

# SKRIPSI

## ANALISA OPTIMASI PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV DENGAN METODE HYBRID GA-SA PADA GI PIER PENYULANG SYIAR PASURUAN



*Disusun Oleh :*  
**HADI SURYA**  
**01.12.095**



**JURUSAN TEKNIK ELEKTRO S1**  
**KONSENTRASI TEKNIK ENERGI LISTRIK**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

**MARET 2008**

1954

THE NATIONAL WATER RESOURCES BOARD  
WASHINGTON, D. C. 20540  
OFFICE OF THE SECRETARY  
WASHINGTON, D. C. 20540

1954  
1954  
1954

TO THE NATIONAL WATER RESOURCES BOARD  
WASHINGTON, D. C. 20540  
FROM THE NATIONAL WATER RESOURCES BOARD  
WASHINGTON, D. C. 20540

1954

**LEMBAR PERSETUJUAN**  
**ANALISA OPTIMASI PENEMPATAN KAPASITOR PADA JARINGAN**  
**DISTRIBUSI PRIMER 20 kV DENGAN METODE *HYBRID GA-SA***  
**PADA G.I PIER, PENYULANG SYIAR PASURUAN**

**SKRIPSI**

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


*Disusun Oleh :*

**HADI SURYA**

**01.12.095**

*Diperiksa dan Disetujui,*

**Dosen Pembimbing I**



**Ir. H. TAUFIK HIDAYAT, M.T**

**NIP. Y. 1018700151**

*Diperiksa dan Disetujui,*

**Dosen Pembimbing II**



**IRRINE BUDI S, S.T, M.T**

**NIP. 132314400**



*Mengetahui,*

**Ketua Jurusan Teknik Elektro S-1**

**Ir. F. Yudi Limpraptono, M.T**

**NIP. Y. 1039500274**

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



## A B S T R A K S I

### ANALISA OPTIMASI PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 kV DENGAN METODE *HYBRID GA-SA* PADA G.I PIER, PENYULANG SYIAR PASURUAN

Hadi Surya

Ir. H. Taufik Hidayat, M.T

Irrine Budi Sulistiawati, S.T, M.T

**Kata Kunci** : Sistem Distribusi Radial, Penempatan Kapasitor, Pengurangan Rugi-rugi, *Simulated Annealing, Genetics Algorithm, Metode GA-SA.*

Meningkatnya usaha di sektor industri dan meningkatnya taraf hidup masyarakat maka kebutuhan akan energi listrik semakin meningkat pula, sehingga diperlukan penyediaan energi listrik beserta jaringan dan penyaluran yang sangat baik. 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 memperbesar komponen rugi-rugi daya, disamping itu dapat memperburuk kondisi tegangan.

Alternatif yang sering dipakai untuk memperbaiki kondisi jaringan akibat adanya rugi-rugi tersebut adalah memasang sumber daya reaktif tambahan disisi beban, salah satunya adalah pemasangan kapasitor.

Banyak metode yang dipakai dalam menganalisa masalah penentuan penempatan dan besar kapasitor pada jaringan distribusi, dan kesemuanya telah menghasilkan ciri dan kelemahan yang berbeda. Hal ini mendorong untuk mengangkat permasalahan dengan menggunakan metode *Hybrid GA-SA*, guna mendapatkan hasil yang lebih optimal dari metode-metode yang telah ada dalam penempatan kapasitor pada jaringan distribusi primer.

Skripsi ini menjelaskan masalah penempatan kapasitor yang optimal dalam sebuah sistem distribusi. Bagaimana menentukan lokasi secara optimal dalam memasang kapasitor, jenis dan ukuran kapasitor, pemasangan dari kapasitor ini untuk mendapatkan fungsi tujuan yang diinginkan yaitu mengurangi rugi-rugi daya, memperbaiki profil tegangan dan menentukan nilai penghematan. Sebuah metode *hybrid GA-SA algorithm* digunakan untuk memilih lokasi pemasangan kapasitor. Dimana sebelum penempatan, jatuh tegangan sebesar  $0.94716 pu$  dan setelah penempatan menjadi  $0.96602 pu$ , untuk rugi daya aktif dari  $76.75926 kW$  menjadi  $56.50346 kW$ . Untuk rugi daya reaktif dari  $117.3255 kVAR$ , menjadi  $86.36124 kVAR$ . Juga terjadi penghematan sebesar  $US\$ 3.127.41$  dimana sebelum penempatan biaya total kehilangan energi sebesar  $US\$ 25.242.21$ , dan setelah penempatan terjadi penurunan sebesar  $US\$ 22.114.81$  atau sebesar  $14.14\%$ . Algoritma solusi diuji dalam sebuah sistem daya 49 bus di GI.PIER, Penyulang SYIAR Pasuruan.



## **KATA PENGANTAR**

Dengan mengucapkan puji syukur kehadiran ALLAH SWT, atas rahmat dan karunia-Nya memungkinkan penulis dapat menyelesaikan skripsi yang berjudul, **“ANALISA OPTIMASI PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 kV DENGAN METODE HYBRID GA-SA PADA JARINGAN DISTRIBUSI PRIMER 20 kV DI GL. PIER, PENYULANG SYIAR PASURUAN”**.

Skripsi ini bertujuan untuk memenuhi kurikulum akademik yang harus ditempuh oleh setiap mahasiswa ITN Malang guna menyelesaikan pendidikan pada jenjang strata satu jurusan Teknik Elektro konsentrasi Teknik Energi Listrik di Institut Teknologi Nasional Malang.

Atas segala bimbingan, pengarahan dan bantuan yang diberikan, sehingga tersusunnya skripsi ini, maka penulis menyampaikan terima kasih kepada :

1. **Prof. Dr. Eng. Ir. Abraham Lomi, MSEE**, selaku Rektor ITN Malang.
2. **Ir. Mochtar Asroni, MSME**, selaku Dekan FTI ITN Malang.
3. **Ir. F Yudi Limpraptono, MT**, selaku Ketua Jurusan Teknik Elektro (S1) ITN Malang.
4. **Yusuf Ismail Nakhoda, ST. MT**, selaku Sekretaris Jurusan Teknik Elektro (S1) ITN Malang.

5. **Ir. H. Taufik Hidayat, MT**, selaku bapak dan dosen pembimbing I.
6. **Irrine Budi Sulistiawati, ST. MT**, selaku dosen pembimbing II.
7. **Hamsyah dan Yayuk Tri Rahayu**, selaku kedua orang tua, beserta seluruh keluarga.
8. Seluruh dosen yang ada di jurusan elektro khususnya.
9. Seluruh rekan mahasiswa ITN jurusan Elektro khususnya angkatan 2001.

Akhirnya penulis mengharapkan skripsi ini dapat berguna dan bermanfaat bagi rekan-rekan mahasiswa khususnya pada jurusan Teknik Elektro S-1 konsentrasi Teknik Energi Listrik.

Malang, Maret 2008

Penulis



## DAFTAR ISI

	<b>Halaman:</b>
<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PERSETUJUAN</b> .....	ii
<b>ABSTRAKSI</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiv
<b>DAFTAR GRAFIK</b> .....	xv
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan.....	3
1.4. Batasan Masalah.....	3
1.5. Metodologi Penelitian.....	4
1.6. Sistematika Penulisan.....	5
<b>BAB II SISTEM DISTRIBUSI TENAGA LISTRIK DAN APLIKASI KAPASITOR</b>	
2.1. Sistem Tenaga Listrik.....	7
2.2. Sistem Distribusi Tenaga Listrik.....	8

2.1.1. Sistem Distribusi Primer ( Jaringan Tegangan Menengah ) .....	10
2.1.2. Sistem Distribusi Sekunder ( Jaringan Tegangan Rendah ).....	10
2.3. Struktur Jaringan Distribusi Tenaga Listrik.....	10
2.3.1. Sistem Jaringan Distribusi Radial.....	10
2.3.2. Sistem Jaringan Distribusi Tertutup ( Loop ) .....	12
2.3.3. Sistem Jaringan Distriobusi <i>Mesh</i> .....	12
2.4. Daya Dalam Sistem Tenaga Listrik .....	13
2.4.1. Daya Nyata ( <i>Real Power</i> ).....	13
2.4.2. Daya Reaktif ( <i>Reactive Power</i> ) .....	14
2.4.3 Daya Semu ( <i>Apparentl Power</i> ).....	14
2.5. Daya Reaktif dan Faktor Daya.....	14
2.6. Kapasitor Daya.....	17
2.6.1. Kapasitor Seri dan Kapasitor <i>Shunt</i> .....	17
2.6.2. Faktor-Faktor Pemilihan Kapasitor Seri dan Kapasitor Shunt ..	18
2.7. Pengaruh Pemasangan Kapasitor <i>Shunt</i> .....	20
2.7.1. Pengurangan Rugi-Rugi Saluran dengan Kapasitor <i>Shunt</i> .....	21
2.7.2. Perbaikan Tegangan.....	22
2.7.3. Perbaikan Faktor Daya dan Kenaikan Kapasitas Sistem .....	23
2.7.4. Perhitungan Pengaruh Perbaikan Faktor Daya .....	25
2.7.5. Penentuan Rating Kapasitor untuk Perbaikan Faktor Daya Beban.....	26
2.8. Sistem <i>Per-Unit</i> .....	27
2.9. Variasi Tegangan .....	28



### **BAB III METODE *HYBRID GA-SA* DAN PENERAPAN**

Analisa Aliran Daya Jaringan Radial.....	30
Tujuan.....	30
3.1. Metode <i>Newton Raphson</i> .....	30
3.1.2. Algoritma Aliran Daya <i>Newton Raphson</i> .....	33
3.1.3. <i>Flowchart</i> Algoritma Aliran Daya <i>Newton Rahpson</i> .....	34
3.2. Metode <i>Hybrid GA-SA</i> .....	35
3.2.1. Algoritma Genetika.....	35
3.2.1.1. Istilah-istilah Algoritma Genetika.....	36
3.2.1.2. Proses Algoritma Genetika.....	39
3.2.1.3. Elitism.....	43
3.2.1.4. Crossover.....	44
3.2.1.5. Mutasi.....	46
3.2.2. Metode <i>Simulated Annealing</i> .....	48
3.2.2.1. Jadwal Pendinginan.....	49
3.2.2.2. Kriteria Penghentian.....	50
3.2.3 <i>Hybrid GA-SA</i> .....	50
3.3. Fungsi Tujuan ( <i>Objective Function</i> ).....	51
3.3.1 Algoritma Program <i>Genetic Algorithm</i> .....	53
3.3.2 Algoritma Program <i>Simulated Annealing</i> .....	54
3.3.3. <i>Flowchart Genetic Algorithm</i> .....	56
3.3.4. <i>Flowchart Simulated Annealing</i> .....	57
3.6. Data Saluran.....	58

3.7.	Pembebanan Sistem 20 kV Penyulang Syiar .....	60
3.8.	Data Kapasitas Kapasitor Dan Harga ( <i>cost/kVAR</i> ).....	62

#### **BAB IV ANALISA DAN HASIL**

4.1.	Sistem Distribusi Tenaga Listrik 20kV GI PIER, Penyulang SYIAR.....	63
4.2.	Analisa Perhitungan.....	65
4.3.	Prosedur Pelaksanaan Program Perhitungan .....	67
4.4.	Analisa Hasil Sebelum Optimasi. ....	83
4.5.	Analisa Hasil Setelah Optimasi. ....	93

#### **BAB V KESIMPULAN**

5.1.	Kesimpulan .....	98
------	------------------	----

#### **DAFTAR PUSTAKA**

#### **LAMPIRAN**



## DAFTAR GAMBAR

Gambar :	Halaman :
2.1 Skema Penyaluran Energi Listrik .....	7
2.2 Jaringan Distribusi Tegangan Menengah (JTM), Jaringan Tegangan Rendah (JTR) dan Sambungan Rumah ke Pelanggan .....	9
2.3 Jaringan Tegangan Menengah Sistem Distribusi Radial .....	11
2.4 Jaringan Tegangan Menengah Struktur Loop yang dipasok dari satu sumber.....	12
2.5 Jaringan Tegangan Menengah Struktur <i>Mesh</i> .....	13
2.6 Segi Tiga Daya.....	15
2.7 Segi Tiga Arus .....	16
2.8 Saluran Primer Dengan Beban Tepusat.....	17
2.9 Kapasitor Shunt Yang Dipasang Pada Saluran.....	18
2.10 Vektor arus dan tegangan sebelum dan setelah pemasangan kapasitor .....	21
2.11 Diagram vektor pada rangkaian dengan <i>pf lagging</i> (a) dan (c) tanpa kapasitor shunt, (b) dan (d) dengan kapasitor <i>shunt</i> .....	22
2.12 Perbaikan Faktor Daya.....	24
2.13 Diagram Fasor dan Sudut Daya Beban Distribusi .....	25
3.1 <i>Flowchart</i> Algoritma Aliran Daya <i>Newton Raphson</i> .....	34
3.2 <i>Roulette-Wheel</i> .....	42
3.3 Pembentukan <i>Next Generation</i> dalam Algoritma Genetika.....	44

3.3.	Pembentukan <i>Next Generation</i> dalam Algoritma Genetika.....	44
3.4.	Ilustrasi Operator Dengan <i>One Point Crossover</i> .....	45
3.5.	Ilustrasi Operator Dengan <i>Two Point Crossover</i> .....	45
3.6.	Ilustrasi Operator Dengan <i>Uniform Crossover</i> .....	45
3.7.	Ilustrasi Operator Mutasi Untuk <i>Representasi String Binner</i> .....	46
3.8.	Ilustrasi Operator Mutasi Untuk <i>Representasi Integer</i> .....	47
3.9.	<i>Flowchart</i> Algoritma Solusi <i>Genetic Algorithm</i> .....	56
3.10.	<i>Flowchart</i> Algoritma Solusi <i>Simulated Annealing</i> .....	57
4.1.	<i>Single Line Diagram</i> Penyulang Syiar Sebelum Penempatan Kapasitor...64	
4-2.	<i>Flowchart</i> Pemecahan Masalah .....	66
4.3.	Tampilan Utama Program Matlab 7.0.4 .....	67
4.4.	Tampilan Inputan Data ( <i>General</i> ).....	68
4.5.	Tampilan Inputan Data (Data Pembebanan).....	69
4.6.	Tampilan Inputan Data (Data Saluran).....	71
4.7.	Tampilan Data <i>Objective Function</i> .....	73
4.8.	Tampilan <i>Parameter GA</i> Dan <i>SA</i> .....	74
4.9.	Tampilan Untuk Menjalankan Program.....	75
4.10.	Tampilan Hasil Aliran Daya Untuk Mengetahui Harga Tegangan dan Sudut Fasa Tiap-tiap Bus Sebelum Kompensasi .....	76
4.11.	Tampilan Hasil Aliran Daya Untuk Daya tiap Saluran Sebelum Kompensasi.....	78
4.12.	Tampilan Hasil Rugi Daya Antar Saluran Sebelum Optimasi.....	80

4.13.	Tampilan Hasil Jumlah Pembangkitan, Pembebanan, Rugi-rugi Sebelum Optimasi.....	82
4.14.	Tampilan Hasil Penempatan Kapasitor Dengan Menggunakan Metode <i>Hybrid GA-SA</i> .....	84
4.15.	Gambar <i>Single line</i> Diagram Penyulang Syiar Setelah Penempatan Kapasitor.....	85
4.16.	Tampilan Hasil Untuk Mengetahui Tegangan Dan Sudut <i>Fasa</i> Tiap Bus Setelah Optimasi.....	86
4.17.	Tampilan Hasil Aliran Daya Untuk Daya tiap Saluran Setelah Kompensasi .....	88
4.18.	Tampilan Hasil Rugi Daya Antar Saluran Setelah Optimasi.....	90
4.19.	Tampilan Hasil Jumlah Pembangkitan, Pembebanan, Rugi-rugi dan <i>Cost Total</i> Sebelum Dan Sesudah Serta Selisih <i>Total Cost</i> Setelah Optimasi Menggunakan Metode <i>Hybrid GA-SA</i> .....	92

## DAFTAR TABEL

<b>Tabel :</b>	<b>Halaman :</b>
2-1 Kapasitor Seri dan Kapasitor Shunt.....	19
3-1 Istilah Yang Digunakan Dalam Algoritma Genetika.....	37
3-2. Spesifikasi Saluran.....	58
3-3. Data Hasil Perhitungan Saluran Penyulang Syiar.....	59
3-4. Data Hasil Perhitungan Pembebanan Sistem 20kV Penyulang Syiar.....	61
4-1. Data Hasil Perhitungan Pembebanan Sistem 20kV Penyulang Syiar.....	70
4-2. Data Hasil Perhitungan Saluran Penyulang Syiar.....	72
4-3. Tegangan dan Sudut <i>Fasa</i> Tiap Bus Penyulang Syiar Sebelum Optimasi.....	77
4-4. Daya Tiap Saluran Penyulang Syiar Sebelum Optimasi.....	79
4-5. Rugi Daya Tiap Saluran Penyulang Syiar Sebelum optimasi.....	81
4-6. Tegangan Dan Sudut Fasa Tiap Bus Penyulang Syiar Setelah Optimasi.....	87
4-7. Daya Tiap Saluran Penyulang Syiar Setelah Optimasi.....	89
4-8. Rugi Daya Tiap Saluran Penyulang Syiar Setelah Optimasi.....	91
4-9. Hasil Program Metode <i>Hybrid GA-SA</i> .....	95

## DAFTAR GRAFIK

<b>Grafik :</b>	<b>Halaman :</b>
4-1 Tegangan Tiap-tiap Bus Terhadap Asumsi Tegangan Awal, Tegangan Sebelum Dan Setelah Kompensasi.....	96
4-2 Harga Rugi Daya Sebelum Dan Sesudah Kompensasi.....	97

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang<sup>[1]</sup>

Dengan meningkatnya usaha di sektor industri dan meningkatnya taraf hidup masyarakat maka kebutuhan akan energi listrik semakin meningkat pula, sehingga diperlukan penyediaan energi listrik beserta jaringan dan penyaluran yang sangat baik. Bertambahnya industri-industri menyebabkan peran penggunaan alat-alat listrik akan semakin luas, misalnya motor-motor listrik, trafo, AC, lampu-lampu penerangan dan lain-lain. Beban industri sangat banyak membutuhkan daya reaktif induktif. Dengan meningkatnya beban-beban induktif, maka memperbesar komponen rugi-rugi daya, disamping itu dapat memperburuk kondisi tegangan.

Alternatif yang sering dipakai untuk memperbaiki kondisi jaringan akibat adanya rugi-rugi tersebut adalah memasang daya reaktif tambahan disisi beban salah satunya adalah pemasangan kapasitor.

Banyak metode yang dipakai dalam menganalisa masalah penentuan penempatan dan besar kapasitor pada jaringan distribusi, dan kesemuanya telah menghasilkan ciri dan kelemahan yang berbeda.

Hal ini mendorong untuk mengangkat permasalahan dengan menggunakan metode *Hybrid GA-SA*, guna mendapatkan hasil yang lebih optimal dari metode-metode yang telah ada dalam penempatan kapasitor pada jaringan distribusi primer.



## 1.2 Rumusan Masalah

Pemasangan kapasitor sebagai sumber daya reaktif tambahan pada sisi beban dilakukan dengan pemasangan kapasitor pada sistem distribusi primer.

Berdasarkan latar belakang, maka dapat dirumuskan permasalahan, yaitu :

1. Penentuan lokasi secara optimal, jumlah, jenis, dan ukuran kapasitor serta investasi dalam peralatan daya reaktif yang akan dipasang atau ditempatkan pada jaringan distribusi radial penyulang Syiar Pasuruan ?
2. Profil tegangan yang didapat dengan pemasangan kapasitor ?
3. Berapa rugi-rugi daya pada saluran dengan pemasangan kapasitor ?

Sehubungan dengan permasalahan diatas maka skripsi ini diberi judul :

**“ANALISA OPTIMASI PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN METODE *HYBRID GA - SA* PADA JARINGAN DISTRIBUSI PRIMER 20 kV DI GL. PIER, PENYULANG SYIAR PASURUAN”**

### 1.3 Tujuan Pembahasan

Adapun tujuan dalam penulisan skripsi ini adalah :

1. Menentukan letak, jenis dan kapasitas kapasitor yang optimal pada jaringan distribusi radial penyulang Syiar.
2. Memperbaiki profil tegangan dengan pemasangan kapasitor.
3. Mengurangi rugi-rugi daya dengan pemasangan kapasitor.
4. Menentukan *total cost* akibat kehilangan energi dengan pemasangan kapasitor.

### 1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan, maka pembahasan skripsi dibatasi oleh hal – hal sebagai berikut :

1. Jaringan yang akan di analisa adalah jaringan distribusi primer 20 kV tipe radial di penyulang Syiar Pasuruan.
2. Hanya Kapasitor *Shunt* yang digunakan dalam perbaikan profil tegangan dan mengurangi rugi-rugi daya.
3. Analisa penempatan letak dan kapasitas kapasitor menggunakan metode *Hybrid GA - SA*.
4. Analisa dilakukan pada kondisi normal pada saat beban puncak.
5. Metode aliran daya yang digunakan yaitu metode *Newton Rhapson*.
6. Nilai faktor daya diasumsikan 0.85.

## Metodologi Penelitian

Metodologi yang digunakan dalam penyusunan skripsi ini adalah:

1. Kajian literatur.

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

2. Pengumpulan Data.

Bentuk data yang digunakan adalah:

- Data kuantitatif, yaitu data yang dapat dihitung atau data yang berbentuk angka-angka.
- Data kualitatif, yaitu data yang berbentuk diagram. Dalam hal ini single line diagram penyulang.

3. Simulasi dan pembahasan masalah.

Untuk analisa perhitungan tegangan, sudut fasa tegangan, rugi-rugi daya dan aliran daya tiap saluran menggunakan metode *Newton Raphson*, untuk menentukan penempatan kapasitor serta *total cost* yang optimal menggunakan metode *Hybrid GA-SA* yang disimulasikan dengan program matlab 7.0.4 pada komputer.

## 1.6. Sistematika Penulisan

Sistematika dari pembahasan di dalam skripsi ini adalah sebagai berikut :

### **BAB I : PENDAHULUAN**

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

### **BAB II : SISTEM DISTRIBUSI TENAGA LISTRIK DAN APLIKASI KAPASITOR**

Disini akan dibahas masalah sistem jaringan distribusi, pengertian jatuh tegangan, pengaruh rendahnya jatuh daya dan penjelasan teori tentang kapasitor serta faktor daya, dan juga penjelasan sistem *per-unit*, serta variasi tegangan.

### **BAB III : METODE *HYBRID GA-SA* DAN PENERAPAN**

Pada bab ini akan dibahas metode aliran daya *Newton Raphson*, teori-teori mengenai metode *GA* dan *SA*, algoritma metode *Hybrid GA-SA*. Data saluran dan data pembebanan pada penyulang Syiar serta data kapasitas dan harga kapasitor (*Cost/kVAR*)

### **BAB IV : ANALISA DAN HASIL**

Pada bab ini akan dibahas mengenai analisa dari hasil simulasi program matlab 7.0.4. mengenai penentuan letak dan kapasitas kapasitor menggunakan metode *Hybrid GA-SA*, serta perbandingan profil tegangan, rugi daya dan *total cost* sebelum dan setelah penempatan kapasitor.

**BAB V : KESIMPULAN**

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan analisa dari hasil simulasi program matlab 7.0.4. yang berisikan kesimpulan

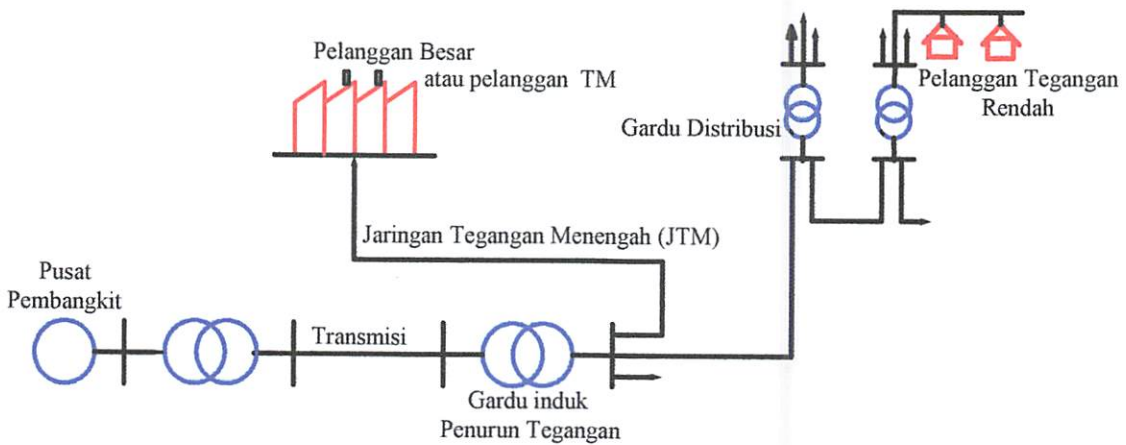
**DAFTAR PUSTAKA**

## BAB II

### SISTEM DISTRIBUSI TENAGA LISTRIK DAN APLIKASI KAPASITOR

#### 2.1. Sistem Tenaga Listrik<sup>[2]</sup>

Sistem tenaga listrik merupakan suatu sistem terpadu yang terbentuk oleh hubungan-hubungan peralatan dan komponen-komponen listrik. Sistem tenaga listrik ini mempunyai peranan utama untuk menyalurkan energi listrik yang dibangkitkan oleh generator ke konsumen yang membutuhkan energi listrik.



**GAMBAR 2-1 SKEMA PENYALURAN ENERGI LISTRIK<sup>[2]</sup>**

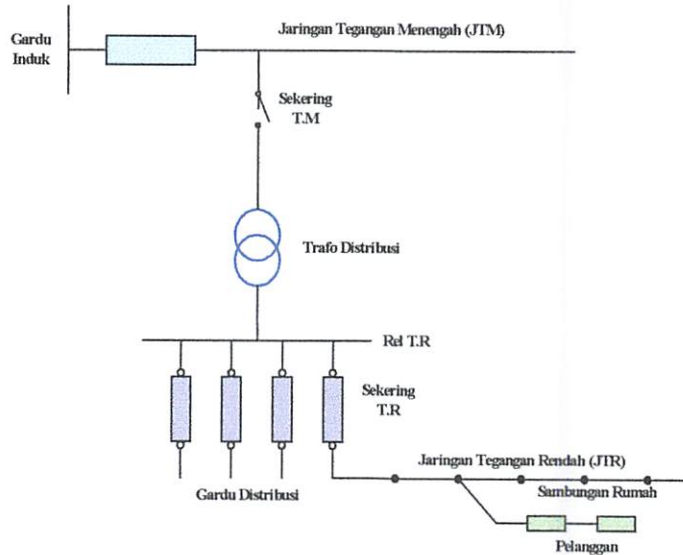


Secara garis besar suatu sistem tenaga listrik dapat dikelompokkan menjadi 3 sub sistem, yaitu :

1. Sistem Pembangkitan : Berperan sebagai sumber daya tenaga listrik dan disebut juga prodaktor energi.
2. Sistem Transmisi atau Penyaluran : Berfungsi sebagai penyalur daya listrik secara besar-besaran dari pembangkit ke bagian sistem distribusi atau konsumen.
3. Sistem Distribusi dan Beban : Berperan sebagai distributor energi ke konsumen yang memerlukan energi tersebut.

## **2.2. Sistem Distribusi Tenaga Listrik<sup>[3]</sup>**

Jaringan distribusi berada pada akhir dari sistem tenaga listrik, perannya mendistribusikan tenaga listrik dari gardu Induk ke konsumen melalui gardu distribusi. jaringan setelah keluar dari gardu induk bisa disebut jaringan distribusi. Setelah tenaga listrik disalurkan melalui jaringan distribusi primer maka kemudian tenaga listrik diturunkan tegangannya dalam gardu-gardu distribusi menjadi tegangan rendah, kemudian disalurkan melalui jaringan tegangan rendah untuk selanjutnya disalurkan kerumah-rumah pelanggan (konsumen) PLN melalui sambungan rumah.



**GAMBAR 2-2 JARINGAN DISTRIBUSI TEGANGAN MENENGAH (JTM), JARINGAN TEGANGAN RENDAH (JTR) DAN SAMBUNGAN RUMAH KE PELANGGAN<sup>[3]</sup>**

Dalam pendistribusian tenaga listrik ke konsumen, tegangan yang digunakan bervariasi tergantung dari jenis konsumen yang membutuhkan. Untuk konsumen industri biasanya digunakan tegangan menengah 20 kV sedangkan untuk konsumen perumahan digunakan tegangan rendah 220/380 Volt, yang merupakan tegangan siap pakai untuk peralatan-peralatan rumah tangga. Dengan demikian maka sistem distribusi tenaga listrik dapat diklasifikasikan menjadi dua bagian sistem yaitu:

1. Sistem Distribusi Primer (Jaringan Tegangan Menengah)
2. Sistem Distribusi Sekunder (Jaringan Tegangan Rendah)

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

### **2.2.1. Sistem Distribusi Primer (Jaringan Tegangan Menengah)**

Tingkat Tegangan yang digunakan pada sistem distribusi primer adalah meliputi tegangan 20 kV, Oleh karena itu sistem distribusi ini sering disebut dengan sistem distribusi tegangan menengah.

### **2.2.2. Sistem distribusi Sekunder (Jaringan Tegangan Rendah)**

Tingkat Tegangan yang digunakan pada sistem distribusi sekunder adalah tegangan rendah yaitu 127/220 volt atau 220/380 volt, oleh karena itu sistem distribusi ini sering disebut dengan sistem distribusi tegangan rendah.

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.

## **2.3. Struktur Jaringan Distribusi Tenaga Listrik<sup>[2]</sup>**

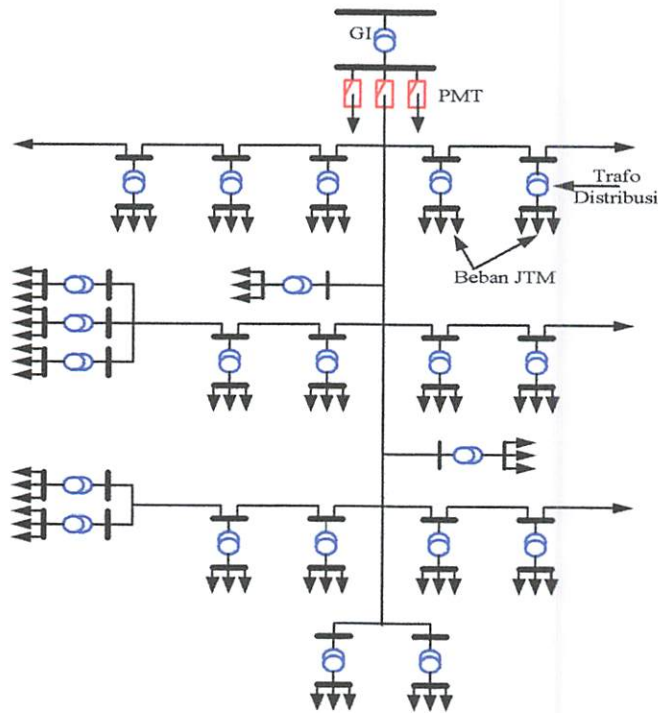
Ada beberapa bentuk jaringan yang umum digunakan untuk menyalurkan dan mendistribusikan tenaga listrik, yaitu :

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

### **2.3.1. Sistem Jaringan Distribusi Radial**

Bentuk jaringan ini merupakan bentuk dasar yang paling sederhana dan paling banyak dipergunakan. Sistem ini dikatakan radial karena dari kenyataannya bahwa jaringan ini ditarik secara radial dari gardu induk ke pusat-

pusat beban atau konsumen yang dilayani. Sistem ini terdiri dari saluran utama (*tunk line*) dan saluran cabang (*lateral*) seperti pada gambar 2.3.



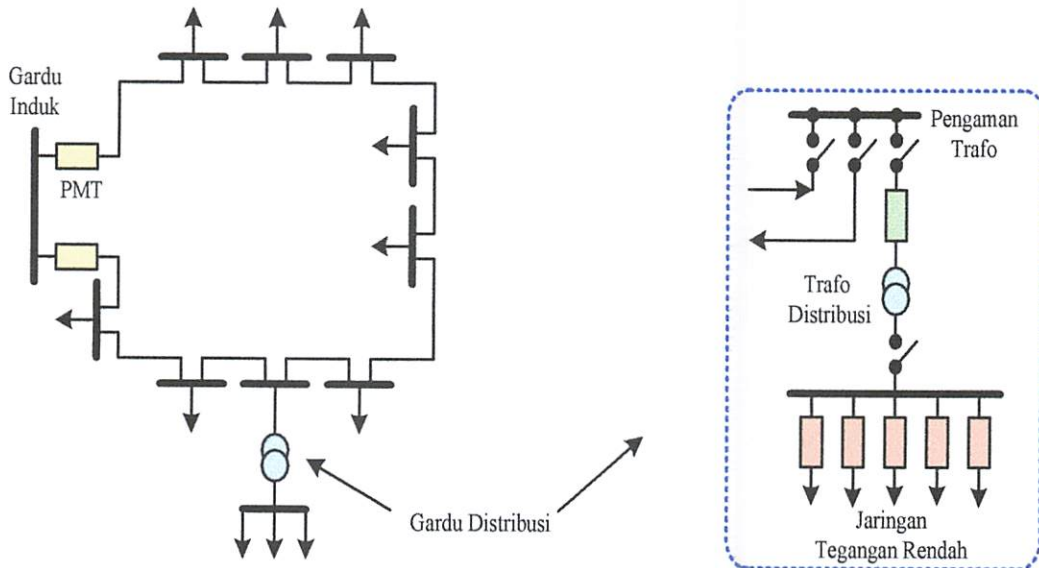
**GAMBAR 2-3 JARINGAN TEGANGAN MENENGAH SISTEM DISTRIBUSI RADIAL<sup>[2]</sup>**

Pelayanan tenaga listrik untuk suatu daerah beban tertentu dilaksanakan dengan memasang transformator pada sembarang 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 salah satu *feeder* maka semua pelanggan yang terhubung pada *feeder* tersebut akan terganggu.

### 2.3.2. Sistem Jaringan Distribusi Tertutup (Loop).

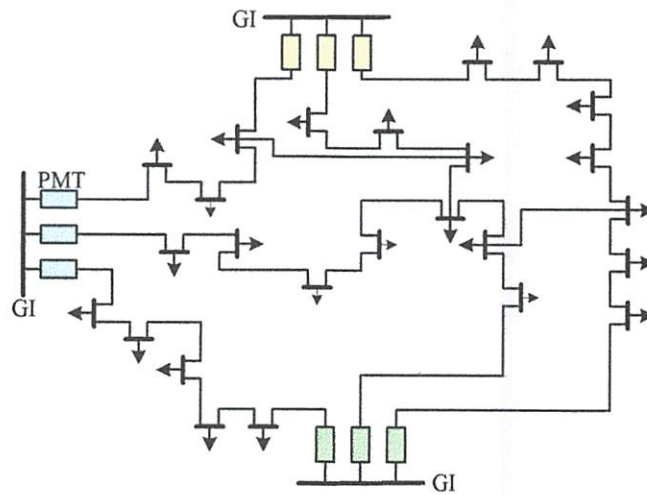
Sistem ini disebut jaringan distribusi *loop* karena saluran primer yang menyalurkan daya sepanjang daerah beban yang dilayani, membentuk suatu rangkaian *loop*.



**GAMBAR 2-4 JARINGAN TEGANGAN MENENGAH STRUKTUR LOOP YANG DIPASOK DARI SATU SUMBER<sup>[2]</sup>**

### 2.3.3. Sistem Jaringan Distribusi *Mesh*.

Jaringan Distribusi *Mesh* merupakan jaringan yang strukturnya kompleks, dimana kelangsungan penyaluran dan pelayanannya diutamakan. Struktur jaringan ini umumnya digunakan pada jaringan tegangan rendah yang kepadatan bebannya cukup tinggi.



**GAMBAR 2-5 JARINGAN TEGANGAN MENENGAH STRUKTUR MESH<sup>[2]</sup>**

**2.4. Daya Dalam Sistem Tenaga Listrik<sup>[2]</sup>**

Dalam sistem tenaga listrik, pembangkit-pembangkit tenaga listrik harus mampu menyediakan tenaga listrik kepada pelanggan sesuai dengan permintaan beban listrik yang ada. dan hal yang harus diperhatikan adalah sistem yang tetap konstan, dalam hal ini tegangan dan frekuensi harus tetap konstan karena berhubungan dengan daya.

Daya listrik yang dibangkitkan dikenal dengan istilah:

**2.4.1. Daya Nyata ( Real Power )**

Daya Nyata dinyatakan dalam persamaan :

$$P = [ V ] [ I ] \cos \varphi \dots\dots\dots (2.1)$$

Daya nyata untuk beban 3 fasa seimbang

$$P = \sqrt{3} | V_{\text{jala-jala}} | | I_{\text{jala-jala}} | \cos \varphi \dots\dots\dots (2.2)$$



### 2.4.2. Daya Reaktif ( *Reactive power* )

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

Daya reaktif dinyatakan dalam persamaan :

$$Q = | V | | I | \sin \phi \dots\dots\dots (2.3)$$

Daya reaktif untuk beban 3 fasa seimbang :

$$Q = \sqrt{3} | V_{\text{jala-jala}} | | I_{\text{jala-jala}} | \sin \phi \dots\dots\dots (2.4)$$

### 2.4.3. Daya Semu ( *Apparent Power* )

Daya semu dinyatakan dalam persamaan :

$$S = | V | | I | \dots\dots\dots (2.5)$$

Daya semu untuk beban 3 fasa seimbang :

$$S = \sqrt{3} | V_{\text{jala-jala}} | | I_{\text{jala-jala}} | \dots\dots\dots (2.6)$$

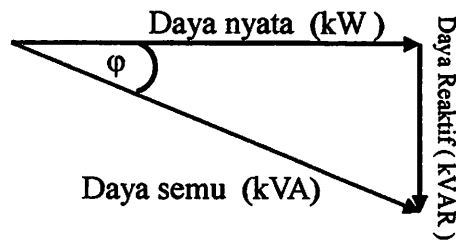
## 2.5. Daya Reaktif dan Faktor Daya<sup>[2]</sup>

Setiap pemakaian daya reaktif akan menyebabkan turunnya faktor daya yang kemudian menyebabkan memburuknya karakteristik kerja peralatan-peralatan sistem pada umumnya, baik dari segi teknik operasional maupun segi ekonomisnya, faktor daya adalah perbandingan antara daya nyata dan daya semu.

$$\text{Faktor daya} = \frac{\text{DayaNyata}(kW)}{\text{Dayasemu}(kVA)} \dots\dots\dots (2.7)$$

Untuk daya semu sendiri dibentuk oleh dua kompoen daya nyata (kW) dan komponen daya reaktif (kVAR).

Hubungan ini dapat digambarkan sebagai berikut:



**GAMBAR 2-6 SEGITIGA DAYA<sup>[2]</sup>**

Dengan Faktor daya =  $\text{Cos } \varphi = \frac{P}{S}$

$$\text{Cos } \varphi = \frac{P}{\sqrt{P^2 + Q_1^2}} \dots\dots\dots (2.8)$$

P = Daya Nyata (kW)

S = Daya semu (kVA)

Q = Daya reaktif (kVar)

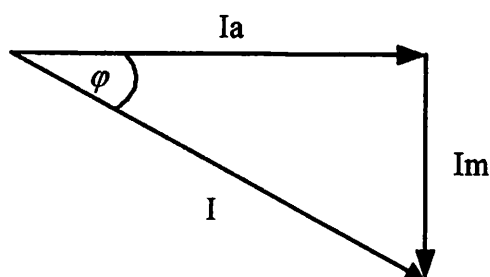
$\varphi$  = Sudut Phase

Dari gambar 2.6 daya semu terdiri dari komponen daya nyata dan daya reaktif.

Jadi faktor daya dapat dilihat dari hubungan antara arus nyata, arus magnetisasi dan arus total.

- Arus nyata ( $I_a$ ) adalah arus yang dibeban dan diubah kedalam energi panas.
- Arus magnetisasi ( $I_m$ ) adalah arus yang mengalir dibeban untuk menimbulkan medan magnet.
- Arus total ( $I$ ) adalah arus yang mengalir di jaringan dan merupakan penjumlahan vektor dari arus nyata dengan arus magnetisasi.

Dalam bentuk vektor hubungan tersebut digambarkan sebagai berikut :

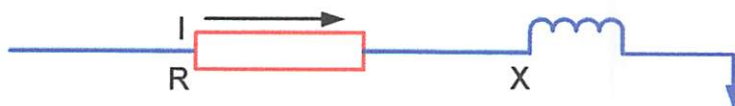


**GAMBAR 2-7 SEGITIGA ARUS <sup>[2]</sup>**

Beberapa sebab sistem distribusi mempunyai faktor daya yang rendah, yaitu :

- Banyaknya pemakaian motor asinkron terutama pada industri.
- Makin meningkatnya pemakaian lampu TL untuk penerangan.
- Pemakaian pemanas air.

Menurunnya faktor daya berarti mengecilnya perbandingan antara daya nyata dengan daya semu atau berarti semakin membesarnya kebutuhan beban dan daya aktif.



**GAMBAR 2-8 SALURAN PRIMER DENGAN BEBAN TEPUSAT<sup>[2]</sup>**

Karena pada saluran terdapat resistansi  $R$  dan reaktansi  $X$ , maka rugi daya ( $P_L$ ) dirumuskan sebagai berikut :

$$\begin{aligned}
 P_L &= I^2 R \\
 &= (I \cos \theta)^2 R + (I \sin \theta)^2 R \dots\dots\dots (2.9)
 \end{aligned}$$

dimana :  $I_R$  adalah arus aktif

$I_X$  adalah arus reaktif

## 2.6. Kapasitor Daya<sup>[2]</sup>

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

### 2.6.1. Kapasitor Seri dan Kapasitor *Shunt*

Kapasitor daya terdiri atas 2 bagian yaitu kapasitor seri dan kapasitor *shunt*:

#### a. Kapasitor Seri

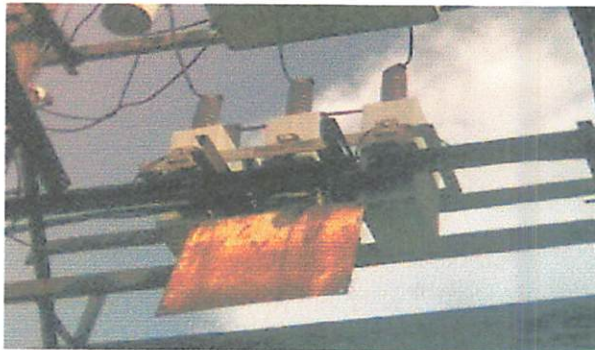
Kapasitor seri adalah kapasitor yang dihubung seri dengan impedansi saluran yang bersangkutan, pemakaiannya amat dibatasi pada saluran distribusi, karena peralatan pengamannya cukup rumit. Jadi secara umum dikatakan biaya

untuk pemasangan kapasitor seri lebih mahal daripada biaya pemasangan kapasitor paralel atau biasa disebut kapasitor *shunt*.

**b. Kapasitor *Shunt***

Kapasitor *shunt* adalah kapasitor yang dihubungkan paralel dengan saluran dan secara intensif digunakan pada saluran distribusi. Kapasitor *shunt* mencatu daya reaktif atau arus yang menentang komponen arus beban induktif.

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



**GAMBAR 2-9 KAPASITOR *SHUNT* YANG DIPASANG PADA SALURAN**

### 2.6.2. Faktor-faktor pemilihan kapasitor Seri dan kapasitor shunt<sup>[2]</sup> :

Faktor-faktor yang mempengaruhi pemilihan kapasitor *shunt* dan seri ditabelkan sebagai berikut :

**TABEL 2-1**  
**KAPASITOR SERI DAN KAPASITOR *SHUNT***

No	Tujuan	Kapasitor	
		Seri	Shunt
1	Memperbaiki faktor daya	Kedua	Pertama
2	Memperbaiki tingkat tegangan pada sistem saluran udara dengan faktor daya normal dan rendah	Pertama	Kedua
3	Memperbaiki tingkat tegangan pada sistem saluran udara dengan faktor daya yang tinggi	Tidak Dipakai	Pertama
4	Memperbaiki tingkat tegangan pada sistem saluran udara bawah tanah dengan faktor daya yang tinggi	Tidak Dipakai	Tidak Dipakai
5	Memperbaiki tingkat tegangan pada sistem saluran bawah tanah dengan faktor daya normal dan rendah	Pertama	Tidak Dipakai
6	Mengurangi rugi-rugi daya dan rugi-rugi energi pada saluran	Kedua	Pertama
7	Mengurangi fluktuasi tegangan	Pertama	Tidak Dipakai



## 2.7. Pengaruh Pemasangan Kapasitor *Shunt*<sup>[2]</sup>

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

Adapun dua cara dalam pemakaian kapasitor *shunt* :

- ✚ Kapasitor Tetap (*Fixed Capacitor*)
- ✚ Kapasitor Saklar (*Switched Capacitor*)

### 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 tegangan yang ditetapkan

### b. Kapasitor Saklar

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

Proses membuka-menutup dari saklar kapasitor *shunt* dapat dilakukan dengan cara manual maupun dengan cara otomatis. Pengendalian secara manual (Pada lokasi atau lokasi jarak jauh) dapat dilakukan pada G.I.

Untuk pengendalian secara otomatis, termasuk didalamnya peralatan pengendali tegangan, arus dan suhu. Tipe yang paling populer adalah pengendalian saklar waktu (*time-switch control*), pengendali tegangan dan pengendali tegangan arus.

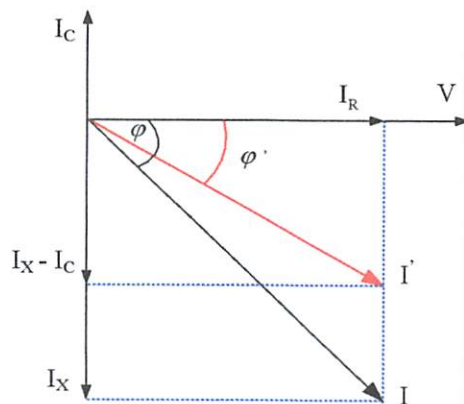
### 2.7.1. Pengurangan Rugi-Rugi Saluran dengan Kapasitor *Shunt*

Rugi-rugi saluran sebelum ada pengaruh pemasangan kapasitor adalah seperti pada persamaan (2.9) diatas.

Sedangkan setelah pemasangan kapasitor, maka sebagian daya reaktif yang dibutuhkan oleh beban akan disuplai oleh kapasitor tersebut, sehingga arus yang mengalir di jaringan akan lebih kecil. Persamaan rugi daya dan rugi energi setelah pemasangan kapasitor adalah :

$$P_L' = (I \cos \theta)^2 R + (I \sin \theta - I_c)^2 R \dots\dots\dots (2.10)$$

Dalam bentuk diagram vektor adalah :



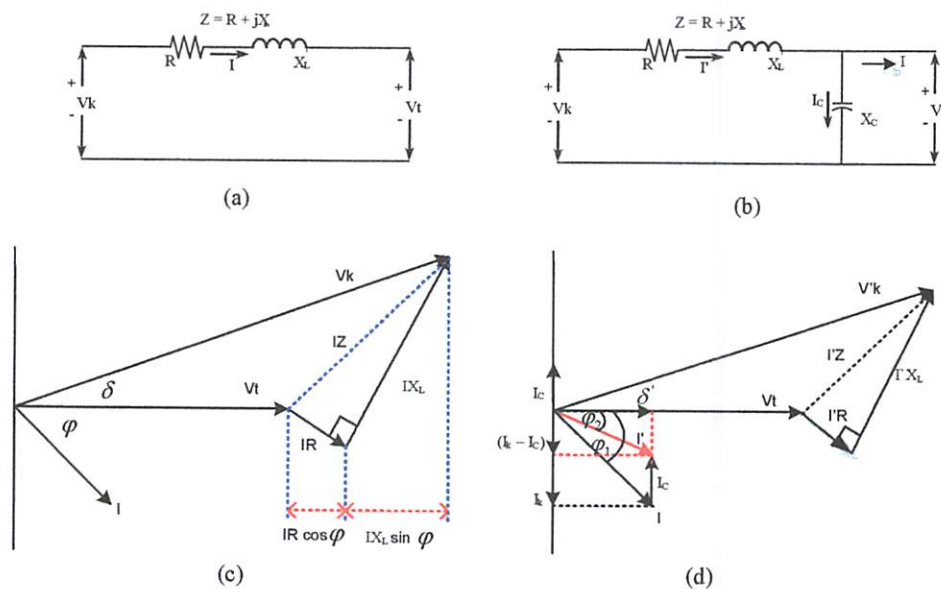
**GAMBAR 2-10 VEKTOR ARUS DAN TEGANGAN SEBELUM DAN SETELAH PEMASANGAN KAPASITOR <sup>[2]</sup>**

Sehingga pengurangan rugi daya dengan adanya pemasangan kapasitor *shunt* adalah :

$$\begin{aligned} \Delta_{pls} &= I^2R - I_1^2R \\ &= (I \cos \theta)^2R + (I \sin \theta)^2R - (I \cos \theta)^2R + (I \sin \theta - I_c)^2R \\ &= 2 (I \sin \theta)I_cR - I_c^2R \dots \dots \dots (2.11) \end{aligned}$$

**2.7.2. Perbaikan Tegangan<sup>[2]</sup>**

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



**GAMBAR 2-11 DIAGRAM VEKTOR PADA RANGKAIAN DENGAN PF LAGGING (a) DAN (c) TANPA KAPASITOR SHUNT, (b) DAN (d) DENGAN KAPASITOR SHUNT<sup>[2]</sup>**

Jatuh tegangan pada jaringan dengan *pf lagging* dapat dihitung sebagai berikut :

Sebelum pemasangan kapasitor :

$$\delta V = IR \cos \phi + IX_L \sin \phi \quad \text{volt} \dots\dots\dots (2.12)$$

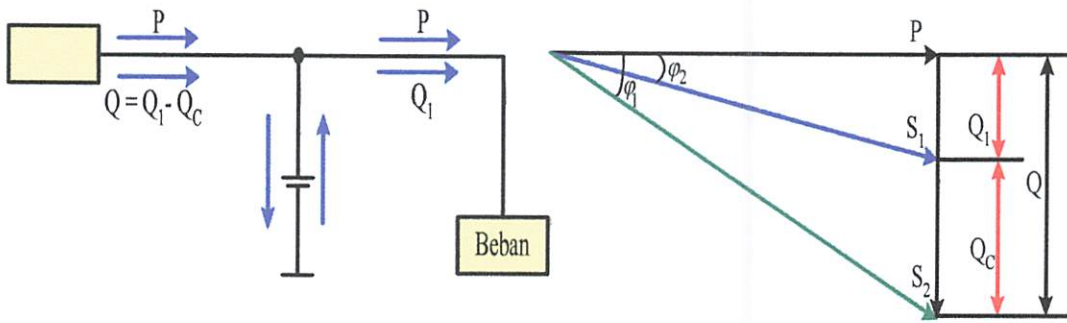
Bila kapasitor dipasang pada ujung penerima dari saluran, seperti yang terlihat pada gambar 2-11b, secara pendekatan jatuh tegangannya menjadi :

$$\delta V = IR \cos \phi + (IX_L - I_C) \sin \phi \quad \text{volt} \dots\dots\dots (2.13)$$

### 2.7.3. Perbaikan Faktor Daya dan Kenaikan 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-rugi dalam sistem. Satu-satunya jalan untuk memperbaiki faktor daya adalah mengurangi daya reaktif di jaringan, Jika komponen arus reaktif dapat dikurangi, maka total arus akan berkurang sedangkan komponen daya aktif tidak berubah, maka faktor daya akan lebih besar sebagai akibat berkurangnya daya reaktif. Faktor daya akan mencapai 100 % jika komponen daya reaktif sama dengan nol (0).

Dengan menambah kapasitor, daya reaktif  $Q$  akan berkurang, gambar 2.12 menunjukkan perbaikan faktor daya pada sistem, kapasitor mensuplai daya reaktif ke beban.



**GAMBAR 2-12 PERBAIKAN FAKTOR DAYA<sup>[2]</sup>**

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

$$\cos \varphi_1 = \frac{P}{S_2} \dots\dots\dots (2.14)$$

Bila suatu kapasitor  $Q_c$  kVAR dipasang pada beban, faktor daya dapat diperbaiki dari  $\cos \varphi_1$  menjadi  $\cos \varphi_2$  dimana:

$$\cos \varphi_2 = \frac{P}{S_1}$$

$$\cos \varphi_2 = \frac{P}{\sqrt{P^2 + Q_1^2}}$$

$$\cos \varphi_2 = \frac{P}{\sqrt{P^2 + (Q_1 - Q_c)^2}} \dots\dots\dots (2.15)$$

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

dapat dihitung menggunakan rumus berikut:

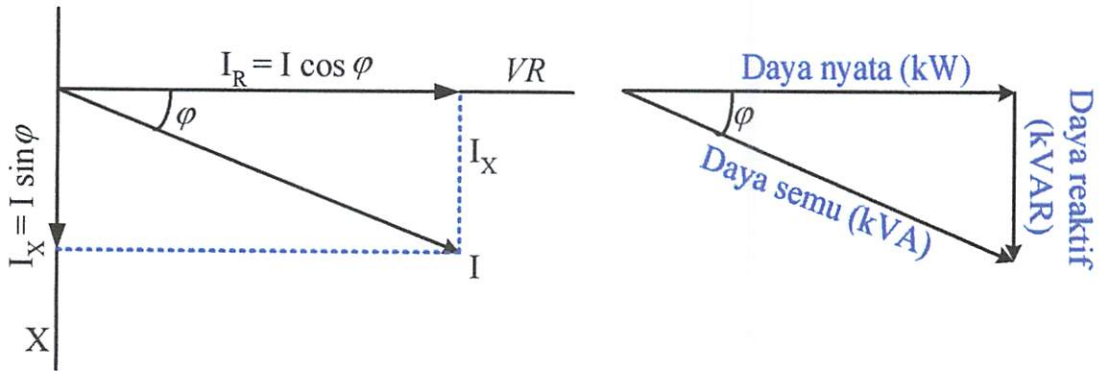
$$\% \text{ Rugi Daya} = 100 \left( \frac{\text{Faktor.daya.mula} - \text{mula}(\cos \varphi_1)}{\text{Faktor.daya.baru}(\cos \varphi_2)} \right) \dots\dots\dots (2.16)$$

%Pengurangan Rugi Daya

$$= 100 \left( 1 - \left( \frac{\text{Faktor.daya.mula} - \text{mula}(\cos \varphi_1)}{\text{Faktor.daya.baru}(\cos \varphi_2)} \right)^2 \right) \dots\dots\dots (2.17)$$

**2.7.4. Perhitungan Pengaruh Perbaikan Faktor Daya**

Diagram *phase* dari dua komponen arus nyata itu arus aktif dan reaktif dapat dilihat pada gambar 2-13 berikut:



**GAMBAR 2-13 DIAGRAM FASOR DAN SUDUT DAYA BEBAN DISTRIBUSI<sup>[2]</sup>**

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

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

$$= \sqrt{(I \cdot \cos \varphi)^2 + (I \cdot \sin \varphi)^2} \dots\dots\dots (2.18)$$

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

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

$$(VI) = \sqrt{(VI.\text{Cos}.\varphi)^2 + (VI.\text{Sin}.\varphi)^2} \dots\dots\dots (2.19)$$

$$\text{Faktor Daya} = \frac{\text{Daya.aktif}}{\text{Daya.Semu}} = \frac{kW}{kVA}$$

Daya Aktif = Daya Semu\* Faktor Daya

$$kW = kVA * \text{Faktor Daya}$$

$$kW = kVA \text{ Cos } \varphi \dots\dots\dots (2.20)$$

### 2.7.5. Penentuan Rating Kapasitor untuk Perbaikan Faktor Daya Beban

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

$$\text{Cos } \varphi = \frac{\text{Daya.aktif}}{\text{Daya.semua}} = \frac{(kW)}{(kVA)} \dots\dots\dots (2.21)$$

$$\text{Sin } \varphi = \frac{\text{Daya.reaktif}}{\text{Daya.semua}} = \frac{(kVAR)}{(kVA)} \dots\dots\dots (2.22)$$

$$\text{Tan } \varphi = \frac{(kVAR)}{(kW)} \dots\dots\dots (2.23)$$

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:



Daya reaktif pada daya mula-mula = Daya aktif x Tan  $\varphi_1$

$$= (\text{kW}) \times \text{Tan } \varphi_1$$

Daya reaktif pada faktor daya baru = Daya aktif x Tan  $\varphi_2$

$$= (\text{kW}) \times \text{Tan } \varphi_2$$

Dengan  $\varphi_1$  = Sudut dari faktor daya mula-mula

$\varphi_2$  = Sudut dari faktor daya yang telah diperbaiki

Rating kapasitor yang dibutuhkan perbaikan factor daya:

Daya reaktif (kVAR) = Daya aktif x (Tan  $\varphi_1 - \varphi_2$ )

$$= (\text{kW}) \times (\text{Tan } \varphi_1 - \varphi_2) \dots \dots \dots (2.24)$$

Untuk penyederhanaan (Tan  $\varphi_1 - \varphi_2$ ) sering ditulis  $\Delta \text{Tan}$ , yang merupakan suatu faktor pengali untuk menentukan daya reaktif.

Daya reaktif (kVAR) = Daya aktif x  $\Delta \text{Tan}$

$$(\text{kVAR}) = (\text{kW}) \times \Delta \text{Tan} \dots \dots \dots (2.25)$$

## 2.8. Sistem Per-Unit<sup>[4]</sup>

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

$$\text{Besaran per - unit} = \frac{\text{besaran sesungguhnya}}{\text{besaran dasar dengan dimensi yang sama}} \dots \dots \dots (2.26)$$



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

- Untuk data 1 fasa

Arus dasar:

$$I_d = \frac{\text{kVA dasar 1 fasa}}{\text{kV dasar L - N}} \dots\dots\dots(2.27)$$

Impedansi dasar:

$$\begin{aligned} Z_d &= \frac{(\text{kV dasar L - N})^2 \times 1000}{\text{kVA dasar 1 fasa}} \dots\dots\dots(2.28) \\ &= \frac{(\text{kV dasar L - N})^2}{\text{MVA dasar 1 fasa}} \end{aligned}$$

Dalam persamaan diatas nilai-nilai besaran diberikan untuk rangkaian satu fasa. Jadi tegangannya adalah tegangan antara fasa ke tanah dan daya setiap fasa.

Setelah besaran-besaran dasar telah ditentukan maka besaran-besaran itu dinormalisasikan terhadap besaran dasar. Dengan demikian impedansi per satuan didefinisikan sebagai berikut :

$$Z = \frac{\text{Impedansi sebenarnya } Z (\Omega)}{\text{Impedansi dasar } Z_d} \dots\dots\dots(2.29)$$

## 2.9. Variasi Tegangan<sup>[2]</sup>

Seperti diketahui, pen-catuan tenaga listrik kepada para pelanggan tegangannya (tegangan pelayanan), tidak dapat konstan, karena adanya impedansi dari jaringan yang mencatu. Dengan kata lain tegangan pelayanan bervariasi pada setiap pelanggan.

Variasi tegangan pelayanan ini, tentu saja ada batas-batas toleransinya. Untuk Indonesia, toleransi tegangan yang diizinkan, didasarkan pada standar PLN, yaitu SPLN 1:1978, dimana ditentukan bahwa variasi tegangan pelayanan, sebagai akibat jatuh tegangan, karena adanya perubahan beban, maksimum +5% dan minimum -10% dari tegangan nominalnya.

### BAB III

## METODE *HYBRID GA-SA* DAN PENERAPAN

Sebelum melakukan optimasi dengan menggunakan metode *Hybrid GA-SA* dilakukan suatu proses analisa aliran daya untuk mengetahui kondisi suatu sistem distribusi radial. Tujuan mempelajari analisa aliran daya adalah :

- ✎ Untuk mengetahui keadaan tegangan pada setiap bus dari sistem jaringan.
- ✎ Untuk mengetahui besarnya daya yang mengalir pada setiap cabang dari struktur jaringan.
- ✎ Untuk mengetahui besar rugi-rugi daya aktif dan daya reaktif pada setiap cabang dari saluran.

### 3.1 Metode Newton Raphson<sup>[4]</sup>

Secara matematis persamaan aliran daya *Newton Raphson* dapat diselesaikan dengan menggunakan koordinat rektanguler, 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 polar, dimana tegangan dinyatakan dalam bentuk polar, yaitu :

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

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

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

Maka persamaan (3.3) 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 diatas akan menghasilkan suatu kumpulan persamaan serempak (simultan) yang tidak linier untuk setiap simpul sistem tenaga listrik. Untuk mengetahui magnitude tegangan (V) dan sudut fasa ( $\delta$ ) disetiap simpul dapat diselesaikan dengan menggunakan persamaan (3.5) dan (3.6) yang dilinierkan dengan metode *Newton Raphson* yang dapat dilihat dari persamaan dibawah 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 didapat 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 Jacqobian yang dibentuk dengan mendefinisikan persamaan (3.5) dan (3.6),

dimana:

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

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

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 :

$$|V|^{k+1} = |V|^k + \Delta|V|^k \dots\dots\dots (3.8)$$

$$\delta^{k+1} = \delta^k + \Delta\delta^k \dots\dots\dots (3.9)$$

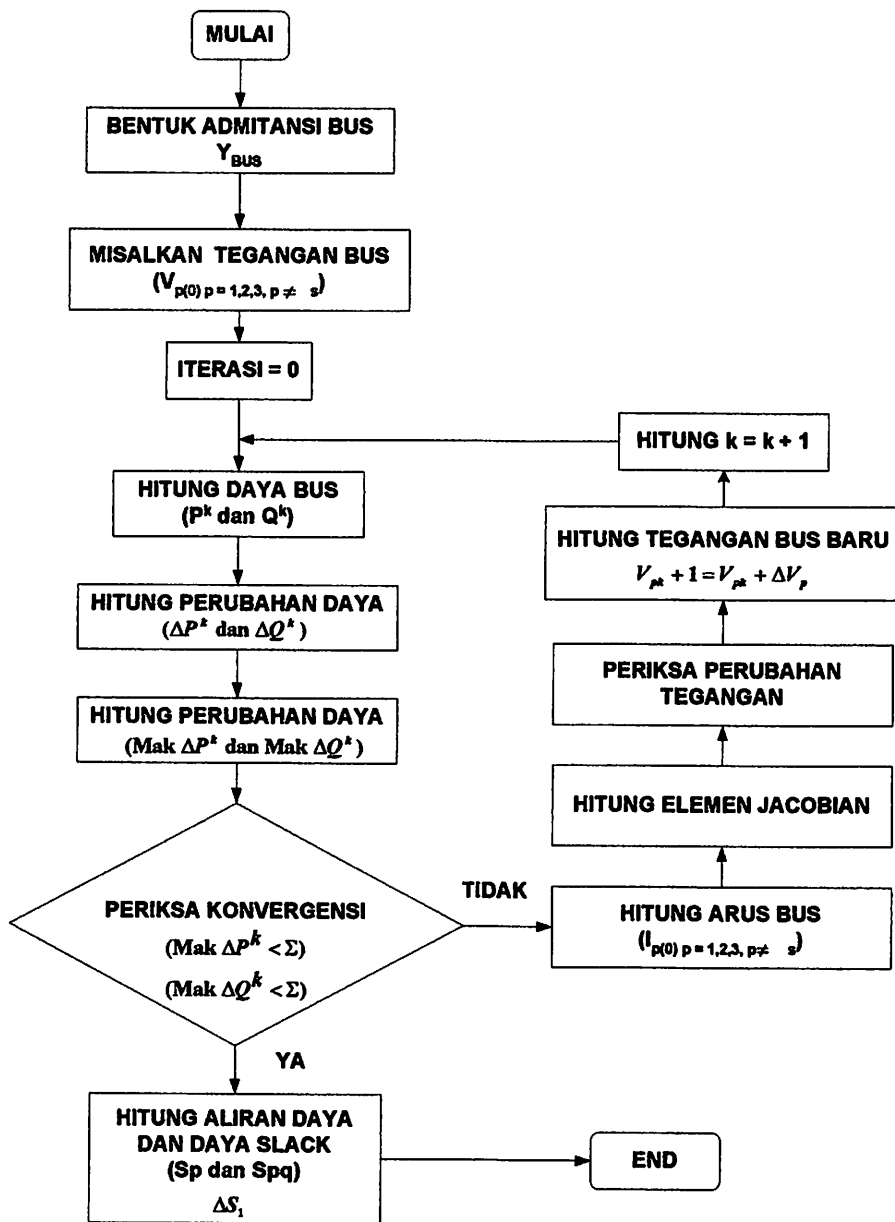
Proses perhitungan akan berulang sampai selisih daya nyata dan daya reaktif antara yang dijadwalkan dengan yang dihitung, yaitu  $\Delta P$  dan  $\Delta Q$  untuk

semua simpul mendekati nilai toleransi atau proses perhitungan iterasi mencapai konvergen.

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

1. Tentukan nilai  $P_{p(\text{ditetapkan})}$  dan  $Q_{p(\text{ditetapkan})}$  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  $\Delta P$  pada setiap rel.
3. Hitung nilai-nilai matrik Jacobian dengan menggunakan nilai-nilai perkiraan atau yang ditentukan dari besar dan sudut tegangan dalam persamaan untuk turunan parsial yang ditentukan dengan differensiasi persamaan (3.5) dan (3.6).
4. Hitung koreksi-koreksi tegangan  $\Delta\delta_q$  dan  $\Delta|V_q|$  pada nilai sebelumnya.
5. Hitung nilai baru dari  $\delta_q$  dan  $|V_q|$  dengan menambahkan  $\Delta\delta_p$  dan  $\Delta|V_q|$  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.

### 3.1.2. Flowchart Algoritma Aliran Daya Newton Raphson



**GAMBAR 3-1 FLOWCHART ALGORITMA ALIRAN DAYA NEWTON RAPHSON**

## 3.2. Metode *Hybrid GA-SA*

### 3.2.1. Algoritma Genetika<sup>[5]</sup>

Algoritma Genetika merupakan metode *adaptive* yang bisa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam makhluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam “siapa yang kuat, dia yang bertahan (*survive*)”. Dengan meniru proses ini, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

Algoritma Genetika ditemukan oleh John Holland pada awal tahun 1970 yang dilandasi oleh sifat-sifat evolusi alam. Holland percaya bahwa ini sangat cocok digabungkan dalam sebuah algoritma komputer, menghasilkan sebuah teknik penyelesaian untuk permasalahan-permasalahan yang sulit dengan langkah alami yaitu melalui evolusi. John Holland mulai bekerja dengan algoritma yang dibentuk dari string-string biner 1 dan 0 yang disebut *kromosom*. Seperti halnya alam, algoritma ini menyelesaikan permasalahan-permasalahan dengan menemukan kromosom-kromosom yang baik dengan memanipulasi materi dan sifat (*gene*) kromosom-kromosom. Algoritma ini tidak mengetahui type permasalahan yang akan diselesaikan. Hanya informasi yang telah diberikan dari *evaluasi* berupa nilai fitness setiap kromosom dengan nilai fitness terbaik yang bertahan hidup dan selalu diproduksi.

Sebelum Algoritma Genetika dijalankan, maka sebuah kode yang sesuai (representasi) untuk persoalan harus dirancang. Titik solusi dalam ruang



permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pemakaian bilangan seperti integer, floating point dan abjad sebagai *allele* (nilai gen) memungkinkan penerapan operator genetika yaitu proses produksi (*reproduction*), pindah silang (*crossover*), mutasi (*mutation*) untuk menciptakan himpunan titik-titik solusi. Untuk memeriksa hasil optimasi, kita membutuhkan fungsi *fitness* yang menandakan gambaran hasil (*solution*) yang sudah dikodekan. Selama proses, induk harus digunakan untuk reproduksi, pindah silang dan mutasi untuk menciptakan keturunan (*offspring*). Jika Algoritma Genetika didesain dengan baik, populasi akan mengalami konvergensi dan akan mendapatkan sebuah solusi yang optimum.

Algoritma Genetika memiliki empat dasar kerja yaitu :

1. Bekerja dengan mengkodekan parameter-parameter permasalahan dan tidak bekerja secara langsung dengan parameter-parameter tersebut.
2. Mencari solusi masalah dari sejumlah populasi kandidat solusi, tidak hanya memproses satu solusi saja.
3. Hanya memperhitungkan fungsi fitness setiap kandidat solusi untuk mendapatkan hasil optimum global.
4. Menggunakan aturan transisi secara probabilistik bukan deterministik.

#### **3.2.1.1. Istilah-Istilah Algoritma Genetika**

Algoritma Genetika menggunakan mekanisme genetika yang ada pada proses alami dan sistem buatan. Istilah-istilah yang digunakan adalah gabungan

dari dua disiplin ilmu, yaitu ilmu Biologi dan ilmu computer. Mitsuo Gen dan Runwei Cheng (1997) menjelaskan istilah-istilah yang digunakan dalam Algoritma Genetika sebagai berikut :

**TABEL 3-1 ISTILAH YANG DIGUNAKAN DALAM ALGORITMA GENETIKA**

<b>Istilah</b>	<b>Keterangan</b>
Kromosom	Individu berupa segmen string yang sudah ditentukan
Gen	Bagian dari string
Loci	Posisi dari gen
Allele	Nilai yang dimasukkan dalam gen
Phenotype	String yang merupakan solusi terakhir
Genotype	Sejumlah string hasil perkawinan yang berpotensi sebagai solusi

Terdapat beberapa parameter yang digunakan dalam Algoritma Genetika. Parameter tersebut digunakan untuk melihat kompleksitas dari Algoritma Genetika. Parameter yang digunakan tersebut adalah :

#### **Jumlah Generasi (*MAXGEN*)**

Merupakan jumlah perulangan (iterasi) dilakukannya rekombinasi dan seleksi. Jumlah generasi ini mempengaruhi kestabilan output dan lama iterasi (waktu proses Algoritma Genetika). Jumlah generasi yang besar dapat mengarahkan kearah solusi yang optimal, namun akan membutuhkan waktu yang lama. Sedangkan jika jumlah generasinya terlalu sedikit maka solusi akan terjebak pada lokal optimum.

### **Ukuran Populasi (*POPSIZE*)**

Ukuran populasi mempengaruhi kinerja dan efektifitas dari Algoritma Genetika. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja Algoritma Genetika menjadi buruk. Dalam hal ini dibutuhkan ruang yang lebih besar untuk mempersentasikan keseluruhan ruang permasalahan. Selain itu penggunaan populasi yang besar dapat mencegah terjasinya konvergensi pada wilayah local. Zbigniew Michalewics (1996) berpendapat banyak aplikasi Algoritma Genetika mempergunakan populasi pada range 50-100.

### **Probabilitas Crossover (*Pc*)**

Probabilitas crossover ini digunakan untuk mengendalikan frekuensi operator crossover. Dalam hal ini, dalam populasi terdapat  $P_c \times POPSIZE$  struktur (individu) yang melakukan pindah silang. Semakin besar nilai probabilitas crossover maka semakin cepat struktur baru yang diperkenalkan dalam populasi. Namun jika probabilitas crossover terlalu besar maka struktur dengan nilai fungsi obyektif yang baik dapat hilang dengan lebih cepat dari seleksi. Sebaliknya jika probabilitas terlalu kecil akan menghalangi proses pencarian dalam proses Algoritma Genetika. Zbigniew Michalewics (1996) berpendapat banyak aplikasi Algoritma Genetika mempergunakan angka probabilitas crossover pada range 0.65 – 1.

### **Probabilitas Mutasi ( $P_m$ )**

Mutasi digunakan untuk meningkatkan variasi populasi digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi  $P_m \times \text{POPSIZE} \times N$ , dimana  $N$  adalah panjang struktur / gen dalam satu individu. Probabilitas mutasi yang rendah akan menyebabkan gen-gen yang berpotensi tidak dicoba. Dan sebaliknya, tingkat mutasi yang tinggi akan menyebabkan keturunan akan semakin mirip dengan induknya. Dalam Algoritma Genetika, mutasi menjalankan aturan penting yaitu :

1. Mengganti gen-gen yang hilang sama proses seleksi.
2. Menyediakan gen-gen yang tidak muncul pada saat inialisasi awal populasi.

Zbigniew Michalewics (1996) berpendapat banyak aplikasi Algoritma Genetika mempergunakan angka probabilitas mutasi pada daerah range 0.001 – 0.01.

### **Panjang Kromosom ( $NVAR$ )**

Panjang kromosom berbeda-beda sesuai dengan model permasalahan. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pengkodean dapat memakai bilangan seperti string biner, integer, floating point dan abjad.

#### **3.2.1.2. Proses Algoritma Genetika**

Sangat perlu untuk mengetahui proses dalam Algoritma Genetika. Dibawah ini akan diuraikan mengenai hal itu, dimana uraian ini merupakan penjabaran dari Algoritma Genetika seperti penjelasan pada bagian berikutnya.

### A. Pengkodean atau Representasi

Langkah pertama kali yang dilakukan dalam penggunaan Algoritma Genetika adalah melakukan pengkodean atau representasi terhadap permasalahan yang akan dilakukan.

Secara umum Algoritma Genetika dibentuk oleh serangkaian kromosom yang ditandai dengan  $x_i$  ( $i = 1, 2 \dots N$ ). Setiap elemen dalam kromosom ini adalah variabel string yang disebut gen, berisi nilai-nilai atau allele. Variabel-variabel ini dapat dinyatakan dalam bentuk bilangan biner, bilangan real (*floating point*), integer, abjad. Pengkodean string biner merupakan pendekatan klasik yang digunakan dalam penelitian Algoritma Genetika karena sederhana. Meskipun representasi dengan cara ini menyulitkan untuk beberapa permasalahan optimasi, misalnya permasalahan graph coloring. Digunakan teknik pengkodean yang lain seperti representasi real number (*floating point*), representasi *order-based* (untuk *bin-patching*, *graph coloring*), *embedded lists* (untuk permasalahan penjadwalan (*scheduling*), *variable element list* (untuk semi konduktor layout), dan even *LISP S-expressions*.

Selanjutnya beberapa kromosom dibentuk dan berkumpul membentuk populasi. Populasi inilah populasi awal bagi Algoritma Genetika untuk awal melakukan pencarian.

### B. Fungsi *Fitness* (Fungsi Evaluasi)

Dalam Algoritma Genetika, sebuah fungsi *fitness*  $f(x)$  harus dirancang untuk masing-masing permasalahan yang akan diselesaikan. Dengan

menggunakan kromosom tertentu, fungsi obyektif atau fungsi evaluasi akan mengevaluasi status masing-masing kromosom. Setiap gen  $x_i$  ( $i = 1, 2, \dots, N$ ) dipergunakan untuk menghitung  $f_k(x)$  ( $k = 1, 2, \dots, \text{POPSIZE}$ ).

Pada permulaan optimasi, biasanya nilai *fitness* masing-masing individu masih mempunyai rentang yang lebar. Seiring dengan bertambah besar generasi, beberapa kromosom mendominasi populasi dan mengakibatkan rentang nilai *fitness* semakin kecil. Hal ini dapat mengakibatkan konvergensi dini (*premature convergence*).

Permasalahan klasik dalam Algoritma Genetika adalah beberapa kromosom dengan nilai *fitness* yang tinggi (tetap bukan nilai optimum) mendominasi populasi dan mengakibatkan Algoritma Genetika konvergen pada lokal optimum. Ketika mencapai konvergen, kemampuan Algoritma Genetika untuk mencari solusi yang lebih baik menghilang. Tukar silang antara kromosom induk yang hampir identik. Dalam hal ini hanya operasi mutasi yang mampu menghasilkan kromosom yang relatif baru dan merupakan cara untuk menghindari kromosom tertentu mendominasi populasi.

### C. Seleksi

Pada Algoritma Genetika terdapat proses seleksi yaitu proses pemilihan kromosom yang akan di-crossover-kan dengan kromosom dari individu lain. Masalah yang paling mendasar pada proses ini adalah bagaimana proses penyeleksiannya. Menurut teori Darwin proses seleksi individu adalah : "*individu terbaik akan tetap hidup dan menghasilkan keturunan*". Pada proses seleksi ini

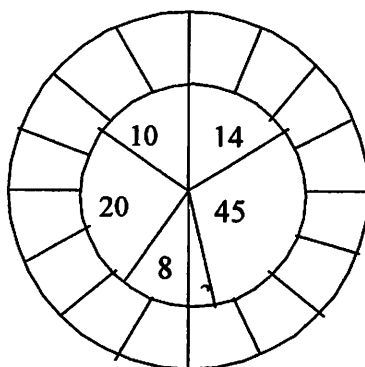
dapat menggunakan banyak metode seperti *rouleete wheel selection*, *rank selection*, *elitesm* dan lain sebagainya.

### Roulette Wheel Selection

Dimana setiap individual memiliki harga fitness sehingga didapatkan probabilitas individual  $(f(t)/\sum f(t))$  tersebut dicopykan pada populasi yang baru. Untuk individual yang memiliki probabilitas 20% untuk jumlah populasi 10 maka kemungkinan individual tersebut dapat terpilih sebanyak dua kali. Ilustrasi kerja operator ini dapat digambarkan seperti pada gambar 3-1.

Adapun algoritma dari *roulette-wheel* adalah sebagai berikut :

1. Menjumlahkan fitness dari seluruh anggota populasi.
2. Membangkitkan nilai k, suatu nilai random antara 0 dan total fitnessnya.
3. Menjumlahkan fitness dari kromosom-kromosom dari populasi mulai 0 hingga total fitness lebih besar atau sama dengan nilai k lalu ambil kromosom tersebut.



**GAMBAR 3-2 ROULETTE-WHEEL<sup>[5]</sup>**

### Rank Selection

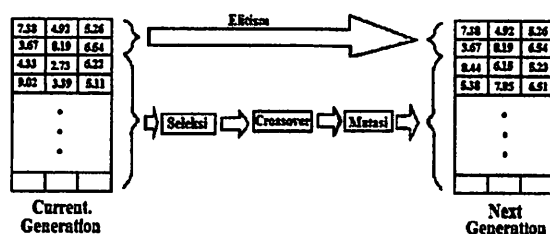
Apabila fitness yang dimiliki oleh suatu kromosom dalam populasi berbeda terlalu jauh dari kromosom lainnya maka hal ini dapat menjadi permasalahan. Misalnya bila kromosom terbaik mempunyai fitness yang menyebabkan besarnya tempat yang dimilikinya dalam *roulette wheel* sebesar 90% maka kromosom-kromosom yang lain akan mempunyai peluang yang terlalu kecil untuk diseleksi.

Rank selection pertama kali merangking populasi dan kemudian setiap kromosom diberi nilai fitness baru berdasarkan hasil rangking tersebut. Yang pertama akan mempunyai *fitness* 1, yang kedua akan mempunyai fitness 2 dan seterusnya sampai yang terakhir akan mempunyai *fitness* N. Dengan demikian semua kromosom akan mempunyai peluang untuk diseleksi..

#### 3.2.1.3. Elitism

Selama membuat populasi baru dengan *crossover* dan mutasi, kemungkinan akan terjadi kehilangan kromosom terbaik (*best/few best*). Elitism adalah metode yang pertama kali meng-copy-kan kromosom terbaik (*best/few best*) kedalam populasi baru. Sisanya dikerjakan dengan cara biasa, yaitu melalui seleksi, *crossover* dan mutasi. Elitism dapat secara cepat meningkatkan performansi dari Algoritma Genetika karena *elitism* menghindarkan hilangnya hilangnya solusi terbaik (*best / few best*) yang telah ditemukan. Ilustrasi kerja operator ini dapat digambarkan seperti pada gambar 3-3.



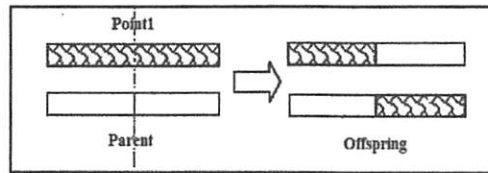


**GAMBAR 3-3 PEMBENTUKAN *NEXT GENERATION* DALAM ALGORITMA GENETIKA<sup>[5]</sup>**

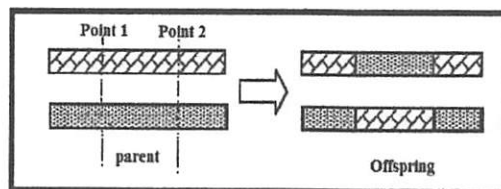
#### 3.2.1.4 *Crossover* (Pindah Silang)

Fungsi dari *crossover* adalah menghasilkan kromosom anak dari kombinasi materi-msteri gen dua kromosom induk. Cara kerjanya dengan membangkitkan sebuah nilai random  $r_k$  dimana  $k = 1, 2, \dots, \text{POPSIZE}$ . Probabilitas *crossover* ( $P_c$ ) ditentukan dan digunakan untuk mengendalikan frekuensi operator *crossover*. Apabila nilai  $r_k < P_c$  maka kromosom ke- $k$  terpilih untuk mengalami *crossover*. *Crossover* yang paling sederhana adalah *one point crossover*. Posisi titik persilangan (point) ditentukan secara random pada range satu sampai panjang kromosom. Kemudian nilai *offspring* diambil dari dua parent tersebut dengan batas titik persilangan tersebut. Ilustrasi kerja operator ini digambarkan seperti pada gambar 3-4.

Kemudian ditingkatkan lagi dengan menggunakan *two point crossover*. Penentuan posisi titik persilangan sama seperti sama seperti *one point crossover* sebelumnya. Pemilihan secara random dilakukan 2 kali. Kemudian nilai *offspring* diambil dari dua parent tersebut dengan batas dua titik persilangan tersebut. Ilustrasi kerja operator ini digambarkan seperti pada gambar 3-5.

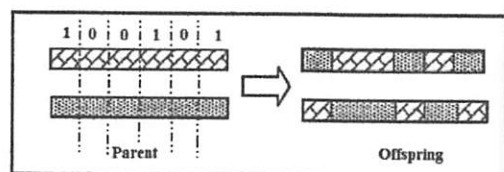


**GAMBAR 3-4 ILLUSTRASI OPERATOR DENGAN *ONE POINT Crossover*<sup>[5]</sup>**



**GAMBAR 3-5 ILLUSTRASI OPERATOR DENGAN *TWO POINT Crossover*<sup>[5]</sup>**

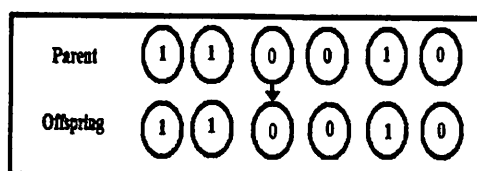
Untuk *crossover* uniform dibangkitkan suatu nilai random 0 dan 1 sepanjang jumlah kromosom untuk nilai loci. Jika nilai yang dibangkitkan mempunyai nilai 1 maka *allele* parent 2 dan offspring 2 untuk loci tersebut diambil dari *allele* parent 1 dan offspring 2 untuk loci tersebut diambil dari *allele* parent 2. Ilustrasi kerja operator ini digambarkan seperti pada gambar 3-6.



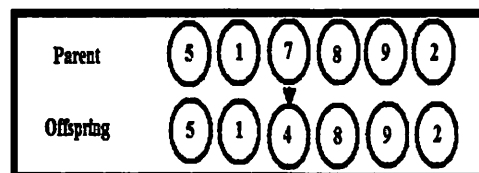
**GAMBAR 3-6 ILLUSTRASI OPERATOR *Crossover* DENGAN *UNIFORM Crossover*<sup>[5]</sup>**

### 3.2.1.5 Mutation (Mutasi)

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam satu individu. Cara kerjanya dengan membangkitkan sebuah nilai random  $r_k$  dimana  $k = 1, 2, \dots, NVAR$  (panjang kromosom). Probabilitas mutasi ( $P_m$ ) ditentukan dan digunakan untuk mengendalikan frekuensi operator mutasi. Apabila nilai random  $r_k$ ,  $P_m$  maka gen ke- $k$  kromosom tersebut terpilih untuk mengalami mutasi. Mutasi dengan mengganti gen 0 dengan 1 atau sebaliknya gen 1 dengan 0. Biasanya disebut flip yaitu membalik nilai ke 1 atau 1 ke 0. Ilustrasi kerja operator untuk representasi string biner digambarkan pada gambar 3-7. Untuk bentuk representasi integer atau floating point, atau selain string biner, seperti gambar 3-8, proses mutasi terjadi apabila nilai  $r_k < P_m$  memenuhi maka gen ke- $k$  digantikan oleh suatu nilai random yang dibangkitkan pada range tertentu sesuai dengan pembentukan populasi awal.



**GAMBAR 3-7 ILLUSTRASI OPERATOR MUTASI UNTUK REPRESENTASI STRING BINER<sup>[5]</sup>**



**GAMBAR 3-8 ILLUSTRASI OPERATOR MUTASI UNTUK REPRESENTASI INTEGER<sup>[5]</sup>**

Untuk kromosom induk seperti gambar 3-8 diatas yaitu 5-1-7-8-9-2, proses mutasi adalah dibangkitkan sebuah nilai random  $r_k$ , [ $k = 1 \dots NVAR$ ]. Misalkan pada saat  $k = 3$  nilai  $r_3 < P_m$  maka gen ke-3 yang bernilai 7 akan bermutasi dengan gen hasil random pada range  $\{1 \dots x\}$  dan diperoleh nilai 4. Maka bentuk kromosom barunya adalah 5-1-4-8-9-2. Dimana  $x$  adalah nilai sembarang integer.

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

### 3.2.2 Simulated Annealing<sup>[6]</sup>

*Simulated Annealing ( SA )*, secara independent diperkenalkan oleh *Kirkpatrick, Gela dan Vecchi* pada tahun 1982 :1983 dan *Cerny* pada tahun 1985.

*Simulated Annealing ( SA)* secara fisika, mengacu pada proses pemanasan benda padat pada temperature yang tinggi kemudian diikuti oleh pendinginan yang dicapai dengan menurunkan temperatur secara bertahap. Pada masing – masing perulangan, dimana suatu kandidat solusi dihasilkan. Kriteria ini dirangkum sebagai berikut :

- Keadaan dengan suatu solusi lebih rendah ( baik ) akan diterima.
- Keadaan dengan solusi lebih tinggi ( jelek ) akan diterima secara terbatas dengan kemungkinan  $\Pr(\Delta)$ . Pernyataan dari kemungkinan penerimaan yang sekarang ini memakai persamaan sebagai berikut :

$$\Pr(\Delta) = [1 / \{1 + \exp(\Delta / T)\}] \dots \dots \dots (3.9)$$

Dimana :

$\Delta$  adalah jumlah penurunan antara solusi lama dengan solusi baru,

$T$  adalah temperatur dimana solusi baru dihasilkan.

Dalam membentuk solusi baru, solusi lama digunakan menurut fungsi distribusi probabilitas *Gaussian ( GPDF )*. *GPDF* diasumsikan sebagai solusi lama dan standar deviasi ditentukan sebagai produk dari temperatur dan faktor skala  $\gamma$ . Solusi baru dibentuk dengan menambahkan jumlah gangguan terhadap solusi lama. Jumlah gangguan tergantung pada temperature ketika faktor skala  $\gamma$  dijaga konstan.

### 3.2.2.1 Jadwal Pendinginan

Jadwal pendinginan diimplementasikan dengan menggunakan jadwal pendinginan *Kirk Patrick*.

- **Nilai awal dari temperatur  $T$**

Nilai awal dari  $T$  biasanya ditentukan pada nilai yang besar sehingga solusi percobaan dengan biaya yang lebih tinggi dapat diterima. Konsekuensinya, ketika  $T$  berkurang secara berangsur – angsur dari nilai awal yang sangat tinggi, maka memungkinkan proses keluar dari titik lokal optimum dalam lingkup ruang percobaan yang sangat besar. Untuk nilai  $T$  yang tepat, maka probabilitas penerimaan dapat diperkirakan dengan rasio  $X_0$ , sebagai berikut :

$$X_0 = 1/(1 + \exp(\Delta C^+ / T)) \dots\dots\dots(3.10)$$

Rasio  $X_0$  didefinisikan sebagai rasio jumlah solusi biaya yang lebih tinggi diterima sebagai jumlah solusi biaya lebih tinggi yang dihasilkan. Sedangkan  $\Delta C$  adalah kenaikan biaya rata – rata dari solusi biaya yang lebih tinggi.

Nilai dari  $X_0$  diketahui, maka nilai awal temperatur dapat diestimasi sebagai berikut :

$$T_0 = [\Delta C^+ / \ln((1 / X_0) - 1)] \dots\dots\dots(3.11)$$

- **Nilai akhir dari temperatur  $T_k$**

Dari penjelasan diatas, maka akan diperoleh nilai akhir dari temperatur  $T_k$  yang diestimasi sebagai berikut :

$$T_k = r^{(k-1)}.T_0 \dots\dots\dots(3.12)$$

Dimana  $T_k$  dan  $T_0$  adalah temperatur ke- $k$  dan temperatur awal.  $k$  adalah counter pendingin dan  $r$  adalah faktor pengurangan temperatur. Proses dari solusi ini dilakukan sampai perulangan maksimum dicapai atau solusi optimum ditemukan.

### 3.2.2.2. Kriteria Penghentian

Proses iterasi dapat diakhiri oleh salah satu jalan berikut :

- 1) Manakala jumlah ditetapkan iterasi tercapai.
- 2) Jika tidak ada peningkatan didalam solusi pada sejumlah iterasi yang ditetapkan.

Kriteria penghentian yang dipakai menggunakan pendekatan (1). Jumlah maksimum iterasi yang ke  $I$  untuk suatu nilai akhir temperatur, ditetapkan yang ditandai oleh  $T_k$  ditentukan dari persamaan, dan  $k=1$ .

$$I = \frac{\log(T_k / T_0)}{\log(r)} + 1 \dots \dots \dots (3.13)$$

Ukuran dari ruang pendekatan didalam iterasi akhir tergantung pada pilihan  $T_k$  dalam menentukan kedekatan atau resolusi dari solusi pada jumlah titik maksimum. Dimana  $I$  adalah nomor maksimum iterasi.

### 3.2.3 Hybrid GA-SA<sup>[1]</sup>

Teknik heuristik yang telah dibahas di atas dikombinasikan untuk membentuk sebuah algoritma hybrid yang memiliki keuntungan. Efisiensi dari algoritma hybrid baru diharapkan meningkat.

Salah satu cara untuk membentuk sebuah algoritma hybrid adalah memulainya dengan sebuah heuristik untuk memberikan solusi yang agak baik bukanya memulai secara acak. Solusi ini sebagai solusi awal dimana heuristik

lainya yang mengikutinya akan memulai pencarian solusi optimal. Dalam skripsi ini, penggabungan algoritma GA dan SA membentuk sebuah algoritma hybrid. Algoritma ini disebut dengan algoritma *hybrid GA – SA*. Algoritma GA diterapkan pertama kali untuk memberikan sebuah solusi awal yang baik. Kemudian, SA memulai pencarian dari solusi ini untuk mencari solusi yang optimal.

### 3.3. Objective Function<sup>[1]</sup>

Tujuan dari masalah penempatan kapasitor adalah mengurangi total kehilangan energi dari semua level beban dan tingkat tegangan dalam batas yang diijinkan, untuk meminimalkan biaya dari kapasitor yang dipasang pada sistem. Fungsi tujuan terdiri dari dua hal. Pertama, biaya penempatan kapasitor dan kedua adalah biaya total kehilangan energi, yang ditunjukkan sebagai berikut:

$$\text{Minimize } \sum_{k=1}^{n_c} C_k(u_k^0) + k_e \sum_{i=1}^{n_l} T_i P_{loss,i}(x^i, u^i) \dots \dots \dots (3.14)$$

Keterangan :

1.  $u_k^0 = I_k * u_s$  dimana  $I_k$  adalah integer non negatif,  $k \in N_C$
2.  $u_k^i =$  variabel diskrit,  $i \in N_T, k \in N_C$
3. batasan aliran daya  $P_{flow}(x^i, u^i) = 0 \quad i \in N_T$
4. batasan operasional  $V_{min} \leq |V_{ik}| \leq V_{max}$
5. untuk  $k \in C_1 =$  fixed kapasitor  $u_k^i = u_k^j$  untuk  $i, j \in N_T$
6. untuk  $k \in C_2 =$  switched kapasitor  $0 \leq u_k^i$  untuk  $i \in N_T$
7.  $I_k \leq (I_k)_{max}$



Dimana :

$\sum_{k=1}^{n_c} C_k(u_k^o)$  menunjukkan biaya pemasangan dan pembelian kapasitor.

$k_e \sum_{i=1}^{n_l} T_i P_{loss,i}(x^i, u^i)$  menunjukkan biaya total kehilangan energi.

Variabel – variabelnya didefinisikan sebagai berikut :

- $i$  : level beban.
- $k$  : lokasi ( bus ).
- $x^i$  : vektor dari variabel – variabel keadaan.
- $u^i$  : vektor dari variabel-variabel kontrol (skema penempatan kapasitor).
- $n_c$  lokasi – lokasi memungkinkan untuk memasang kapasitor.
- $u_k^o$  : nilai  $kVar$  dari kapasitor pada beban puncak.
- $k_c$  : biaya energi per unit (\$/kWh).
- $n_l$  : jumlah level beban.
- $T_i$  : durasi untuk level beban  $i$ .
- $P_{loss,i}$  : kehilangan beban (kW).
- $u_s$  : ukuran standar dari kapasitor.
- $N_c$  : lokasi yang memungkinkan untuk dipasang kapasitor.
- $N_l$  : level beban yang berbeda.
- $|V_{ik}|$  : total tegangan rms pada bus ke-k selama level beban ke-i.
- $V_{min}$  : tegangan operasional minimum yang diijinkan.

$V_{\max}$  : tegangan operasional maksimum yang diijinkan.

$C_1$  : set dari kapasitor fixed.

$C_2$  : set dari kapasitor switched.

$(I_k)_{\max}$  : jumlah maks dari kapasitor yang akan dipasang pada bus(k).

### 3.3.1. Algoritma Program *Genetic Algorithm*

1. Masukkan dan baca data sistem dan jaringan berupa data beban P, Q. Data saluran R,X.  $P_{\text{dasar}}$ ,  $V_{\text{dasar}}$ . Biaya dan kapasitas kapasitor, tegangan operasional minimum dan maksimum yang diperkenankan. Input parameter GA. Jumlah populasi (*Pop Size*), Maksimum Generasi (*Max Gen*), Nilai kemungkinan *Crossover*( $P_c$ ), Nilai kemungkinan Mutasi( $P_m$ ), dan panjang kromosom tiap individu.
2. Hitunglah *losses* daya selama masing-masing level beban, total kehilangan energi, tegangan pada masing-masing bus.
3. Hasilkan seperangkat solusi awal yang layak, dengan membentuk populasi awal secara acak.
4. Hitunglah nilai *fitnes* dari masing-masing solusi.
5. Hitung nilai *fitnes* rata-rata, juga hitunglah seleksi probabilitas dari masing-masing individu.
6. pindahkan semua individu dimana nilai *fitnes* lebih kecil dibandingkan nilai *fitnes* rata-rata yang telah dihitung sebagai generasi berikutnya tanpa adanya perubahan
7. Pilih satu induk. Kemudian pilihlah induk lainnya secara acak. Gunakan operator *crossover* dan mutasi untuk menghasilkan keturunan baru.

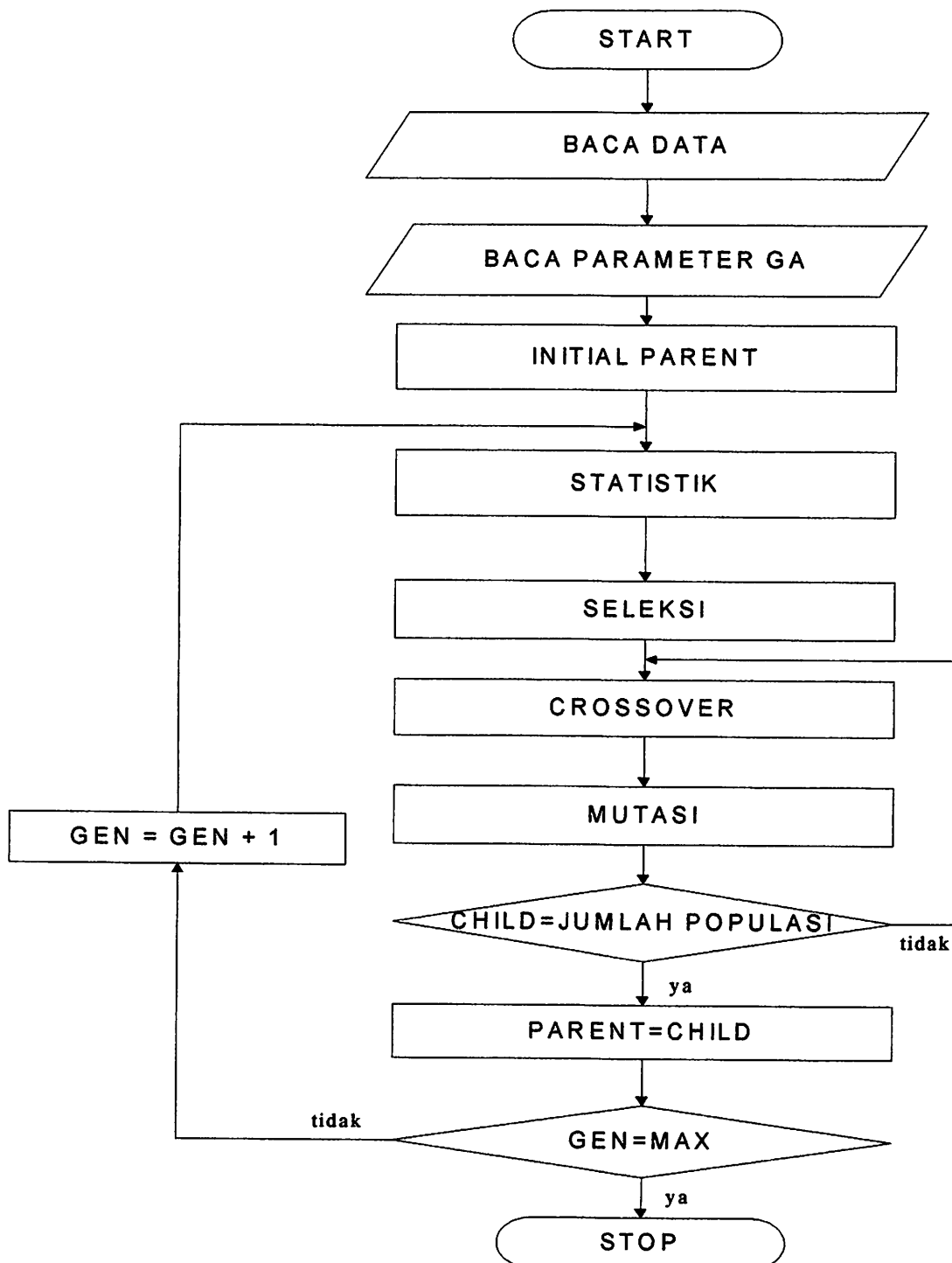
8. Apakah keturunan baru sudah layak. Jika tidak kembalilah ke langkah 6, jika keturunan sudah layak melangkahlah ke tahap selanjutnya.
9. Hitunglah nilai fitness dari keturunan.
10. Hasilkan sebuah individu didalam populasi baru untuk mengganti sebuah individu dimana nilai fitnessnya lebih besar dari rata-rata nilai fitness yang telah dihitung.
11. Ulangi ke langkah (6) hingga (10) untuk mencari semua individu yang tersisa.
12. Ulangi ke langkah (4) hingga (11) apabila kriteria penghentian tidak dipenuhi. Jika sudah terpenuhi, melangkahlah ke langkah (13).
13. STOP (Solusi terbaik pada populasi baru merupakan solusi yang optimal).

### **3.3.2. Algoritma program *Simulated Annealing*.**

1. Masukkan dan baca data sistem dan jaringan berupa data beban P, Q. Data saluran R,X.  $P_{\text{dasar}}$ ,  $V_{\text{dasar}}$ . Biaya dan kapasitas kapasitor, tegangan operasional minimum dan maksimum yang diperkenankan dan parameter-parameter Simulated Annealing. Jumlah iterasi, konstanta pendinginan, konstanta  $X_0$ , konstanta flip.
2. Hitunglah tegangan sistem, *losses* daya selama level beban dan total kehilangan energi pada masing-masing bus.
3. Hasilkan sebuah solusi awal yang layak dan hitunglah *cost function* yang berkaitan.
4. Carilah solusi-solusi yang berdekatan dengan melakukan gerakan yang berbeda berdasarkan persentase yang ditetapkan sebelumnya.

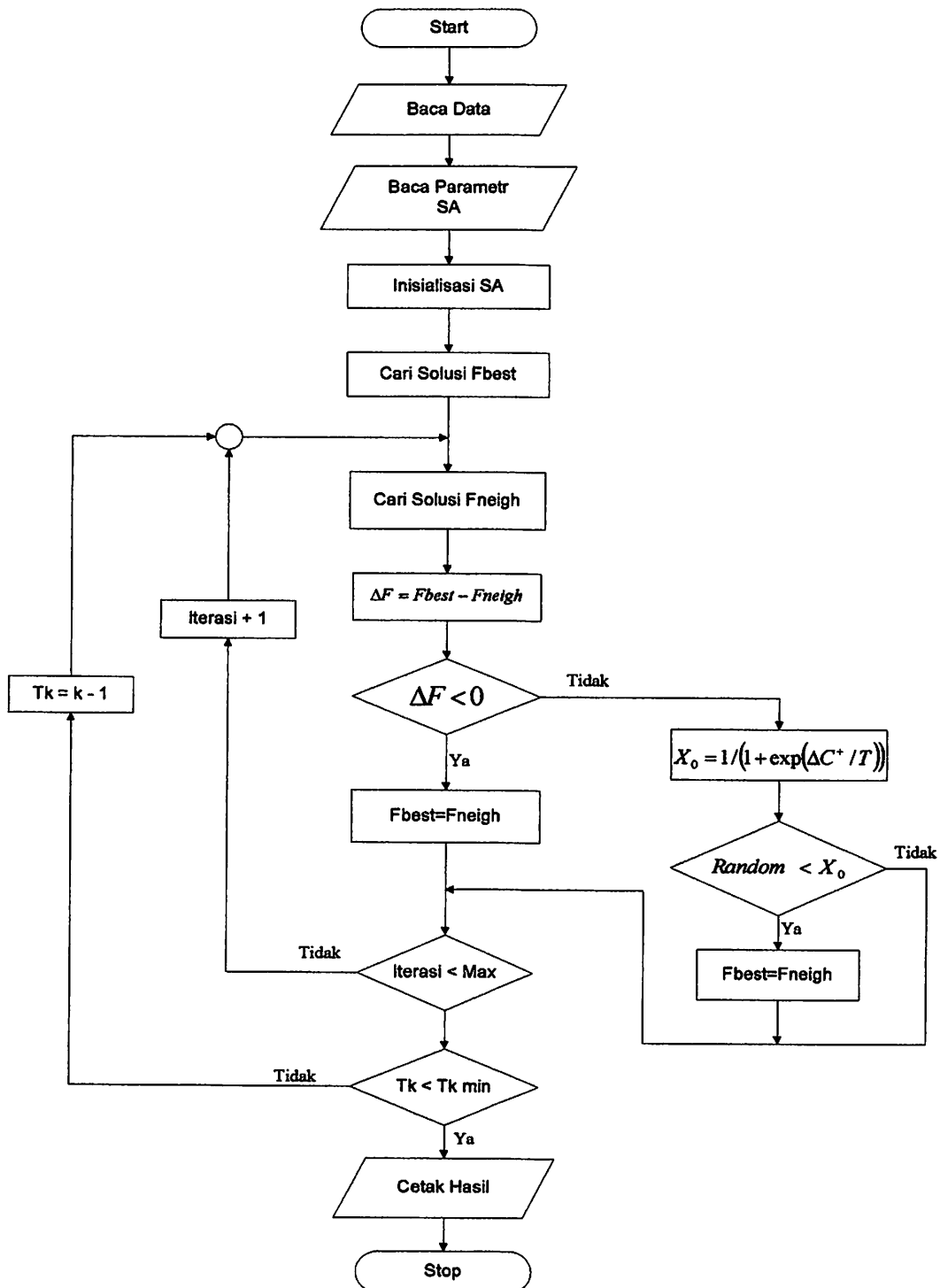
5. Periksalah kelayakan dari konfigurasi baru, jika tidak layak kembali ke langkah 4, jika sebaliknya lanjutkan ke langkah 6.
6. Rancanglah sebuah jadwal pendinginan. Pada masing-masing temperatur, lakukan sejumlah gerakan. Untuk beberapa gerakan lakukan langkah (7) dan (8). Jika sebaliknya lanjutkan ke langkah (9).
7. Hasilkan sebuah konfigurasi baru yang layak.
8. Perbaharui konfigurasi sistem.
9. Periksalah kriteria penghentian. Jika tidak terpenuhi, kembali ke langkah (6). Sebaliknya jika sudah terpenuhi lakukan ke langkah selanjutnya.
10. Cetak hasil dari konfigurasi yang optimal.

### 3.3.3. Flowchart *Genetic Algorithm*



**GAMBAR 3-10** *FLOWCHART ALGORITMA SOLUSI GENETIC ALGORITHM*

### 3.3.4. Flowchart *Simulated Annealing*



**GAMBAR 3-11** *FLOWCHART* ALGORITMA SOLUSI *SIMULATED ANNEALING*

### 3.4. Data Saluran

Jaringan distribusi Penyulang Syiar menggunakan kabel saluran udara dengan spesifikasi yang seperti pada tabel.3-1.

**TABEL 3-2**  
**SPESIFIKASI SALURAN**

Jenis Konduktor	Penampang nominal (mm <sup>2</sup> )	Impedansi saluran (Ω /km)
AAAC	150	0.2162 + j 0.3305
AAAC	120	0.2688 + j 0.3376

Dari data spesifikasi saluran pada penyulang Syiar diketahui bahwa jenis konduktor yang digunakan adalah AAAC (*all-aluminium-alloyconductors*) dengan penampang nominal 150 mm<sup>2</sup> dengan impedansi saluran 0.2162 + j0.3305 Ω/km, untuk saluran utama dan 120 mm<sup>2</sup> dengan impedansi saluran 0.2688 + j0.3376 Ω/km untuk sub saluran.

Untuk perhitungan Impedansi saluran diberikan contoh pada bus dari 1 ke 2 dan dari 8 ke 9 sebagai berikut:

Jarak antara bus dari 1 ke 2 dengan panjang = 2120 m = 2.12 km

$$R = 2.12 \text{ km} \times 0.2162 \text{ } \Omega/\text{km} = 0.45834 \text{ } \Omega$$

$$X = 2.12 \text{ km} \times 0.3305 \text{ } \Omega/\text{km} = 0.70066 \text{ } \Omega$$

Jarak antara bus dari 8 ke 9 dengan panjang = 780 m = 0,78 km

$$R = 0.78 \text{ km} \times 0.2688 \text{ } \Omega/\text{km} = 0.20966 \text{ } \Omega$$

$$X = 0.78 \text{ km} \times 0.3376 \text{ } \Omega/\text{km} = 0.26333 \text{ } \Omega$$

Dengan cara yang sama seperti pada contoh diatas untuk saluran yang lainnya maka diperoleh hasil seperti pada tabel 3-2.

**TABEL 3-3**  
**HASIL PERHITUNGAN SALURAN PENYULANG SYIAR**

No Saluran	Saluran		Panjang (m)	R ( $\Omega$ )	X ( $\Omega$ )
	Dari	Ke			
1	1	2	2120	0.45834	0.70066
2	2	3	1150	0.24863	0.38008
3	3	4	750	0.16215	0.24788
4	4	5	680	0.14702	0.22474
5	5	6	430	0.09297	0.14212
6	5	7	570	0.12323	0.18839
7	3	8	320	0.06918	0.10576
8	8	9	780	0.20966	0.26333
9	8	10	1890	0.40862	0.62465
10	10	11	2350	0.50807	0.77668
11	11	12	1260	0.27241	0.41643
12	12	13	1740	0.37619	0.57507
13	13	14	930	0.20107	0.30737
14	14	15	2490	0.66931	0.84062
15	14	16	1960	0.42375	0.64778
16	16	17	1480	0.31998	0.48914
17	17	18	2490	0.53834	0.82295
18	18	19	430	0.09297	0.14212
19	19	20	1600	0.43008	0.54016
20	19	21	510	0.11026	0.16856
21	21	22	920	0.19890	0.30406
22	21	23	1320	0.28538	0.43626
23	23	24	2460	0.66125	0.83050
24	23	25	1270	0.34138	0.42875
25	18	26	1120	0.24214	0.37016
26	26	27	1670	0.36105	0.55194
27	27	28	2590	0.69619	0.87438
28	27	29	2850	0.61617	0.94193
29	29	30	100	0.02162	0.03305
30	30	31	120	0.02594	0.03966
31	31	32	250	0.05405	0.08263
32	32	33	1450	0.31349	0.47923
34	33	34	2610	0.56428	0.86261
35	32	35	720	0.19354	0.24307
36	30	36	233	0.05037	0.07701
37	36	37	820	0.22042	0.27683
38	36	38	130	0.02811	0.04297
39	38	39	590	0.12756	0.19500
40	39	40	1090	0.23566	0.36025
41	40	41	1100	0.29568	0.37136
42	41	42	1070	0.28762	0.36123
43	40	43	950	0.20539	0.31398
44	43	44	790	0.17080	0.26110
45	44	45	750	0.16215	0.24788
46	44	46	1460	0.39245	0.49290
47	43	47	2210	0.47780	0.73041
48	47	48	2250	0.48645	0.74363
49	48	49	2500	0.54050	0.82625



### 3.5. Pembebanan Sistem 20 kV Penyulang Syiar

Pembebanan diperoleh dengan mengambil data dari masing-masing trafo distribusi, dimana besarnya beban pada masing-masing fasa diasumsikan seimbang. Jika besarnya pembebanan adalah nol, maka pada bus tidak terdapat trafo distribusi tetapi hanya merupakan simpul. Pada tahap ini rugi-rugi yang terjadi pada trafo distribusi diabaikan, dengan mengasumsi faktor daya 0,85. dibawah ini diberikan contoh perhitungan pembebanan pada bus 2:

Kapasitas Trafo	= 150 kVA
Beban Gardu	= 57,58%
Factor Daya	= $\text{Cos } \theta = 0,85$ $\text{Sin } \theta = 0,5268$
Pembebanan	= $150 \times 57,58\% = 86,37 \text{ kVA}$
Beban Aktif	= $86,37 \times \text{Cos } \theta = 86,37 \times 0,85 = 73,4145 \text{ kW}$
Beban Reaktif	= $86,37 \times \text{Sin } \theta = 86,37 \times 0,5268 = 45,4982 \text{ kVAR}$

Untuk bus 3 sampai dengan 49 terlihat seperti pada tabel 3-3 dibawah ini

**TABEL 3-4**  
**HASIL PERHITUNGAN PEMBEBANAN SISTEM 20KV PENYULANG**  
**SYIAR**

No Node	Kode Trafo	Lokasi	Pembebanan			Type Bus
			S (kVA)	P (kW)	Q (kVAR)	
1	-	-	0	0	0	Slack
2	T.71	Mojokopek	86.37	73.4145	45.4982	Load
3	-	-	0	0	0	Load
4	T.69	Mojoparon	85.68	72.8280	45.1347	Load
5	-	-	0	0	0	Load
6	T.90	PT. Metabisulfid	555	471.7500	292.3644	Load
7	T.203	Ds. Blawi	121.01	102.8585	63.7460	Load
8	-	-	0	0	0	Load
9	T.26	Ds. Pekoren	106	90.1000	55.8390	Load
10	T.130	Ds. Pekoren	117.31	99.7135	61.7969	Load
11	T.202	Ds. Rembang	100.92	85.7820	53.1629	Load
12	T.70	Ds. Ketapaan	72.28	61.4380	38.0759	Load
13	T.199	Ds. Kdg Banteng	95.61	81.2685	50.3657	Load
14	-	-	0	0	0	Load
15	T.200	Ds. Kdg Banteng	29.79	25.3215	15.6929	Load
16	T.251	Ds. Cengkelek	25.63	21.7855	13.5014	Load
17	T.252	Ds. Candiroboh	31.87	27.0895	16.7886	Load
18	-	-	0	0	0	Load
19	-	-	0	0	0	Load
20	T.144	Pejangkungan	66.05	56.1425	34.7940	Load
21	-	-	0	0	0	Load
22	T.125	Sbr. Glagah	43.18	36.7030	22.7465	Load
23	-	-	0	0.0000	0.0000	Load
24	T.182	Roh Wangkal	26.56	22.5760	13.9913	Load
25	T.246	Ds. Oyoran	30.71	26.1035	16.1775	Load
26	T.41	Ds. Rembang	75.05	63.7925	39.5350	Load
27	-	-	0	0.0000	0.0000	Load
28	T.186	Oro-Oro Bulu	80.13	68.1105	42.2111	Load
29	T.134	Kalisat	69.05	58.6925	36.3743	Load
30	-	-	0	0.0000	0.0000	Load
31	T.135	Ds. Kalisat	39.26	33.3710	20.6815	Load
32	-	-	0	0.0000	0.0000	Load
33	T.301	Pohkpan/Sisipan	57.27	48.6795	30.1688	Load
34	T.227	PT. Charun Pohkpan	345	293.2500	181.7400	Load
35	T.25	Ds. Kdg Lingit	52.65	44.7525	27.7351	Load
36	-	-	0	0.0000	0.0000	Load
37	T.245	Pejaten	71.59	60.8515	37.7124	Load
38	T.154	Tampung Barat	74.36	63.2060	39.1716	Load
39	T.94	Ds. Tampung	149.41	126.9985	78.7066	Load

40	-		0	0.0000	0.0000	Load
41	T.95	Ds. Kanigoro	65.59	55.7515	34.5517	Load
42	T.145	DS. Krenggih	77.13	65.5605	40.6307	Load
43	-		0	0.0000	0.0000	Load
44	-		0	0.0000	0.0000	Load
45	T.239	Dinas Pertanian	450	382.5000	237.0522	Load
46	T.185	Ds. Slambrit	54.5	46.3250	28.7097	Load
47	T.117	Ds. Syiar	48.73	41.4205	25.6701	Load
48	T.86	Genenganwaru	96.76	82.2460	50.9715	Load
49	T.87	Ds. Wangkal	31.64	26.8940	16.6674	Load

### 3.6. Data Kapasitas kapasitor dan Harga (*Cost/ kVAR*).

Pada analisa menentukan nilai biaya, nilai kapasitas kapasitor dan harga sudah ditentukan dalam \$ /kVAR. Data yang tersedia disesuaikan dengan data harga yang ada di journal<sup>[1]</sup>, yaitu dengan ukuran (*size*) 300kVAR/Unit dan harga kapasitor yang tersedia 900\$/Bank dengan harga pemasangan 1000\$/Lokasi.

## **BAB IV**

### **ANALISA DAN HASIL**

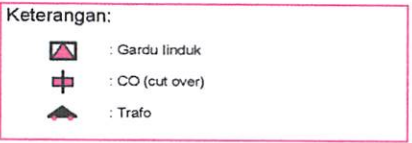
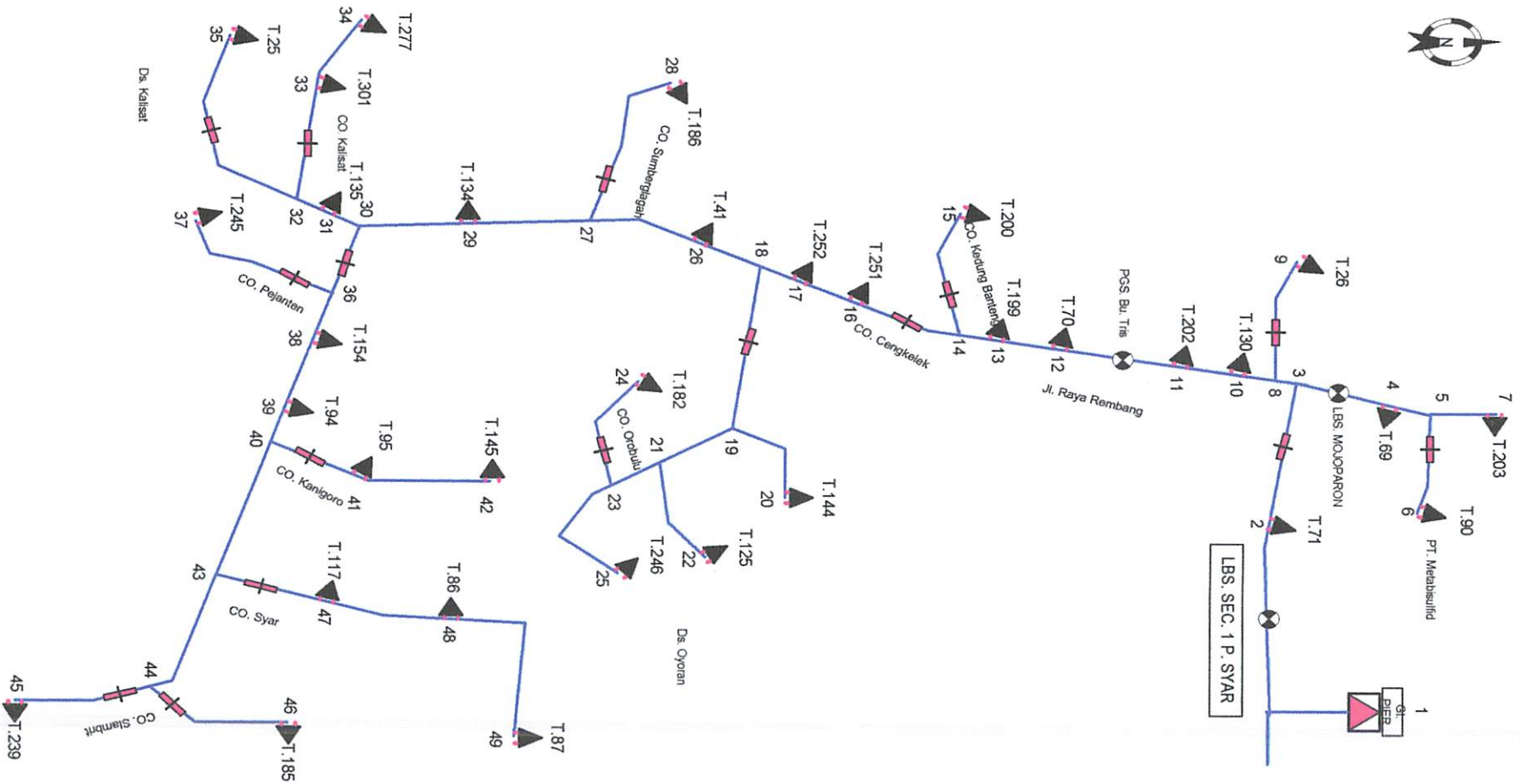
#### **4.1. Sistem Distribusi Tenaga listrik 20kV GI PIER Penyulang Syiar**

Dalam perhitungan aliran daya Data yang diambil dari GI. PIER Pasuruan yang melayani 6 (enam) buah penyulang.

Namun pada skripsi ini hanya menganalisa satu penyulang saja, yaitu pada penyulang Syiar. Sistem Distribusi Radial GI. PIER yang memakai tegangan distribusi 20 kV. Untuk menyelesaikan perhitungan Aliran Daya terlebih dahulu ditetapkan *single line diagram* sesuai dengan gambar (4.1) yang akan dianalisis, dimana dasar yang digunakan:

- Tegangan Dasar : 20kV
- Daya Dasar : 50MVA

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



Gambar 4-1  
Single Line Diagram Penyulang Syiar Sebelum  
Penempatan Kapasitor

## 4.2. Analisa Perhitungan

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

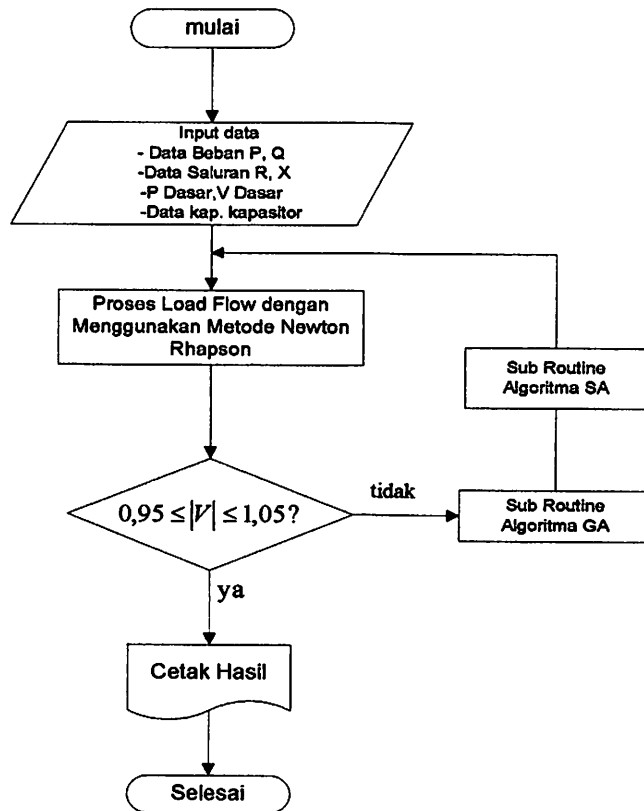
Untuk menentukan lokasi, kapasitas, dan setting kapasitor terlebih dahulu menentukan rugi daya saluran terbesar yang digunakan sebagai input untuk menjalankan metode *Hybrid GA-SA* agar kapasitor dapat diletakan pada lokasi yang memberikan profil tegangan yang paling optimum. Sedangkan untuk perhitungan besar nilai penghematan ditentukan 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. Harga yang dipilih adalah 20kV dan 50MVA. 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 0.85.

Perhitungan diawali dengan menampilkan *single line diagram* dari penyulang yang mewakili keadaan sistem yang sesungguhnya. Dari gambar 4-1 diperlihatkan *single line diagram* dari penyulang *Syiar*. Pada penyulang ini jumlah bus dan jumlah saluran masing-masing adalah:

- Slack bus = 1
- Load bus = 49
- Jumlah saluran = 48

## Flow Chart Penyelesaian Masalah



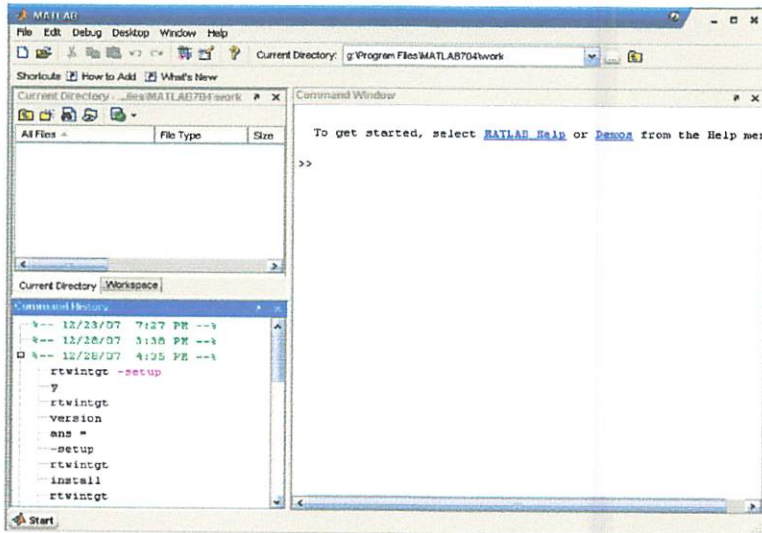
**GAMBAR 4-2** *FLOWCHART* PENYELESAIAN MASALAH



### 4.3. Prodesur Pelaksanaan Program Perhitungan.

Produser menjalankan program perhitungan dengan menggunakan bahasa pemrograman MATLAB 7.0.4 dapat dilakukan sebagai berikut :

#### 1. Tampilan Utama Program MATLAB 7.0.4

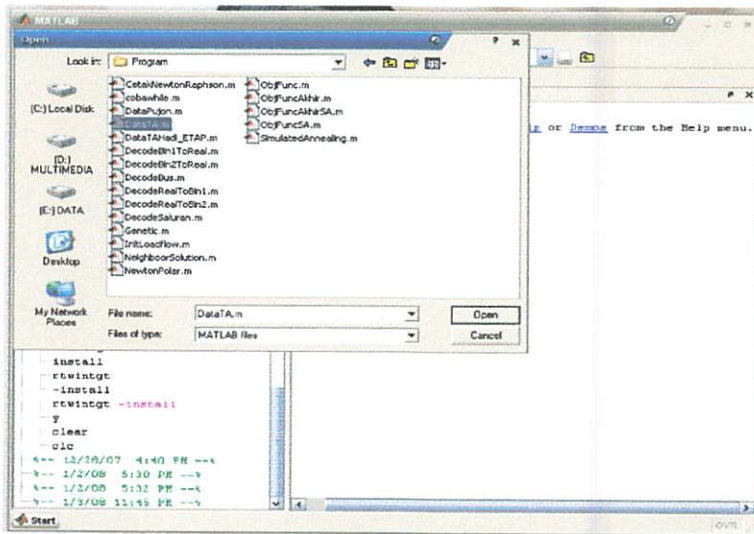


**GAMBAR 4-1 TAMPILAN UTAMA PROGRAM MATLAB 7.0.4**

Pada gambar 4-2 merupakan tampilan utama program simulasi matlab 7.0.4 sebelum menjalankan program yang mana terlebih dahulu telah dimasukkan data-data yang dibutuhkan, yaitu data pembebanan, data saluran, data *objective function* serta parameter-parameter dari metode *Hybrid GA-SA*.



2. Tekan *open file* untuk membuka data yang sudah tersimpan, pilih data kemudian klik *open* untuk membuka data inputan



**GAMBAR 4-3 TAMPILAN INPUTAN DATA**

Pada gambar 4-3 merupakan tampilan dari inputan data yang tersimpan dan untuk membuka data yang tersimpan klik *open file* yang ada pada pojok kiri atas, kemudian akan muncul pilihan *file* seperti yang terlihat pada gambar tampilan tandai dan kemudian klik *open* untuk membuka inputan data yang dibutuhkan. Setelah itu lanjut ke langkah 3.

3. Tampilan inputan data pembebanan, gerakan *scroll* ke bawah untuk melihat tampilan inputan data saluran

```

1
2 cto
3 Vbase=20.000;
4 Xbase=1000;
5 Rbase=10.000;
6 Xkonst=1000;
7 Rkonst=1000;
8 Zbase=(Vbase*Xkonst)^2/(Pbase*Xkonst);
9 Xbase=(Vbase*Xkonst)/Zbase;
10 Rbase=(Vbase*Rkonst)/Zbase;
11
12
13 -----
14 bus=1:1.00000 0.00000 0.000 0.000 0.000 0.000 0.000 0.0000 3;
15 1.00000 0.00000 0.000 0.000 73.415 45.498 0.0000 3;
16 1.00000 0.00000 0.000 0.000 0.000 0.000 0.0000 3;
17 1.00000 0.00000 0.000 0.000 72.828 45.135 0.0000 3;
18 1.00000 0.00000 0.000 0.000 491.750 292.344 0.0000 3;
19 1.00000 0.00000 0.000 0.000 102.859 63.746 0.0000 3;
20 1.00000 0.00000 0.000 0.000 0.000 0.000 0.0000 3;
21 1.00000 0.00000 0.000 0.000 90.100 55.839 0.0000 3;
22 1.00000 0.00000 0.000 0.000 99.714 61.797 0.0000 3;
23 1.00000 0.00000 0.000 0.000 85.782 53.163 0.0000 3;
24 1.00000 0.00000 0.000 0.000 61.438 39.076 0.0000 3;
25 1.00000 0.00000 0.000 0.000 81.259 50.346 0.0000 3;
26 1.00000 0.00000 0.000 0.000 0.000 0.000 0.0000 3;
27 1.00000 0.00000 0.000 0.000 25.322 15.493 0.0000 3;
28 1.00000 0.00000 0.000 0.000 21.786 13.501 0.0000 3;
29 1.00000 0.00000 0.000 0.000 27.090 16.789 0.0000 3;
30 1.00000 0.00000 0.000 0.000 0.000 0.000 0.0000 3;
31 1.00000 0.00000 0.000 0.000 0.000 0.000 0.0000 3;
32 1.00000 0.00000 0.000 0.000 56.143 34.794 0.0000 3;
33 1.00000 0.00000 0.000 0.000 0.000 0.000 0.0000 3;
34 1.00000 0.00000 0.000 0.000 36.703 22.746 0.0000 3;
35 1.00000 0.00000 0.000 0.000 0.000 0.000 0.0000 3;
36 1.00000 0.00000 0.000 0.000 23.576 13.991 0.0000 3;
37 1.00000 0.00000 0.000 0.000 26.104 16.177 0.0000 3;
38

```

**GAMBAR 4-4 TAMPILAN INPUTAN DATA PEMBEBANAN**

Pada gambar 4-4 terlihat tampilan inputan data pembebanan yang mana pembebanan diperoleh dengan mengambil data dari masing – masing trafo distribusi, dimana besarnya beban pada masing – masing fasa diasumsikan seimbang. Dengan mengasumsi faktor daya  $0,85$ . berikut adalah contoh perhitungan pembebanan pada bus 2

$$\text{Kapasitas Trafo} = 150 \text{ kVA}$$

$$\text{Beban Gardu} = 57,58\%$$

$$\text{Faktor Daya} = \cos \theta = 0,85$$

$$\sin \theta = 0,5268$$

$$\text{Pembebanan} = 150 \times 57,58\% = 86,37 \text{ kVA}$$

$$\text{Beban Aktif} = 86,37 \times \cos \theta = 86,37 \times 0,85 = 73,4145 \text{ kW}$$

$$\text{Beban Reaktif} = 86,37 \times \sin \theta = 86,37 \times 0,5268 = 45,4982 \text{ kVAR}$$

Untuk bus 3 sampai dengan 49 dapat dilihat pada table 4-1

**TABEL 4-1**  
**HASIL PERHITUNGAN PEMBEBANAN SISTEM 20KV PENYULANG**  
**SYIAR**

No Node	Kode Trafo	Lokasi	Pembelianan			Type Bus
			S (kVA)	P (kW)	Q (kVAR)	
1	-	-	0	0	0	Slack
2	T.71	Mojokopek	86.37	73.4145	45.4982	Load
3	-	-	0	0	0	Load
4	T.69	Mojoparon	85.68	72.8280	45.1347	Load
5	-	-	0	0	0	Load
6	T.90	PT. Metabisulfid	555	471.7500	292.3644	Load
7	T.203	Ds. Blawi	121.01	102.8585	63.7460	Load
8	-	-	0	0	0	Load
9	T.26	Ds. Pekoren	106	90.1000	55.8390	Load
10	T.130	Ds. Pekoren	117.31	99.7135	61.7969	Load
11	T.202	Ds. Rembang	100.92	85.7820	53.1629	Load
12	T.70	Ds. Ketapaan	72.28	61.4380	38.0759	Load
13	T.199	Ds. Kdg Banteng	95.61	81.2685	50.3657	Load
14	-	-	0	0	0	Load
15	T.200	Ds. Kdg Banteng	29.79	25.3215	15.6929	Load
16	T.251	Ds. Cengkelek	25.63	21.7855	13.5014	Load
17	T.252	Ds. Candiroboh	31.87	27.0895	16.7886	Load
18	-	-	0	0	0	Load
19	-	-	0	0	0	Load
20	T.144	Pejangkungan	66.05	56.1425	34.7940	Load
21	-	-	0	0	0	Load
22	T.125	Sbr. Glagah	43.18	36.7030	22.7465	Load
23	-	-	0	0.0000	0.0000	Load
24	T.182	Roh Wangkal	26.56	22.5760	13.9913	Load
25	T.246	Ds. Oyoran	30.71	26.1035	16.1775	Load
26	T.41	Ds. Rembang	75.05	63.7925	39.5350	Load
27	-	-	0	0.0000	0.0000	Load
28	T.186	Oro-Oro Bulu	80.13	68.1105	42.2111	Load
29	T.134	Kalisat	69.05	58.6925	36.3743	Load
30	-	-	0	0.0000	0.0000	Load
31	T.135	Ds. Kalisat	39.26	33.3710	20.6815	Load
32	-	-	0	0.0000	0.0000	Load
33	T.301	Pohkpan/Sisipan	57.27	48.6795	30.1688	Load
34	T.227	PT. Charun Pohkpan	345	293.2500	181.7400	Load
35	T.25	Ds. Kdg Lingit	52.65	44.7525	27.7351	Load
36	-	-	0	0.0000	0.0000	Load
37	T.245	Pejaten	71.59	60.8515	37.7124	Load
38	T.154	Tampung Barat	74.36	63.2060	39.1716	Load
39	T.94	Ds. Tampung	149.41	126.9985	78.7066	Load



40	-		0	0.0000	0.0000	Load
41	T.95	Ds. Kanigoro	65.59	55.7515	34.5517	Load
42	T.145	DS. Krenggih	77.13	65.5605	40.6307	Load
43	-		0	0.0000	0.0000	Load
44	-		0	0.0000	0.0000	Load
45	T.239	Dinas Pertanian	450	382.5000	237.0522	Load
46	T.185	Ds. Slambricit	54.5	46.3250	28.7097	Load
47	T.117	Ds. Syiar	48.73	41.4205	25.6701	Load
48	T.86	Genenganwaru	96.76	82.2460	50.9715	Load
49	T.87	Ds. Wangkal	31.64	26.8940	16.6674	Load

#### 4. Gerakan scroll ke bawah untuk melihat tampilan inputan data saluran.

Bus	Le	EP	EA	Z	X	LP	EP	ADAPR	SIKAPR
64									
65									
66									
67									
68	2	1	2	0.45834	0.70266	0.00000	0.00000	0.00000	10000.000
69	2	3	3	0.24683	0.38008	0.00000	0.00000	0.00000	10000.000
70	3	4	4	0.16215	0.24788	0.00000	0.00000	0.00000	10000.000
71	4	5	5	0.14701	0.22474	0.00000	0.00000	0.00000	10000.000
72	5	6	6	0.09297	0.14212	0.00000	0.00000	0.00000	10000.000
73	5	7	7	0.12313	0.18819	0.00000	0.00000	0.00000	10000.000
74	3	8	8	0.05818	0.10576	0.00000	0.00000	0.00000	10000.000
75	8	9	9	0.20966	0.36333	0.00000	0.00000	0.00000	10000.000
76	8	10	10	0.40862	0.62465	0.00000	0.00000	0.00000	10000.000
77	10	11	11	0.50807	0.77468	0.00000	0.00000	0.00000	10000.000
78	11	12	12	0.27241	0.41443	0.00000	0.00000	0.00000	10000.000
79	12	13	13	0.37619	0.57507	0.00000	0.00000	0.00000	10000.000
80	13	14	14	0.20107	0.30737	0.00000	0.00000	0.00000	10000.000
81	14	15	15	0.69911	0.84022	0.00000	0.00000	0.00000	10000.000
82	14	16	16	0.42276	0.44778	0.00000	0.00000	0.00000	10000.000
83	16	17	17	0.31998	0.40914	0.00000	0.00000	0.00000	10000.000
84	17	18	18	0.53834	0.62295	0.00000	0.00000	0.00000	10000.000
85	18	19	19	0.09297	0.14212	0.00000	0.00000	0.00000	10000.000
86	19	20	20	0.43008	0.64016	0.00000	0.00000	0.00000	10000.000
87	19	21	21	0.11026	0.16856	0.00000	0.00000	0.00000	10000.000
88	21	22	22	0.18080	0.30405	0.00000	0.00000	0.00000	10000.000
89	21	23	23	0.28838	0.43626	0.00000	0.00000	0.00000	10000.000
90	23	24	24	0.64125	0.93250	0.00000	0.00000	0.00000	10000.000
91	23	25	25	0.34130	0.42975	0.00000	0.00000	0.00000	10000.000
92	18	26	26	0.24214	0.37016	0.00000	0.00000	0.00000	10000.000
93	26	27	27	0.36105	0.55194	0.00000	0.00000	0.00000	10000.000
94	27	28	28	0.59419	0.87438	0.00000	0.00000	0.00000	10000.000
95	27	29	29	0.41617	0.64193	0.00000	0.00000	0.00000	10000.000
96	29	30	30	0.02162	0.03305	0.00000	0.00000	0.00000	10000.000
97	30	31	31	0.02394	0.03966	0.00000	0.00000	0.00000	10000.000
98	31	32	32	0.05405	0.08263	0.00000	0.00000	0.00000	10000.000
99	32	33	33	0.11318	0.17923	0.00000	0.00000	0.00000	10000.000
100	33	34	34	0.56428	0.86261	0.00000	0.00000	0.00000	10000.000

**GAMBAR 4-5 TAMPILAN INPUTAN DATA SALURAN**

Pada gambar 4-5 terlihat tampilan inputan data saluran yang mana jaringan distribusi Penyulang Syiar menggunakan kabel saluran udara dengan jenis konduktor yang digunakan adalah AAAC (*all-aluminium-alloyconductors*) dengan penampang nominal  $150 \text{ mm}^2$  dengan impedansi saluran  $0.2162 + j0.3305 \Omega/\text{km}$ , untuk saluran utama dan  $120 \text{ mm}^2$  dengan impedansi saluran  $0.2688 + j0.3376 \Omega/\text{km}$  untuk sub saluran.

Untuk perhitungan Impedansi saluran diberikan contoh pada bus dari 1 ke 2 dan dari 8 ke 9 sebagai berikut:

Jarak antara bus dari 1 ke 2 dengan panjang = 2120 m = 2.12 km

$$R = 2.12 \text{ km} \times 0.2162 \text{ } \Omega/\text{km} = 0.45834 \text{ } \Omega$$

$$X = 2.12 \text{ km} \times 0.3305 \text{ } \Omega/\text{km} = 0.70066 \text{ } \Omega$$

Jarak antara bus dari 8 ke 9 dengan panjang = 780 m = 0,78 km

$$R = 0.78 \text{ km} \times 0.2688 \text{ } \Omega/\text{km} = 0.20966 \text{ } \Omega$$

$$X = 0.78 \text{ km} \times 0.3376 \text{ } \Omega/\text{km} = 0.26333 \text{ } \Omega$$

Dengan cara yang sama seperti pada contoh diatas untuk saluran yang lainnya maka diperoleh hasil seperti pada tabel 4-2.

**TABEL 4-2**

**HASIL PERHITUNGAN SALURAN PENYULANG SYIAR**

No Saluran	Saluran		Panjang (m)	R ( $\Omega$ )	X ( $\Omega$ )
	Dari	Ke			
1	1	2	2120	0.45834	0.70066
2	2	3	1150	0.24863	0.38008
3	3	4	750	0.16215	0.24788
4	4	5	680	0.14702	0.22474
5	5	6	430	0.09297	0.14212
6	5	7	570	0.12323	0.18839
7	3	8	320	0.06918	0.10576
8	8	9	780	0.20966	0.26333
9	8	10	1890	0.40862	0.62465
10	10	11	2350	0.50807	0.77668
11	11	12	1260	0.27241	0.41643
12	12	13	1740	0.37619	0.57507
13	13	14	930	0.20107	0.30737
14	14	15	2490	0.66931	0.84062
15	14	16	1960	0.42375	0.64778
16	16	17	1480	0.31998	0.48914
17	17	18	2490	0.53834	0.82295
18	18	19	430	0.09297	0.14212
19	19	20	1600	0.43008	0.54016
20	19	21	510	0.11026	0.16856
21	21	22	920	0.19890	0.30406
22	21	23	1320	0.28538	0.43626
23	23	24	2460	0.66125	0.83050
24	23	25	1270	0.34138	0.42875
25	18	26	1120	0.24214	0.37016
26	26	27	1670	0.36105	0.55194
27	27	28	2590	0.69619	0.87438
28	27	29	2850	0.61617	0.94193
29	29	30	100	0.02162	0.03305

30	30	31	120	0.02594	0.03966
31	31	32	250	0.05405	0.08263
32	32	33	1450	0.31349	0.47923
34	33	34	2610	0.56428	0.86261
35	32	35	720	0.19354	0.24307
36	30	36	233	0.05037	0.07701
37	36	37	820	0.22042	0.27683
38	36	38	130	0.02811	0.04297
39	38	39	590	0.12756	0.19500
40	39	40	1090	0.23566	0.36025
41	40	41	1100	0.29568	0.37136
42	41	42	1070	0.28762	0.36123
43	40	43	950	0.20539	0.31398
44	43	44	790	0.17080	0.26110
45	44	45	750	0.16215	0.24788
46	44	46	1460	0.39245	0.49290
47	43	47	2210	0.47780	0.73041
48	47	48	2250	0.48645	0.74363
49	48	49	2500	0.54050	0.82625

## 5. Gerakan *scroll* ke bawah untuk melihat tampilan data *objective function*.

```

116 %-----
117 - [Nbuss, Z, Lc, Tr, Tc, CapSel]=beodeSaluran(Nbus, Sel);
118 - global Gen
119 - awal=cputime;
120 - global Nbus I Lo Tr Tc Cap Beban Phase Zbase TypSel CapSel;
121 - [V, Iw, AItc, Sg, SL, Smd, Suml, Rr, RgltB, Iq]=NewtonPolar(Nbus, Z, Lc, Tr, Tc, Cap, Beban, Phase, Zbase, TypSel);
122 - akhir=cputime;
123 - selang=akhir-awal;
124 - [Nbus]=CetakNewtonRaphson(Nbus, Iw, selang, V, Sg, SL, Z, AItc, RgltB, Smd, Suml, Rr);
125 - disp(' ');
126 - kLow=0.8;
127 - kMed=0.0;
128 - kPes=1.0;
129 - global kLow kMed kPes;
130 - k=0.04;
131 - global kq;
132 - Tlow=1000;
133 - Tmed=6760;
134 - Tpes=1000;
135 - global kcap Tlow Tmed Tpes;
136 - [Fitness]=initLoadflow;
137 %-----
138 %Data Objective Function
139 %Obj=1;
140 - lenda0=1000000000;
141 - lenda=1000;
142 - lendaCap=1000000;
143 - @lenda lendaV lenda0 lendaCap;
144 - CapBase=700;
145 - Nbank=4;
146 - costFixed=3;
147 - costSwitch=3;
148 - InstCost=1000;
149 - global Nbank CapBank costFixed costSwitch InstCost;
150 - Vmin=0.95;
151 - Vmax=1.05;
152 - global Vmin Vmax;

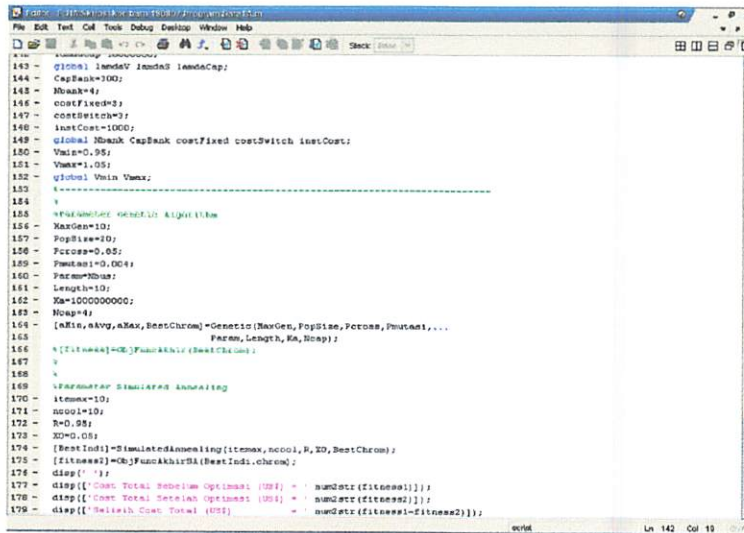
```

**GAMBAR 4-6 TAMPILAN DATA *OBJECTIVE FUNCTION***

Pada gambar 4-6 adalah tampilan data *objective function* atau fungsi tujuan. Dimana fungsi tujuan terdiri dari 2 hal. Pertama, biaya penempatan kapasitor dan kedua adalah biaya total kehilangan energi.



6. Gerakan *scroll* ke bawah untuk melihat tampilan parameter *GA* dan *SA*.



```

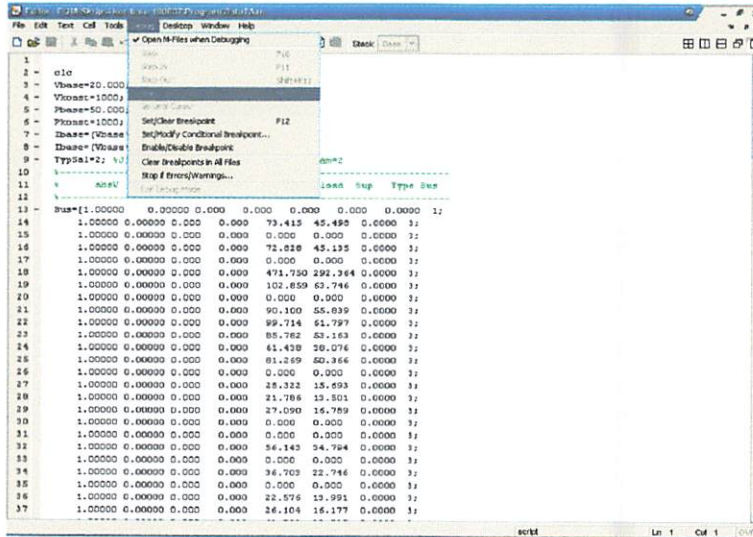
143 - @linda() lendaV lendaS lendaCap;
144 - CapBank=100;
145 - Nbank=4;
146 - costFixed=3;
147 - costSwitch=3;
148 - InstCost=1000;
149 - global Nbank CapBank costFixed costSwitch InstCost;
150 - Vmax=0.95;
151 - Vmax=1.05;
152 - global Vmin Vmax;
153 - %-----
154 - %
155 - %Parameter genetika algoritma
156 - MaxGen=10;
157 - PopSize=10;
158 - Pcross=0.05;
159 - Pmutasi=0.004;
160 - ParamNbas;
161 - Length=10;
162 - Max100000000;
163 - Npop=4;
164 - [nRn, nAvg, nMax, BestChrom]=Genetic(MaxGen, PopSize, Pcross, Pmutasi, ...
165 - Param, Length, Nn, Npop);
166 - % [Fitness]=@FuncAkhir(BestChrom);
167 - %
168 - %
169 - %Parameter Simulasi Annealing
170 - Iterasi=10;
171 - ncool=10;
172 - X=0.05;
173 - XD=0.05;
174 - [BestInd1]=SimulasiAnnealing(Iterasi, ncool, X, XD, BestChrom);
175 - [Fitness2]=@FuncAkhirSA(BestInd1.chrom);
176 - disp(' ');
177 - disp(['Cost Total Sebelum Optimas (10) = ' num2str(Fitness1)]);
178 - disp(['Cost Total Setelah Optimas (10) = ' num2str(Fitness2)]);
179 - disp(['Selisih Cost Total (10) = ' num2str(Fitness1-Fitness2)]);

```

**GAMBAR 4-7 TAMPILAN PARAMETER *GA* DAN *SA***

Pada gambar 4-7 adalah tampilan dari parameter metode yang digunakan yaitu *GA* dan *SA*. Dimana parameter – parameter rancangan untuk *GA* yang diterapkan pada penempatan kapasitor parameternya adalah *Pop Size*, *Max Gen*, *Mutasi*, dan *Crossover*. Sedangkan parameter – parameter rancangan untuk *SA* adalah jumlah gerakan dalam masing – masing iterasi, faktor pendinginan, dan rasio penerimaan awal.

5. Kemudian klik *debug* setelah itu tampil pilihan, pilih *run* untuk menjalankan program.

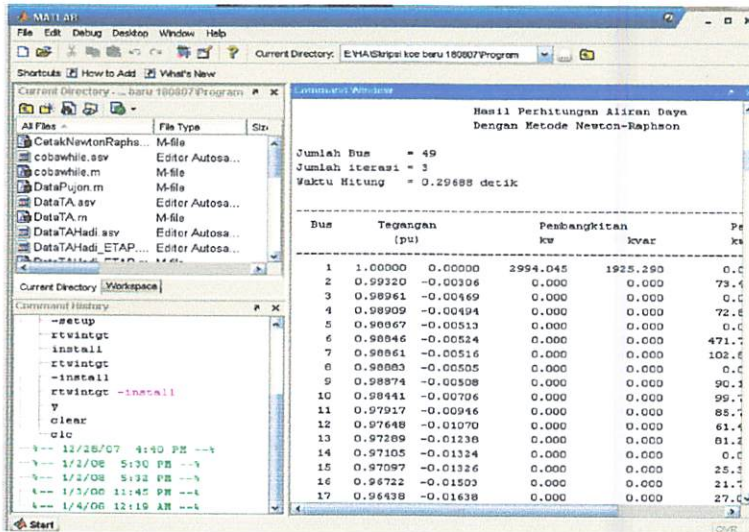


**GAMBAR 4-8 TAMPILAN UNTUK MENJALANKAN PROGRAM**

Pada gambar 4-8 adalah tampilan untuk menjalankan program simulasi matlab 7.0.4. Pilih file dengan nama *debug* kemudian klik, setelah itu muncul seperti yang terlihat pada tampilan kemudian pilih *run* dan klik untuk menjalankan program simulasi matlab 7.0.4. setelah itu lanjut ke langkah berikutnya.



6. Setelah menunggu beberapa saat muncul tampilan hasil perhitungan aliran daya dengan metode *Newton Rhapson* untuk mengetahui tegangan dan sudut fasa tiap bus sebelum optimasi.



**GAMBAR 4-9 TAMPILAN HASIL UNTUK MENGETAHUI TEGANGAN DAN SUDUT FASA TIAP BUS SEBELUM OPTIMASI**

Pada gambar 4-9 terlihat tampilan hasil dari tegangan dan sudut fasa tiap bus, juga nilai P, Q dari pembangkitan dan pembebanan sebelum optimasi hasil dari perhitungan aliran daya dengan menggunakan *metode newton rhapson*.

Dari hasil perhitungan ini dapat terlihat jatuh tegangan yang tidak di izinkan terdapat pada bus 33, bus 34 kemudian bus 37 s/d bus 49 yaitu sebesar 0.94 pu atau 18 kV dimana tegangan yang di izinkan yaitu 0.95 pu s/d 1.05 pu. Pada tabel 4-3 diperlihatkan bus-bus yang terdapat jatuh tegangan.

**Tabel 4-3**  
**Tegangan dan Sudut Fasa Tiap Bus Penyulang Syiar**  
**Sebelum Kompensasi**

Bus	Tegangan abs (pu)	Sudut Fasa (rad)
1	1	0
2	0.9932	-0.00306
3	0.98961	-0.00469
4	0.98909	-0.00494
5	0.98867	-0.00513
6	0.98846	-0.00524
7	0.98861	-0.00516
8	0.98883	-0.00505
9	0.98874	-0.00508
10	0.98441	-0.00706
11	0.97917	-0.00946
12	0.97648	-0.0107
13	0.97289	-0.01238
14	0.97105	-0.01324
15	0.97097	-0.01326
16	0.96722	-0.01503
17	0.96438	-0.01638
18	0.95966	-0.01863
19	0.9596	-0.01867
20	0.95948	-0.01871
21	0.95955	-0.01869
22	0.95951	-0.01871
23	0.95948	-0.01872
24	0.95941	-0.01875
25	0.95944	-0.01874
26	0.95772	-0.01957
27	0.95494	-0.02091
28	0.95472	-0.021
29	0.95041	-0.02312
30	0.95025	-0.0232
31	0.9502	-0.02322
32	0.95009	-0.02328
33	0.94954	-0.02355
34	0.94869	-0.02396
35	0.95005	-0.02329
36	0.95001	-0.02332
37	0.94995	-0.02334
38	0.94988	-0.02338
39	0.94934	-0.02365

40	0.94849	-0.02406
41	0.94832	-0.02413
42	0.94823	-0.02416
43	0.94788	-0.02436
44	0.9475	-0.02455
45	0.94718	-0.02471
46	0.94741	-0.02458
47	0.94751	-0.02455
48	0.94723	-0.02468
49	0.94716	-0.02472

7. Gerakan *scroll* ke bawah untuk melihat tampilan aliran daya antar saluran sebelum optimasi.

Bus from	Bus to	Daya ku	Daya kvar	Bus from	Bus to	Daya
1	2	2994.045	1925.290	2	1	-2979.526
2	3	2906.111	1857.595	3	2	-2899.615
3	4	647.927	401.595	4	3	-647.487
3	8	2250.688	1444.143	8	3	-2249.425
4	5	574.859	356.492	5	4	-574.687
5	6	471.023	292.476	6	5	-471.750
5	7	102.064	63.753	7	5	-102.859
8	9	90.106	55.847	9	8	-90.100
8	10	2159.319	1386.365	10	8	-2152.439
10	11	2052.725	1314.052	11	10	-2044.939
11	12	1959.137	1246.986	12	11	-1955.323
12	13	1893.885	1205.046	13	12	-1890.915
13	14	1807.646	1147.065	14	13	-1805.212
14	15	25.324	15.695	15	14	-25.322
14	16	1779.888	1127.669	16	14	-1774.900
16	17	1753.114	1106.543	17	16	-1749.439
17	18	1722.349	1084.136	18	17	-1716.355
18	19	141.547	87.730	19	18	-141.540
18	26	1574.809	987.236	26	18	-1572.538
19	20	56.140	34.000	20	19	-56.143

**GAMBAR 4-10 TAMPILAN HASIL ALIRAN DAYA ANTAR SALURAN SEBELUM OPTIMASI**

Untuk gambar 4-10 memperlihatkan tampilan hasil aliran daya antar saluran dari bus ke bus untuk daya yang di alirkan sepanjang saluran pada penyulang Syiar, Pada tabel 4-4 memperlihatkan daya tiap saluran sebelum optimasi.



**Tabel 4-4**  
**Daya Tiap Saluran Penyulang Syiar**  
**Sebelum Optimasi**

Saluran		P (kW)	Q (kVAR)	Saluran		P (kW)	Q (kVAR)
From	To			From	To		
1	2	2994.045	1925.29	2	1	-2979.526	-1903.094
2	3	2906.111	1857.596	3	2	-2898.615	-1846.137
3	4	647.927	401.995	4	3	-647.687	-401.627
3	8	2250.688	1444.143	8	3	-2249.425	-1442.212
4	5	574.859	356.492	5	4	-574.687	-356.229
5	6	471.823	292.476	6	5	-471.75	-292.364
5	7	102.864	63.753	7	5	-102.859	-63.746
8	9	90.106	55.847	9	8	-90.1	-55.839
8	10	2159.319	1386.365	10	8	-2152.439	-1375.849
10	11	2052.725	1314.052	11	10	-2044.939	-1302.149
11	12	1959.157	1248.986	12	11	-1955.323	-1243.124
12	13	1893.885	1205.048	13	12	-1888.915	-1197.451
13	14	1807.646	1147.085	14	13	-1805.212	-1143.364
14	15	25.324	15.695	15	14	-25.322	-15.693
14	16	1779.888	1127.669	16	14	-1774.9	-1120.044
16	17	1753.114	1106.543	17	16	-1749.439	-1100.925
17	18	1722.349	1084.136	18	17	-1716.355	-1074.974
18	19	141.547	87.738	19	18	-141.54	-87.727
18	26	1574.809	987.236	26	18	-1572.538	-983.764
19	20	56.148	34.8	20	19	-56.143	-34.794
19	21	85.392	52.927	21	19	-85.389	-52.922
21	22	36.704	22.748	22	21	-36.703	-22.746
21	23	48.685	30.175	23	21	-48.682	-30.171
23	24	22.577	13.993	24	23	-22.576	-13.991
23	25	26.105	16.178	25	23	-26.104	-16.177
26	27	1508.745	944.229	27	26	-1505.627	-939.464
27	28	68.123	42.226	28	27	-68.111	-42.211
27	29	1437.504	897.237	29	27	-1432.653	-889.822
29	30	1373.96	853.448	30	29	-1373.804	-853.209
30	31	420.432	260.902	31	30	-420.414	-260.875
30	36	953.372	592.307	36	30	-953.197	-592.039
31	32	387.043	240.194	32	31	-387.012	-240.146
32	33	342.257	212.41	33	32	-342.117	-212.194
32	35	44.754	27.737	35	32	-44.753	-27.735
33	34	293.437	182.025	34	33	-293.25	-181.74
36	37	60.855	37.716	37	36	-60.852	-37.712
36	38	892.342	554.323	38	36	-892.256	-554.191
38	39	829.05	515.019	39	38	-828.713	-514.505
39	40	701.714	435.798	40	39	-701.268	-435.116

40	41	121.335	75.21	41	40	-121.318	-75.189
40	43	579.933	359.906	43	40	-579.667	-359.499
41	42	65.566	40.637	42	41	-65.561	-40.631
43	44	429.041	266.091	44	43	-428.92	-265.906
43	47	150.627	93.408	47	43	-150.585	-93.344
44	45	382.591	237.192	45	44	-382.5	-237.052
44	46	46.328	28.714	46	44	-46.325	-28.71
47	48	109.164	67.674	48	47	-109.142	-67.64
48	49	26.896	16.669	49	48	-26.894	-16.667

8. Gerakan *scroll* ke bawah lagi untuk melihat tampilan rugi daya antar saluran sebelum optimasi

Bus from	Bus to	Rugi kw	Rugi kvar
1	2	14.519	22.195
2	3	7.496	11.459
3	4	0.241	0.360
3	6	1.269	1.991
4	5	0.172	0.263
5	6	0.073	0.112
5	7	0.005	0.007
8	9	0.006	0.008
8	10	6.879	10.516
10	11	7.786	11.903
11	12	3.034	5.062
12	13	4.970	7.596
13	14	2.434	3.721
14	15	0.002	0.002
14	16	4.968	7.625
16	17	3.675	5.610
17	18	5.994	9.162
18	19	0.007	0.011
18	20	2.271	3.471
19	20	0.005	0.006
19	21	0.003	0.005

**GAMBAR 4-11 TAMPILAN HASIL RUGI DAYA ANTAR SALURAN SEBELUM OPTIMASI**

Pada gambar 4-11 memperlihatkan hasil rugi daya antar saluran sepanjang penyulang Syiar sebelum optimasi yang diakibatkan rugi-rugi saluran,. Tabel 4-5 memperlihatkan rugi daya tiap saluran.

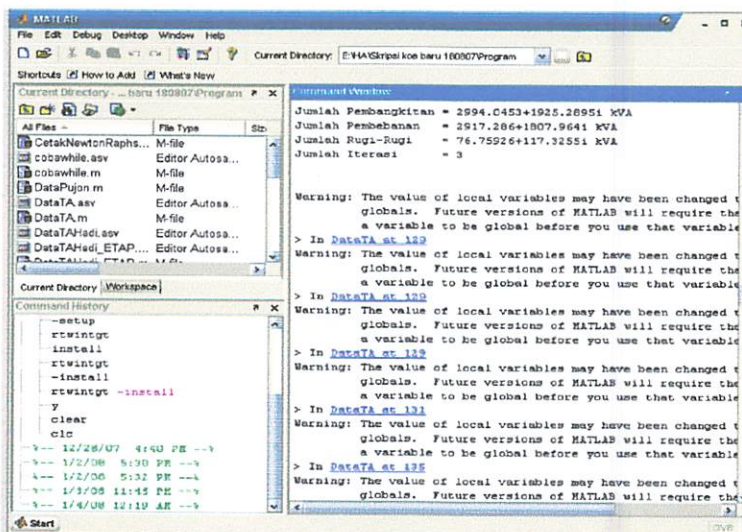


**Tabel 4-5**  
**Rugi Daya Tiap Saluran Penyulang Syiar**  
**Sebelum Kompensasi**

Saluran		Rugi Daya	
Dari	Ke	kW	kVAR
1	2	14.519	22.195
2	3	7.496	11.459
3	4	0.241	0.368
3	8	1.263	1.931
4	5	0.172	0.263
5	6	0.073	0.112
5	7	0.005	0.007
8	9	0.006	0.008
8	10	6.879	10.516
10	11	7.786	11.903
11	12	3.834	5.862
12	13	4.97	7.598
13	14	2.434	3.721
14	15	0.002	0.002
14	16	4.988	7.625
16	17	3.675	5.618
17	18	5.994	9.162
18	19	0.007	0.011
18	26	2.271	3.471
19	20	0.005	0.006
19	21	0.003	0.005
21	22	0.001	0.002
21	23	0.003	0.004
23	24	0.001	0.002
23	25	0.001	0.001
26	27	3.117	4.766
27	28	0.012	0.015
27	29	4.851	7.415
29	30	0.157	0.239
30	31	0.018	0.027
30	36	0.176	0.269
31	32	0.031	0.047
32	33	0.141	0.215
32	35	0.001	0.002
33	34	0.187	0.285
36	37	0.003	0.004
36	38	0.086	0.131
38	39	0.337	0.515
39	40	0.446	0.682
40	41	0.017	0.021
40	43	0.266	0.406

41	42	0.005	0.006
43	44	0.121	0.185
43	47	0.042	0.064
44	45	0.091	0.14
44	46	0.003	0.004
47	48	0.022	0.034
48	49	0.002	0.002

9. Gerakan lagi *scroll* ke bawah untuk melihat tampilan hasil jumlah pembangkitan, pembebanan, dan jumlah rugi-rugi sebelum optimasi.



**GAMBAR 4-12 TAMPILAN HASIL JUMLAH PEMBANGKITAN, PEMBEBANAN, RUGI-RUGI SEBELUM OPTIMASI**

Pada gambar 4-12 memperlihatkan jumlah pembangkitan, jumlah pembebanan, dan jumlah rugi – rugi sebelum optimasi dari hasil perhitungan aliran daya dengan metode *Newton Rhapson*. Dimana untuk Jumlah pembangkitan sebesar  $2994.0453+1925.2895i$  kVA, Jumlah pembebanan sebesar  $2917.286+1807.964i$  kVA dan Jumlah rugi-rugi sebesar  $76.75926+117.3255i$  kVA

#### 4.4. Analisa Hasil Sebelum Optimasi

Setelah dilakukan perhitungan dengan menggunakan bahasa pemrograman MATLAB 7.0.4. untuk hasil *load flow* atau aliran daya sebelum dilakukan optimasi dengan menggunakan metode *newton raphson* terdapat jatuh tegangan atau yang beroperasi diluar batas yang diijinkan oleh PLN (0.95 pu s/d 1.05 pu) yang terlihat pada gambar 4-9. Dimana jatuh tegangan sebesar 0.94 pu atau 18 kV terjadi pada bus 33, 34 kemudian bus 37 s/d 49. Sedangkan pada gambar 4-12 adalah tampilan hasil dari jumlah pembangkitan, jumlah pembebanan, serta rugi-rugi daya sebelum optimasi.

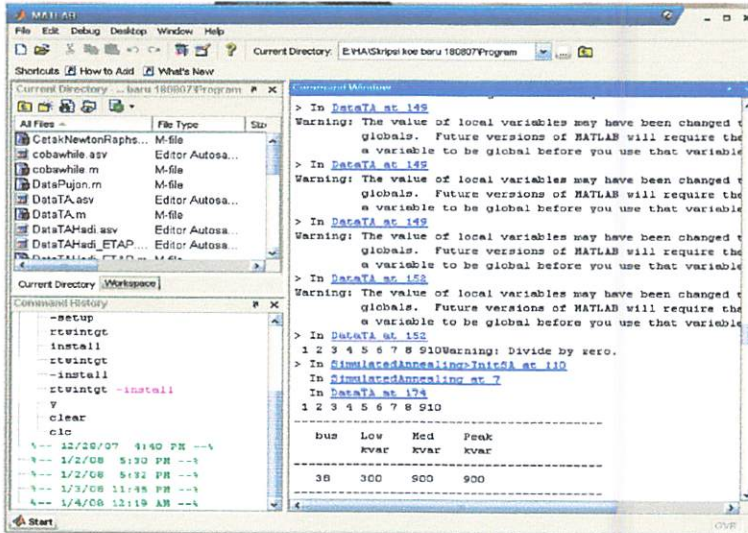
- Jumlah pembangkitan sebelum optimasi sebesar  $2994.0453+1925.2895i$  kVA.
- Jumlah rugi-rugi daya sebelum optimasi sebesar  $76.75926+117.3255i$  kVA.
- Jumlah pembebanan sebelum optimasi sebesar  $2917.286+1807.964i$  kVA.

Besarnya rugi-rugi daya serta buruknya kondisi tegangan yang telah dijabarkan di atas akibat dari meningkatnya beban-beban induktif, misalnya motor-motor listrik, trafo, lampu-lampu penerangan dan lain-lain.

Dari hasil analisa diatas dibutuhkan suatu cara untuk memperbaiki kondisi jaringan yang dianggap kritis yang beroperasi diluar batas yang diijinkan oleh PLN (0.95 pu s/d 1.05 pu). Dan cara alternatif yang sering dipakai adalah dengan cara memasang kapasitor untuk menaikkan profil tegangan dan menekan seminimal mungkin rugi-rugi daya.

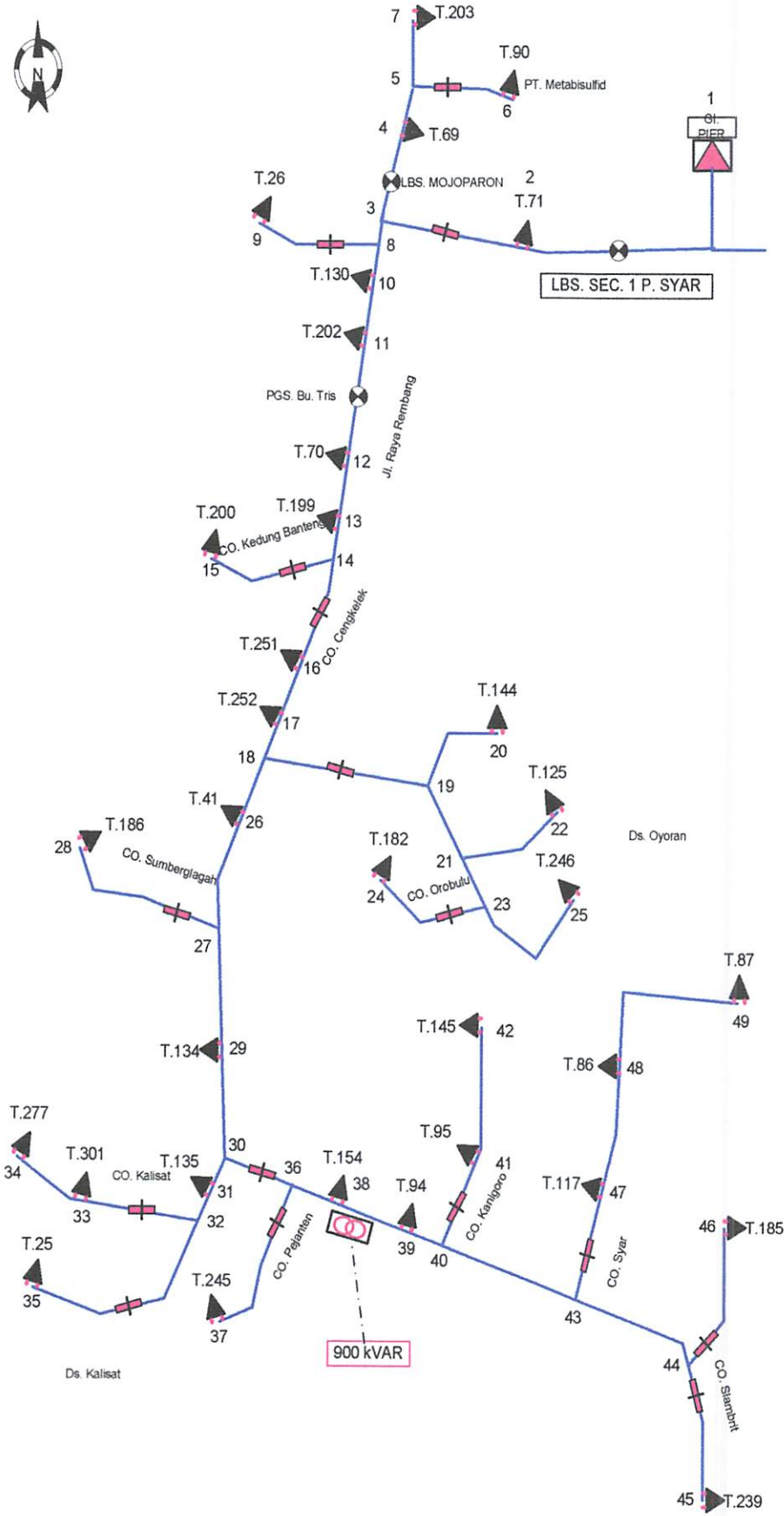


10. Gerakan *scroll* ke bawah untuk melihat hasil tampilan penempatan kapasitor dengan menggunakan metode *Hybrid GA-SA*







**GAMBAR 4-13 TAMPILAN HASIL PENEMPATAN KAPASITOR DENGAN MENGGUNAKAN METODE *HYBRID GA-SA***

Pada gambar 4-13 memperlihatkan hasil dari penempatan kapasitor dengan menggunakan metode *hybrid GA-SA*, dimana kapasitor yang optimal ditempatkan pada bus 38 dengan jenis kapasitor yang digunakan adalah tipe *switch*. dengan kapasitas kapasitor sebesar 900 kVAR. Berikut diperlihatkan *single line diagram* penyulang syiar setelah penempatan kapasitor.



Gambar 4-16  
Single Line Diagram Penyalang Syjar Setelah  
Penempatan Kapasitor

**Keterangan:**

-  : Gardu lindung
-  : CO (cut over)
-  : Trafo
-  : Kapasitor

11. Gerakan *scroll* ke bawah untuk melihat hasil tampilan dari hasil perhitungan aliran daya untuk mengetahui tegangan dan sudut fasa setelah optimasi.

Bus	Tegangan (pu)	Pembangkitan kw	Jarak	P	Q
1	1.00000	0.00000	2973.789	994.325	0.0
2	0.99486	-0.00409	0.000	0.000	73.4
3	0.99217	-0.00629	0.000	0.000	0.0
4	0.99165	-0.00653	0.000	0.000	72.6
5	0.99124	-0.00672	0.000	0.000	0.0
6	0.99102	-0.00682	0.000	0.000	471.7
7	0.99117	-0.00675	0.000	0.000	102.4
8	0.99164	-0.00679	0.000	0.000	0.0
9	0.99155	-0.00683	0.000	0.000	90.1
10	0.99070	-0.00973	0.000	0.000	99.7
11	0.98532	-0.01328	0.000	0.000	85.7
12	0.98362	-0.01514	0.000	0.000	61.4
13	0.98140	-0.01766	0.000	0.000	81.2
14	0.98029	-0.01898	0.000	0.000	0.0
15	0.98022	-0.01900	0.000	0.000	25.3
16	0.97802	-0.02173	0.000	0.000	21.7
17	0.97635	-0.02380	0.000	0.000	27.0

**GAMBAR 4-14 TAMPILAN HASIL UNTUK MENGETAHUI TEGANGAN DAN SUDUT FASA TIAP BUS SETELAH OPTIMASI**

Pada gambar 4-14 terlihat tampilan hasil dari tegangan dan sudut fasa tiap bus, juga nilai P, Q dari pembangkitan dan pembebanan setelah optimasi, hasil dari perhitungan aliran daya dengan menggunakan metode *newton rhapsion*.

Dari hasil perhitungan aliran daya setelah optimasi terdapat kenaikan tegangan disetiap bus yang sebelumnya terjadi jatuh tegangan yaitu di bus 33, dan bus 34 kemudian bus 37 s/d bus 49 yaitu sebesar 0.96 pu atau 19,2 kV dimana tegangan yang di izinkan yaitu 0.95 pu s/d 1.05 pu, yang mana artinya bahwa pada bus-bus tersebut telah masuk pada batas tegangan yang diizinkan oleh PLN. Tabel 4-6 memperlihatkan keadaan tegangan dan sudut fasa setelah optimasi.

**Tabel 4-6**  
**Tegangan dan Sudut Fasa Tiap Bus Penyulang Syiar**  
**Setelah Optimasi**

Bus	Tegangan abs (pu)	Sudut fasa (rad)
1	1	0
2	0.99486	-0.00409
3	0.99217	-0.00629
4	0.99165	-0.00653
5	0.99124	-0.00672
6	0.99102	-0.00682
7	0.99117	-0.00675
8	0.99164	-0.00679
9	0.99155	-0.00683
10	0.9887	-0.00973
11	0.98532	-0.01328
12	0.98362	-0.01514
13	0.9814	-0.01766
14	0.98029	-0.01898
15	0.98022	-0.019
16	0.97802	-0.02173
17	0.97635	-0.0238
18	0.97362	-0.02727
19	0.97355	-0.0273
20	0.97344	-0.02734
21	0.97351	-0.02732
22	0.97347	-0.02734
23	0.97344	-0.02736
24	0.97337	-0.02738
25	0.9734	-0.02737
26	0.97256	-0.02875
27	0.97111	-0.03092
28	0.97089	-0.031
29	0.96885	-0.03452
30	0.96878	-0.03464
31	0.96873	-0.03467
32	0.96862	-0.03472
33	0.96808	-0.03498
34	0.96725	-0.03538
35	0.96858	-0.03473
36	0.96872	-0.03488
37	0.96866	-0.0349
38	0.96869	-0.03501
39	0.96816	-0.03526
40	0.96733	-0.03566



41	0.96716	-0.03572
42	0.96708	-0.03576
43	0.96673	-0.03595
44	0.96636	-0.03613
45	0.96605	-0.03628
46	0.96627	-0.03616
47	0.96636	-0.03613
48	0.9661	-0.03626
49	0.96602	-0.03629

12. Gerakan *scroll* ke bawah untuk melihat tampilan hasil aliran daya antar saluran setelah optimasi.

Bus from	Bus to	Daya kw	Daya kvar	Bus from	Bus to	Daya kw
1	2	2973.789	594.325	2	1	-2962.523
2	3	2069.108	931.605	3	2	-2803.321
3	4	647.925	401.991	4	3	-647.686
3	8	2235.396	520.767	8	3	-2234.471
4	5	574.858	356.490	5	4	-574.686
5	6	471.823	292.475	6	5	-471.750
5	7	102.864	63.753	7	5	-102.859
8	9	90.106	55.847	9	8	-90.100
8	10	2144.365	463.506	10	8	-2139.365
10	11	2039.651	394.065	11	10	-2034.043
11	12	1948.261	332.930	12	11	-1945.521
12	13	1884.083	290.066	13	12	-1880.551
13	14	1799.282	234.300	14	13	-1797.564
14	15	25.324	15.695	15	14	-25.322
14	16	1772.240	215.978	16	14	-1768.726
16	17	1746.940	197.105	17	16	-1744.355
17	18	1717.265	176.365	18	17	-1713.058
18	19	141.546	87.737	19	18	-141.539
18	26	1571.512	82.197	26	18	-1569.930
19	20	56.140	34.800	20	19	-56.143

**GAMBAR 4-15 TAMPILAN HASIL ALIRAN DAYA ANTAR SALURAN SETELAH OPTIMASI**

Untuk gambar 4-15 terlihat tampilan hasil aliran daya antar saluran dari bus ke bus untuk daya yang dialirkan sepanjang saluran pada penyulang Syiar setelah optimasi. Pada tabel 4-7 memperlihatkan daya tiap saluran setelah di optimasi

**Tabel 4-7**  
**Daya Tiap Saluran Penyulang Syiar**  
**Setelah Optimasi**

Saluran		Daya		Saluran		Daya	
From	To	(kW)	(kVAR)	From	To	(kW)	(kVAR)
1	2	2973.789	994.325	2	1	-2962.523	-977.103
2	3	2889.108	931.605	3	2	-2883.321	-922.758
3	4	647.925	401.991	4	3	-647.686	-401.625
3	8	2235.396	520.767	8	3	-2234.471	-519.352
4	5	574.858	356.49	5	4	-574.686	-356.228
5	6	471.823	292.475	6	5	-471.75	-292.364
5	7	102.864	63.753	7	5	-102.859	-63.746
8	9	90.106	55.847	9	8	-90.1	-55.839
8	10	2144.365	463.506	10	8	-2139.365	-455.862
10	11	2039.651	394.065	11	10	-2034.043	-385.493
11	12	1948.261	332.33	12	11	-1945.521	-328.142
12	13	1884.083	290.066	13	12	-1880.551	-284.666
13	14	1799.282	234.3	14	13	-1797.564	-231.673
14	15	25.324	15.695	15	14	-25.322	-15.693
14	16	1772.24	215.978	16	14	-1768.726	-210.606
16	17	1746.94	197.105	17	16	-1744.355	-193.154
17	18	1717.265	176.365	18	17	-1713.058	-169.934
18	19	141.546	87.737	19	18	-141.539	-87.727
18	26	1571.512	82.197	26	18	-1569.93	-79.779
19	20	56.148	34.8	20	19	-56.143	-34.794
19	21	85.391	52.926	21	19	-85.389	-52.922
21	22	36.704	22.747	22	21	-36.703	-22.746
21	23	48.685	30.174	23	21	-48.682	-30.171
23	24	22.577	13.993	24	23	-22.576	-13.991
23	25	26.105	16.178	25	23	-26.104	-16.177
26	27	1506.137	40.244	27	26	-1503.971	-36.932
27	28	68.123	42.226	28	27	-68.111	-42.211
27	29	1435.848	-5.293	29	27	-1432.481	10.442
29	30	1373.788	-46.816	30	29	-1373.679	46.982
30	31	420.417	260.88	31	30	-420.4	-260.854
30	36	953.262	-307.862	36	30	-953.127	308.068
31	32	387.029	240.173	32	31	-386.999	-240.127
32	33	342.245	212.391	33	32	-342.109	-212.183
32	35	44.754	27.737	35	32	-44.753	-27.735
33	34	293.429	182.014	34	33	-293.25	-181.74
36	37	60.855	37.716	37	36	-60.852	-37.712
36	38	892.272	-345.783	38	36	-892.203	345.888
38	39	828.997	514.94	39	38	-828.674	-514.445
39	40	701.675	435.738	40	39	-701.246	-435.083
40	41	121.334	75.209	41	40	-121.318	-75.189



40	43	579.912	359.874	43	40	-579.657	-359.483
41	42	65.566	40.637	42	41	-65.561	-40.631
43	44	429.033	266.078	44	43	-428.916	-265.9
43	47	150.624	93.404	47	43	-150.584	-93.343
44	45	382.588	237.186	45	44	-382.5	-237.052
44	46	46.328	28.714	46	44	-46.325	-28.71
47	48	109.163	67.673	48	47	-109.141	-67.64
48	49	26.895	16.669	49	48	-26.894	-16.667

13. Gerakan *scroll* ke bawah untuk melihat tampilan hasil rugi daya antar saluran setelah optimasi.

Rugi Daya Antar Saluran			
Bus	from to	Rugi	
		ku	kwax
1	2	11.266	17.222
2	3	5.787	8.047
3	4	0.239	0.366
3	6	0.926	1.415
4	5	0.171	0.261
5	6	0.073	0.111
5	7	0.005	0.007
8	9	0.006	0.008
8	10	5.000	7.644
10	11	5.607	8.572
11	12	2.740	4.189
12	13	3.532	5.400
13	14	1.718	2.627
14	15	0.002	0.002
14	16	3.514	5.372
16	17	2.585	3.951
17	18	4.207	6.432
18	19	0.007	0.010
18	20	1.581	2.418
19	20	0.003	0.006
19	21	0.003	0.004

**GAMBAR 4-16 TAMPILAN HASIL RUGI DAYA ANTAR SALURAN SETELAH OPTIMASI**

Pada gambar 4-16 memperlihatkan hasil rugi daya antar saluran yang diakibatkan rugi-rugi saluran, dimana setelah dilakukan optimasi terjadi penurunan rugi daya antar saluran, Tabel 4-8 memperlihatkan penurunan nilai rugi daya tiap saluran setelah optimasi.

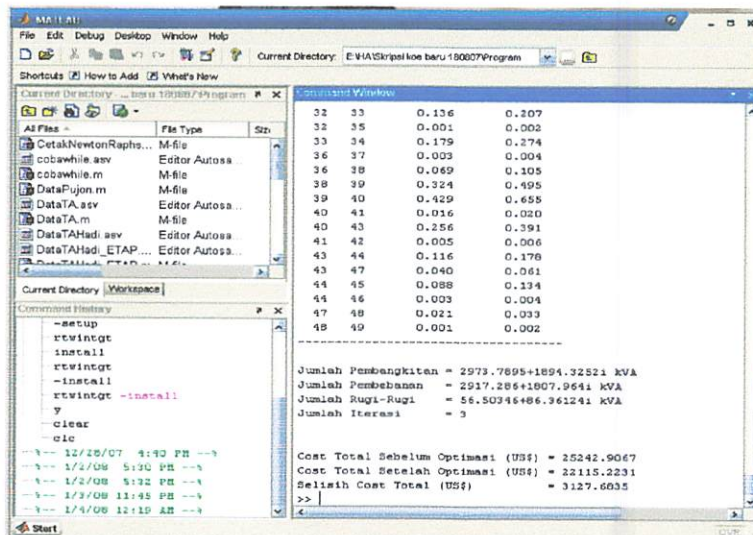
**Tabel 4-8**  
**Rugi Daya Tiap Saluran Penyulang Syiar**  
**Setelah Kompensasi**

Saluran		Rugi Daya	
Dari	Ke	kW	kVAR
1	2	11.266	17.222
2	3	5.787	8.847
3	4	0.239	0.366
3	8	0.926	1.415
4	5	0.171	0.261
5	6	0.073	0.111
5	7	0.005	0.007
8	9	0.006	0.008
8	10	5	7.644
10	11	5.607	8.572
11	12	2.74	4.189
12	13	3.532	5.4
13	14	1.718	2.627
14	15	0.002	0.002
14	16	3.514	5.372
16	17	2.585	3.951
17	18	4.207	6.432
18	19	0.007	0.01
18	26	1.581	2.418
19	20	0.005	0.006
19	21	0.003	0.004
21	22	0.001	0.001
21	23	0.002	0.004
23	24	0.001	0.002
23	25	0.001	0.001
26	27	2.166	3.312
27	28	0.012	0.015
27	29	3.368	5.148
29	30	0.109	0.166
30	31	0.017	0.026
30	36	0.135	0.206
31	32	0.03	0.046
32	33	0.136	0.207
32	35	0.001	0.002
33	34	0.179	0.274
36	37	0.003	0.004
36	38	0.069	0.105
38	39	0.324	0.495
39	40	0.429	0.655
40	41	0.016	0.02
40	43	0.256	0.391



41	42	0.005	0.006
43	44	0.116	0.178
43	47	0.04	0.061
44	45	0.088	0.134
44	46	0.003	0.004
47	48	0.021	0.033
48	49	0.001	0.002

14. Gerakan *scroll* ke bawah untuk melihat hasil tampilan jumlah pembangkitan, pembebanan, dan rugi-rugi setelah optimasi juga *cost total* sebelum optimasi, setelah optimasi dan selisih *cost total* dengan menggunakan metode *Hybrid GA-SA*.



**GAMBAR 4-17 TAMPILAN HASIL JUMLAH PEMBANGKITAN, PEMBEBANAN, RUGI-RUGI DAN COST TOTAL SEBELUM DAN SESUDAH SERTA SELISIH TOTAL COST SETELAH OPTIMASI MENGGUNAKAN METODE *HYBRID GA-SA***

Pada gambar 4-17 dimana jumlah pembangkitan, dan jumlah rugi-rugi setelah optimasi dengan menggunakan Metode *HYBRID GA-SA* pada penyulang SYIAR terjadi penurunan setelah dilakukan optimasi yaitu untuk :

- jumlah pembangkitan menjadi  $2973.7895 + j 1894.3252kVA$ .
- jumlah pembebanan  $2917.286 + j1807.964kVA$ .

- jumlah rugi-rugi menjadi  $56.50346 + j 86.36124kVA$ .

Selain jumlah pembangkitan, serta rugi-rugi yang turun akibat dilakukan optimasi juga terdapat keuntungan lain yaitu *total cost* akibat kehilangan energi setelah pemasangan kapasitor, dimana *total cost* akibat kehilangan energi sebelum optimasi akibat rugi-rugi daya sebesar (US\$) 25.242.9067, dan setelah dilakukan optimasi *total cost* akibat kehilangan energi dari turunya rugi-rugi daya menjadi sebesar (US\$) 22.115.2231 sehingga diperoleh penghematan sebesar (US\$) 3.127.6835 pertahun.

#### 4.5. Analisa Hasil Setelah Optimasi

Setelah dilakukan perhitungan dengan menggunakan bahasa pemrograman MATLAB 7.0.4. Dengan metode *Hybrid GA-SA* penempatan kapasitor yang optimal terletak pada bus 38 dengan kapasitas kapasitor sebesar 900 kVAR dengan jenis kapasitor yang digunakan adalah *tipe switch*, yang seperti terlihat gambar 4-13. Untuk hasil *load flow* atau aliran daya setelah dilakukan optimasi dengan menggunakan metode *newton rhapson* seperti yang terlihat pada gambar 4-14 terdapat kenaikan tegangan di setiap bus yang sebelumnya terjadi jatuh tegangan yaitu di bus 33, 34 dan bus 37 s/d 49 sebesar 0.96 pu atau sebesar 19.2kV. Begitu juga untuk jumlah pembangkitan dan rugi-rugi daya terjadi penurunan seperti yang terlihat pada gambar 4-17.

- Untuk jumlah pembangkitan sebesar  $2973.7895+1894.3252i$  kVA dari semula sebesar  $2994.0453+1925.2895i$  kVA.

- Jumlah rugi-rugi daya sebesar  $56.50346+86.36124i$  kVA dari semula sebesar  $76.75926+117.3255i$  kVA.

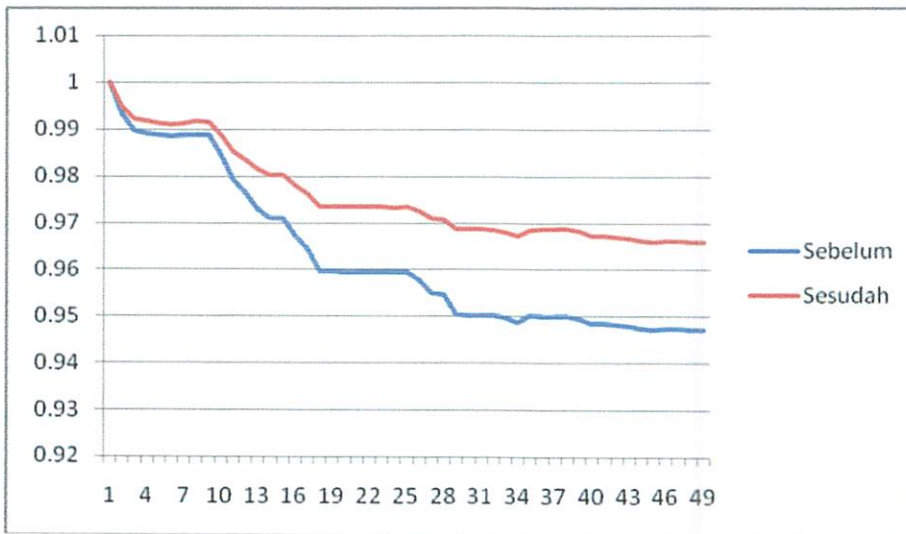
Selain itu didapat penghematan sebesar *US\$ 3.127.6835* pertahun. akibat dari pemasangan kapasitor, dimana sebelum penempatan, *total cost* akibat kehilangan energi dari besarnya rugi-rugi daya sebesar *US\$ 25.242.9067* dan setelah penempatan kapasitor terjadi penurunan *total cost* akibat kehilangan energi dari turunya rugi-rugi daya pada jaringan sebesar *14.14%* atau sebesar *US\$ 22.115.2231*.

Dilihat dari hasil tersebut diatas bahwa kondisi jaringan setelah di pasang kapasitor dengan menggunakan metode *Hybrid GA-SA* terjadi perbaikan terutama untuk profil tegangan menjadi naik sesuai dengan batas operasi yang diijinkan oleh pihak PLN, dan untuk rugi-rugi daya terjadi penurunan yang cukup signifikan. Selain itu juga didapat penghematan akibat dari optimasi penempatan kapasitor dengan menggunakan metode *Hybrid GA-SA*.

Berikut adalah hasil dari program dengan *metode Hybrid GA-SA* setelah dilakukan penempatan kapasitor pada jaringan 20kV sistem distribusi radial di Penyulang Syiar Pasuruan.

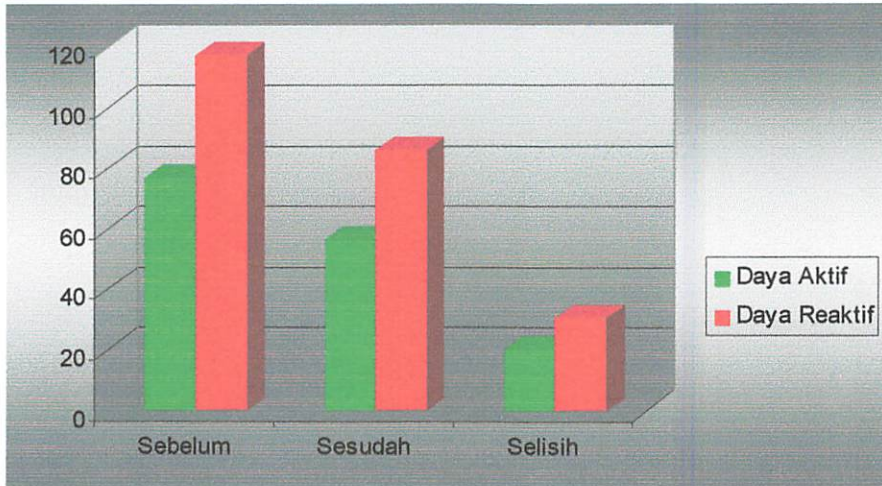
**TABEL 4-9**  
**HASIL PROGRAM METODE *HYBRID GA-SA***

No	Keterangan	Sebelum Penempatan	Setelah Penempatan Dengan Metode Hybrid GA-SA	
			Bus	Kapasitas
1	Lokasi dan Kapasitas Kapasitor			(kVAR)
			38	900
2	Tegangan terendah pada (pu)	0.94716 pu atau 18.94 kV	0.96602 pu atau 19.32 kV	
3	Rugi Daya: Aktif (kW)	76.75926	56.50346	
	Pengurangan (%)		26.38%	
3	Reaktif (kVAR)	117.3255	86.36124	
	Pengurangan (%)		26.39%	
4	Total Cost US\$	25.242.91	22.115.22	
	Selisih Total Cost US\$		14.14%	
			3.127.68	



**GRAFIK 4-1 TEGANGAN TIAP BUS TERHADAP TEGANGAN AWAL, TEGANGAN SEBELUM DAN SETELAH OPTIMASI**

Grafik 4-1 adalah grafik perbandingan tegangan tiap bus terhadap tegangan awal, sebelum dan sesudah dikompensasi dapat terlihat bahwa tegangan terendah terjadi pada bus 49 sebesar  $0.94716 pu$  atau  $18.9432 kV$  dan setelah dilakukan kompensasi naik menjadi  $0.96602 pu$  atau  $19.32 kV$ . Dimana batas tegangan yang diijinkan oleh PLN adalah sebesar  $0.95 pu$  sampai  $1.05 pu$ .



**GRAFIK 4-2 RUGI DAYA SEBELUM DAN SETELAH OPTIMASI**

Dari grafik 4-2 terlihat bahwa penurunan rugi daya aktif adalah 20.2558 kW dari 76.75926 kW menjadi 56.50346 kW sehingga terjadi penurunan sebesar 26.38%, Sedangkan untuk daya reaktif juga terjadi penurunan sebesar 30.96426 kVAR dari 117.3255 kVAR menjadi 86.36124 kVAR sehingga terjadi penurunan sebesar 26.39%.

## BAB V

### KESIMPULAN

#### 5.1 Kesimpulan

Dari hasil analisa diatas maka dapat ditarik kesimpulan sebagai berikut:

1. Letak kapasitor yang optimal terletak pada *bus 38* di lokasi Tampung Barat dengan jenis kapasitor yang digunakan adalah tipe *switch*. Dengan kapasitas sebesar *900 kVAR*.
2. Tegangan terendah pada saluran sebesar *0.94716 pu* atau sebesar *18.9432 kV* terdapat pada *bus 49* dan setelah optimasi naik menjadi *0.96602 pu* atau sebesar *19.32 kV*.
3. Besarnya penurunan rugi-rugi daya setelah kompensasi : untuk rugi daya aktif sebesar *26.38%* dari *76.75926 kW* menjadi *56.50346 kW*, dan untuk daya reaktif sebesar *26.39%* dari *117.3255 kVAR* menjadi *86.36124 kVAR*.
4. Untuk *total cost* akibat kehilangan energi sebelum optimasi, akibat rugi-rugi daya sebesar *US\$ 25.242.90*. dan setelah optimasi, akibat turunnya rugi-rugi daya sehingga *total cost* akibat kehilangan energi menjadi sebesar *US\$ 22.115.22* atau terjadi penurunan sebesar *14.14 %*. Sehingga diperoleh nilai penghematan sebesar *US\$ 3.127.68*. pertahun.

## DAFTAR PUSTAKA

- [1]. Al-Mohammed Ali Hasan, Elamin Ibrahim, "Capacitor Placement In Distribution Systems Using Artificial Intelligent Techniques". IEEE Bologna Power Tech Conference, Bologna, Italy, June 23th-26<sup>th</sup> 3003.
- [2] Basri Hasan, "Sistem Distribusi Tenaga Listrik".
- [3]. Marsudi Djiteng, "Operasi sistem tenaga Listrik", Balai Penerbit Dan Humas ISTN 1990.
- [4] Jr, Stevenson D William. "Analisa Sistem Tenaga Listrik". Edisi ke empat, Erlangga, Jakarta, 1993.
- [5]. Mitsuo Gen, Runwei Cheng, "Genetic Algorithm And Engineering Design". (John Willey & Son, Inc, 1994) p-7
- [6]. Kirkpatrick S, Gellat. C and Vecchi. M., "Optimization by Simulated Annealing". May 1983.



# Capacitor Placement In Distribution Systems Using Artificial Intelligent Techniques

Hassan Al-Mohammed Non-Member, IEEE

Ibrahim Elamin, Senior Member, IEEE

Abstract

This paper undertakes the problem of optimal capacitor placement in a distribution system. The problem is how to initially determine the locations to install capacitors, the types and sizes of capacitors to be installed and, during each load level, the control settings of these capacitors in order that a desired objective function is minimized while the load constraints, network constraints and operational constraints (e.g. voltage profile) at different load levels are satisfied. The problem is formulated as a combinatorial optimization problem with a non-differentiable objective function. Four solution methodologies based on simulated annealing (SA), genetic algorithms (GA), tabu search (TS), and hybrid GA-SA algorithms are presented. The solution methodologies are preceded by a sensitivity analysis to select the candidate capacitor installation locations. The effect of non-linear loads on the optimal solution is studied. The proposed solution algorithms have been implemented into and tested on a real power system.

Keywords: capacitor placement, simulated annealing, genetic algorithm, tabu search combinatorial optimization, non-linear loads, artificial intelligence

## I. INTRODUCTION

The application of shunt capacitors in distribution systems has always been an important subject to distribution engineers [1]. The general capacitor placement problem, (CPP), consists of determining the number, location, type, size and control settings at different load levels of the capacitors to be installed. The objective is to minimize energy losses while considering capacitor installation costs. The subject of optimal capacitor placement is a well-researched topic, which has been addressed by many authors [1-8]. Different problem formulations and solution methodologies have been proposed. Analytical methods, gradient search techniques, sensitivity-based methods, dynamic programming and various heuristic techniques have all been applied to solve the capacitor placement problem, resulting in different procedures and shortcomings.

---

Hassan Al-Mohammed is with Saudi Electricity Company (East Region), Jeddah, Saudi Arabia & Ibrahim Elamin is with King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia ( e-mail: [elamin@kfupm.edu.sa](mailto:elamin@kfupm.edu.sa))

Nowadays, because of the widespread use of power electronics and solid-state devices, the non-linear portion of distribution loads has increased, which can cause a significant amount of harmonic distortion in voltage and current. If not properly sized and placed, capacitors may amplify harmonic currents and voltages due to possible resonance. This may lead to additional stress on equipment insulation, an increased rate of capacitor failure, and interference with communication systems [7].

This paper reports on a solution of the CPP by means of general-purpose heuristic optimization techniques. The objective function is minimized subject to power flow constraints, minimum and maximum allowable operating voltages, load variation, use of both fixed and switched capacitors, maximum number of capacitors to be installed at a particular bus, and maximum allowable total harmonic distortion caused by the non-linear loads.

The paper also studies the impact of non-linear loads on the optimal solution.

## II. PROBLEM FORMULATION

In this section, a formulation of the CPP is presented. By taking the non-linear loads into consideration, this new formulation is an extension of the problem formulation proposed by many researchers.

### A. Assumptions

The following assumptions are considered while formulating the problem [7].

1. The system is balanced.
2. Loads vary in a conforming manner.
3. The forecasted active and reactive powers provided by the load duration curve represent fundamental frequency powers. Additional powers at harmonic frequencies are negligible.
4. Loads at bus 'j' are partitioned into  $w_j$  nonlinear loads and  $(1-w_j)$  linear loads.
5. Both clusters of linear and nonlinear loads at bus 'j' have the same displacement factor, i.e., power factor at fundamental frequency.

### B. Objective Function

The objective of the capacitor placement problem is to reduce the total energy losses of the system during all load levels while striving to minimize the cost of the capacitors installed in the system. The objective function consists of two terms. The first is the cost of capacitor placement and the second is the cost of the total energy losses.

cost associated with capacitor placement is composed of a fixed installation cost and a purchase cost. The cost function described in this way is a step-like function rather than a continuously differentiable function since capacitors in practice are grouped in banks of standard discrete capacities. The second term in the objective function represents the total cost of the energy losses. This term is obtained by summing the real power losses for each load level multiplied by the responding duration.

#### Mathematical Representation

The capacitor placement problem is expressed mathematically as shown below:

$$\text{minimize } \sum_{k=1}^{n_k} C_k(u_k^o) + k_e \sum_{i=1}^{n_l} T_i P_{loss,i}(x^i, u^i) \quad (1)$$

subject to:

$$u_k^o = l_k * u_s, \quad \text{where } l_k \text{ is a non-negative integer,}$$

$$l_k \in N_C$$

$$u_k^i = \text{discrete variable, } i \in N_T, k \in N_C$$

$$\text{power flow constraints } P_{flow}(x^i, u^i) = 0 \quad i \in N_T$$

$$\text{operational constraints } V_{\min} \leq |V_{ik}| \leq V_{\max}$$

$$\text{for } k \in C_1 = \text{fixed cap } u_k^i = u_k^j \leq u_k^o \text{ for } i, j \in N_T$$

$$\text{for } k \in C_2 = \text{Switched cap. } 0 \leq u_k^i \leq u_k^o \text{ for } i \in N_T$$

$$THD_{ik} \leq THD_{\max}$$

$$l_k \leq (l_k)_{\max}$$

where:

$C_k(u_k^o)$  represents the installation cost of capacitors;

$\sum_{i=1}^{n_l} T_i P_{loss,i}(x^i, u^i)$  represents the cost of the total energy losses;

and

variables are defined as follows:

$n_l$ : load level

$n_k$ : location(bus)

$x^i$ : a vector of state variables

$u^i$ : a vector of control variables (capacitor placement scheme)

$C_1$ : possible locations to install capacitors;

$u_s$ : kVar value of the capacitor at the peak load;

$k_e$ : energy cost per unit;

$n_l$ : number of load levels;

$T_i$ : duration for load level  $i$ ;

$C_2$ : standard size of capacitor bank;

$N_C$ : set of possible locations to install capacitors;

$N_T$ : set of different load levels;

$V_{ik}$ : total rms voltage at the  $k$ -th bus during the  $i$ -th load level;

$V_{\min}$ :

$V_{\min}$ : minimum allowable operating voltage;

$V_{\max}$ : maximum allowable operating voltage;

$C_1$ : set of fixed capacitors;

$C_2$ : set of switched capacitors;

$THD_{ik}$ : total harmonic distortion at the  $k$ -th bus during the  $i$ -th load level;

$THD_{\max}$ : maximum allowable total harmonic distortion;

$(l_k)_{\max}$ : maximum number of capacitor banks to be installed at bus( $k$ );

### III. SOLUTION ALGORITHMS

Combinatorial optimization problems can be solved either by exact or by approximate methods. In exact methods, all the feasible solutions are evaluated and the best one is selected as the optimal solution. However, exact methods are impractical when a real-life problem is to be solved. This is due to the large number of feasible solutions to be evaluated. This paper reports on the application of simulated annealing (SA), genetic algorithm (GA), and the tabu search (TS) for the solution of the CPP problem. The methods are well documented in the literature and will not be reviewed here [9-11].

#### A. Application of SA to the Capacitor Placement Problem

The following steps summarize the procedure of an SA-based solution algorithm:

1. Input system and network data, integer seed, cost of capacitors and parameters of the annealing algorithm.
2. Calculate the system voltages, power losses during each load level and the total harmonic distortion for the case of nonlinear loads prior to capacitor placement.
3. Generate an initial feasible solution and calculate the associated cost function.
4. Obtain neighboring solutions by executing different types of moves based on their pre-specified percentages.
5. Check the feasibility of the new configuration. If not feasible go to 4, otherwise proceed to 6.
6. Design a proper cooling schedule. At each temperature, perform a number of moves. For any move do step (7) and step (8). Otherwise, proceed to step (9).
7. Generate a new feasible configuration.
8. Update the system configuration.
9. Check the stopping criterion. If not satisfied, go to (6); otherwise proceed to the next step.
10. Print out the optimal configuration.

#### B. Application of GA to the Capacitor Placement Problem

The algorithm procedure can be summarized as follows:

1. Read system and network data. Input the cost of capacitors, minimum and maximum allowable operating voltages. Input algorithm parameters, i.e. population size, crossover and mutation rates.
2. Calculate system power losses during each load level, total energy losses, bus voltages and total harmonic distortion at each bus for the case of nonlinear loads prior to capacitor installation.
3. Generate a set of initial feasible solution(s), forming the initial population, randomly.

4. Calculate the associated fitness value of each solution.
  5. Calculate the average fitness value of the population. Also, calculate the probability selection of each individual.
  6. Transfer all individuals whose fitness values are less than the calculated average fitness value to the next generation without change.
  7. Select one parent. Choose the other parent randomly. Apply crossover and mutation operators to generate new offspring.
  8. If the offspring is not feasible go to 7, else go to 9.
  9. Calculate the fitness value of the offspring.
  10. Generate an individual in the new population to replace an individual whose fitness value is greater than the calculated average fitness value.
  11. Repeat steps (7) to (10) to find all remaining individuals.
  12. Repeat steps (5) to (11) if the stopping criterion is not satisfied. Otherwise go to (13).
- Best solution in the new population is the optimal solution.

#### Application of TS to the Capacitor Placement Problem

TS solution method has been implemented to the CPP. The algorithm procedure is summarized as follows:

1. Input system and network data, capacitor cost, allowable operating voltage limits, and tabu search parameters including number of moves per iteration and tabu list size.
2. Calculate bus voltages, power losses, and total energy losses during all load levels and total harmonic distortion at each bus for the case of nonlinear loads.
3. Begin the TS algorithm with some initial solution, from the solution space. If not feasible, generate another solution at random. The initial solution is the current best solution.
4. Go through a sample set of candidate moves to generate neighborhood solutions from the neighborhood structure of the current solution.
5. Evaluate the current move. If the move produces a higher evaluation than any other so far found admissible, go to (6). Otherwise, go to (9).
6. Check tabu status. If the candidate move is tabu, go to (7). Otherwise, go to (8).
7. Check aspiration criterion. If the move passes the aspiration criterion, go to (8). Otherwise, go to (9).
8. Check if the move is admissible. Store as new current best move.
9. Check sampling criteria, if another move in the list is to be examined, go to (5).
10. Make the chosen best move.
11. If the stopping criterion is not satisfied, update the tabu list, tabu status and aspiration level. Otherwise, go to (4).
12. Print out the best solution found so far.

#### Application of a Hybrid SA-GA to the CPP

heuristic techniques that have been discussed above may be combined with each other to form a hybrid having

advantages from each one of these heuristics. The efficiency of the new hybrid algorithm is expected to increase.

One way to form a hybrid algorithm is to start with a heuristic to give a fairly good feasible solution rather than starting completely at random. This solution will be considered as the initial solution from which the other heuristic, following the first one, will start the search for the optimal solution. In the paper, combining both GA and SA forms a hybrid algorithm. The algorithm is referred to as GA-SA hybrid algorithm. The GA is applied first to provide a good initial solution. Then, SA begins the search from this solution to find the optimal one.

## IV. NUMERICAL RESULTS

The solution algorithms have been implemented into software packages using FORTRAN-77. Several numerical results are presented to illustrate the performance of the solution algorithms. The test system is a 69-bus distribution system with one main branch and seven laterals. The schematic diagram and load and network data are given in [6]. Table 1 shows the cost, operating data and the operating constraints. The peak, medium, and light load durations are considered 1000 hrs, 6760 hrs and 1000 hrs respectively. The medium and light loads are assumed to be 0.8 and 0.5 of the peak value respectively.

Table 1 System Cost and Related Data

N	$V_{min}$	$V_{max}$	TH $D_{max}$ x (%)	$(I_k)_{max}$ (bank)	$K_e$ (\$/kWh)	Inst. Cost(\$)	Cost (\$) bank	Size kVar
11	0.93	1.05	5	4	0.06	1000	900	300

Three different cases have been studied as described below:

Case-I: With fixed capacitors only and all loads are linear.

Case-II: With fixed and switched capacitors and all loads are linear.

Case-III: With both fixed and switched capacitors and with both linear and non-linear loads.

In the case of non-linear loads, buses # 5,14,20,30,39,50,61 and 64 are assumed to have nonlinear loads with a percentage of 20,10,10,20,30,40,30, and 50% respectively.

#### A. System Conditions Without Capacitors

Prior to capacitor placement, system conditions have been calculated. Minimum bus voltage ( $V_{min}$ ), maximum bus voltage ( $V_{max}$ ), real power losses ( $P_{loss}$ ), cost of energy losses during each load level ( $E_{loss}$ ) and the total cost of energy losses for the case with all the loads linear are shown in Table 2. Table 3 shows the same for the case of the combined linear and nonlinear loads. Maximum total harmonic distortion ( $THD_{max}$ ) during each load level is calculated for this case.

Table 2: System Conditions (linear loads only)

	Load Case		
	Light	Medium	Peak
$V_{min}$ (p.u.)	0.9569	0.9290	0.9094
$V_{max}$ (p.u.)	1.0000	1.0000	1.0000
$P_{loss}$ (kW)	50.4991	137.1431	222.5266
$E_{loss}$ Cost	3029.947	55625.23	13351.59
Total Cost of Energy Losses =US\$72006.770			

Table 3: System Conditions (linear & non-linear loads)

	Load Case		
	Light	Medium	Peak
$V_{min}$ (p.u.)	0.9573	0.9295	0.9099
$V_{max}$ (p.u.)	1.0005	1.0006	1.0006
THD <sub>max</sub> (%)	3.419	3.792	3.884
$P_{loss}$ (kW)	50.6559	138.2789	223.8575
$E_{loss}$ Cost	3039.355	56085.93	13431.45
Total Cost of Energy Losses = US\$72556.73			

*Design Parameters of the Solution Algorithms*

*Design Parameters for the SA*

Design parameters for the SA algorithm are given in Table 4 for three study cases, ( $T$ ) represents the initial temperature, ( $AR$ ) is the initial acceptance ratio, ( $M$ ) is number of moves in 1 iteration, ( $CF$ ) is the cooling factor and  $SC$  represents the number of consecutive iterations, during which no improvement is encountered in the optimal solution, executed before the algorithm is terminated.

Table 4: SA Design Parameters

Parameter	T	AR	M	CF	SC
- I	13000	0.8	5	0.85	15
- II	33000	1.0	7	0.85	15
- III	100	0.53	15	0.88	8

*Design Parameters for the GA*

Table 5 shows the design parameters of the GA algorithm applied to the CPP. The parameters are the population size, ( $P_r$ ), mutation rate, ( $M_r$ ), crossover rate,  $C_r$ , and the number of iterations before the algorithm is terminated, ( $G$ ).

Table 5 GA Design Parameters

Parameter	$P_r$	$M_r$	$G$	$C_r$
- I	30	0.1	20	1.0
- II	32	0.08	20	1.0
- III	20	0.2	15	1.0

*B. 3 Design Parameters for the TS*

Table 6 shows the design parameter for the TS algorithm as applied to the CPP. The parameters are the tabu list, ( $T_s$ ), number of moves in each iteration, ( $V$ ), and the number of consecutive iterations during which no improvement is achieved.

Table 6 TS Design Parameters

Parameter	$T_s$	$V$	$I_t$
Case -I	16	3	40
Case -II	13	7	10
Case -III	13	4	40

*B.4 Design Parameters for the GA-SA Hybrid*

Finally, a hybrid algorithm has been designed by combining both the GA and SA. The algorithm is referred to as GA-SA algorithm. In this hybrid algorithm, the GA is applied first to obtain a fairly good feasible solution. This solution is used as the initial solution for the SA algorithm in its search for the optimal solution. The design parameters of the GA-SA algorithm are given in Table 7.

Table 7- Test System-1: GA-SA Design Parameters

Parameter	$T_o$	$A_{ro}$	$M$	$C_f$	$S_c$	$P_s$	$M_r$	$G$	$C_r$
Case -I	1000	0.83	6	0.90	15	6	0.15	2	1.0
Case -II	3000	1.0	5	0.92	30	6	0.10	3	1.0
Case -III	3000	0.57	7	0.90	20	20	0.2	5	1.0

*C. System Solution for Study Case I*

Table 8 shows the results of the simulation for the study case-I. This case assumes that the system has linear loads and that only fixed capacitors can be used. The data in Table 8 shows the capacitor location and size for different load levels when the various solution algorithms are used. Table 8 also shows the cost of system losses, cost of the capacitor installations, and the system savings. The savings result from the reduction of system losses. The solution calls for the installation of 300 kVar and 1200 kVar capacitors at buses 21 and 61 respectively. The solution algorithms produce the same results with system savings of US\$17028.40.

Table 8 Comparison of the results for Case-I

Solution Method							
SA		GA		TS		GA-SA	
21	61	21	61	21	61	21	61
300	1200	300	1200	300	1200	300	1200
300	1200	300	1200	300	1200	300	1200
300	1200	300	1200	300	1200	300	1200
48478.33	48478.33	48478.33	48478.33	48478.33	48478.33	48478.33	48478.33
6500	6500	6500	6500	6500	6500	6500	6500
54978.33	54978.33	54978.33	54978.33	54978.33	54978.33	54978.33	54978.33
17028.44	17028.44	17028.44	17028.44	17028.44	17028.44	17028.44	17028.44

**System Solution for Study Case II**

Study case-II assumes linear loads but allows for the installation of switchable and fixed capacitors. Table 9 shows results when the four solution algorithms are applied. As case-I, capacitors are needed at buses 21 and 61. However, in this case the sizes vary, depending on the load levels. Column two indicates that a switchable capacitor of 300 kVar is required at bus 21. However, this capacitor should be switched on only during medium and peak load levels. The requirements at bus 61 are 900 kVar capacitors for both light and medium loads. At the peak load, the capacitor should be 1200 kVar. The other algorithms produced different solutions. The hybrid GA-SA shows similar results to the SA's for bus 21. The GA-SA solution requires the installation of 600 kVar, 900 kVar and 1200 kVar for light, medium and peak loads respectively. This algorithm also produced the highest savings of US\$18033.0.

Table 9 Comparison of the Results for Case-II

Location(Bus)	Solution Method							
	SA		GA		TS		GA-SA	
	21	61	21	61	21	61	21	61
Cap(kVar) light load	0	900	600	600	300	900	0	600
Cap(kVar) medium load	300	900	600	900	300	1200	300	900
Cap(kVar) peak load	300	1200	600	1200	300	1200	300	1200
Cost of Energy Losses (US\$)	47807.47		47449.47		47908.653		47472.95	
Installation Cost (US\$)	6500		7400		6500		6500	
Total System Cost (US\$)	54307.47		54849.47		54408.65		53972.95	
Savings (US\$)	17699.3		17157.3		17598.12		18033.82	

**C. System Solution for Study Case III**

Case-III deals with a system of linear and nonlinear loads. Buses 5, 14, 20, 30, 39, 50, 61, and 64 contain linear and nonlinear loads with different percentages as indicated at the beginning of Section 5. The study also allows for the use of switchable and fixed capacitors. The inclusion of the nonlinear loads and harmonic constraints has produced substantial changes in the optimal solution as indicated in Table 10. The SA algorithm produces a solution that requires capacitors at buses 8, 21, 48, 49, 59, and 61. The capacitor sizes and modes are shown in column two. For example, a 1200 kVar has to be placed at bus 8. During the light load, only 600 kVar needs to be switched on. The GA and TS algorithms show that there is a need for capacitors at buses 8, 11, 12, 48, 49, 50, 59 and 61. Columns three and four show system losses and savings. Both methods produce very low savings. As in case -III, the hybrid GA-SA produced the highest savings when capacitors are placed at buses 8, 21, 48, 50, and 61. The savings are US\$4591.95.

**V. CONCLUSIONS**

The following conclusions may be drawn :

- The energy losses of the system increase if nonlinear loads are considered.

- The installation of fixed capacitors has proven to be very economical as it reduces the overall system cost. Installation of switched capacitors is even more economical.
- The optimal solution is changed substantially when the nonlinear loads are taken into account, due to the possible resonance that may be caused by capacitors. If the resonance condition takes place, the distortion levels will be amplified. In this case, the algorithm will search for locations and sizes that do not violate the maximum THD constraint. This, however, will be at the expense of the overall savings of the system. Therefore, the effect of non-linear loads should be incorporated during the planning and design phases of the system.

## VI. ACKNOWLEDGEMENTS

The authors acknowledge the support and facilities of King Fahd University of Petroleum and Minerals, Dhahran Saudi Arabia.

## VII. REFERENCES

- 1]. M. Kaplan, 'Optimization of number, location, size, control type, and control setting of shunt capacitors on radial distribution feeders', IEEE Trans. on PAS, Vol. PAS - 103, No. 9, Sept. 1984, pp. 2659-2665.
- 2]. Y. Baghzouz, 'Effects of nonlinear loads on optimal capacitor placement in radial feeders', IEEE Trans. on Power Delivery, Vol. 6, No. 1, Jan. 1991, pp. 245-251.
- 3]. Hsiao-Dong Chiang, 'Optimal capacitor placement in distribution systems: Part 1: A new formulation and the overall problem', IEEE Trans. on PWRD, Vol. 5, No. 2, April 1990, pp. 634-642.
- 4]. Hsiao-Dong Chiang, 'Optimal capacitor placement in distribution systems: Part 2: Solution algorithm and numerical results', IEEE Trans. on PWRD, Vol. 5, No. 2, April 1990, pp. 643-649.
- 5]. Srinivasan Sundhararajan, 'Optimal selection of capacitors for radial distribution systems using a genetic algorithm', IEEE Trans. on PS, Vol. 9, No. 3, August 1994, pp. 1499-1507
- 6]. Yann-Chang Huang, 'Solving the capacitor placement problem in a radial distribution system using Tabu Search approach', IEEE Trans. on PS, Vol. 11, No. 4, Nov. 1996, pp. 1868- 1873.
- 7]. Z. Q. Wu, 'Optimal choice of fixed and switched capacitors in radial distributors with distorted substation voltage', IEE Proc.-Part C, Vol. 142, No. 1, Jan. 1995, pp. 24-28.
- 8]. Karen Man Min, 'Capacitor Placement, Replacement and Control in a Large-scale Distribution System by GA-based Two State Algorithm', IEEE Trans. on PS, Vol. 12, no.3, August 1997, pp. 1160-1166.
- 9]. Colin R. Reeves, *Modern Heuristic Techniques for Combinational Problems*, McGraw-Hill International, UK, 1995.
- 10]. Glover, Fred, 'New approaches for heuristic search: A bilateral linkage with artificial intelligence', *European Journal of Operation Research*, Vol. 39, No.2, 1989, pp. 119 - 130.
- 11]. S. Kirkpatrick, C. Gelatt and M. Vecchi, 'Optimization by simulated annealing', *Science* Vol. 220, No. 4598, May 1983, pp. 671-680.

**Ali Hassan Al-Mohammed** received his B.Sc. & M.Sc. in electrical engineering from KFUPM in 1994 & 1999 respectively. He is currently with the Saudi Electricity Company, Dammam, Saudi Arabia

**Ibrahim El-Amin** graduated with B.Sc. in electrical engineering from University of Khartoum, Sudan, in 1971. He received his M.Sc. & Ph.D. from UMIST, UK in 1975 & 1978 respectively. He is currently a Professor at KFUPM, Dhahran, Saudi Arabia. Dr. El-Amin's research interests include power system, HVDC transmission, Power quality and system Harmonics and energy conservation.

## VIII Biographies

Table 10 Comparison of the results for Case-III

		Heuristic Method																					
		SA					GA					TS					GA-SA						
Location(Bus)		8	21	48	49	59	61	8	12	48	49	61	11	48	50	59	61	8	21	48	50	61	
Cap(kVar) light load		600	0	900	300	0	900	900	600	300	0	600	300	600	900	600	900	600	300	300	1200	900	600
Cap(kVar) medium load		1200	300	1200	300	0	900	1200	600	900	900	900	300	900	900	600	900	600	300	300	1200	900	900
Cap(kVar) lpeak load		1200	300	1200	300	300	1200	1200	600	900	900	1200	900	900	1200	600	900	900	1200	300	1200	900	1200
Cost of Energy Losses		49469.7					51574.98					52702.2					48564.78						
Installation Cost of Capacitors		19500					19400					18500					19400						
Total System Cost		68969.70					70974.98					71202.20					67964.78						
Savings		3587.03					1581.75					1354.53					4591.95						



# *Sampiran 1*

*Langkah & prosedur pelaksanaan Menya*



*lankan & program & T&T.*

*Hasil & perbandingan & perhitungan & re*

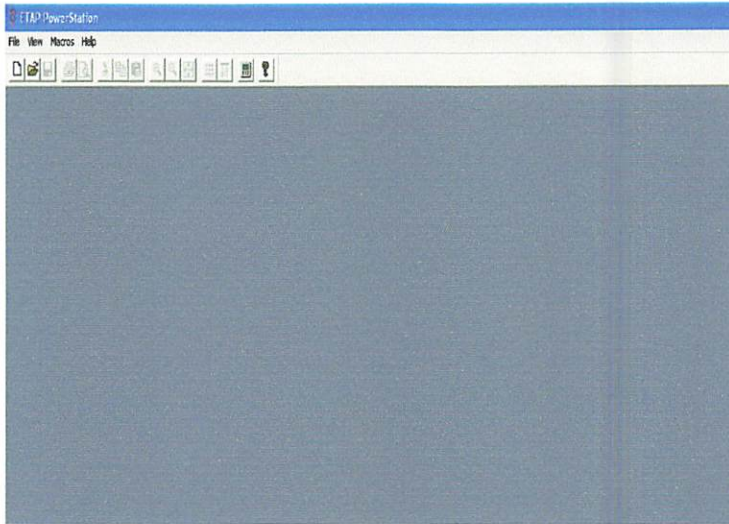


*gram & T&T & dan M&T & T&T.*

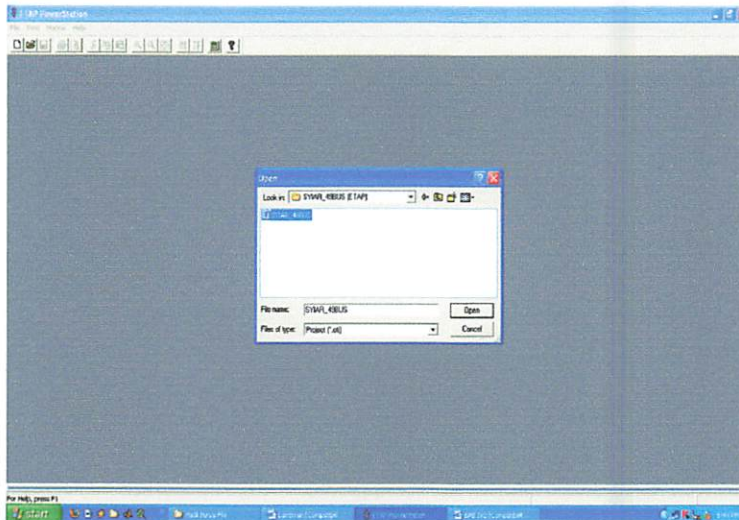


## ➤ **Prosedur Pelaksanaan program ETAP**

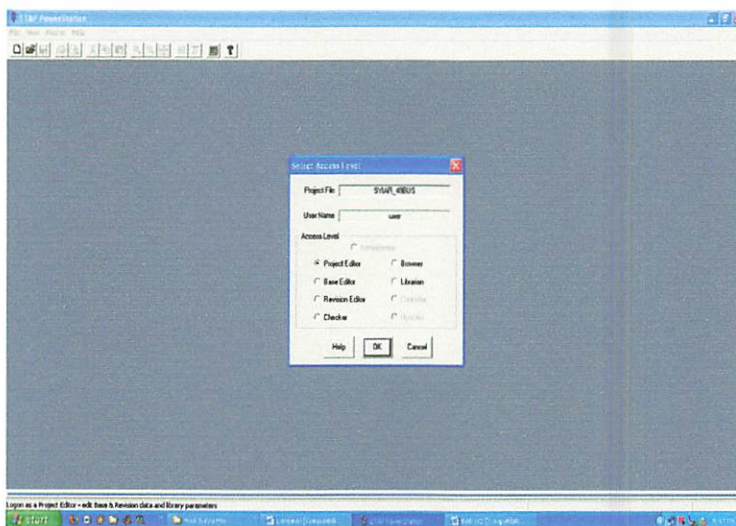
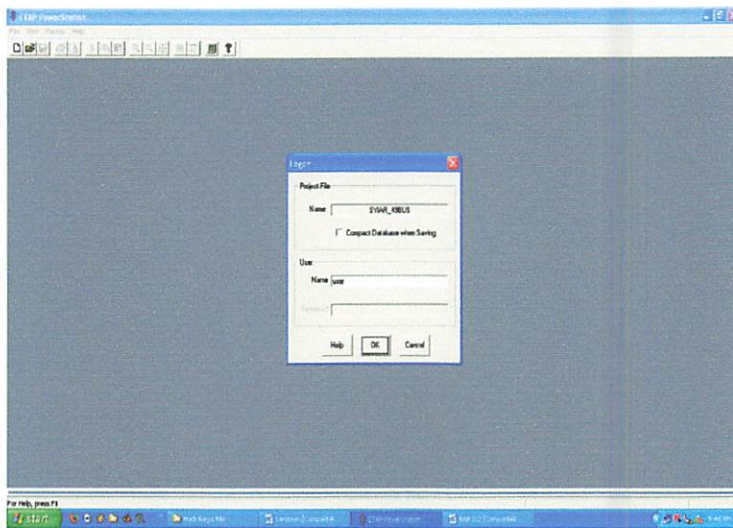
### 1) **Tampilan Utama Program ETAP**



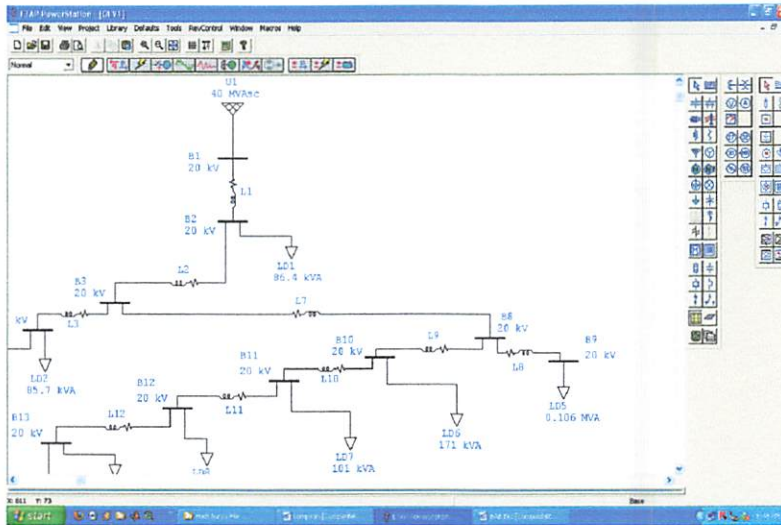
- 2) Untuk menjalankan program ETAP klik open untuk membuka data yang tersimpan kemudian pilih data yang akan digunakan dan klik open.



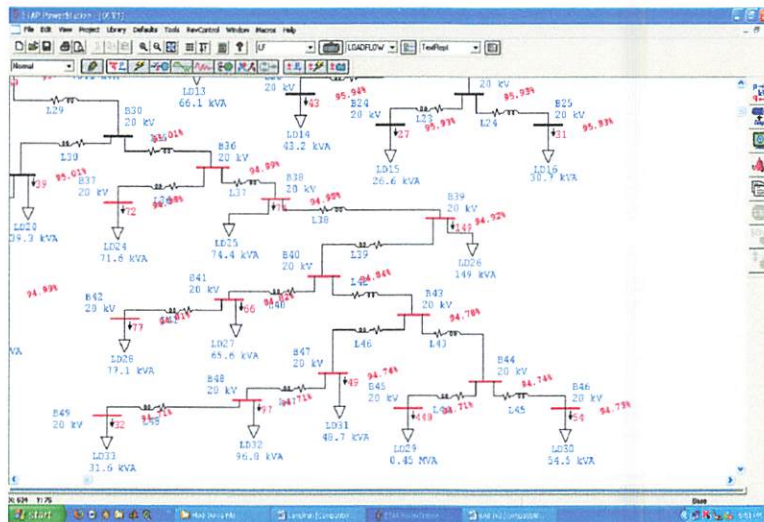
- 3) Setelah itu akan tampil seperti yang terlihat pada tampilan kemudian klik OK



- 4) Setelah mengikuti langkah 3 kemudian akan muncul tampilan seperti berikut yaitu gambar dari single line diagram penyulang Syiar yang sebelumnya telah digambar di ETAP. Dengan data yang digunakan sama dengan data yang digunakan pada program MATLAB.



- 5) Langkah selanjutnya klik loadflow analysis dan run loadflow, untuk melihat bus-bus yang terdapat jatuh tegangan.

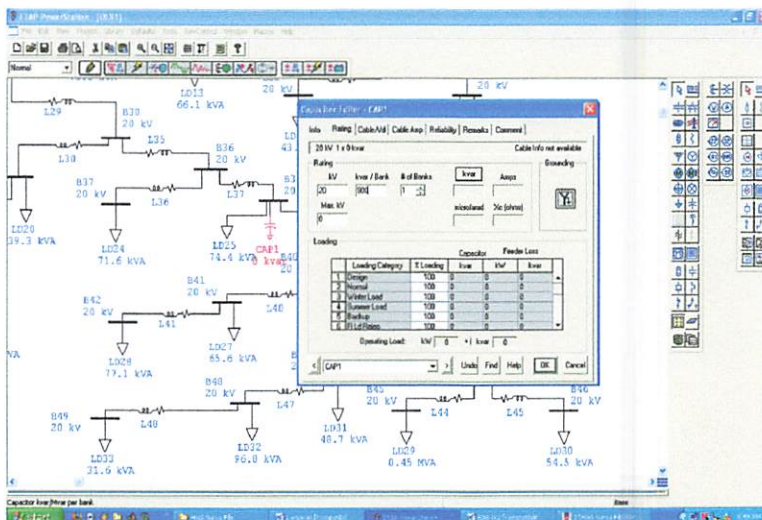




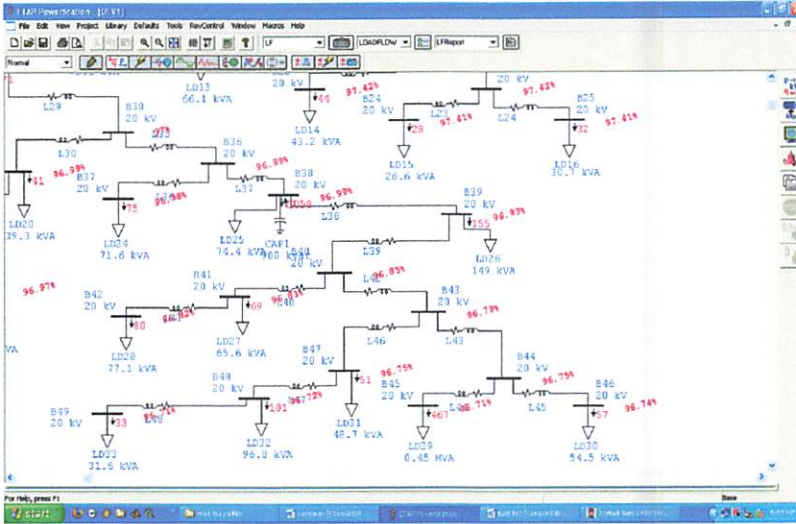
- 6) Kemudian pada text report pilih LfReport dan klik report manager untuk melihat load flow report sebelum optimasi seperti yang terlihat pada tampilan.

Bus	V(kV)	Generation	Wate Load	Heat Load	Load Flow	PF%
B1	20.00	0.00	142.1	1.0	142.1	0.99
B2	20.00	0.00	0.0	0.0	147.7	0.99
B3	20.00	0.00	0.0	0.0	150.0	0.99
B4	20.00	0.00	0.0	0.0	150.0	0.99
B5	20.00	0.00	0.0	0.0	150.0	0.99
B6	20.00	0.00	0.0	0.0	150.0	0.99
B7	20.00	0.00	0.0	0.0	150.0	0.99
B8	20.00	0.00	0.0	0.0	150.0	0.99
B9	20.00	0.00	0.0	0.0	150.0	0.99
B10	20.00	0.00	0.0	0.0	150.0	0.99
B11	20.00	0.00	0.0	0.0	150.0	0.99
B12	20.00	0.00	0.0	0.0	150.0	0.99
B13	20.00	0.00	0.0	0.0	150.0	0.99
B14	20.00	0.00	0.0	0.0	150.0	0.99
B15	20.00	0.00	0.0	0.0	150.0	0.99
B16	20.00	0.00	0.0	0.0	150.0	0.99
B17	20.00	0.00	0.0	0.0	150.0	0.99
B18	20.00	0.00	0.0	0.0	150.0	0.99
B19	20.00	0.00	0.0	0.0	150.0	0.99
B20	20.00	0.00	0.0	0.0	150.0	0.99
B21	20.00	0.00	0.0	0.0	150.0	0.99

- 7) Langkah berikutnya memasang kapasitor dengan cara pilih edit dan klik kemudian pilih capacitor dan pasang sesuai dengan hasil dari program MATLAB, yaitu pada bus 38 dengan kapasitas 900kVAR dengan cara pada kapasitor yang terpasang klik 2x dan akan muncul seperti yang terlihat pada tampilan, kemudian pilih rating untuk men-setting kapasitor sesuai dengan hasil perhitungan program MATLAB.



8) Pada langkah 8 dan 9 akan terlihat hasil load flow setelah dipasang kapasitor, cara ataupun langkah sama dengan langkah 5 dan 6.



The report provides a summary of the load flow analysis. Below is a simplified version of the 'LOAD FLOW REPORT' table:

Bus	V	W	Generation	Meter Load	End Flow	PF
B39	20.00	39.3	0	39.3	39.3	0.97
B42	20.00	77.1	0	77.1	77.1	0.97
B44	20.00	149	0	149	149	0.97
B45	20.00	48.7	0	48.7	48.7	0.97
B46	20.00	54.5	0	54.5	54.5	0.97
B47	20.00	6.45	0	6.45	6.45	0.97
B48	20.00	96.0	0	96.0	96.0	0.97
B49	20.00	31.6	0	31.6	31.6	0.97

**ETAP PowerStation**  
4.0.0C

Page: 9  
Date: 03-16-2008  
SN: KLGCONSULT  
Revision: Base  
Config.: Normal

SYIAR\_49BUS

Study Case: LF

**LOAD FLOW REPORT**

Bus		Voltage		Generation		Motor Load		Static Load		Load Flow					XFMR	
ID	kV	%Mag	Ang.	MW	Mvar	MW	Mvar	MW	Mvar	ID	MW	Mvar	Amp	%PF	% Tap	
	20.000	100.000	0.0	3.03	1.95	0	0	0	0	B2	3.03	1.95	104	84.1		
	20.000	99.312	-0.2	0	0	0	0	0.07	0.05	B1	-3.02	-1.93	104	84.3		
										B3	2.94	1.88	101	84.3		
	20.000	98.948	-0.3	0	0	0	0	0	0	B2	-2.94	-1.87	101	84.4		
										B4	0.65	0.40	22	85.0		
										B8	2.29	1.47	79	84.2		
	20.000	98.896	-0.3	0	0	0	0	0.07	0.05	B3	-0.65	-0.40	22	85.0		
										B5	0.57	0.36	19	85.0		
	20.000	98.855	-0.3	0	0	0	0	0	0	B4	-0.57	-0.35	19	85.0		
										B6	0.47	0.29	16	85.0		
										B7	0.10	0.06	3	85.0		
	20.000	98.833	-0.3	0	0	0	0	0.47	0.29	B5	-0.47	-0.29	16	85.0		
	20.000	98.848	-0.3	0	0	0	0	0.10	0.06	B5	-0.10	-0.06	3	85.0		
	20.000	98.868	-0.3	0	0	0	0	0	0	B3	-2.29	-1.47	79	84.2		
										B9	0.09	0.06	3	85.0		
										B10	2.20	1.41	76	84.2		
	20.000	98.860	-0.3	0	0	0	0	0.09	0.06	B8	-0.09	-0.06	3	85.0		
	20.000	98.419	-0.4	0	0	0	0	0.15	0.09	B8	-2.19	-1.40	76	84.3		
										B11	2.05	1.31	71	84.2		
	20.000	97.897	-0.5	0	0	0	0	0.08	0.05	B10	-2.04	-1.30	71	84.4		
										B12	1.95	1.25	68	84.3		
	20.000	97.628	-0.6	0	0	0	0	0.06	0.04	B11	-1.95	-1.24	68	84.4		
										B13	1.89	1.20	66	84.4		
	20.000	97.270	-0.7	0	0	0	0	0.08	0.05	B12	-1.88	-1.19	66	84.5		
										B14	1.80	1.14	63	84.4		
	20.000	97.086	-0.8	0	0	0	0	0	0	B13	-1.80	-1.14	63	84.5		
										B15	0.03	0.02	0	85.0		
										B16	1.77	1.12	62	84.5		
	20.000	97.078	-0.8	0	0	0	0	0.03	0.02	B14	-0.03	-0.02	0	85.0		
	20.000	96.705	-0.9	0	0	0	0	0.02	0.01	B14	-1.77	-1.12	62	84.6		
										B17	1.75	1.10	61	84.6		
	20.000	96.422	-0.9	0	0	0	0	0.03	0.02	B16	-1.74	-1.10	61	84.6		
										B18	1.72	1.08	60	84.6		
	20.000	95.952	-1.1	0	0	0	0	0	0	B17	-1.71	-1.07	60	84.8		
										B19	0.14	0.09	5	85.0		
										B26	1.57	0.98	55	84.7		



ETAP PowerStation

Page: 10  
 Date: 03-16-2008  
 SN: KLGCONSULT  
 Revision: Base  
 Config: Normal

4.0.0C

Study Case: LF

SYIAR\_49BUS

Bus ID	kV	Voltage		Generation		Motor Load		Static Load		Load Flow			XFMR		
		%Mag	Ang.	MW	Mvar	MW	Mvar	MW	Mvar	MW	Mvar	Amp	%PF	%Tap	
20.000	20.000	95.945	-1.1	0	0	0	0	0	0	0	0	-0.14	-0.09	5	85.0
												0.06	0.03	1	85.0
												0.09	0.05	3	85.0
20.000	20.000	95.934	-1.1	0	0	0	0	0.06	0.03	0.03	0.03	-0.06	-0.03	1	85.0
20.000	20.000	95.940	-1.1	0	0	0	0	0	0	0	0	-0.09	-0.05	3	85.0
												0.04	0.02	1	85.0
												0.05	0.03	1	85.0
												-0.04	-0.02	1	85.0
												-0.05	-0.03	1	85.0
												0.02	0.01	0	85.0
												0.03	0.02	0	85.0
												-0.02	-0.01	0	85.0
												-0.03	-0.02	0	85.0
												-1.57	-0.98	55	84.8
												1.50	0.94	53	84.8
												-1.50	-0.94	53	84.8
												0.07	0.04	2	85.0
												1.43	0.89	51	84.8
												-0.07	-0.04	2	85.0
												-1.43	-0.89	51	84.9
												1.37	0.85	48	84.9
												-1.37	-0.85	48	85.0
												0.42	0.26	14	85.0
												0.95	0.59	34	84.9
												-0.42	-0.26	14	85.0
												0.38	0.24	13	85.0
												-0.38	-0.24	13	85.0
												0.34	0.21	12	85.0
												0.04	0.03	1	85.0
												-0.34	-0.21	12	85.0
												0.29	0.18	10	85.0
												-0.29	-0.18	10	85.0
												-0.04	-0.03	1	85.0
												-0.95	-0.59	34	84.9
												0.06	0.04	2	85.0
												0.89	0.55	31	84.9
												-0.06	-0.04	2	85.0
												-0.89	-0.55	31	84.9

ETAP PowerStation  
4.0.0C

Page: 11  
Date: 03-16-2008  
SN: KLGCONSULT  
Revision: Base  
Config.: Normal

SYIAR\_49BUS

Study Case: LF

Bus		Voltage		Generation		Motor Load		Static Load		Load Flow					XFMR	
ID	kV	%Mag	Ang.	MW	Mvar	MW	Mvar	MW	Mvar	ID	MW	Mvar	Amp	%PF	% Tap	
										B39	0.83	0.51	29	84.9		
	20.000	94.923	-1.4	0	0	0	0	0.13	0.08	B38	-0.83	-0.51	29	85.0		
										B40	0.70	0.44	25	85.0		
	20.000	94.838	-1.4	0	0	0	0	0	0	B39	-0.70	-0.43	25	85.0		
										B41	0.12	0.08	4	85.0		
										B43	0.58	0.36	20	85.0		
	20.000	94.822	-1.4	0	0	0	0	0.06	0.03	B40	-0.12	-0.08	4	85.0		
										B42	0.07	0.04	2	85.0		
	20.000	94.813	-1.4	0	0	0	0	0.07	0.04	B41	-0.07	-0.04	2	85.0		
	20.000	94.777	-1.4	0	0	0	0	0	0	B40	-0.58	-0.36	20	85.0		
										B44	0.43	0.26	15	85.0		
										B47	0.15	0.09	5	85.0		
	20.000	94.740	-1.4	0	0	0	0	0	0	B43	-0.43	-0.26	15	85.0		
										B45	0.38	0.24	13	85.0		
										B46	0.05	0.03	1	85.0		
	20.000	94.708	-1.4	0	0	0	0	0.38	0.24	B44	-0.38	-0.24	13	85.0		
	20.000	94.731	-1.4	0	0	0	0	0.05	0.03	B44	-0.05	-0.03	1	85.0		
	20.000	94.740	-1.4	0	0	0	0	0.04	0.03	B43	-0.15	-0.09	5	85.0		
										B48	0.11	0.07	3	85.0		
	20.000	94.713	-1.4	0	0	0	0	0.08	0.05	B47	-0.11	-0.07	3	85.0		
										B49	0.03	0.02	0	85.0		
	20.000	94.705	-1.4	0	0	0	0	0.03	0.02	B48	-0.03	-0.02	0	85.0		

voltage regulated bus ( voltage controlled or swing type machine connected to it)

bus with a load mismatch of more than 0.1 MVA

**Tabel 1-1**  
**Perbandingan Perhitungan Program ETAP dan MATLAB Sebelum Optimasi**

No. Bus	ETAP		MATLAB	Selisih
1	100.000	1.00000	1	0.0000000
2	99.312	0.99312	0.9932	0.0000827
3	98.948	0.98948	0.98961	0.0001331
4	98.896	0.98896	0.98909	0.0001291
5	98.855	0.98855	0.98867	0.0001238
6	98.833	0.98833	0.98846	0.0001290
7	98.848	0.98848	0.98861	0.0001260
8	98.868	0.98868	0.98883	0.0001455
9	98.860	0.98860	0.98874	0.0001402
10	98.419	0.98419	0.98441	0.0002225
11	97.897	0.97897	0.97917	0.0002041
12	97.628	0.97628	0.97648	0.0001967
13	97.270	0.97270	0.97289	0.0001935
14	97.086	0.97086	0.97105	0.0001875
15	97.078	0.97078	0.97097	0.0001858
16	96.705	0.96705	0.96722	0.0001661
17	96.422	0.96422	0.96438	0.0001645
18	95.952	0.95952	0.95966	0.0001420
19	95.945	0.95945	0.9596	0.0001489
20	95.934	0.95934	0.95948	0.0001401
21	95.940	0.95940	0.95955	0.0001470
22	95.937	0.95937	0.95951	0.0001439
23	95.933	0.95933	0.95948	0.0001486
24	95.926	0.95926	0.95941	0.0001494
25	95.929	0.95929	0.95944	0.0001504
26	95.758	0.95758	0.95772	0.0001392
27	95.481	0.95481	0.95494	0.0001304
28	95.459	0.95459	0.95472	0.0001317
29	95.030	0.95030	0.95041	0.0001112
30	95.015	0.95015	0.95025	0.0001028
31	95.009	0.95009	0.9502	0.0001083
32	94.999	0.94999	0.95009	0.0001046
33	94.944	0.94944	0.94954	0.0000994
34	94.860	0.94860	0.94869	0.0000901
35	94.994	0.94994	0.95005	0.0001052
36	94.990	0.94990	0.95001	0.0001087
37	94.984	0.94984	0.94995	0.0001116
38	94.977	0.94977	0.94988	0.0001071
39	94.923	0.94923	0.94934	0.0001087
40	94.838	0.94838	0.94849	0.0001061
41	94.822	0.94822	0.94832	0.0001047
42	94.813	0.94813	0.94823	0.0001029
43	94.777	0.94777	0.94788	0.0001063
44	94.740	0.94740	0.9475	0.0001012
45	94.708	0.94708	0.94718	0.0000987
46	94.731	0.94731	0.94741	0.0000963
47	94.740	0.94740	0.94751	0.0001070
48	94.713	0.94713	0.94723	0.0001012
49	94.705	0.94705	0.94716	0.0001063

**Jumlah Selisih ETAP dengan MATLAB : 0.0062904**  
**Selisih rata-rata ETAP dengan MATLAB : .00012838**

**ETAP PowerStation**  
4.0.0C

Page: 9  
Date: 03-16-2008  
SN: KLGCONSULT  
Revision: Base  
Config.: Normal

Study Case: LF

SYIAR\_49BUS

**LOAD FLOW REPORT**

Bus		Voltage		Generation		Motor Load		Static Load		Load Flow					XFMR	
ID	kV	%Mag	Ang.	MW	Mvar	MW	Mvar	MW	Mvar	ID	MW	Mvar	Amp	%PF	% Tap	
	20.000	100.000	0.0	3.10	0.91	0	0	0	0	B2	3.10	0.91	93	95.9		
	20.000	99.486	-0.3	0	0	0	0	0.07	0.05	B1	-3.08	-0.90	93	96.0		
										B3	3.01	0.85	90	96.2		
	20.000	99.217	-0.4	0	0	0	0	0	0	B2	-3.00	-0.84	90	96.3		
										B4	0.65	0.40	22	85.0		
										B8	2.35	0.44	69	98.3		
	20.000	99.165	-0.4	0	0	0	0	0.07	0.05	B3	-0.65	-0.40	22	85.0		
										B5	0.58	0.36	19	85.0		
	20.000	99.124	-0.4	0	0	0	0	0	0	B4	-0.58	-0.36	19	85.0		
										B6	0.47	0.29	16	85.0		
										B7	0.10	0.06	3	85.0		
	20.000	99.102	-0.4	0	0	0	0	0.47	0.29	B5	-0.47	-0.29	16	85.0		
	20.000	99.118	-0.4	0	0	0	0	0.10	0.06	B5	-0.10	-0.06	3	85.0		
	20.000	99.164	-0.4	0	0	0	0	0	0	B3	-2.35	-0.44	69	98.3		
										B9	0.09	0.06	3	85.0		
										B10	2.26	0.38	66	98.6		
	20.000	99.156	-0.4	0	0	0	0	0.09	0.06	B8	-0.09	-0.06	3	85.0		
	20.000	98.872	-0.6	0	0	0	0	0.15	0.09	B8	-2.26	-0.37	66	98.7		
										B11	2.11	0.28	62	99.1		
	20.000	98.546	-0.8	0	0	0	0	0.09	0.05	B10	-2.10	-0.27	62	99.2		
										B12	2.02	0.22	59	99.4		
	20.000	98.384	-0.9	0	0	0	0	0.06	0.04	B11	-2.02	-0.21	59	99.4		
										B13	1.95	0.18	57	99.6		
	20.000	98.172	-1.1	0	0	0	0	0.08	0.05	B12	-1.95	-0.17	57	99.6		
										B14	1.87	0.12	55	99.8		
	20.000	98.067	-1.2	0	0	0	0	0	0	B13	-1.86	-0.12	55	99.8		
										B15	0.03	0.02	0	85.0		
										B16	1.84	0.10	54	99.9		
	20.000	98.059	-1.2	0	0	0	0	0.03	0.02	B14	-0.03	-0.02	0	85.0		
	20.000	97.852	-1.3	0	0	0	0	0.02	0.01	B14	-1.84	-0.09	54	99.9		
										B17	1.81	0.08	53	99.9		
	20.000	97.694	-1.5	0	0	0	0	0.03	0.02	B16	-1.81	-0.08	53	99.9		
										B18	1.78	0.06	52	99.9		
	20.000	97.437	-1.7	0	0	0	0	0	0	B17	-1.78	-0.05	52	100.0		
										B19	0.15	0.09	5	85.0		
										B26	1.63	-0.04	48	-100.0		

ETAP PowerStation

4.0.0C

Page: 10

Date: 03-16-2008

SN: KIGCONSULT

Revision: Base

Config: Normal

SYLAR\_49BUS

Study Case: LF

Bus ID	Voltage			Generation		Motor Load		Static Load		ID	Load Flow			XFMR %Tap
	kV	%Reg	Ang	MW	MVar	MW	MVar	MW	MVar		MW	MVar	Amp	
20000	97.430	-1.7		0	0	0	0	0	0	B18	-0.15	-0.09	5	85.0
										B20	0.06	0.04	2	85.0
										B21	0.09	0.05	3	85.0
20000	97.419	-1.7		0	0	0	0	0.06	0.04	B19	-0.06	-0.04	2	85.0
20000	97.425	-1.7		0	0	0	0	0	0	B19	-0.09	-0.05	3	85.0
										B22	0.04	0.02	1	85.0
										B23	0.05	0.03	1	85.0
20000	97.421	-1.7		0	0	0	0	0.04	0.02	B21	-0.04	-0.02	1	85.0
20000	97.418	-1.7		0	0	0	0	0	0	B21	-0.05	-0.03	1	85.0
										B24	0.02	0.01	0	85.0
										B25	0.03	0.02	0	85.0
20000	97.411	-1.7		0	0	0	0	0.02	0.01	B23	-0.02	-0.01	0	85.0
20000	97.414	-1.7		0	0	0	0	0.03	0.02	B23	-0.03	-0.02	0	85.0
20000	97.339	-1.8		0	0	0	0	0.07	0.04	B18	-1.63	0.04	48	-100.0
										B27	1.56	-0.08	46	-99.9
20000	97.206	-1.9		0	0	0	0	0	0	B26	-1.56	0.09	46	-99.9
										B28	0.07	0.04	2	85.0
										B29	1.49	-0.13	44	-99.6
20000	97.184	-1.9		0	0	0	0	0.07	0.04	B27	-0.07	-0.04	2	85.0
20000	97.002	-2.1		0	0	0	0	0.06	0.04	B27	-1.49	0.13	44	-99.6
										B30	1.43	-0.17	42	-99.3
20000	96.995	-2.2		0	0	0	0	0	0	B29	-1.43	0.17	42	-99.3
										B31	0.43	0.27	15	85.0
										B36	0.99	-0.44	32	-91.3
20000	96.990	-2.2		0	0	0	0	0.03	0.02	B30	-0.43	-0.27	15	85.0
										B32	0.40	0.25	14	85.0
20000	96.979	-2.2		0	0	0	0	0	0	B31	-0.40	-0.25	14	85.0
										B33	0.35	0.22	12	85.0
										B35	0.05	0.03	1	85.0
20000	96.973	-2.2		0	0	0	0	0.05	0.03	B32	-0.35	-0.22	12	85.0
										B34	0.30	0.19	10	85.0
20000	96.837	-2.2		0	0	0	0	0.30	0.19	B33	-0.30	-0.19	10	85.0
20000	96.975	-2.2		0	0	0	0	0.05	0.03	B32	-0.05	-0.03	1	85.0
20000	96.991	-2.2		0	0	0	0	0	0	B30	-0.99	0.44	32	-91.3
										B37	0.06	0.04	2	85.0
										B38	0.93	-0.48	31	-88.8
20000	96.985	-2.2		0	0	0	0	0.06	0.04	B36	-0.06	-0.04	2	85.0
20000	96.990	-2.2		0	0	0	0	0.07	-1.02	B36	-0.93	0.48	31	-88.8

ETAP PowerStation

4.0.0C

Study Case: LF

Page: 11  
 Date: 03-16-2008  
 SN: KLGCONSULT  
 Revision: Base  
 Config.: Normal

SYIAR\_49BUS

Bus		Voltage		Generation		Motor Load		Static Load		Load Flow					XFMR	
ID	kV	%Mag	Ang.	MW	Mvar	MW	Mvar	MW	Mvar	ID	MW	Mvar	Amp	%PF	% Tap	
										B39	0.86	0.54	30	84.9		
20.000	96.934	-2.2		0	0	0	0	0.13	0.08	B38	-0.86	-0.54	30	85.0		
										B40	0.73	0.45	25	85.0		
20.000	96.848	-2.2		0	0	0	0	0	0	B39	-0.73	-0.45	25	85.0		
										B41	0.13	0.08	4	85.0		
										B43	0.60	0.37	21	85.0		
20.000	96.831	-2.2		0	0	0	0	0.06	0.04	B40	-0.13	-0.08	4	85.0		
										B42	0.07	0.04	2	85.0		
20.000	96.822	-2.2		0	0	0	0	0.07	0.04	B41	-0.07	-0.04	2	85.0		
20.000	96.786	-2.2		0	0	0	0	0	0	B40	-0.60	-0.37	21	85.0		
										B44	0.45	0.28	15	85.0		
										B47	0.16	0.10	5	85.0		
20.000	96.747	-2.2		0	0	0	0	0	0	B43	-0.45	-0.28	15	85.0		
										B45	0.40	0.25	13	85.0		
										B46	0.05	0.03	1	85.0		
20.000	96.715	-2.3		0	0	0	0	0.40	0.25	B44	-0.40	-0.25	13	85.0		
20.000	96.739	-2.2		0	0	0	0	0.05	0.03	B44	-0.05	-0.03	1	85.0		
20.000	96.748	-2.2		0	0	0	0	0.04	0.03	B43	-0.16	-0.10	5	85.0		
										B48	0.11	0.07	4	85.0		
20.000	96.720	-2.3		0	0	0	0	0.09	0.05	B47	-0.11	-0.07	4	85.0		
										B49	0.03	0.02	0	85.0		
20.000	96.712	-2.3		0	0	0	0	0.03	0.02	B48	-0.03	-0.02	0	85.0		

voltage regulated bus ( voltage controlled or swing type machine connected to it)



bus with a load mismatch of more than 0.1 MVA

**Tabel 1-2**  
**Perbandingan Perhitungan Program ETAP dan MATLAB Setelah Optimasi**

No.Bus	ETAP		MATLAB	Selisih
1	100.000	1.00000	1	0.000000
2	99.486	0.99486	0.99486	0.000011
3	99.217	0.99217	0.99217	0.000008
4	99.165	0.99165	0.99165	0.0000034
5	99.124	0.99124	0.99124	0.0000024
6	99.102	0.99102	0.99102	0.0000018
7	99.118	0.99118	0.99117	0.0000053
8	99.164	0.99164	0.99164	0.0000038
9	99.156	0.99156	0.99155	0.0000089
10	98.872	0.98872	0.9887	0.0000189
11	98.546	0.98546	0.98532	0.0001418
12	98.384	0.98384	0.98362	0.0002169
13	98.172	0.98172	0.9814	0.0003157
14	98.067	0.98067	0.98029	0.0003776
15	98.059	0.98059	0.98022	0.0003686
16	97.852	0.97852	0.97802	0.0005003
17	97.694	0.97694	0.97635	0.0005904
18	97.437	0.97437	0.97362	0.0007484
19	97.430	0.97430	0.97355	0.0007505
20	97.419	0.97419	0.97344	0.0007475
21	97.425	0.97425	0.97351	0.0007416
22	97.421	0.97421	0.97347	0.0007442
23	97.418	0.97418	0.97344	0.0007388
24	97.411	0.97411	0.97337	0.0007371
25	97.414	0.97414	0.9734	0.0007364
26	97.339	0.97339	0.97256	0.0008329
27	97.206	0.97206	0.97111	0.0009506
28	97.184	0.97184	0.97089	0.0009453
29	97.002	0.97002	0.96885	0.0011678
30	96.995	0.96995	0.96878	0.0011729
31	96.990	0.96990	0.96873	0.0011663
32	96.979	0.96979	0.96862	0.0011678
33	96.923	0.96923	0.96808	0.0011516
34	96.837	0.96837	0.96725	0.0011234
35	96.975	0.96975	0.96858	0.0011663
36	96.991	0.96991	0.96872	0.0011919
37	96.985	0.96985	0.96866	0.0011877
38	96.990	0.96990	0.96869	0.0012080
39	96.934	0.96934	0.96816	0.0011849
40	96.848	0.96848	0.96733	0.0011495
41	96.831	0.96831	0.96716	0.0011474
42	96.822	0.96822	0.96708	0.0011373
43	96.786	0.96786	0.96673	0.0011263
44	96.747	0.96747	0.96636	0.0011134
45	96.715	0.96715	0.96605	0.0010993
46	96.739	0.96739	0.96627	0.0011167
47	96.748	0.96748	0.96636	0.0011178
48	96.720	0.96720	0.9661	0.0010978
49	96.712	0.96712	0.96602	0.0011011

**Jumlah Selisih ETAP dengan MATLAB : 0.0353264**  
**Selisih rata-rata ETAP dengan MATLAB :0.00072095**



*Laporan Beban Engkuran UPT*   
*Bangkit*  
*Listing Program Metode Hybrid*   
*B.H.M*

*Lampiran 2*



LAPORAN PENGURUAN BEBAN UPT BANGSIL  
 BULAN JULI 2007

NO DUN	ALAMAT	KAPA GRTM	SISTEM TBAJUAN	TOL / BULAN PENGU KURAN	AMPERE												BEBAN PADA TITIK			BEBAN TITIK			KIRI MAYAN																
					LINE A			LINE B			LINE C			LINE D			R	T	N	R	T	N																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	DA KEMEP	100	20	400	1602/2007	27	47	66	39	-	-	-	-	-	50	63	66	25	-	-	-	77	111	102	64	231	232	228	78.02	78.52	AB87C1	222	97.37	C169	222	97.37	REMON		
42	WANONG BRT.	180	20	400	1602/2007	-	-	-	-	-	14	18	9	6	-	-	-	-	74	81	107	48	90	63	111	43	232	232	231	87.89	42.43	B1	220	95.24	BCC17	220	95.24	REMON	
43	PAIDREJED	180	20	400	2302/2007	-	-	-	-	-	103	127	124	30	-	-	-	-	30	16	16	18	183	130	130	33	224	224	224	91.13	91.34	B1A4D2A7	220	98.21	D2	220	98.21	REMON	
44	JL ALUNG TIMUR	180	20	400	2302/2007	-	-	-	-	-	-	-	-	-	138	133	113	36	-	-	-	158	133	113	36	233	234	234	88.68	85.42	A683	222	94.87	C1	222	94.87	REMON		
45	JL KARTINI	200	20	400	2308/2007	-	-	-	-	-	40	61	48	1	-	-	-	-	211	178	186	66	231	206	241	67	234	225	227	161.10	90.60	D1A6	220	98.92	D1A6	220	98.92	REMON	
46	DA KRSEKIAN	200	20	400	2308/2007	89	35	69	42	-	-	-	-	-	124	92	175	42	-	-	-	213	127	233	64	232	232	132.32	91.19	A6	222	94.87	C9	222	94.87	REMON			
47	JL PANTURA	180	20	400	2302/2007	-	-	-	-	-	60	64	26	17	-	-	-	-	66	70	39	62	165	74	67	229	228	232	81.75	61.09	DAATD2	222	96.69	-	222	96.69	REMON		
48	LUMUMBEL	200	20	400	1602/2007	-	-	-	-	-	69	80	130	35	-	-	-	-	100	170	69	84	169	242	226	70	227	227	144.87	73.44	B1A21	220	97.78	DCC035	220	97.78	REMON		
49	KEMALOKO	180	20	400	2202/2007	-	-	-	-	-	132	85	48	24	-	-	-	-	60	104	69	31	168	189	134	65	227	226	234	120.32	75.20	B1A1A0D2	220	94.02	DA3388	220	94.02	REMON	
50	DS NAYANGKING	180	20	400	2202/2007	-	-	-	-	-	77	86	135	23	-	-	-	-	61	49	56	44	136	145	191	70	223	227	109.46	68.41	B4A397	220	98.66	DA4A86	220	98.66	REMON		
51	ND RUMIN UTJ.	180	20	400	2307/2007	-	-	-	-	-	65	67	43	32	-	-	-	-	103	62	65	6	162	133	102	69	223	225	224	92.14	67.99	B17	221	98.66	DAC1	221	98.66	REMON	
52	DA KRSEKIAN	180	20	400	2302/2007	-	-	-	-	-	-	-	-	-	-	-	-	-	130	168	176	64	130	169	138	58	224	227	104.64	67.64	B4A3	222	97.37	DCC1	222	97.37	REMON		
53	KALIKUNTING	180	20	400	1602/2007	-	-	-	-	-	-	-	-	-	-	-	-	-	116	108	136	55	116	108	136	55	231	231	228	83.83	65.49	C1096	223	97.38	-	223	97.38	REMON	
54	RACI	200	20	400	2302/2007	-	-	-	-	-	44	71	66	32	-	-	-	-	65	68	73	10	105	143	140	43	230	234	230	89.60	44.90	B0A5	220	95.65	-	220	95.65	REMON	
55	KEDANTEN	180	20	400	1602/2007	-	-	-	-	-	0	71	37	45	-	-	-	-	125	128	180	45	128	189	102	45	230	234	231	114.85	73.03	D2A8	221	95.67	DCC17	220	95.24	REMON	
56	DS POOLAR	180	20	400	2202/2007	-	-	-	-	-	121	62	134	42	-	-	-	-	14	15	1	16	135	77	135	59	231	232	231	80.13	60.08	D12A8	221	95.67	-	221	95.67	REMON	
57	DALELUK	200	20	400	1602/2007	184	113	90	77	-	-	-	-	-	30	29	39	10	148	117	62	20	162	143	99	48	231	234	230	68.60	62.00	B710	220	93.62	-	220	93.62	REMON	
58	DS BACI MAMUK	180	20	400	2302/2007	-	-	-	-	-	-	-	-	-	-	-	-	-	144	144	148	70	227	230	230	227	230	230	66.66	63.85	A1	220	95.66	CA1BCCB3	220	95.66	REMON		
59	MOJOPARON	180	20	400	0406/2007	40	31	39	38	-	-	-	-	-	102	63	100	35	-	-	-	116	117	161	76	228	232	235	117.78	78.52	A1B1	222	94.47	C2B2	222	94.47	REMON		
60	DA KEMALOKO	180	20	400	12/21/2007	30	22	21	2	-	2	6	1	13	-	-	-	-	37	62	66	21	116	66	110	30	228	229	229	72.66	43.18	B1C0B7	222	97.37	DA3A83	222	97.37	REMON	
61	MOJOKOPAK	3360	20	400	2204/2007	64	104	66	-	-	-	-	-	-	28	30	31	7	-	-	-	111	134	128	30	227	225	226	66.37	67.68	A10B2	220	97.78	C4	220	97.78	REMON		
62	DS ROHOKERUH	180	20	400	2402/2007	70	62	64	30	-	-	-	-	-	4	10	16	01	-	-	-	74	110	111	32	232	230	232	68.13	49.66	A2BCC3	223	98.12	C1B4	223	98.12	REMON		
63	PULANGARI	3360	20	400	2402/2007	69	45	67	32	-	-	-	-	-	111	67	76	40	-	-	-	212	117	161	76	228	232	235	117.78	78.52	A1B1	222	94.47	C2B2	222	94.47	REMON		
64	KWANGARI	3360	20	400	2302/2007	01	79	54	16	-	-	-	-	-	22	21	60	13	-	-	-	110	100	127	29	232	237	238	60.60	61.89	B4C3A4	220	96.49	DAATB2	220	96.49	REMON		
65	DS KRABDEMAN	3360	20	400	2302/2007	01	79	54	16	-	-	-	-	-	22	21	60	13	-	-	-	110	100	127	29	232	237	238	60.60	61.89	B19C1	222	94.07	D1C4	222	94.07	REMON		
66	ND RUMIN TENG	3323	20	400	0406/2007	-	-	-	-	-	7	8	-	-	-	-	-	-	19	29	19	13	19	29	27	25	232	232	232	17.32	28.69	A8D6	223	96.12	DA3A92	223	96.12	REMON	
67	WIKOTYOROGO	200	20	400	1602/2007	183	110	70	74	-	-	-	-	-	112	79	110	24	-	-	-	268	188	160	99	231	234	232	146.10	73.09	A18A1A1B7	222	98.65	C9	222	98.65	REMON		
68	ND RUMIN SEL.	250	20	400	2302/2007	64	89	89	38	-	-	-	-	-	37	45	67	-	-	-	-	118	133	160	38	228	229	229	64.82	47.11	A18A1A1B7	222	98.65	C9	222	98.65	REMON		
69	DS DAUDAYANG	180	20	400	0804/2007	43	121	59	69	-	-	-	-	-	108	120	127	8	164	156	149	32	230	230	232	107.62	67.40	B1A5C3	223	96.12	D1C3	223	96.12	REMON					
70	DS GOMBAYEK	250	20	400	1604/2007	60	93	167	50	-	-	-	-	-	153	74	64	69	-	-	-	218	189	278	113	216	219	217	101.72	78.96	BCCD5	220	101.39	D4	220	101.39	REMON		
71	PEMA KALIREJO	160	20	400	2302/2007	-	-	-	-	-	66	40	130	20	-	-	-	-	148	63	60	63	124	102	141	141	228	228	228	124.09	78.31	B1C8B4A2	221	98.81	C1B7	221	98.81	REMON	
72	Pem. POKROEN	90	20	400	2205/2007	10	27	4	13	-	-	-	-	-	23	35	17	32	-	-	-	42	62	42	21	30	227	225	228	82.87	67.73	A2B7	220	96.49	C2B10	220	96.49	REMON	
73	GEMERKAWARU	3500	20	400	1702/2007	01	47	104	71	-	-	-	-	-	66	67	62	18	-	-	-	132	133	154	74	227	229	226	98.79	64.81	A4BCCB33	220	97.38	C4	220	97.38	REMON		
74	DAWANGAL	50	20	400	1702/2007	-	-	-	-	-	14	28	13	18	-	-	-	-	28	39	30	20	39	61	97	28	228	228	228	91.64	63.28	ASD2A7	223	97.38	C13	223	97.38	REMON	
75	LATER THUR	100	20	400	2302/2007	-	-	-	-	-	12	2	22	33	-	-	-	-	49	72	91	33	65	76	116	60	229	230	230	88.86	65.69	A10	222	94.47	C4C2	222	94.47	REMON	
76	DA TAMPUNG	200	20	400	2306/2007	-	-	-	-	-	66	68	111																										







## Coba While

```
%coba while
clc
pos=zeros(4,1);
min=1;
max=100;
for i=1:4
    pos(i)=round(min+rand*(max-min));
    cek=1;
    while cek==1
        if i==1
            break;
        end
        for j=1:(i-1)
            if pos(j)==pos(i)
                pos(i)=round(min+rand*(max-min));
                cek=1;
            else
                cek=0;
            end
        end
    end
end
disp(pos);
pos1=round(min+rand*(max-min));
pos2=round(min+rand*(max-min));
while pos2==pos1
    pos2=round(min+rand*(max-min));
end
disp(pos1);
disp(pos2);
```

## Decodebin1 ToReal

```
function [val]=DecodeBin1ToReal(chrom1,min,max)
nparam=length(chrom1);
sum=0;
powerof2=1;
sa=2^nparam-1;
for i=nparam:-1:1
    if chrom1(i)==1
        sum=sum+powerof2;
    end
    powerof2=powerof2*2;
end
sum=sum/sa;
val=min+sum*(max-min);
```

## Decodebin2 ToReal

```
function [val]=DecodeBin2ToReal(chrom2,min,max)
x=chrom2(:,1);
nparam=length(x);
x=chrom2(1,:);
nlength=length(x);
val=zeros(nparam,1);
```

```

chrom1=zeros(nlength,1);
for i=1:nparam
    for j=1:nlength
        chrom1(j)=chrom2(i,j);
    end
    val(i)=DecodeBin1ToReal(chrom1,min(i),max(i));
end

```

## DecodeReal ToBin1

```

function [chrom1]=DecodeRealToBin1(val,min,max,nlength)
chrom1=zeros(nlength,1);
sa=2^nlength-1;
sum=(val-min)/(max-min);
value=round(sum*sa);
chrom1=zeros(nlength,1);
for i=1:nlength
    chrom1(i)=0;
end
i=nlength;
while value>0
    cek=mod(value,2);
    if cek==1
        chrom1(i)=1;
    else
        chrom1(i)=0;
    end
    value=fix(value/2);
    i=i-1;
end

```

## DecodeReal ToBin2

```

function [chrom2]=DecodeRealToBin2(val,min,max,nlength)
nparam=length(val);
chrom2=zeros(nparam,nlength);
chrom1=zeros(nlength,1);
for i=1:nparam
    chrom1=DecodeRealToBin1(val(i),min(i),max(i),nlength);
    for j=1:nlength
        chrom2(i,j)=chrom1(j);
    end
end

```

## Decode Saluran

```

function [Nsal,Z,Lc,Tr,Ts,CapSal]=DecodeSaluran(Nbus,Sal)
X=Sal(:,1);
Nsal=length(X);
Z=zeros(Nbus,Nbus);
Lc=zeros(Nbus,Nbus);
Tr=zeros(Nbus,Nbus);
Ts=zeros(Nbus,Nbus);
CapSal=zeros(Nbus,Nbus);
for i=1:Nsal
    dari=round(Sal(i,1));
    ke=round(Sal(i,2));

```



```

Z(dari,ke)=complex(Sal(i,3),Sal(i,4));
Lc(dari,ke)=Sal(i,5);
Tr(dari,ke)=Sal(i,6);
Ts(dari,ke)=complex(Sal(i,7),Sal(i,8));
CapSal(dari,ke)=Sal(i,9);
CapSal(ke,dari)=Sal(i,9);
end

```

## Decode Bus

```

function [Nbus,Beban,Cap]=DecodeBus(Bus)
X=Bus(:,1);
Nbus=length(X);
Beban=zeros(Nbus,7);
Cap=zeros(Nbus,1);
for i=1:Nbus
    Beban(i,1)=Bus(i,1);
    Beban(i,2)=Bus(i,2);
    Beban(i,3)=Bus(i,3);
    Beban(i,4)=Bus(i,4);
    Beban(i,5)=Bus(i,5);
    Beban(i,6)=Bus(i,6);
    Beban(i,7)=Bus(i,8);
    Cap(i,1)=Bus(i,7);
end

```

## Cetak Newton Raphson

```

function
[Nbus]=CetakNewtonRaphson(Nbus,ite,selang,V,Sg,SL,Z,Alir,RugiS,SumG,SumL,Rr)
disp(' ');
disp(['          Hasil Perhitungan Aliran Daya']);
disp(['          Dengan Metode Newton-Raphson']);
disp(' ');
disp(['Jumlah Bus    = ' num2str(Nbus)]);
disp(['Jumlah iterasi = ' num2str(ite)]);
disp(['Waktu Hitung   = ' num2str(selang) ' detik']);
disp(' ');
disp(['-----']);
disp([' Bus      Tegangan      Pembangkitan      Pembebanan      ']);
disp(['      (pu)          kw      kvar      kw      kvar      ']);
disp(['-----']);
Sa=[(1:Nbus)' real(V) imag(V) real(Sg) imag(Sg) real(SL) imag(SL)];
fprintf('%5.0f %8.5f %8.5f %8.3f %8.3f %8.3f %8.3f\n',Sa);
disp(['-----']);
disp(' ');
disp(['Aliran Daya Antar Saluran']);
disp(' ');
s=0;
disp(['-----']);
disp([' Bus      Daya      Bus      Daya      ']);
disp([' from to   kw      kvar   from to   kw      kvar      ']);
disp(['-----']);

for i=1:Nbus
    for j=1:Nbus
        x=imag(RugiS(i,j));
    end
end

```

```

    if x~=0
        S=[i j real(Alir(i,j)) imag(Alir(i,j)) j i real(Alir(j,i)) imag(Alir(j,i))];
        fprintf(' %3.0f %3.0f %9.3f %9.3f %3.0f %3.0f %9.3f %9.3fn',S');
    end
end
end
disp(['-----']);
disp(' ');
disp(['Rugi Daya Antar Saluran']);
disp(' ');
s=0;
disp(['-----']);
disp([' Bus      Rugi      ']);
disp([' from to   kw      kvar  ']);
disp(['-----']);
for i=1:Nbus
    for j=1:Nbus
        x=imag(Z(i,j));
        if x~=0
            S=[i j real(RugiS(i,j)) imag(RugiS(i,j))];
            fprintf(' %3.0f %3.0f %8.3f %8.3fn',S');
        end
    end
end
disp(['-----']);
disp(' ');
disp(['Jumlah Pembangkitan = ' num2str(SumG) ' kVA']);
disp(['Jumlah Pembebanan   = ' num2str(SumL) ' kVA']);
disp(['Jumlah Rugi-Rugi     = ' num2str(Rr) ' kVA']);
disp(['Jumlah Iterasi       = ' num2str(ite)]);
disp(' ');

```

## Newton Polar

```

function
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(N,z,Lc,Tr,Ts,Cap,Beban,Pbase
,Zbase,TypSal)
%Loadflow Metode Newton Raphson Polar
%Copyright (c) 2002 by Ugro Software

%Baca Data-----
ite=0;
[V,Sg,SL,TypeBus]=BacaData(N,Beban,Pbase);
[z,Lc]=UpdateSaluran(N,z,Lc,Zbase,TypSal);
[Y]=Admitansi(N,z,Lc,Tr,Ts,Cap);
for i=1:15
    [dS]=PowerMismatch(N,Y,Lc,Sg,SL,TypeBus,V);
    [Cek]=CekKonvergen(N,dS,TypeBus);
    if Cek==1
        break;
    end
    [Jq]=Jaqobian(N,V,Y,TypeBus);
    [V]=UpdateTegangan(N,V,Jq,dS,TypeBus);
    ite=ite+1;
end

[Vr]=UbahRectangular(N,V);
[Alir]=AlirDaya(N,Vr,Y,Lc,Pbase);
[Sg]=Daya(N,Vr,TypeBus,Sg,SL,Y,Pbase,Alir);

```

```

SL=Pbase*SL;
[SumG,SumL,Rr]=RugiRugi(N,Sg,SL);
[RugiS]=RugiSaluran(N,z,Alir);

```

```

function [V,Sg,SL,TypeBus]=BacaData(N,Beban,Pbase)
V=zeros(N,1);
Sg=zeros(N,1);
SL=zeros(N,1);
TypeBus=zeros(N,1);
for i=1:N
    V(i)=complex(Beban(i,1),Beban(i,2));
    Sg(i)=complex(Beban(i,3)/Pbase,Beban(i,4)/Pbase);
    SL(i)=complex(Beban(i,5)/Pbase,Beban(i,6)/Pbase);
    TypeBus(i)=Beban(i,7);
end

```

```

function [z,Lc]=UpdateSaluran(N,z,Lc,Zbase,TypSal)
if TypSal==2
    for i=1:N
        for j=1:N
            r=real(z(i,j));
            x=imag(z(i,j));
            if x~=0
                z(i,j)=complex(r/Zbase,x/Zbase);
                Lc(i,j)=Lc(i,j)/Zbase;
            end
        end
    end
end
end

```

```

function [Y]=Admitansi(N,z,Lc,Tr,Ts,Cap)
Y=zeros(N,N);
C=zeros(N,N);
for i=1:N
    for j=1:N
        xa=imag(z(i,j));
        if xa~=0
            z(j,i)=z(i,j);
            Lc(j,i)=Lc(i,j);
        end
    end
end
for i=1:N
    for j=1:N
        xa=imag(z(i,j));
        if xa~=0
            C(i,j)=1/z(i,j);
        end
    end
end
for i=1:N
    for j=1:N
        if i==j
            sumA=0;
            sumB=0;
            for k=1:N
                if i~=k
                    sumA=sumA+real(C(i,k));
                    sumB=sumB+imag(C(i,k))+Lc(i,k);
                end
            end

```

```

    end
    if Cap(i)~=0
        sumB=sumB+Cap(i);
    end
    Y(i,j)=complex(sumA,sumB);
else
    Y(i,j)=complex(-real(C(i,j)), -imag(C(i,j)));
end
end
end
for i=1:N
    for j=1:N
        if Tr(i,j)~=0
            Tr(i,j)=1/Tr(i,j);
            Y(i,i)=Y(i,i)-C(i,j);
            Y(i,i)=Y(i,i)+C(i,j)*Tr(i,j)^2;
            Y(i,j)=Tr(i,j)*Y(i,j);
            Y(j,i)=Y(i,j);
        end
    end
end
for i=1:N
    for j=1:N
        ax=real(Ts(i,j));
        if ax~=0
            ax=real(Ts(i,j))*cos(imag(Ts(i,j)));
            bx=real(Ts(i,j))*sin(imag(Ts(i,j)));
            ts=complex(ax,bx);
            Y(i,j)=-C(i,j)/ts;
            Y(j,i)=-C(i,j)/conj(ts);
            Y(i,i)=Y(i,i)-C(i,j);
            Y(i,i)=Y(i,i)+C(i,j)/(ax^2+bx^2);
        end
    end
end
for i=1:N
    if Cap(i)~=0
        re=real(Y(i,i));
        im=imag(Y(i,i))+Cap(i);
        Y(i,i)=complex(re,im);
    end
end
end

```

function [dS]=PowerMismatch(N,Y,Lc,Sg,SL,TypBus,V)

%Fungsi ini untuk menghitung selisih daya

PV=0;

for i=1:N

if TypBus(i)==2

PV=PV+1;

end

end

n=N-1+N-1-PV;

dS=zeros(n,1);

Pc=zeros(N,1);

Qc=zeros(N,1);

sp=0;

sq=0;

for i=1:N

if TypBus(i)~=1

sum1=0;

```

for j=1:N
    Gij=real(Y(i,j));
    Bij=imag(Y(i,j));
    Ui=real(V(i));
    Uj=real(V(j));
    dij=imag(V(i))-imag(V(j));
    sum1=sum1+Ui*Uj*(Gij*cos(dij)+Bij*sin(dij));
end
Pc(i)=sum1;
end
if TypBus(i)==3
    sum2=0;
    for j=1:N
        Gij=real(Y(i,j));
        Bij=imag(Y(i,j));
        Ui=real(V(i));
        Uj=real(V(j));
        dij=imag(V(i))-imag(V(j));
        sum2=sum2+Ui*Uj*(Gij*sin(dij)-Bij*cos(dij));
    end
    Qc(i)=sum2;
end
end
sp=0;
sq=N-1;
for i=1:N
    if TypBus(i)~=1
        sp=sp+1;
        dS(sp)=real(Sg(i))-real(SL(i))-Pc(i);
    end
    if TypBus(i)==3
        sq=sq+1;
        dS(sq)=imag(Sg(i))-imag(SL(i))-Qc(i);
    end
end
end

```

```
function [Jq]=Jaqobian(N,V,Y,TypBus)
```

```
PV=0;
```

```
for i=1:N
```

```
    if TypBus(i)==2
```

```
        PV=PV+1;
```

```
    end
```

```
end
```

```
NJq=N-1+N-1-PV;
```

```
Jq=zeros(NJq,NJq);
```

```
%Pembentukan Jaqobian H
```

```
row=0;
```

```
for i=1:N
```

```
    if TypBus(i)~=1
```

```
        row=row+1;
```

```
        col=0;
```

```
        for j=1:N
```

```
            if TypBus(j)~=1
```

```
                col=col+1;
```

```
                Ui=real(V(i));
```

```
                di=imag(V(i));
```

```
                Uj=real(V(j));
```

```
                dj=imag(V(j));
```

```
                Gij=real(Y(i,j));
```

```
                Bij=imag(Y(i,j));
```









```

for i=1:N
    if TypeBus(i)==2
        PV=PV+1;
    end
end
n=N-1+N-1-PV;
dV=zeros(n,1);
dV=Jq\dS;
sp=0;
sq=0;
for i=1:N
    if TypeBus(i)~=1
        sp=sp+1;
        V(i)=complex(real(V(i)),imag(V(i))+dV(sp));
    end
end
sq=N-1;
for i=1:N
    if TypeBus(i)==3
        sq=sq+1;
        dt=dV(sq)*real(V(i));
        V(i)=complex(real(V(i))+dt,imag(V(i)));
    end
end

```

```

function [Cek]=CekKonvergen(N,dS,TypeBus)
Cek=1;
PV=0;
for i=1:N
    if TypeBus(i)==2
        PV=PV+1;
    end
end
n=N-1+N-1-PV;
Tol=0.0001;
for i=1:n
    if abs(dS(i))>Tol
        Cek=0;
    end
end

```

```

function [Vr]=UbahRectangular(N,V)
Vr=zeros(N,1);
for i=1:N
    Vreal=real(V(i))*cos(imag(V(i)));
    Vimag=real(V(i))*sin(imag(V(i)));
    Vr(i)=complex(Vreal,Vimag);
end

```

```

function [Alir]=AlirDaya(N,V,Y,Lc,Pbase)
Alir=zeros(N,N);
for i=1:N
    for j=1:N
        Lc(j,i)=Lc(i,j);
    end
end
for i=1:N
    for j=1:N
        if j~=i
            Yx=imag(Y(i,j));

```

```

    if Yx~=0
        Ys=complex(-real(Y(i,j)), -imag(Y(i,j)));
        Ls=complex(0, Lc(i,j));
        Al=conj(V(i))*(V(i)-V(j))*Ys+conj(V(i))*V(i)*Ls;
        Alir(i,j)=complex(real(Al)*Pbase, -imag(Al)*Pbase);
    end
end
end
end
end

```

```
function [Sg]=Daya(N,V,Typ,Sg,SL,Y,Pbase,Alir)
```

```

for i=1:N
    sum=0;
    if Typ(i)==1
        for j=1:N
            Al=imag(Y(i,j));
            if Al~=0
                sum=sum+Alir(i,j);
            end
        end
        Sg(i)=sum+SL(i)*Pbase;
    end
    sum=0;
    if Typ(i)==2
        for j=1:N
            sum=sum+(imag(V(i))*(real(V(j))*real(Y(i,j))+imag(V(j))*-imag(Y(i,j)))-
real(V(i))*(imag(V(j))*real(Y(i,j))-real(V(j))*-imag(Y(i,j))));
        end
        Sg(i)=complex(real(Sg(i))*Pbase, (sum+imag(SL(i)))*Pbase);
    end
    if Typ(i)==3
        Cap=imag(Sg(i));
        if Cap~=0
            Sg(i)=Pbase*Sg(i);
        end
    end
end
end
end

```

```
function [SumG,SumL,Rr]=RugiRugi(N,Sg,SL)
```

```

Rr=zeros(N,1);
SumG=0+0i;
SumL=0+0i;
for i=1:N
    SumG=SumG+Sg(i);
    SumL=SumL+SL(i);
end
Rr=SumG-SumL;

```

```
function [RugiS]=RugiSaluran(N,z,Alir)
```

```

RugiS=zeros(N,N);
for i=1:N
    for j=1:N
        Xa=imag(z(i,j));
        if Xa~=0
            RugiS(i,j)=Alir(i,j)+Alir(j,i);
        end
    end
end
end
end

```

## Initial LoadFlow

```
function [fitness]=InitLoadflow()
global Ncap Tlow Tmed Tpea kLow kMed kPea;
global Nbus Z Lc Tr Ts Cap Beban Pbase Zbase TypSal CapSal;
global ke;
[BebanLow,BebanMed,BebanPea]=DecodeBeban(Beban,kLow,kMed,kPea);
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
                                                    Cap,BebanLow,Pbase,Zbase,TypSal);

Plow=real(Rr);
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
                                                    Cap,BebanMed,Pbase,Zbase,TypSal);

Pmed=real(Rr);
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
                                                    Cap,BebanPea,Pbase,Zbase,TypSal);

Ppea=real(Rr);
fitness=ke*(Tlow*Plow+Tmed*Pmed+Tpea*Ppea);

function [BebanLow,BebanMed,BebanPea]=DecodeBeban(Beban,kLow,kMed,kPea)
x=Beban(:,1);
rows=length(x);
x=Beban(1,:);
cols=length(x);
BebanLow=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanLow(i,j)=Beban(i,j);
    end
    BebanLow(i,5)=kLow*Beban(i,5);
    BebanLow(i,6)=kLow*Beban(i,6);
end
BebanMed=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanMed(i,j)=Beban(i,j);
    end
    BebanMed(i,5)=kMed*Beban(i,5);
    BebanMed(i,6)=kMed*Beban(i,6);
end
BebanPea=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanPea(i,j)=Beban(i,j);
    end
    BebanPea(i,5)=kPea*Beban(i,5);
    BebanPea(i,6)=kPea*Beban(i,6);
end
End
```

## Genetic Algorithm

```
function [aMin,aAvg,aMax,BestChrom]=Genetic(MaxGen,PopSize,Pcross,Pmutasi,...
                                           Param,Length,Ka,Ncap)
aMin=zeros(MaxGen,1);
aMax=zeros(MaxGen,1);
aAvg=zeros(MaxGen,1);
[Parent]=InitialIndividu(PopSize,Param,Length);
[Child]=InitialIndividu(PopSize,Param,Length);
[Parent]=InitParent(Parent,PopSize,Param,Length,Ncap,Ka);
```

```

MaxIndi=CariMaxIndividu(PopSize,Parent);
[min,avg,max,SumFitness]=Statistik(PopSize,Parent);
for i=1:MaxGen
    Child=Generasi(PopSize,Param,Length,Pcross,Pmutasi,Parent,...
        Child,SumFitness,Ka);
    [Parent]=FindNewParent(PopSize,Param,Length,Parent,Child);
    [min,avg,max,SumFitness]=Statistik(PopSize,Parent);
    tmpIndi=CariMaxIndividu(PopSize,Parent);
    if MaxIndi.fitness<tmpIndi.fitness
        MaxIndi=GetIndividu(tmpIndi);
    end
    aMin(i)=min;
    aAvg(i)=avg;
    aMax(i)=max;
    fprintf('%2.0f',i);
end
BestChrom=GetBestChrom(MaxIndi);

```

```

function [allele]=GetRandom(min,max)
allele=min+rand*(max-min);

```

```

function [chrom]=GetRandomChrom(Param,Length,Ncap)
chrom=zeros(Param,Length);
for i=1:Param
    for j=2:Length
        rnd=rand;
        if rnd<=0.5
            chrom(i,j)=1;
        else
            chrom(i,j)=0;
        end
    end
end
pos=zeros(Ncap,1);
min=1;
max=Param;
for i=1:4
    pos(i)=round(GetRandom(min,max));
    cek=1;
    while cek==1
        if i==1
            break;
        end
        for j=1:(i-1)
            if pos(j)==pos(i)
                pos(i)=round(GetRandom(min,max));
                cek=1;
            else
                cek=0;
            end
        end
    end
end
end
for i=1:Ncap
    ia=pos(i);
    chrom(ia,1)=1;
end

```

```

function [Individu]=InitialIndividu(PopSize,Param,Length)
for i=1:PopSize

```

```
    Individu(i).chrom=zeros(Param,Length);
    Individu(i).fitness=0;
end
```

```
function [NewIndi]=GetIndividu(OldIndi)
x=OldIndi.chrom(:,1);
rows=length(x);
x=OldIndi.chrom(1,:);
cols=length(x);
NewIndi.chrom=zeros(rows,cols);
NewIndi.fitness=0;
for i=1:rows
    for j=1:cols
        NewIndi.chrom(i,j)=OldIndi.chrom(i,j);
    end
end
NewIndi.fitness=OldIndi.fitness;
```

```
function [MaxIndi]=CariMaxIndividu(PopSize,CurrIndi)
MaxIndi=GetIndividu(CurrIndi(1));
for i=2:PopSize
    if MaxIndi.fitness<CurrIndi(i).fitness
        MaxIndi=GetIndividu(CurrIndi(i));
    end
end
```

```
function [Parent]=InitParent(Parent,PopSize,Param,Length,Ncap,Ka)
for i=1:PopSize
    Parent(i).chrom=GetRandomChrom(Param,Length,Ncap);
    Parent(i).fitness=Ka/ObjFunc(Parent(i).chrom);
end
```

```
function [min,avg,max,sumFitness]=Statistik(PopSize,Parent)
min=Parent(1).fitness;
max=Parent(1).fitness;
sumFitness=Parent(1).fitness;
for i=2:PopSize
    if max<Parent(i).fitness
        max=Parent(i).fitness;
    end
    if min>Parent(i).fitness
        min=Parent(i).fitness;
    end
    sumFitness=sumFitness+Parent(i).fitness;
end
avg=sumFitness/PopSize;
```

```
function [mate]=Seleksi(PopSize,Parent,SumFitness)
partsum=0;
i=0;
mate=0;
rnd=rand*SumFitness;
for i=1:PopSize
    partsum=partsum+Parent(i).fitness;
    if partsum>rnd
        mate=i;
        break;
    end
end
```



```

if mate==0
    mate=PopSize;
end

```

```

function [AlleleChild]=Mutasi(AlleleParent,Pmutasi)
rnd=rand;
if rnd<=Pmutasi
    if AlleleParent==1
        AlleleChild=0;
    else
        AlleleChild=1;
    end
else
    AlleleChild=AlleleParent;
end

```

```

function [chromC]=MutasiChrom(chromP,Pmutasi)
x=chromP(:,1);
rows=length(x);
x=chromP(1,:);
cols=length(x);
chromC=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        chromC(i,j)=Mutasi(chromP(i,j),Pmutasi);
    end
end

```

```

function [chromC1,chromC2]=Crossover(Param,Length,chromP1,chromP2,Pcross)
chromC1=zeros(Param,Length);
chromC2=zeros(Param,Length);
rnd=rand;
if rnd<=Pcross
    pos1=round(1+rand*((Param-1)-1));
    pos2=round(1+rand*((Param-1)-1));
    while pos1==pos2
        pos2=round(1+rand*((Param-1)-1));
    end
    if pos2<pos1
        tmp=pos1;
        pos1=pos2;
        pos2=tmp;
    end
    for i=1:pos1
        for j=1:Length
            chromC1(i,j)=chromP2(i,j);
            chromC2(i,j)=chromP1(i,j);
        end
    end
    for i=(pos1+1):pos2
        for j=1:Length
            chromC1(i,j)=chromP1(i,j);
            chromC2(i,j)=chromP2(i,j);
        end
    end
    for i=(pos2+1):Param
        for j=1:Length
            chromC1(i,j)=chromP2(i,j);
            chromC2(i,j)=chromP1(i,j);
        end
    end
end

```

```

    end
else
    for i=1:Param
        for j=1:Length
            chromC1(i,j)=chromP1(i,j);
            chromC2(i,j)=chromP2(i,j);
        end
    end
end
end

```

```

function [Child]=Generasi(PopSize,Param,Length,Pcross,Pmutasi,Parent,...
    Child,SumFitness,Ka)

```

```

i=1;
while (i<=PopSize)
    mate1=Seleksi(PopSize,Parent,SumFitness);
    mate2=Seleksi(PopSize,Parent,SumFitness);
    [Child(i).chrom,Child(i+1).chrom]=Crossover(Param,Length,...
        Parent(mate1).chrom,...
        Parent(mate2).chrom,Pcross);
    [Child(i).chrom]=MutasiChrom(Child(i).chrom,Pmutasi);
    [Child(i+1).chrom]=MutasiChrom(Child(i+1).chrom,Pmutasi);
    Child(i).fitness=Ka/ObjFunc(Child(i).chrom);
    Child(i+1).fitness=Ka/ObjFunc(Child(i+1).chrom);
    i=i+2;
end

```

```

function [Parent]=FindNewParent(PopSize,Param,Length,Parent,Child)

```

```

[tmp]=InitialIndividu(PopSize,Param,Length);

```

```

for i=1:PopSize
    pos=i;
    while (pos==i)
        rnd=rand;
        pos=round(1+rnd*(PopSize-1));
    end
    if Child(i).fitness>Parent(pos).fitness
        tmp(i)=GetIndividu(Child(i));
    else
        tmp(i)=GetIndividu(Parent(pos));
    end
end
end
for i=1:PopSize
    Parent(i)=GetIndividu(tmp(i));
end

```

```

function [chrom]=GetBestChrom(MaxIndi)

```

```

x=MaxIndi.chrom(:,1);
rows=length(x);
x=MaxIndi.chrom(1,:);
cols=length(x);
chrom=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        chrom(i,j)=MaxIndi.chrom(i,j);
    end
end
end

```



## Simulated Annealing

```
function [BestIndi]=SimulatedAnnealing(itemax,ncool,R,X0,chromGA)
%ubah chromosome biner dari ga ke chromosome ga
[chromSA]=DecodeChromGA(chromGA);
%cari inisial best individu
[BestIndi]=InitIndividu(chromSA);
%inisialisasi sa untuk mencari suhu awal
[theta0]=InitSA(BestIndi,X0,itemax);
%copy best individu ke optimal individu sementara
[OptIndi]=GetIndividu(BestIndi);
%inisial suhu untuk awal perhitungan
the=theta0;
for i=1:ncool
    for j=1:itemax
        %cari neighborhood solution dari optimal solusi
        CekIndi=GetNeighboorIndi(OptIndi);
        %apakah cek solusi lebih baik daripada best solusi
        if CekIndi.fitness<BestIndi.fitness
            BestIndi=GetIndividu(CekIndi);
        end
        %apakah cek solusi lebih baik daripada optimal solusi
        if CekIndi.fitness<OptIndi.fitness
            OptIndi=GetIndividu(CekIndi);
        %jika cek solusi lebih jelek cek apakah cek solusi
        %bisa diterima sebagai optimal solusi walaupun lebih jelek
        else
            dF=CekIndi.fitness-OptIndi.fitness;
            rnd=rand;
            PdF=1/(1+exp(dF/the));
            %cek solusi DITERIMA walaupun lebih jelek!!!!
            if rnd<PdF
                OptIndi=GetIndividu(CekIndi);
            end
        end
    end
    %turunkan (dinginkan) suhu
    the=R^(i-1)*theta0;
    fprintf('%2.0f,i);
end
```

```
function [chromSA]=DecodeChromGA(chromGA)
global Nbank;
x=chromGA(:,1);
param=length(x);
chromSA=zeros(param,4);
for i=1:param
    chromSA(i,1)=chromGA(i,1);
    ch=zeros(3,1);
    for j=2:4
        ch(j-1)=chromGA(i,j);
    end
    chromSA(i,2)=round(DecodeBin1ToReal(ch,1,Nbank));
    ch=zeros(3,1);
    for j=5:7
        ch(j-4)=chromGA(i,j);
    end
    chromSA(i,3)=round(DecodeBin1ToReal(ch,1,Nbank));
    ch=zeros(3,1);
```

```

for j=8:10
    ch(j-7)=chromGA(i,j);
end
chromSA(i,4)=round(DecodeBin1ToReal(ch,1,Nbank));
end

```

```

function [indi]=InitIndividu(chromSA)
x=chromSA(:,1);
rows=length(x);
x=chromSA(1,:);
cols=length(x);
indi.chrom=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        indi.chrom(i,j)=chromSA(i,j);
    end
end
[indi.fitness]=ObjFuncSA(indi.chrom);

```

```

function [indiN]=GetNeighborIndi(indiP)
[indiN.chrom]=NeighborSolution(indiP.chrom);
[indiN.fitness]=ObjFuncSA(indiN.chrom);

```

```

function [indi]=GetIndividu(indiP)
x=indiP.chrom(:,1);
rows=length(x);
x=indiP.chrom(1,:);
cols=length(x);
indi.chrom=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        indi.chrom(i,j)=indiP.chrom(i,j);
    end
end
indi.fitness=indiP.fitness;

```

```

function [theta0]=InitSA(BestIndi,X0,itemax)
X1=0;
dC=0;
IndiP=GetIndividu(BestIndi);
for i=1:itemax
    IndiCek=GetNeighborIndi(IndiP);
    if IndiCek.fitness<IndiP.fitness
        IndiP=GetIndividu(IndiCek);
    else
        rnd=rand;
        if rnd<=X0
            X1=X1+1;
            dC=dC+(IndiCek.fitness-IndiP.fitness);
        end
    end
end
dC=dC/X1;
theta0=abs(dC/log(1/X0-1));

```



## Neighbor Solution

```
function [chrom]=NeighborSolution(chromP)
x=chromP(:,1);
rows=length(x);
x=chromP(1,:);
cols=length(x);
chrom=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        chrom(i,j)=chromP(i,j);
    end
end
sa=0;
for i=1:rows
    if chrom(i,1)==1
        sa=sa+1;
    end
end
pos=zeros(sa,1);
sa=0;
for i=1:rows
    if chrom(i,1)==1
        sa=sa+1;
        pos(sa)=i;
    end
end
lenpos=length(pos);
typ=round(1+rand*(4-1));
global Nbank;
%ubah posisi kapasitor ke bus lain yang berdekatan
if typ==1
    no=round(1+rand*(lenpos-1));
    bus1=pos(no);
    chpos=round(1+rand*(2-1));
    if chpos==1
        bus2=FindBackBus(bus1);
        cek=CekBus(pos,bus2);
        while (cek==1) & (bus1~=bus2) & (bus1==0) & (bus2==0)
            posc=round(1+rand*(lenpos-1));
            bus1=pos(posc);
            bus2=FindBackBus(bus1);
            cek=CekBus(pos,bus2);
        end
    elseif chpos==2
        bus2=FindForwardBus(bus1);
        cek=CekBus(pos,bus2);
        while (cek==1) & (bus1~=bus2) & (bus1==0) & (bus2==0)
            posc=round(1+rand*(lenpos-1));
            bus1=pos(posc);
            bus2=FindForwardBus(bus1);
            cek=CekBus(pos,bus2);
        end
    end
end
tmp=zeros(cols,1);
for j=1:cols
    tmp(j)=chrom(bus1,j);
end
for j=1:cols
```

```

    chrom(bus1,j)=chrom(bus2,j);
end
for j=1:cols
    chrom(bus2,j)=tmp(j);
end
%ubah kapasitas kapasitor
elseif typ==2
    no=round(1+rand*(lenpos-1));
    bus1=pos(no);
    chpos=round(1+rand*(2-1));
    if chpos==1
        kap=chrom(bus1,2);
        kap=kap+1;
        if kap>Nbank
            kap=Nbank;
        end
        chrom(bus1,2)=kap;
    elseif chpos==2
        kap=chrom(bus1,2);
        kap=kap-1;
        if kap<1
            kap=1;
        end
        chrom(bus1,2)=kap;
    end
    chpos=round(1+rand*(2-1));
    if chpos==1
        kap=chrom(bus1,3);
        kap=kap+1;
        if kap>Nbank
            kap=Nbank;
        end
        chrom(bus1,3)=kap;
    elseif chpos==2
        kap=chrom(bus1,3);
        kap=kap-1;
        if kap<1
            kap=1;
        end
        chrom(bus1,3)=kap;
    end
    chpos=round(1+rand*(2-1));
    if chpos==1
        kap=chrom(bus1,4);
        kap=kap+1;
        if kap>Nbank
            kap=Nbank;
        end
        chrom(bus1,4)=kap;
    elseif chpos==2
        kap=chrom(bus1,4);
        kap=kap-1;
        if kap<1
            kap=1;
        end
        chrom(bus1,4)=kap;
    end
end
%tambah kapasitor
elseif typ==3
    bus1=round(1+rand*(rows-1));

```

```

cek=CekBus(pos,bus1);
while cek==1
    bus1=round(1+rand*(rows-1));
    cek=CekBus(pos,bus1);
end
chrom(bus1,1)=1;
chrom(bus1,2)=round(1+rand*(Nbank-1));
chrom(bus1,3)=round(1+rand*(Nbank-1));
chrom(bus1,4)=round(1+rand*(Nbank-1));
%kurangi kapasitor
elseif typ==4
    no=round(1+rand*(lenpos-1));
    bus1=pos(no);
    chrom(bus1,1)=0;
end

```

```
function [bus]=FindBackBus(nobus)
```

```

global Z;
x=Z(:,1);
nbus=length(x);
dari=1;
ke=1;
if nobus==1
    bus=1;
else
    for i=1:nbus
        for j=1:nbus
            x=imag(Z(i,j));
            if x~=0
                if j==nobus
                    ke=j;
                    break;
                end
            end
        end
    end
    for i=1:nbus
        for j=1:nbus
            x=imag(Z(i,j));
            if x~=0
                if j==ke
                    dari=i;
                    break;
                end
            end
        end
    end
    bus=dari;
end

```

```
function [bus]=FindForwardBus(nobus)
```

```

global Z;
x=Z(:,1);
nbus=length(x);
dari=1;
ke=1;
if nobus==1
    bus=1;
else
    for i=1:nbus

```



```

for j=1:nbus
    x=imag(Z(i,j));
    if x~=0
        if i==nbus
            dari=j;
            break;
        end
    end
end
end
for i=1:nbus
    for j=1:nbus
        x=imag(Z(i,j));
        if x~=0
            if i==dari
                ke=i;
                break;
            end
        end
    end
end
end
bus=ke;
end

```

```

function [cek]=CekBus(pos,bus)
rows=length(pos);
cek=0;
if cek==0
    for i=1:rows
        if bus==pos(i)
            cek=1;
            break;
        end
    end
end
end

```

## Objective Function GA

```

function [fitness]=ObjFunc(chrom)
global Ncap Tlow Tmed Tpea kLow kMed kPea;
global Nbus Z Lc Tr Ts Cap Beban Pbase Zbase TypSal CapSal;
global lamdaV lamdaS lamdaCap;
global CapBank costFixed costSwitch;
global ke;
%Decoding data chromosome menjadi data capasitor pada low med peak dan
%hitung juga jumlah pemasangan kapasitor
[capLow,capMed,capPea,countCap]=DecodeDataChrom(chrom,CapBank);
%cari pelanggaran jumlah pemasangan kapasitor
dcap=0;
if countCap>Ncap
    dcap=countCap-Ncap;
end
pinCap=dcap*lamdaCap;
%cari beban low med peak dan masukkan data pemasangan kapasitor pada.
%data beban low med peak dan cari type capasitor fixed atau switch
[BebanLow,BebanMed,BebanPea,typCap]=DecodeBeban(Beban,capLow,capMed,...
        capPea,kLow,kMed,kPea);
%hitung total cost capasitor yang terpasang
[costCap]=HitungCostCapasitor(capLow,capMed,capPea,typCap);

```



```

dV=0;
dSal=0;
%Hitung loadflow beban low
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
    Cap,BebanLow,Pbase,Zbase,TypSal);

PlossLow=real(Rr);
%hitung pelanggaran tegangan tiap bus pada beban low
[dVt]=HitungCostV(V);
dV=dV+dVt;
%hitung pelanggaran kapasitor saluran pada beban low
[dS]=HitungConstSaluran(Alir,CapSal);
dSal=dSal+dS;
%Hitung loadflow beban Med
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
    Cap,BebanMed,Pbase,Zbase,TypSal);

PlossMed=real(Rr);
%hitung pelanggaran tegangan tiap bus pada beban med
[dVt]=HitungCostV(V);
dV=dV+dVt;
%hitung pelanggaran kapasitor saluran pada beban med
[dS]=HitungConstSaluran(Alir,CapSal);
dSal=dSal+dS;
%Hitung loadflow beban Peak
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
    Cap,BebanPea,Pbase,Zbase,TypSal);

PlossPea=real(Rr);
%hitung pelanggaran tegangan tiap bus pada beban peak
[dVt]=HitungCostV(V);
dV=dV+dVt;
%hitung pelanggaran kapasitor saluran pada beban peak
[dS]=HitungConstSaluran(Alir,CapSal);
dSal=dSal+dS;

%hitung pinalty pelanggaran tegangan
pinV=dV*lamdaV;
%hitung pinalty pelanggaran kapasitas saluran
pinS=dSal*lamdaS;

%hitung fitness (rumus [1] di jurnal)
fitness=costCap+ke*(Tlow*PlossLow+Tmed*PlossMed+Tpea*PlossPea)+...
    pinV+pinS+pinCap;

function [capLow,capMed,capPea,countCap]=DecodeDataChrom(chrom,CapBank)
global Nbank;
x=chrom(:,1);
param=length(x);
x=chrom(1,:);
len=length(x);
countCap=0;
capLow=zeros(param,1);
capMed=zeros(param,1);
capPea=zeros(param,1);
for i=1:param
    if chrom(i,1)==1
        countCap=countCap+1;
        ch=zeros(3,1);
        for j=2:4
            ch(j-1)=chrom(i,j);
        end
        capLow(i)=round(DecodeBin1ToReal(ch,1,Nbank))*CapBank;
    end
end

```

```

ch=zeros(3,1);
for j=5:7
    ch(j-4)=chrom(i,j);
end
capMed(i)=round(DecodeBin1ToReal(ch,1,Nbank))*CapBank;
ch=zeros(3,1);
for j=8:10
    ch(j-7)=chrom(i,j);
end
capPea(i)=round(DecodeBin1ToReal(ch,1,Nbank))*CapBank;
end
end

```

```

function [BebanLow,BebanMed,BebanPea,typCap]=DecodeBeban(Beban,capLow,...
    capMed,capPea,kLow,kMed,kPea)

```

```

x=Beban(:,1);
rows=length(x);
x=Beban(1,:);
cols=length(x);
BebanLow=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanLow(i,j)=Beban(i,j);
    end
    BebanLow(i,4)=capLow(i);
    BebanLow(i,5)=kLow*Beban(i,5);
    BebanLow(i,6)=kLow*Beban(i,6);
end
BebanMed=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanMed(i,j)=Beban(i,j);
    end
    BebanMed(i,4)=capMed(i);
    BebanMed(i,5)=kMed*Beban(i,5);
    BebanMed(i,6)=kMed*Beban(i,6);
end
BebanPea=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanPea(i,j)=Beban(i,j);
    end
    BebanPea(i,4)=capPea(i);
    BebanPea(i,5)=kPea*Beban(i,5);
    BebanPea(i,6)=kPea*Beban(i,6);
end
typCap=zeros(rows,1);
%type 1 = Fixed cap type = 2 Switch Cap type = 0 tidak ada capasitor
for i=1:rows
    if capLow(i)~=0
        if (capLow(i)==capMed(i)) & (capLow(i)==capPea(i))
            typCap(i)=1;
        else
            typCap(i)=2;
        end
    end
end
end

```

```

function [max]=CariMax(data1,data2,data3)
max=data1;

```

```

if max<data2
    max=data2;
end
if max<data3
    max=data3;
end

```

```

function [costCap]=HitungCostCapasitor(capLow,capMed,capPea,typCap)
global costFixed costSwitch;
global instCost;
rows=length(capLow);
costCap=0;
for i=1:rows
    if typCap(i)==1
        costCap=costCap+(capLow(i)*costFixed+instCost);
    elseif typCap(i)==2
        max=CariMax(capLow(i),capMed(i),capPea(i));
        costCap=costCap+(max*costSwitch+instCost);
    end
end
end

```

```

function [dV]=HitungCostV(V)
global Vmin Vmax
rows=length(V);
dV=0;
for i=1:rows
    absV=real(V(i));
    if absV>Vmax
        dV=dV+(absV-Vmax);
    end
    if absV<Vmin
        dV=dV+(Vmin-absV);
    end
end
end

```

```

function [dSal]=HitungConstSaluran(Alir,CapSal)
x=Alir(1,:);
rows=length(x);
x=Alir(:,1);
cols=length(x);
dSal=0;
for i=1:rows
    for j=1:cols
        P=real(Alir(i,j));
        if P>0
            absP=abs(P);
            if absP>CapSal(i,j)
                dSal=dSal+(absP-CapSal(i,j));
            end
        end
    end
end
end
end

```

## Objective Function SA

```

function [fitness]=ObjFuncSA(chrom)
global Ncap Tlow Tmed Tpea kLow kMed kPea;
global Nbus Z Lc Tr Ts Cap Beban Pbase Zbase TypSal CapSal;
global lamdaV lamdaS lamdaCap;

```



```

global CapBank costFixed costSwitch;
global ke;
%Decoding data chromosome menjadi data kapasitor pada low med peak dan
%hitung juga jumlah pemasangan kapasitor
[capLow, capMed, capPea, countCap]=DecodeDataChrom(chrom, CapBank);
%cari pelanggaran jumlah pemasangan kapasitor
dcap=0;
if countCap>Ncap
    dcap=countCap-Ncap;
end
pinCap=dcap*lamdaCap;
%cari beban low med peak dan masukkan data pemasangan kapasitor pada.
%data beban low med peak dan cari type kapasitor fixed atau switch
[BebanLow, BebanMed, BebanPea, typCap]=DecodeBeban(Beban, capLow, capMed, ...
    capPea, kLow, kMed, kPea);
%hitung total cost kapasitor yang terpasang
[costCap]=HitungCostCapasitor(capLow, capMed, capPea, typCap);
dV=0;
dSal=0;
%Hitung loadflow beban low
[V, ite, Alir, Sg, SL, SumG, SumL, Rr, RugiS, Jq]=NewtonPolar(Nbus, Z, Lc, Tr, Ts, ...
    Cap, BebanLow, Pbase, Zbase, TypSal);
PlossLow=real(Rr);
%hitung pelanggaran tegangan tiap bus pada beban low
[dVt]=HitungCostV(V);
dV=dV+dVt;
%hitung pelanggaran kapasitor saluran pada beban low
[dS]=HitungConstSaluran(Alir, CapSal);
dSal=dSal+dS;
%Hitung loadflow beban Med
[V, ite, Alir, Sg, SL, SumG, SumL, Rr, RugiS, Jq]=NewtonPolar(Nbus, Z, Lc, Tr, Ts, ...
    Cap, BebanMed, Pbase, Zbase, TypSal);
PlossMed=real(Rr);
%hitung pelanggaran tegangan tiap bus pada beban med
[dVt]=HitungCostV(V);
dV=dV+dVt;
%hitung pelanggaran kapasitor saluran pada beban med
[dS]=HitungConstSaluran(Alir, CapSal);
dSal=dSal+dS;
%Hitung loadflow beban Peak
[V, ite, Alir, Sg, SL, SumG, SumL, Rr, RugiS, Jq]=NewtonPolar(Nbus, Z, Lc, Tr, Ts, ...
    Cap, BebanPea, Pbase, Zbase, TypSal);
PlossPea=real(Rr);
%hitung pelanggaran tegangan tiap bus pada beban peak
[dVt]=HitungCostV(V);
dV=dV+dVt;
%hitung pelanggaran kapasitor saluran pada beban peak
[dS]=HitungConstSaluran(Alir, CapSal);
dSal=dSal+dS;

%hitung pinalty pelanggaran tegangan
pinV=dV*lamdaV;
%hitung pinalty pelanggaran kapasitas saluran
pinS=dSal*lamdaS;

%hitung fitness (rumus [1] di jurnal)
fitness=costCap+ke*(Tlow*PlossLow+Tmed*PlossMed+Tpea*PlossPea)+...
    pinV+pinS+pinCap;

function [capLow, capMed, capPea, countCap]=DecodeDataChrom(chrom, CapBank)

```

```

x=chrom(:,1);
param=length(x);
x=chrom(1,:);
len=length(x);
countCap=0;
capLow=zeros(param,1);
capMed=zeros(param,1);
capPea=zeros(param,1);
for i=1:param
    if chrom(i,1)==1
        capLow(i)=chrom(i,2)*CapBank;
        capMed(i)=chrom(i,3)*CapBank;
        capPea(i)=chrom(i,4)*CapBank;
    end
end
end

```

```

function [BebanLow,BebanMed,BebanPea,typCap]=DecodeBeban(Beban,capLow,...
    capMed,capPea,kLow,kMed,kPea)

```

```

x=Beban(:,1);
rows=length(x);
x=Beban(1,:);
cols=length(x);
BebanLow=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanLow(i,j)=Beban(i,j);
    end
    BebanLow(i,4)=capLow(i);
    BebanLow(i,5)=kLow*Beban(i,5);
    BebanLow(i,6)=kLow*Beban(i,6);
end
BebanMed=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanMed(i,j)=Beban(i,j);
    end
    BebanMed(i,4)=capMed(i);
    BebanMed(i,5)=kMed*Beban(i,5);
    BebanMed(i,6)=kMed*Beban(i,6);
end
BebanPea=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanPea(i,j)=Beban(i,j);
    end
    BebanPea(i,4)=capPea(i);
    BebanPea(i,5)=kPea*Beban(i,5);
    BebanPea(i,6)=kPea*Beban(i,6);
end
typCap=zeros(rows,1);
%type 1 = Fixed cap type = 2 Switch Cap type = 0 tidak ada capacitor
for i=1:rows
    if capLow(i)~=0
        if (capLow(i)==capMed(i)) & (capLow(i)==capPea(i))
            typCap(i)=1;
        else
            typCap(i)=2;
        end
    end
end
end
end

```

```

function [max]=CariMax(data1,data2,data3)
max=data1;
if max<data2
    max=data2;
end
if max<data3
    max=data3;
end

```

```

function [costCap]=HitungCostCapasitor(capLow,capMed,capPea,typCap)
global costFixed costSwitch;
global instCost;
rows=length(capLow);
costCap=0;
for i=1:rows
    if typCap(i)==1
        costCap=costCap+(capLow(i)*costFixed+instCost);
    elseif typCap(i)==2
        max=CariMax(capLow(i),capMed(i),capPea(i));
        costCap=costCap+(max*costSwitch+instCost);
    end
end

```

```

function [dV]=HitungCostV(V)
global Vmin Vmax
rows=length(V);
dV=0;
for i=1:rows
    absV=real(V(i));
    if absV>Vmax
        dV=dV+(absV-Vmax);
    end
    if absV<Vmin
        dV=dV+(Vmin-absV);
    end
end

```

```

function [dSal]=HitungConstSaluran(Alir,CapSal)
x=Alir(1,:);
rows=length(x);
x=Alir(:,1);
cols=length(x);
dSal=0;
for i=1:rows
    for j=1:cols
        P=real(Alir(i,j));
        if P>0
            absP=abs(P);
            if absP>CapSal(i,j)
                dSal=dSal+(absP-CapSal(i,j));
            end
        end
    end
end

```





## Objective Function Akhir GA

```
function [fitness]=ObjFuncAkhir(chrom)
global Ncap Tlow Tmed Tpea kLow kMed kPea;
global Nbus Z Lc Tr Ts Cap Beban Pbase Zbase TypSal CapSal;
global lamdaV lamdaS lamdaCap;
global CapBank costFixed costSwitch;
global ke;
%Decoding data chromosome menjadi data kapasitor pada low med peak dan
%hitung juga jumlah pemasangan kapasitor
[capLow, capMed, capPea, countCap]=DecodeDataChrom(chrom, CapBank);
disp(' ');
disp('-----');
disp(' bus Low Med Peak ');
disp(' kvar kvar kvar ');
disp('-----');
for i=1:Nbus
    if capLow(i)~=0
        %if (capLow(i)==capMed(i)) & (capLow(i)==capPea(i))
        %
        %else
        %
        %end
        Sa=[i capLow(i) capMed(i) capPea(i)];
        fprintf(' %3.0f %5.0f %5.0f %5.0f\n', Sa);
    end
end
disp('-----');
disp(' ');

%cari beban low med peak dan masukkan data pemasangan kapasitor pada.
%data beban low med peak dan cari type kapasitor fixed atau switch
[BebanLow, BebanMed, BebanPea, typCap]=DecodeBeban(Beban, capLow, capMed, ...
    capPea, kLow, kMed, kPea);
%hitung total cost kapasitor yang terpasang
[costCap]=HitungCostCapasitor(capLow, capMed, capPea, typCap);

%Hitung loadflow beban low
[V, ite, Alir, Sg, SL, SumG, SumL, Rr, RugiS, Jq]=NewtonPolar(Nbus, Z, Lc, Tr, Ts, ...
    Cap, BebanLow, Pbase, Zbase, TypSal);
PlossLow=real(Rr);

%Hitung loadflow beban Med
[V, ite, Alir, Sg, SL, SumG, SumL, Rr, RugiS, Jq]=NewtonPolar(Nbus, Z, Lc, Tr, Ts, ...
    Cap, BebanMed, Pbase, Zbase, TypSal);
PlossMed=real(Rr);

%Hitung loadflow beban Peak
mulai=cputime;
[V, ite, Alir, Sg, SL, SumG, SumL, Rr, RugiS, Jq]=NewtonPolar(Nbus, Z, Lc, Tr, Ts, ...
    Cap, BebanPea, Pbase, Zbase, TypSal);
selesai=cputime;
selang=selesai-mulai;
[Nbus]=CetakNewtonRaphson(Nbus, ite, selang, V, Sg, SL, Z, Alir, RugiS, SumG, SumL, Rr);
PlossPea=real(Rr);

%hitung fitness (rumus [1] di jurnal)
fitness=costCap+ke*(Tlow*PlossLow+Tmed*PlossMed+Tpea*PlossPea);
```

```
disp(['Cost Total Setelah Optimasi (US$) = ' num2str(fitness)]);
```

```
function [capLow, capMed, capPea, countCap]=DecodeDataChrom(chrom, CapBank)
global Nbank;
x=chrom(:, 1);
param=length(x);
x=chrom(1, :);
len=length(x);
countCap=0;
capLow=zeros(param, 1);
capMed=zeros(param, 1);
capPea=zeros(param, 1);
for i=1:param
    if chrom(i, 1)==1
        countCap=countCap+1;
        ch=zeros(3, 1);
        for j=2:4
            ch(j-1)=chrom(i, j);
        end
        capLow(i)=round(DecodeBin1ToReal(ch, 1, Nbank))*CapBank;
        ch=zeros(3, 1);
        for j=5:7
            ch(j-4)=chrom(i, j);
        end
        capMed(i)=round(DecodeBin1ToReal(ch, 1, Nbank))*CapBank;
        ch=zeros(3, 1);
        for j=8:10
            ch(j-7)=chrom(i, j);
        end
        capPea(i)=round(DecodeBin1ToReal(ch, 1, Nbank))*CapBank;
    end
end
```

```
function [BebanLow, BebanMed, BebanPea, typCap]=DecodeBeban(Beban, capLow, ...
    capMed, capPea, kLow, kMed, kPea)

x=Beban(:, 1);
rows=length(x);
x=Beban(1, :);
cols=length(x);
BebanLow=zeros(rows, cols);
for i=1:rows
    for j=1:cols
        BebanLow(i, j)=Beban(i, j);
    end
    BebanLow(i, 4)=capLow(i);
    BebanLow(i, 5)=kLow*Beban(i, 5);
    BebanLow(i, 6)=kLow*Beban(i, 6);
end
BebanMed=zeros(rows, cols);
for i=1:rows
    for j=1:cols
        BebanMed(i, j)=Beban(i, j);
    end
    BebanMed(i, 4)=capMed(i);
    BebanMed(i, 5)=kMed*Beban(i, 5);
    BebanMed(i, 6)=kMed*Beban(i, 6);
end
BebanPea=zeros(rows, cols);
for i=1:rows
    for j=1:cols
```

```

    BebanPea(i,j)=Beban(i,j);
end
BebanPea(i,4)=capPea(i);
BebanPea(i,5)=kPea*Beban(i,5);
BebanPea(i,6)=kPea*Beban(i,6);
end
typCap=zeros(rows, 1);
%type 1 = Fixed cap type = 2 Switch Cap type = 0 tidak ada kapasitor
for i=1:rows
    if capLow(i)~=0
        if (capLow(i)==capMed(i)) & (capLow(i)==capPea(i))
            typCap(i)=1;
        else
            typCap(i)=2;
        end
    end
end
end

```

```

function [max]=CariMax(data1,data2,data3)
max=data1;
if max<data2
    max=data2;
end
if max<data3
    max=data3;
end
end

```

```

function [costCap]=HitungCostCapasitor(capLow,capMed,capPea,typCap)
global costFixed costSwitch;
global instCost;
rows=length(capLow);
costCap=0;
for i=1:rows
    if typCap(i)==1
        costCap=costCap+(capLow(i)*costFixed+instCost);
    elseif typCap(i)==2
        max=CariMax(capLow(i),capMed(i),capPea(i));
        costCap=costCap+(max*costSwitch+instCost);
    end
end
end

```



## Objective Function Akhir SA

```

function [fitness]=ObjFuncAkhirSA(chrom)
global Ncap Tlow Tmed Tpea kLow kMed kPea;
global Nbus Z Lc Tr Ts Cap Beban Pbase Zbase TypSal CapSal;
global lamdaV lamdaS lamdaCap;
global CapBank costFixed costSwitch;
global ke;
%Decoding data chromosome menjadi data kapasitor pada low med peak dan
%hitung juga jumlah pemasangan kapasitor
[capLow,capMed,capPea,countCap]=DecodeDataChrom(chrom,CapBank);
disp(' ');
disp('-----');
disp(' bus Low Med Peak');
disp(' kvar kvar kvar');
disp('-----');
for i=1:Nbus
    if capLow(i)~=0

```



```

    %if (capLow(i)==capMed(i)) & (capLow(i)==capPea(i))
    %
    %else
    %
    %end
    Sa=[i capLow(i) capMed(i) capPea(i)];
    fprintf(' %3.0f %5.0f %5.0f %5.0f\n',Sa);
end
end
disp('-----');
disp(' ');
% cari beban low med peak dan masukkan data pemasangan kapasitor pada.
% data beban low med peak dan cari type kapasitor fixed atau switch
[BebanLow,BebanMed,BebanPea,typCap]=DecodeBeban(Beban,capLow,capMed,...
        capPea,kLow,kMed,kPea);
% hitung total cost kapasitor yang terpasang
[costCap]=HitungCostCapasitor(capLow,capMed,capPea,typCap);

% Hitung loadflow beban low
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
        Cap,BebanLow,Pbase,Zbase,TypSal);
PlossLow=real(Rr);
% hitung pelanggaran tegangan tiap bus pada beban low

% Hitung loadflow beban Med
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
        Cap,BebanMed,Pbase,Zbase,TypSal);
PlossMed=real(Rr);
% hitung pelanggaran tegangan tiap bus pada beban med

% Hitung loadflow beban Peak
mulai=cputime;
[V,ite,Alir,Sg,SL,SumG,SumL,Rr,RugiS,Jq]=NewtonPolar(Nbus,Z,Lc,Tr,Ts,...
        Cap,BebanPea,Pbase,Zbase,TypSal);
selesai=cputime;
selang=selesai-mulai;
[Nbus]=CetakNewtonRaphson(Nbus,ite,selang,V,Sg,SL,Z,Alir,RugiS,SumG,SumL,Rr);
PlossPea=real(Rr);

% hitung fitness (rumus [1] di jurnal)
fitness=costCap+ke*(Tlow*PlossLow+Tmed*PlossMed+Tpea*PlossPea);

function [capLow,capMed,capPea,countCap]=DecodeDataChrom(chrom,CapBank)
x=chrom(:,1);
param=length(x);
x=chrom(1,:);
len=length(x);
countCap=0;
capLow=zeros(param,1);
capMed=zeros(param,1);
capPea=zeros(param,1);
for i=1:param
    if chrom(i,1)==1
        capLow(i)=chrom(i,2)*CapBank;
        capMed(i)=chrom(i,3)*CapBank;
        capPea(i)=chrom(i,4)*CapBank;
    end
end
end

```

```
function [BebanLow,BebanMed,BebanPea,typCap]=DecodeBeban(Beban,capLow,...
    capMed,capPea,kLow,kMed,kPea)
```

```
x=Beban(:,1);
rows=length(x);
x=Beban(1,:);
cols=length(x);
BebanLow=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanLow(i,j)=Beban(i,j);
    end
    BebanLow(i,4)=capLow(i);
    BebanLow(i,5)=kLow*Beban(i,5);
    BebanLow(i,6)=kLow*Beban(i,6);
end
BebanMed=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanMed(i,j)=Beban(i,j);
    end
    BebanMed(i,4)=capMed(i);
    BebanMed(i,5)=kMed*Beban(i,5);
    BebanMed(i,6)=kMed*Beban(i,6);
end
BebanPea=zeros(rows,cols);
for i=1:rows
    for j=1:cols
        BebanPea(i,j)=Beban(i,j);
    end
    BebanPea(i,4)=capPea(i);
    BebanPea(i,5)=kPea*Beban(i,5);
    BebanPea(i,6)=kPea*Beban(i,6);
end
typCap=zeros(rows,1);
%type 1 = Fixed cap type = 2 Switch Cap type = 0 tidak ada capasitor
for i=1:rows
    if capLow(i)~=0
        if (capLow(i)==capMed(i)) & (capLow(i)==capPea(i))
            typCap(i)=1;
        else
            typCap(i)=2;
        end
    end
end
end
```

```
function [max]=CariMax(data1,data2,data3)
max=data1;
if max<data2
    max=data2;
end
if max<data3
    max=data3;
end
```

```
function [costCap]=HitungCostCapasitor(capLow,capMed,capPea,typCap)
global costFixed costSwitch;
global instCost;
rows=length(capLow);
costCap=0;
for i=1:rows
```

```
if typCap(i)==1
    costCap=costCap+(capLow(i)*costFixed+instCost);
elseif typCap(i)==2
    max=CariMax(capLow(i),capMed(i),capPea(i));
    costCap=costCap+(max*costSwitch+instCost);
end
end
```



# LAMPIRAN 3

- ✚ surat persetujuan survey
- ✚ Form. S-1A ( Permohonan persetujuan skripsi )
- ✚ Form. S-2 ( Lembar Pengajuan Judul skripsi )
- ✚ Form. S-3c ( Berita Acara Seminar Proposal skripsi )
- ✚ Form. S-4a ( Bimbingan skripsi )
- ✚ Form. S-4b ( Formulir Bimbingan skripsi )



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

## BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

1. **Nama** : HADI SURYA
2. **NIM** : 01.12.095
3. **Jurusan** : Teknik Elektro S-1
4. **Konsentrasi** : Teknik Energi Listrik
5. **Judul Skripsi** : ANALISA OPTIMASI PENEMPATAN KAPASITOR  
PADA JARINGAN DISTRIBUSI PRIMER 20kV  
DENGAN METODE *HYBRID GA-SA* DI GI. PIER,  
PENYULANG SYIAR PASURUAN

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

**Hari** : Senin  
**Tanggal** : 17 Maret 2008  
**Dengan Nilai** : 76,3 ( B+ ) *Buy*



Ir. Mochtar Asroni, M.S.M.E  
Ketua

### Panitia Ujian Skripsi

Ir. F. Yudi Limpraptono, M.T  
Sekretaris

### Anggota Penguji

Ir. Teguh Herbasuki, M.T

Ir. Widodo Pudji M. M.T



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

## PERSETUJUAN PERBAIKAN SKRIPSI

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

Hari : Senin  
Tanggal : 17 maret 2008

Telah dilakukan perbaikan skripsi oleh :

1. Nama : Hadi Surya
2. NIM : 01 12 095
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Energi Listrik S-1
5. Judul Skripsi : ANALISA OPTIMASI PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20kV DENGAN METODE HYBRID GA-SA DI GI PIER, PENYULANG SYIAR PASURUAN.

Perbaikan meliputi :

No	Materi Perbaikan	Ket
1.	Perbaikan faktor daya ( hal. 24 )	
2.	Diperjelas masalah <i>total cost</i> pada tujuan	

**Anggota Penguji**

**Ir. Teguh Herbasuki, M.T**

**Dosen Pembimbing I**

**Ir. H. Taufik Hidayat, M.T**

**Dosen Pembimbing II**

**Irrine Budi S. S.T, M.T**

THE UNIVERSITY OF CHICAGO  
DEPARTMENT OF CHEMISTRY  
5708 SOUTH CAMPUS DRIVE  
CHICAGO, ILLINOIS 60637

RECEIVED

TO: [Name] FROM: [Name]

DATE: [Date]

RE: [Subject]

1. [Text]

2. [Text]

3. [Text]

4. [Text]

5. [Text]

6. [Text]

7. [Text]

8. [Text]

9. [Text]

10. [Text]

11. [Text]

12. [Text]

13. [Text]

14. [Text]



15. [Text]

16. [Text]

17. [Text]

**PT. PLN (PERSERO)  
DISTRIBUSI JAWA TIMUR  
AREA PELAYANAN & JARINGAN PASURUAN**

Jl. Sudirman No. 69 Pasuruan 67115

Telp : (0343) 426515 s/d 426517

Telex :

Faximile : (0343) 426518

Alamat Kawat : PLN PSR

No. : 0749/041/APJ-PSR/2007

12 JUL 2007

No. :  
No. :  
No. :

Subjek : *Persetujuan Survey.*

KEPADA Yth. :

**DEKAN FAKULTAS TEKNOLOGI INDUSTRI  
I T N MALANG  
Jl. Bendungan Sigura-gura No. 2  
MALANG - 65145**

Menjawab surat Saudara Nomor : ITN-1330/III.TA-2/2/07 dan No.: ITN-1331/III.TA-2/2/07 tanggal 14 Juni 2007 perihal Survey, dengan ini disampaikan bahwa pada dasarnya kami dapat mengijinkan pelaksanaan Survey kepada mahasiswa Saudara sebagai berikut :

1. Nama : HADI SURYA  
NIM : 01.12.095
  
2. Nama : M. KHOIRUL ANAM  
NIM : 01.12.089

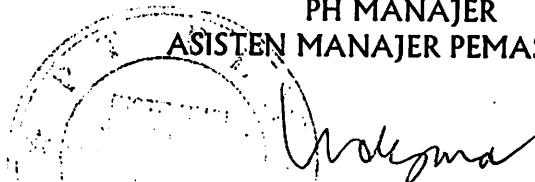
Adapun pelaksanaannya mulai tanggal 16 Juli 2007 sampai dengan selesai dan ditempatkan di Unit Pelayanan Dan Jaringan Bangil PT PLN (Persero) Area Pelayanan dan Jaringan Pasuruan.

Untuk penjelasan lebih lanjut, diminta yang bersangkutan menghubungi langsung **ASISTEN MANAJER SDM DAN ADMINISTRASI PT PLN (Persero) Area Pelayanan Dan Jaringan Pasuruan** Jl. Panglima Sudirman No. 69 Pasuruan dengan membawa salinan surat ini.

Perlu kami sampaikan bahwa apabila diperlukan data, kami hanya akan memberikan yang bersifat **TIDAK RAHASIA**.

Demikian, atas perhatiannya disampaikan terima kasih.

**PH MANAJER  
ASISTEN MANAJER PEMASARAN,**



Tembusan Yth. :

Ir. WAHJUDI

- ♦ MANAJER UPJ BANGIL



# PERMOHONAN PERSETUJUAN SKRIPSI

yang betanda tangan dibawah ini :

Nama : HADI SURYA  
 M : 01.12.095  
 Semester :  
 Fakultas : Teknologi Industri  
 Jurusan : Teknik Elektro S-1  
 Konsentrasi : Teknik Elektronika / Teknik Energi Listrik  
 Alamat : Jl. Bend. Nawangan 12 Malang

Sehubungan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama ini lampirkan persyaratan-persyaratan yang harus dipenuhi. Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

- Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
- Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
- Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
- Telah menempuh mata kuliah  $\geq 134$  sks dengan IPK  $\geq 2$  dan tidak ada nilai E (.....)
- Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
- Memenuhi persyaratan administrasi (.....)

Sehubungan dengan permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas  
 Recording Teknik Elektro

*[Signature]*  
 (..... Puji Handayani.....)

Malang, .....200

Pemohon

*[Signature]*  
 (..... Hadi Surya.....)

Disetujui

Ketua Jurusan Teknik Elektro

*[Signature]*  
 Ir. F. Yudi Limpraptono, MT  
 NIP. P. 1039500274

Mengetahui  
 Dosen Wali

*[Signature]*  
 (.....)

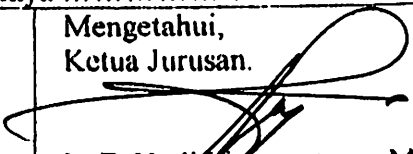
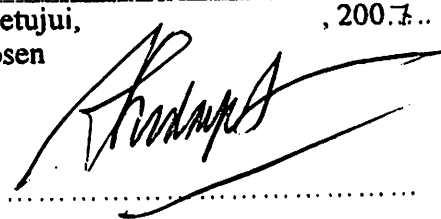
Keperluan :  
 Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

IPK  $\geq 2.43$   
 137  
 Praktikum lengkap



**LEMBAR PENGAJUAN JUDUL SKRIPSI  
 JURUSAN TEKNIK ELEKTRO S-1**

Spesialisasi : Teknik Energi Listrik/Teknik Elektronika \*)

Nama Mahasiswa : <u>HADI SURYA</u>		Nim : <u>01.12.095</u>	
Waktu pengajuan	Tanggal :	Bulan :	Tahun : <u>2007</u>
Spesifikasi judul ( berilah tanda silang )			
<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 .....		
Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) :		Mengetahui, Ketua Jurusan.	
<u>Ir. Taufik Hidayat, MT Fey</u>		 Ir. F. Yudi Limpraptono, MT Nip. Y. 1039500274	
Judul yang diajukan mahasiswa :	<u>"ANALISA OPTIMASI PENEMPATAN KAPASITOR... MENGGUNAKAN TEKNIK ARTIFICIAL INTELLIGENT PADA JARINGAN DISTRIBUSI 20 KV DI GI PER. PASURUAN"</u>		
Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu	..... ..... .....		
Catatan :	..... ..... ..... .....		
Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui, <span style="float: right;">, 2007..</span> Dosen 		

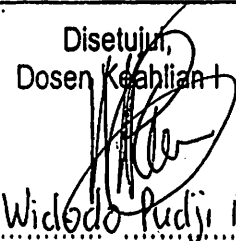
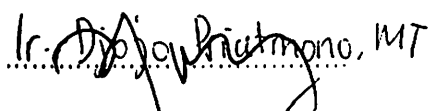
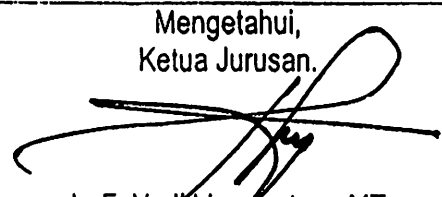
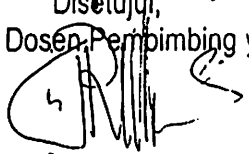
**Perhatian :**

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



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

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika\*)

1.	Nama Mahasiswa: <u>HADI SUWA</u>	Nim: <u>01.12.095</u>			
2.	Keterangan	Tanggal	Waktu	Tempat	
	Pelaksanaan	<u>19 Juli 2007</u>	<u>12.00 - 12.45</u>	<u>Ruang: SEMINAR</u>	
<b>Spesifikasi Judul (berilah tanda silang)**)</b>					
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. Elektronika Digital & Komputer <input type="checkbox"/> g. Elektronika Komunikasi <input type="checkbox"/> h. lainnya .....			
4.	Judul Proposal yang diseminarkan Mahasiswa	<u>OPTIMASI PENEMPATAN KAPASITOR PADA JARINGAN DISTRIBUSI PRIMER 20 KV MENGGUNAKAN METODE HYBRID GA-SA DI GI PIER PENYULANG SYAR PASURUAN</u>			
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	..... ..... .....			
6.	Catatan: .....				
<b>Persetujuan Judul Skripsi</b>					
7.	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II			
	 <u>Ir. Widodo Sudji, M, MT</u>	 <u>Ir. Djoko Priatomo, MT</u>			
Mengetahui, Ketua Jurusan.		Disetujui, Calon Dosen Pembimbing ybs			
 <u>Ir. F. Yudi Limapratono, MT</u> NIP. P. 1039500274		 <u>Ir. H. Budi S, ST, MT</u> NIP. 132314400			

Perhatian:

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