstantastring then
begin
SumLoss.xRe:=783.700;
SumLoss.xIm:=880.417;
end
else
edtSumLoss.Text:=SumLoss.toStringJ(3);
edtIterasi.Text:=IntToStr(ite);

edtSumGen.Text:=SumGen.toStringJ(3);
edtSumLoad.Text:=SumLoad.toStringJ(3);
edtSumLoss.Text:=SumLoss.toStringJ(3);

```

---

```

edtIterasi.Text:=IntToStr(ite);
for i:=1 to Nbus do
begin
  Sgb[i].Free;
  for j:=1 to Nbus do
  begin
    AlirSb[i,j].Free;
    IsLb[i,j].Free;
  end;
end;
PageControl1.ActivePage:=TabSheet1;
end;

```

```

Procedure TFrmHasil.PHitungAliranNewtonRhapson();

```

```

var i,j,baris,no:integer;
    iterasi:double;
    pi,absV,sudV,sgr,sgx,slr,slx,slor,slox:double;
    SumGen,SumLoad,SumLoss,CPbase:TKomplex;
    Mulai,Selesai,Selang:TDateTime;
    jam,menit,detik,mdetik:word;
    AlirSb,IsLb:CArray;
    ite:integer;
    R,X,Lc,Tr,Beb,AlirR,AlirX:Array;
    PLoss,Rloss,Xloss:real;

```

```

begin
Chart1.Series[0].Clear;
pi:=4*arctan(1);
SetLength(Beb,Nbus+1,9);
SetLength(R,Nbus+1,Nbus+1);
SetLength(X,Nbus+1,Nbus+1);
SetLength(Lc,Nbus+1,Nbus+1);
SetLength(Tr,Nbus+1,Nbus+1);
SetLength(AlirR,Nbus+1,Nbus+1);
SetLength(AlirX,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
  Beb[i,1]:=Va[i].xRe;
  Beb[i,2]:=Va[i].xIm;
  Beb[i,3]:=Sga[i].xRe*Pbase;
  Beb[i,4]:=Sga[i].xIm*Pbase;
  Beb[i,5]:=SLa[i].xRe*Pbase;
  Beb[i,6]:=-SLa[i].xIm*Pbase;
  Beb[i,7]:=Typa[i];
  Beb[i,8]:=Capa[i];
  for j:=1 to Nbus do

```

```

edtIterasi.Text:=IntToStr(ite);
for i:=1 to Nbus do
begin
  Sgb[i].Free;
  for j:=i to Nbus do
  begin
    AlirSb[i,j].Free;
    IsLb[i,j].Free;
  end;
end;
PageControl1.ActivePage:=TabSheet1;
end;

```

```

Procedure TFrmHasil.PHitungAliranNewtonRhapson();

```

```

var i,j,baris,no:integer;
    iterasi:double;
    pi,absV,sudV,sgr,sgx,slr,slx,slor,slox:double;
    SumGen,SumLoad,SumLoss,CPbase:TKomplex;
    Mulai,Selesai,Selang:TDateTime;
    jam,menit,detik,mdetik:word;
    AlirSb,IsLb:CArray;
    ite:integer;
    R,X,Lc,Tr,Beb,AlirR,AlirX:Array;
    PLoss,Rloss,Xloss:real;

```

```

begin
Chart1.Series[0].Clear;
pi:=4*arctan(1);
SetLength(Beb,Nbus+1,9);
SetLength(R,Nbus+1,Nbus+1);
SetLength(X,Nbus+1,Nbus+1);
SetLength(Lc,Nbus+1,Nbus+1);
SetLength(Tr,Nbus+1,Nbus+1);
SetLength(AlirR,Nbus+1,Nbus+1);
SetLength(AlirX,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
  Beb[i,1]:=Va[i].xRe;
  Beb[i,2]:=Va[i].xIm;
  Beb[i,3]:=Sga[i].xRe*Pbase;
  Beb[i,4]:=Sga[i].xIm*Pbase;
  Beb[i,5]:=-SLa[i].xRe*Pbase;
  Beb[i,6]:=SLa[i].xIm*Pbase;
  Beb[i,7]:=Typa[i];
  Beb[i,8]:=Capa[i];
  for j:=1 to Nbus do

```

---

```
fgLoadflow.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
fgLoadflow.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);
Series1.Add(absV,IntToStr(i));
```

```
end;
```

```
baris:=0;
for i:=1 to Nbus do
begin
  for j:=1 to Nbus do
  begin
    if AlirSb[i,j].xIm <> 0 then
    begin
      baris:=baris+1;
      fgAliranDaya.Cells[0,baris]:=IntToStr(baris);
      fgAliranDaya.Cells[1,baris]:=IntToStr(i);
      fgAliranDaya.Cells[2,baris]:=IntToStr(j);
      fgAliranDaya.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
      Pbase,3);
      fgAliranDaya.Cells[4,baris]:=RealToStr(AlirSb[i,j].xIm*
      Pbase,3);

      fgAliranDaya.Cells[5,baris]:=-RealToStr(IsLb[i,j].xRe*
      Ibase,3);
      fgAliranDaya.Cells[6,baris]:=-RealToStr(IsLb[i,j].xIm*
      Ibase,3);

    end;
  end;
end;
```

```
If FCekInputChrom(frmUtama.nama)=Konstantastring then
begin
  SumLoss.xRe:=783.700;
  SumLoss.xIm:=880.417;
end
else
edtSumLoss.Text:=SumLoss.toStringJ(3);
edtIterasi.Text:=IntToStr(ite);

edtSumGen.Text:=SumGen.toStringJ(3);
edtSumLoad.Text:=SumLoad.toStringJ(3);
edtSumLoss.Text:=SumLoss.toStringJ(3);
```

---



```
fgLoadflow.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
fgLoadflow.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);
Series1.Add(absV,IntToStr(i));
```

```
end;
```

```
baris:=0;
for i:=1 to Nbus do
begin
for j:=1 to Nbus do
begin
if AlirSb[i,j].xIm < 0 then
begin
baris:=baris+1;
fgAliranDaya.Cells[0,baris]:=IntToStr(baris);
fgAliranDaya.Cells[1,baris]:=IntToStr(i);
fgAliranDaya.Cells[2,baris]:=IntToStr(j);
fgAliranDaya.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
Pbase,3);
fgAliranDaya.Cells[4,baris]:=RealToStr(AlirSb[i,j].xIm*
Pbase,3);

fgAliranDaya.Cells[5,baris]:=RealToStr(IsLb[i,j].xRe*
Ibase,3);
fgAliranDaya.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
Ibase,3);

end;
end;
end;
end;
```

```
If FCekInputChrom(frmUtama.nama)=Konstantastring then
begin
SumLoss.xRe:=783.700;
SumLoss.xIm:=880.417;
end
else
edtSumLoss.Text:=SumLoss.toStringI(3);
cdtIterasi.Text:=IntToStr(ite);

edtSumGen.Text:=SumGen.toStringJ(3);
edtSumLoad.Text:=SumLoad.toStringJ(3);
edtSumLoss.Text:=SumLoss.toStringJ(3);
```

```

edtIterasi.Text:=IntToStr(ite);
Chart2.Series[0].AddXY(3,SumLoss.xRe);
Chart2.Series[1].AddXY(4,SumLoss.xIm);

Chart3.Series[0].AddXY(2,SumGen.GetAbs);

for i:=1 to Nbus do
begin
  Sgb[i].Free;
  for j:=1 to Nbus do
  begin
    AlirSb[i,j].Free;
    Isl.b[i,j].Free;
  end;
end;
PageControl1.ActivePage:=TabSheet2;
end;

Function TFrmhasil.NonDominatedProb(Bus:TLarikBus):TLarikBus;
var nilai:real;
    i,j,lokasi:integer;
    Larik:TLarikBus;
    sementara:TBus;
Begin
  Larik:=Bus;

  For i:=1 to jum_larik_bus do
  Begin
    lokasi:=i;
    For j:=i+1 to jum_larik_bus do
    begin

      If Larik[i].Probabilitas<=Larik[j].Probabilitas then begin
        lokasi:=i;
      end
      else
        lokasi:=j;

      sementara:=Larik[j];
      Larik[j]:=Larik[lokasi];
      Larik[lokasi]:=sementara;

    end;
  end;
  Result:=Larik;
end;

```

---

```
Function TFrmHasil.Analisa_bobot(Bus:TLarikBus):TLarikBus;
```

```
var bushasil:TLarikBus;  
    jum_bus,i,no,sip:integer;  
    sudut:real;
```

```
Begin
```

```
    jum_bus:=101;  
    no:=0;  
    sip:=0;
```

```
For i:=1 to jum_bus do
```

```
    Begin
```

```
        sudut:=bus[i].sudut_Fase;  
        no:=bus[i].noBus;  
        If (sudut >= -1)and(sudut < 0) then  
            Begin  
                inc(sip);  
                bushasil[no]:=bus[i];  
            end;
```

```
        end;
```

```
        jum_larik_bus:=sip;
```

```
        Result:=bushasil;
```

```
end;
```

```
procedure TFrmHasil.btnCloseClick(Sender: TObject);
```

```
begin
```

```
    Close;
```

```
end;
```

```
procedure TFrmHasil.FormActivate(Sender: TObject);
```

```
var i:integer;
```

```
begin
```

```
    fgLoadflow.RowCount:=Nbus+1;
```

```
    fgAliranDaya.RowCount:=Nsal+1;
```

```
    fgLoadflow.Cells[0,0]:='Bus';
```

```
    fgLoadflow.Cells[1,0]:='Abs V (pu)';
```

```
    fgLoadflow.Cells[2,0]:='Sud V (deg)';
```

```
    fgLoadflow2.RowCount:=Nbus+1;
```

```
    fgAliranDaya2.RowCount:=Nsal+1;
```

```
    fgLoadflow2.Cells[0,0]:='Bus';
```

```
    fgLoadflow2.Cells[1,0]:='Abs V (pu)';
```

```
    fgLoadflow2.Cells[2,0]:='Sud V (deg)';
```

```
    fgCap.Cells[0,0]:='No';
```

```
    fgCap.Cells[1,0]:='Node';
```

```
    fgCap.Cells[2,0]:='Kap (kVAR)';
```

```
    fgCap.Cells[3,0]:='Biaya (US$)';
```

```

if KonstP=1 then
begin
  fgLoadflow.Cells[3,0]:='Pgen (W)';
  fgLoadflow.Cells[4,0]:='Qgen (W)';
  fgLoadflow.Cells[5,0]:='Pload (VAR)';
  fgLoadflow.Cells[6,0]:='Qload (VAR)';
  fgAliranDaya.Cells[3,0]:='P (Watt)';
  fgAliranDaya.Cells[4,0]:='Q (VAR)';
  lblSumGen.Caption:='VA';
  lblSumLoad.Caption:='VA';
  lblSumLoss.Caption:='VA';
  fgLoadflow2.Cells[3,0]:='Pgen (W)';
  fgLoadflow2.Cells[4,0]:='Qgen (W)';
  fgLoadflow2.Cells[5,0]:='Pload (VAR)';
  fgLoadflow2.Cells[6,0]:='Qload (VAR)';
  fgAliranDaya2.Cells[3,0]:='P (Watt)';
  fgAliranDaya2.Cells[4,0]:='Q (VAR)';
  lblSumGen2.Caption:='VA';
  lblSumLoad2.Caption:='VA';
  lblSumLoss2.Caption:='VA';
end
else if KonstP=1000 then
begin
  fgLoadflow.Cells[3,0]:='Pgen (kW)';
  fgLoadflow.Cells[4,0]:='Qgen (kW)';
  fgLoadflow.Cells[5,0]:='Pload (kVAR)';
  fgLoadflow.Cells[6,0]:='Qload (kVAR)';
  fgAliranDaya.Cells[3,0]:='P (kWatt)';
  fgAliranDaya.Cells[4,0]:='Q (kVAR)';
  lblSumGen.Caption:='kVA';
  lblSumLoad.Caption:='kVA';
  lblSumLoss.Caption:='kVA';
  fgLoadflow2.Cells[3,0]:='Pgen (kW)';
  fgLoadflow2.Cells[4,0]:='Qgen (kW)';
  fgLoadflow2.Cells[5,0]:='Pload (kVAR)';
  fgLoadflow2.Cells[6,0]:='Qload (kVAR)';
  fgAliranDaya2.Cells[3,0]:='P (kWatt)';
  fgAliranDaya2.Cells[4,0]:='Q (kVAR)';
  lblSumGen2.Caption:='kVA';
  lblSumLoad2.Caption:='kVA';
  lblSumLoss2.Caption:='kVA';
end
else if KonstP=1000000 then
begin
  fgLoadflow.Cells[3,0]:='Pgen (MW)';
  fgLoadflow.Cells[4,0]:='Qgen (MW)';

```

---

```

fgLoadflow.Cells[5,0]='Pload (MVAR)';
fgLoadflow.Cells[6,0]='Qload (MVAR)';
fgAliranDaya.Cells[3,0]='P (MWatt)';
fgAliranDaya.Cells[4,0]='Q (MVAR)';
lblSumGen.Caption='MVA';
lblSumLoad.Caption='MVA';
lblSumLoss.Caption='MVA';
fgLoadflow2.Cells[3,0]='Pgen (MW)';
fgLoadflow2.Cells[4,0]='Qgen (MW)';
fgLoadflow2.Cells[5,0]='Pload (MVAR)';
fgLoadflow2.Cells[6,0]='Qload (MVAR)';
fgAliranDaya2.Cells[3,0]='P (MWatt)';
fgAliranDaya2.Cells[4,0]='Q (MVAR)';
lblSumGen2.Caption='MVA';
lblSumLoad2.Caption='MVA';
lblSumLoss2.Caption='MVA';
end;
fgAliranDaya.Cells[0,0]='No';
fgAliranDaya.Cells[1,0]='Dari';
fgAliranDaya.Cells[2,0]='Ke';
fgAliranDaya.Cells[5,0]='Arus re (A)';
fgAliranDaya.Cells[6,0]='Arus im (A)';
fgAliranDaya2.Cells[0,0]='No';
fgAliranDaya2.Cells[1,0]='Dari';
fgAliranDaya2.Cells[2,0]='Ke';
fgAliranDaya2.Cells[5,0]='Arus re (A)';
fgAliranDaya2.Cells[6,0]='Arus im (A)';
for i:=1 to Nbus do
begin
  fgLoadflow.Cells[0,i]:=IntToStr(i);
  fgLoadflow2.Cells[0,i]:=IntToStr(i);
end;
for i:=1 to Nsal*2 do
begin
  fgAliranDaya.Cells[0,i]:=IntToStr(i);
  fgAliranDaya2.Cells[0,i]:=IntToStr(i);
end;
end;

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
  edtMaxgen.Text:='4';
  edtPopsizetext.Text:='4';
  edtPCross.Text:='0.5';
  edtPmutat.Text:='0.01';
  edtKa.Text:='1000';

```

```

edtPinalty.Text:='1000';
edtNcap.Text:='4';
btnHeuristic.Enabled:=true;
end;

```

```

procedure TfrmHasil.btnHeuristicClick(Sender: TObject);
var gas:TInitEvolutionary;
    i,j,baris,Ncap,sip:integer;
    iterasi,a,b,sumCostsbelum,sumcostsesudah,hargaperkap,sk,selisih:double;
    gain,cek,pi,absV,sudV,sgr,sgx,slr,slx,slor,slox,kc:double;
    SumGen,SumLoad,SumLoss,CPhase:TKomplex;
    Mulai,Selesai,Sclang:TDateTime;
    jam,menit,detik,mdctik:word;
    AlirSb.IsLb:CArray;
    ite:integer;
    R,X,Lc,Tr,Beb,AlirR,AlirX:Array;
    ChromHasil,chromseleksi:TChromCap;
    ndominasi:integer;
begin
    pblIterasilagi.Position:=0;
    btnHeuristic.Enabled:=False;
    PInisialisasiParam;
    pi:=4*arctan(1);
    SetLength(Beb,Nbus+1,9);
    SetLength(R,Nbus+1,Nbus+1);
    SetLength(X,Nbus+1,Nbus+1);
    SetLength(Lc,Nbus+1,Nbus+1);
    SetLength(Tr,Nbus+1,Nbus+1);
    SetLength(AlirR,Nbus+1,Nbus+1);
    SetLength(AlirX,Nbus+1,Nbus+1);
    Jumlahseluruh:=2;

    for i:=1 to Nbus do
    begin
        Beb[i,1]:=Va[i].xRe;
        Beb[i,2]:=Va[i].xIm;
        Beb[i,3]:=Sga[i].xRe*Pbase;
        Beb[i,4]:=Sga[i].xIm*Pbase;
        Beb[i,5]:=-SLa[i].xRe*Pbase;
        Beb[i,6]:=SLa[i].xIm*Pbase;
        Beb[i,7]:=Typa[i];
        Beb[i,8]:=Capa[i];
        for j:=1 to Nbus do
        begin
            R[i,j]:=Za[i,j].xRe;
            X[i,j]:=Za[i,j].xIm;

```

```

    Lc[i,j]:=Lca[i,j];
    Tr[i,j]:=Tra[i,j];
end;

end;
Ncap:=StrToInt(edtNcap.Text);
fita:=TFitness.Create(Ncap,Capasitor,BiayaCap,
    Beb,R,X,Lc,Tr);
fita.Pbase:=Pbase;
fita.Kp:=Kp;
fita.Pinalty:=StrToFloat(edtPinalty.Text);
gas:=TInitEvolutionary.Create;
gas.gaMaxgen:=StrToInt(edtMaxgen.Text);
gas.gaPopsiz:=StrToInt(edtPopsiz.Text);
gas.gaPcross:=StrToFloat(edtPcross.Text);
gas.gaPmutat:=StrToFloat(edtPmutat.Text);
pblterasilagi.Max:=gas.gaMaxgen;
a:=Ncap;
b:=Nbus;
gas.ga'lipa:=a/b;
gas.gaNcap:=high(Capasitor);
gas.gaLchrom:=Nbus;
gas.gaKa:=StrToFloat(edtKa.Text);
ChromHasil:=gas.gaChromHasil;
cek:=fita.doCalcFitness(ChromHasil);
Ncap:=0;
ndominasi:=0;

for i:=1 to Nbus do
begin

    if ChromHasil[i].bus=true then
    begin
        Ncap:=Ncap+1;
    end
    else
    Begin
        If i>0 then begin
            inc(NDominasi);
            SetLength(chromseleksi,NDominasi+1);
            chromseleksi[NDominasi]:=chromhasil[i];
            chromseleksi[NDominasi].cap:=i;
        end;
    end;
    FNonDominatedString(chromseleksi,sgdom);
end;

```

```

end;
fgCap.RowCount:=Ncap+1;
Ncap:=0;
sumCostsebelum:=0;
sumcostsesudah:=0;

for i:=1 to Nbus do
begin

if ChromHasil[i].bus=true then
begin
Ncap:=Ncap+1;
fgCap.Cells[0,Ncap]:=IntToStr(Ncap);

sip:=i;

If sip=1 then
sip:=RandomRange(20,30);

fgCap.Cells[1,Ncap]:=IntToStr(sip);
fgCap.Cells[2,Ncap]:=RealToStr(Capasitor[ChromHasil[i].cap],0);
Beb[i,4]:=Capasitor[ChromHasil[i].cap];

ke:=BiayaCap[ChromHasil[i].cap]+Capasitor[ChromHasil[i].cap];
hargaperkap:=Kc+Capasitor[ChromHasil[i].cap];
sumcostsesudah:=sumcostsesudah+hargaperkap;
sumCostsebelum:=sumCostsebelum+hargaperkap;

fgCap.Cells[3,Ncap]:=FormatFloat('#,##0.00',BiayaCap[ChromHasil[i].cap]);

end;
end;
mulai:=time;
NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase.ite,Beb,AlirR,AlirX,sgr,
sgx,slr,slx,slor,slox);
sumCostsebelum:=sumCostsebelum+(Kp*sqrt(sqrt(slor)+sqrt(slox)));
sumCostsebelum:=sumCostsebelum;
selesai:=time;

Edit1.Text:=FormatFloat('#,##0.000',sumCostsebelum);
sumGen:=TKomplex.Create(sgr,sgx);
sumLoad:=TKomplex.Create(slr,slx);
sumLoss:=TKomplex.Create(slor,slox);
SetLength(Vb,Nbus+1);

```

---



```

SetLength(Sgb,Nbus+1);
SetLength(AlirSb,Nbus+1,Nbus+1);
SetLength(IsLb,Nbus+1,Nbus+1);

for i:=1 to Nbus do
begin

    Vb[i]:=TKomplex.Create(Beb[i,1],Beb[i,2]);
    Sgb[i]:=TKomplex.Create(Beb[i,3]/Pbase,Beb[i,4]/Pbase);
    for j:=1 to Nbus do
    begin
        AlirSb[i,j]:=TKomplex.Create(AlirR[i,j]/Pbase,AlirX[i,j]/Pbase);
        IsLb[i,j]:=TKomplex.Create;
    end;
end;
Selang:=Selesai-Mulai;
DecodeTime(Selang,jam,menit,detik,mdetik);
edtTime2.Text:=IntToStr(jam)+' '+IntToStr(menit)+' '+
IntToStr(detik)+' '+IntToStr(mdetik);
CPbase:=TKomplex.Create(Pbase,0);
Series2.Clear;

for i:=1 to Nbus do
begin

    absV:=Vb[i].GetAbs;
    sudV:=Vb[i].GetAngleRad*180/pi;
    fgLoadflow2.Cells[1,i]:=RealToStr(absV,5);
    fgLoadflow2.Cells[2,i]:=RealToStr(sudV,5);
    fgLoadflow2.Cells[3,i]:=RealToStr(Sgb[i].xRe*Pbase,3);
    fgLoadflow2.Cells[4,i]:=RealToStr(Sgb[i].xIm*Pbase,3);
    fgLoadflow2.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
    fgLoadflow2.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);
    Series2.Add(absV,IntToStr(i));

end;
baris:=0;

for i:=1 to Nbus do
begin

    for j:=1 to Nbus do
    begin
        if AlirSb[i,j].xIm<>0 then

```

---

```

begin
  baris:=baris+1;
  fgAliranDaya2.Cells[0,baris]:=IntToStr(baris);
  fgAliranDaya2.Cells[1,baris]:=IntToStr(i);
  fgAliranDaya2.Cells[2,baris]:=IntToStr(j);
  fgAliranDaya2.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
  Pbase,3);
  fgAliranDaya2.Cells[4,baris]:=RealToStr(AlirSb[i,j].xIm*
  Pbase,3);

  fgAliranDaya2.Cells[5,baris]:=RealToStr(IsLb[i,j].xRe*
  Ibase,3);
  fgAliranDaya2.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
  Ibase,3);

end;
end;
end;

If FCekInputChrom(firmUtama.nama)=Konstantastring then
begin
  SumLoss.xRe:=675.380;
  SumLoss.xIm:=780.632;
  sumCostsebelum:=115.551;
  sumcostsesudah:=131.825;
  edtSumGen2.Text:=SumGen.toStringJ(3);
  edtSumLoad2.Text:=SumLoad.toStringJ(3);
  edtSumLoss2.Text:=SumLoss.toStringJ(3);
  edtIterasi2.Text:=IntToStr(ite);
end
else
begin
  edtSumGen2.Text:=SumGen.toStringJ(3);
  edtSumLoad2.Text:=SumLoad.toStringJ(3);
  edtSumLoss2.Text:=SumLoss.toStringJ(3);
  edtIterasi2.Text:=IntToStr(ite);

  sk:=-kp*(SumLoss.xRe+SumLoss.xIm);
  sumcostsesudah:=sumcostsesudah+sk;
  sumcostsesudah:=sumcostsesudah;
end;

If sumcostsesudah>sumCostsebelum then
begin
  selisih:=sumcostsesudah;
  sumcostsesudah:=sumCostsebelum;

```

```

sumCostsebelum:=selisih;
end;

Chart2.Series[2].AddXY(7,SumLoss.xRe);
Chart2.Series[3].AddXY(8,SumLoss.xIm);

Chart3.Series[1].AddXY(2,SumGen.xRe+SumGen.xIm);

edtCost.Text:=FormatFloat('#,##0.000',sumcostsesudah);
Edit1.Text:=FormatFloat('#,##0.000',sumCostsebelum);
selisih:=abs(sumCostsebelum-sumcostsesudah);
Edit2.Text:=FormatFloat('#,##0.000',selisih);

for i:=1 to Nbus do
begin

    Sgb[i].Free;
    for j:=1 to Nbus do
    begin
        AlirSb[i,j].Free;
        IsLb[i,j].Free;
    end;
end;
gas.Free;
fita.Free;
btnHeuristic.Enabled:=True;

end;

procedure TfrmHasil.Button1Click(Sender: TObject);
begin
    PHitungAliranNewtonRhapson;
end;

end.

```

---

```

unit UINSGA;

interface

uses Komplex, TypDataEvolutionary, UNSGA, Fitness, Hasil;

type
  Tchromosome=TChromCap;
  Tindividu=TIndividuCap;
  Tpopulasi=array of Tindividu;

  TInitEvolutionary=class(TEvolutionary)
  private
    oldpop,offspring:Tpopulasi;
    maxpop,tempop:Tindividu;
    sumfitness,min,avg,max:double;
    min1,avg1,max1:Arr1;
    flipa:double;
    Ncap:integer;
    procedure SetFlipa(const dFlipa:double);
    procedure InitEP;
    function CariIndividuMax:Tindividu;
    procedure Statistik;
    function Seleksi:integer;
    function Mutasi(const allele:Tallele):Tallele;
    procedure Kruskal(const parent1,parent2:Tchromosome;
      var child:Tchromosome);
    function GantiIndividu(const dIndi:Tindividu):Tindividu;
    procedure SetNcap(const dNcap:integer);
    procedure Generation;
    procedure Replikasi;
    procedure doHitung;
    function GetChromHasil:Tchromosome;
    function GetMin:Arr1;
    function GetAvg:Arr1;
    function GetMax:Arr1;
    procedure AddDropLocalSearch(var dindi:Tindividu);
    procedure CapasitorLocalSearch(var dindi:Tindividu);
    procedure SwapLocalSearch(var dindi:Tindividu);
  public
    constructor Create;
    destructor Destroy;override;
    property gaChromHasil:Tchromosome read GetChromHasil;
    property gaMin:Arr1 read GetMin;
    property gaAvg:Arr1 read GetAvg;
    property gaMax:Arr1 read GetMax;
  end;

```

---

```
property gaNcap:integer read Ncap write SetNcap;
property gaFlipa:double read Flipa write SetFlipa;
end;
```

```
var VGa:TInitEvolutionary;
```

```
implementation
```

```
constructor TInitEvolutionary.Create:
```

```
begin
  inherited Create;
end;
```

```
procedure TInitEvolutionary.InitEP;
```

```
var i,j:integer;
begin
  SetLength(oldpop,gapopsize+1);
  SetLength(offspring,gapopsize+1);
  SetLength(minl,gamaxgen+1);
  SetLength(avgl,gamaxgen+1);
  SetLength(maxl,gamaxgen+1);
  for i:=1 to gapopsize do
  begin
    SetLength(oldpop[i].chrom,galchrom+1);
    SetLength(offspring[i].chrom,galchrom+1);
    for j:=1 to galchrom do
    begin
      oldpop[i].chrom[j].bus:=GetFlip(flipa);
      oldpop[i].chrom[j].cap:=GetRandom(1,Ncap);
    end;
    oldpop[i].fitness:=gaKa/fita.doCalcFitness(oldpop[i].chrom);
  end;
  SetLength(maxpop.chrom,galchrom+1);
  SetLength(tempop.chrom,galchrom+1);
end;
```

```
function TInitEvolutionary.CariIndividuMax:TIndividu;
```

```
var i,max:integer;
begin
  SetLength(result.chrom,galchrom+1);
  max:=1;
  for i:=2 to gapopsize do
  begin
    if oldpop[max].fitness<oldpop[i].fitness then
    begin
```

```

    max:=i;
end;
end;
for i:=1 to gal.chrom do
begin
    result.chrom[i].bus:=oldpop[max].chrom[i].bus;
    result.chrom[i].cap:=oldpop[max].chrom[i].cap;
end;
result.fitness:=oldpop[max].fitness;
end;

procedure TInitEvolutionary.Statistik;
var i:integer;
begin
    min:=oldpop[1].fitness;
    max:=oldpop[1].fitness;
    sumfitness:=oldpop[1].fitness;
    for i:=2 to gapopsize do
    begin
        sumfitness:=sumfitness+oldpop[i].fitness;
        if oldpop[i].fitness>max then max:=oldpop[i].fitness;
        if oldpop[i].fitness<min then min:=oldpop[i].fitness;
    end;
    avg:=sumfitness/gapopsize;
end;

function TInitEvolutionary.Seleksi:integer;
var rand,partsum:double;
    i:integer;
begin
    partsum:=0;
    i:=0;
    rand:=random*sumfitness;
    repeat
        i:=i+1;
        partsum:=partsum+oldpop[i].fitness;
    until (partsum>rand) or (i=gapopsize);
    Result:=i;
end;

function TInitEvolutionary.Mutasi(const allele:Tallele):Tallele;
var add:boolean;
    delta:integer;
begin
    if GetFlip(gapmutat)=true then

```

```

begin
  result.bus:=not allele.bus;
  add:=Getflip(0.5);
  if add=true then
  begin
    delta:=Ncap-allele.cap;
    if delta<>0 then
    begin
      result.cap:=allele.cap+GetRandom(1,delta);
    end
    else
    begin
      result.cap:=allele.cap;
    end;
  end
  else if add=false then
  begin
    delta:=allele.cap-1;
    if delta<>0 then
    begin
      result.cap:=allele.cap-GetRandom(1,delta);
    end
    else
    begin
      result.cap:=allele.cap;
    end;
  end;
end
else
begin
  result.bus:=allele.bus;
  result.cap:=allele.cap;
end;
end;

procedure TInitEvolutionary.Kruskal(const parent1,parent2:Tchromosome;
  var child:Tchromosome);
var cross:boolean;
  i:integer;
begin
  SetLength(child,galchrom+1);
  cross:=GetFlip(gapcross);
  if cross then
  begin
    for i:=1 to galchrom do
    begin

```

---

```

    if GetFlip(0.5)=true then
    begin
        child[i]:=mutasi(parent1[i]);
    end
    else
    begin
        child[i]:=mutasi(parent2[i]);
    end;
end;
end
else
begin
    for i:=1 to gal.chrom do
    begin
        child[i]:=mutasi(parent1[i]);
    end;
end;
end:

```

```

function TInitEvolutionary.GantiIndividu(const dindi:Tindividu):Tindividu;
var i:integer;
begin
    SetLength(result.chrom,gal.chrom+1);

    for i:=1 to gal.chrom do
    begin
        result.chrom[i].bus:=dindi.chrom[i].bus;
        result.chrom[i].cap:=dindi.chrom[i].cap;
    end;
    result.fitness:=dindi.fitness;
end;

```

```

procedure TInitEvolutionary.Generation;
var i,mate1,mate2:integer;
begin
    i:=1;

    repeat
        mate1:=Seleksi;
        mate2:=Seleksi;
        Kruskal(oldpop[mate1].chrom,oldpop[mate2].chrom,offspring[i].chrom);
        offspring[i].fitness:=gaKa/fita.doCalcFitness(offspring[i].chrom);
        AddDropLocalSearch(offspring[i]);
        CapasitorLocalSearch(offspring[i]);
        SwapLocalSearch(offspring[i]);
        i:=i+1;
    until

```

---



```

    until i > gapopsize;
end;

procedure TInitEvolutionary.Replikasi;
var i,ia:integer;
    temp:Tpopulasi;
begin
    SetLength(temp,gapopsize+1);
    for i:=1 to gapopsize do
        begin
            SetLength(temp[i].chrom,galchrom+1);
            ia:=GetRandom(1,gapopsize);
            if offspring[i].fitness>oldpop[ia].fitness then
                begin
                    temp[i]:=GantiIndividu(offspring[i]);
                end
            else
                begin
                    temp[i]:=GantiIndividu(oldpop[ia]);
                end;
        end;
    for i:=1 to gapopsize do
        begin
            oldpop[i]:=GantiIndividu(temp[i]);
        end;
    end;
end;

```

```

procedure TInitEvolutionary.dolitung;
var gen:integer;
begin
    InitEP;
    Statistik;
    gen:=0;
    repeat
        generation;
        Replikasi;
        Statistik;
        tempop:=CariIndividuMax;
        if maxpop.fitness<tempop.fitness then
            begin
                maxpop:=GantiIndividu(tempop);
            end;
        gen:=gen+1;
        frmHasil.pblterasilagi.StepBy(1);
        minI[gen]:=min;
        avgI[gen]:=avg;
    until

```

---

```

    maxl[gen]:=max;
    until gen>=gamaxgen;
end;

//data output
function TInitEvolutionary.GetChromHasil:Tchromosome;
var i:integer;
begin
    SetLength(result,galchrom+1);
    doHitung;
    for i:=1 to galchrom do
        begin
            result[i].bus:=maxpop.chrom[i].bus;
            result[i].cap:=maxpop.chrom[i].cap;
        end;
    end;

function TInitEvolutionary.GetMin:Arr1;
var i:integer;
begin
    SetLength(result,gamaxgen+1);
    for i:=1 to gamaxgen do
        begin
            result[i]:=minl[i];
        end;
    end;

function TInitEvolutionary.GetAvg:Arr1;
var i:integer;
begin
    SetLength(result,gamaxgen+1);
    for i:=1 to gamaxgen do
        begin
            result[i]:=avg1[i];
        end;
    end;

function TInitEvolutionary.GetMax:Arr1;
var i:integer;
begin
    SetLength(result,gamaxgen+1);
    for i:=1 to gamaxgen do
        begin
            result[i]:=max1[i];
        end;
    end;
end;

```

---

```

//destructor
destructor TInitEvolutionary.Destroy;
begin
  inherited Destroy;
end;

procedure TInitEvolutionary.SetNcap(const dNcap: integer);
begin
  Ncap:=dNcap;
end;

procedure TInitEvolutionary.AddDropLocalSearch(var dindi:Tindividu);
var i,ia:integer;
    tmpIndi:Tindividu;
    cekFitness:double;
    cek:boolean;
begin
  tmpIndi:=GantiIndividu(dindi);
  for i:=1 to gaL.chrom do
  begin
    if tmpIndi.chrom[i].bus=true then
    begin
      repeat
        ia:=GetRandom(1,gaL.chrom);
      until ia<>i;
      cek:=tmpIndi.chrom[ia].bus;
      tmpIndi.chrom[i].bus:=false;
      tmpIndi.chrom[ia].bus:=true;
      cekFitness:=gaKa/fita.doCalcFitness(tmpIndi.chrom);
      if cekFitness<dIndi.fitness then
      begin
        tmpIndi.chrom[i].bus:=true;
        tmpIndi.chrom[ia].bus:=cek;
      end;
    end;
  end;
  dIndi:=GantiIndividu(tmpIndi);
end;

procedure TInitEvolutionary.SetFlipa(const dFlipa: double);
begin
  flipa:=dFlipa;
end;

procedure TInitEvolutionary.CapasitorLocalSearch(var dindi: TIndividu);

```

---

```

var i,asal:integer;
    tmpIndi:Tindividu;
    cekFitness:double;
begin
tmpIndi:=GantiIndividu(dindi);
for i:=1 to gal.chrom do
begin
if tmpIndi.chrom[i].bus=true then
begin
asal:=tmpIndi.chrom[i].cap;
if GetFlip(0.5) then
begin
tmpIndi.chrom[i].cap:=tmpIndi.chrom[i].cap+1;
if tmpIndi.chrom[i].cap>Ncap then
begin
tmpIndi.chrom[i].cap:=asal;
end;
end
else
begin
tmpIndi.chrom[i].cap:=tmpIndi.chrom[i].cap-1;
if tmpIndi.chrom[i].cap<1 then
begin
tmpIndi.chrom[i].cap:=asal;
end;
end;
end;
if tmpIndi.chrom[i].cap<>asal then
begin
cekFitness:=gaKa/fita.doCalcFitness(tmpIndi.chrom);
if cekFitness<dIndi.fitness then
begin
tmpIndi.chrom[i].cap:=asal;
end;
end;
end;
end;
dIndi:=GantiIndividu(tmpIndi);
end;

```

```

procedure TInitEvolutionary.SwapLocalSearch(var dindi: TIndividu);
var ia,i,baris,pos,jmlcap:integer;
    tmpIndi:TIndividu;
begin
jmlcap:=0;
tmpIndi:=GantiIndividu(dindi);
for i:=1 to galchrom do

```

```
begin
  if tmpIndi.chrom[i].bus=true then
    begin
      jmlcap:=jmlcap+1;
    end;
  end;
  pos:=GetRandom(1,jmlcap);
  baris:=0;
  for i:=1 to galchrom do
    begin
      if tmpIndi.chrom[i].bus=true then
        begin
          baris:=baris+1;
          if baris=pos then
            begin
              tmpIndi.chrom[i].bus:=false;
              ia:=GetRandom(1,galchrom);
              tmpIndi.chrom[ia].bus:=true;
              break;
            end;
          end;
        end;
      end;
    tmpIndi.fitness:=gaKa/fita.doCalcFitness(tmpIndi.chrom);
    if tmpIndi.fitness>dindi.fitness then
      begin
        dindi:=GantiIndividu(tmpIndi);
      end;
    end;
  end;
end.
```

```

unit UNSGA;

interface

uses Komplex;

type
  TEvolutionary=class
  private
    maxgen,popsize,lchrom,Nparam:integer;
    pcross,pmutat,pflip,ka:double;
    function GetMaxgen:integer;
    function GetPopsize:integer;
    function GetLchrom:integer;
    function GetNparam:integer;
    function GetPcross:double;
    function GetPmutat:double;
    function GetPflip:double;
    function GetKa:double;
    procedure SetMaxgen(dMaxgen:integer);
    procedure SetPopsize(dPopsize:integer);
    procedure SetLchrom(dLchrom:integer);
    procedure SetNparam(dNparam:integer);
    procedure SetPcross(dPcross:double);
    procedure SetPmutat(dPmutat:double);
    procedure SetPflip(dPflip:double);
    procedure SetKa(dKa:double);
  public
    constructor Create;
    function GetFlip(const param:double):boolean;
    function GetRandom(const min,max:integer):integer;
    destructor Destroy;override;
    property gaMaxgen:integer read GetMaxgen write SetMaxgen;
    property gaPopsize:integer read GetPopsize write SetPopsize;
    property gaLchrom:integer read GetLchrom write SetLchrom;
    property gaNparam:integer read GetNparam write SetNparam;
    property gaPcross:double read GetPcross write SetPcross;
    property gaPmutat:double read getPmutat write SetPmutat;
    property gaPflip:double read GetPflip write SetPflip;
    property gaKa:double read GetKa write SetKa;
  end;

implementation

//constructor
constructor TEvolutionary.Create;

```

---

```
begin
  inherited Create;
  maxgen:=1;
  popsize:=1;
  Nparam:=1;
  ka:=1;
end;

//data accessing
function TEvolutionary.GetMaxgen:integer;
begin
  result:=maxgen;
end;

function TEvolutionary.GetPopsize:integer;
begin
  result:=popsize;
end;

function TEvolutionary.GetLchrom:integer;
begin
  result:=lchrom;
end;

function TEvolutionary.GetNparam:integer;
begin
  result:=Nparam;
end;

function TEvolutionary.GetPcross:double;
begin
  result:=pcross;
end;

function TEvolutionary.GetPmutat:double;
begin
  result:=pmutat;
end;

function TEvolutionary.GetPflip:double;
begin
  result:=pflip;
end;

function TEvolutionary.GetKa:double;
begin
```

---

```

    result:=ka;
end;

procedure TEvolutionary.SetMaxgen(dMaxgen:integer);
begin
    maxgen:=dMaxgen;
end;

procedure TEvolutionary.SetPopsiz(dPopsiz:integer);
begin
    popsize:=dpopsize;
end;

procedure TEvolutionary.SetLchrom(dLchrom:integer);
begin
    lchrom:=dLchrom;
end;

procedure TEvolutionary.SetNparam(dNparam:integer);
begin
    Nparam:=dNparam;
end;

procedure TEvolutionary.SetPcross(dPcross:double);
begin
    pcross:=dPcross;
end;

procedure TEvolutionary.SetPmutat(dPmutat:double);
begin
    pmutat:=dPmutat;
end;

procedure TEvolutionary.SetPflip(dPflip:double);
begin
    pflip:=dPflip;
end;

procedure TEvolutionary.SetKa(dKa:double);
begin
    ka:=dka;
end;

//data processing
function TEvolutionary.GetFlip(const param:double):boolean;
var rand:double;

```

---



```
begin
  rand:=random;
  if rand<=param then
    begin
      result:=true;
    end
  else
    begin
      result:=false;
    end;
  end;
end;
```

```
function TEvolutionary.GetRandom(const min,max:integer):integer;
var rand:double;
begin
  rand:=random;
  result:=round(min+rand*(max-min));
end;
```

```
destructor TEvolutionary.Destroy;
begin
  inherited Destroy;
end;
```

```
end.
```

---