

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



**ANALISIS REKONFIGURASI JARINGAN DISTRIBUSI PRIMER 20 kV  
UNTUK MEMINIMALISASI RUGI-RUGI SECARA *MULTI-OBJECTIVE*  
DENGAN METODE *GENETIC ALGORITHM***

**SKRIPSI**

Disusun oleh :  
**YAN ADITYA PERDANA**  
00.12052



**SEPTEMBER 2005**

---

**LEMBAR PERSETUJUAN**

**ANALISIS REKONFIGURASI JARINGAN DISTRIBUSI PRIMER 20 kV  
UNTUK MEMINIMALISASI RUGI-RUGI SECARA *MULTI-OBJECTIVE*  
DENGAN METODE *GENETIC ALGORITHM***

**SKRIPSI**

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

**Disusun Oleh :**

**YAN ADITYA PERDANA  
NIM. 00.12.052**



**Mengetahui,  
Ketua Jurusan Teknik Elektro**

**Ir. F. Yudi Limpraptono, MT  
NIP.Y.103.950.0274**

**Menyetujui,  
Dosen Pembimbing**

**Ir. H. Taufik Hidayat, MT  
NIP. 101.870.00151**

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

**Ya Thanks to :**

# Allah SWT

Atas Segala Petunjuk dan bimbingan-Nya yang tertera pada Al-Quran dan As-sunnah, maupun petunjuk hidup yang tersirat (alam semesta dan kehidupan). Tidak lupa kepada junjungan kita, Nabi besar Muhammad Saw. beserta keluarga, sahabat, serta kepada kaum muslimin sedunia yang sedang berjuang meraih eksistensi kaumnya.



## My parents

H. Agus Hidayat, SH & Hj. Dwi Arti Sulistyowarni

have been a constant and invaluable support to me throughout every stage of my education and certainly no less during the writing of this skripsi Thanks, Mom and Dad.

**Mbak Dian, ST** Makasih atas doa & dukungannya maaf klo ak sering bikin kamu marah moga2x sukses jd arsitek handal & cepet married, I love U

**Mbak Nisam** Ingat selalu nasehat ortu & kakak2mu moga-moga kamu jdi wanita yg soleh+pinter+dewasa ok moga2x cepet lulus ya dek, Ilove U

**Nenekku Hj.Siti Fatimah** Makasih atas tempat tinggalnya(nangka 5) selama kuliah di malang jg doa dan nasehat yang tak henti2xnya moga-moga eyang selalu diberi kesehatan & rejeki oleh Allah SWT ...Amin

## My Best friends

Naser (Makasih supportnya selama ini " basket it's our life man"), Sinyo, ST (Temen curhat sejati soulmate forever), Cahyo cIn ST (Ayo semangat kurang dikit lot), Bakrul (Selamat uda gak jd ketua jomblo), Sugeng, ST(Moga-moga sukses dlm membina keluarga & kejaan dijakarta), Sersan Alfi (Cepet jadl Jenderal \*\*\*\* prend), Nanang (film wisuda tlg diedit), Ustad Hafid (Ayo kita Ngaji lagi), Lina+sinta (Sahabat baruku - saudara baruku !!!!)

\*) Also thank's for my ride honda GL Max 94 & my phone Nokia S210

## SPECIAL THANK'S TO :

**Ir. H. Taufik Hidayat, MT** : Makasih banyak atas Ilmu, nasehat dan masukan yang diberikan kepada saya sehingga skripsi ini dapat selesai (maju terus offroad indonesia !!!) .

**Mas Ugro (My Programer)** : Makasih telah ngerjain program dan ilmunya meskipun gak bisa perfect tapi boleh lah ....heec2x, mudah-mudahan kebaikan mas dibalas oleh Allah S.W.T.

**Dwi.W, ST** : Makasih udah dipinjem laporan skripsinya moga2x cepet dapet kerja suwon2x.

### ~~Temen-temen seperjuangan~~

**Lukman Khakim** (Thanks for everything man jasamu bnyk bgt), **Andy T.D** (Mas gun hajar ae P.M... klo gak di ACC, maaf ya klo srg ngrepotin kamu), **Deny Crut** (Makasih buat kerjasama kita selama ini), **Hendra .K** (Si Pria Metroseksual tp baik hati heee), **Sony K** (Lulusan the best ST '00), **Beni. H** (Pak Beni makasih ats bantuan selama ini sukses for u ), **Sastro** (Motor listrik yang crazy ponaane Zuhal), **Mamad** (suwoooooon), **Hunait** (Endi ole... dr banjar), **Kukuh** (cewekmu jagain coz deny suka curi-curi pandang heeee), **Mia** (!!!!), **Tanto** (To ga ramal manch ta), **Tri Panjen** (pria berhati e... Kentang (motor menggambarkan orangnya lo hee), **Desta** (moga2x sukses thn dpn), **Neni** (RT gmn jd nonton ta heee), **Rai** (Ayo jus semangat), **Tini** (Makasih data2x+disk... moga2x lancar skripsimu), **Johan Mlg** (Lapo mandek bos), **Surya** (Wis ditinggal gengmu lo cepetan lulus), **Toni** (Kemana aja? gak pernah nongol), **Arfan** (Selamat berjuang ), **Farid 98** (Bareng2x dong klo maju jgn ninggal hee), **Mamad 99** (klo blk ke balikpapan ojo lali konco oke), **Diana 99** (makasih saran & masukannya kita sama2x rekonfigurasi mania), **Anton 99** (Ayo ngopi ng AGP), **Dony 99+Dadang 98** (Kompre Sukses ber3 hee).....Dll

# Temen ST '00

Bang Tholip My brother(Friendship forever), Abas(My friend from brebes thank's bro), Kaktus (Pak Rtn Arek kontrakan), Pak Ek (The Fly mania), Johan Gresik (The Kentol from gresik), Joko Klaten (Nguyu tok ae !!), Budie Gresik(Karaoke mania), Lukman Arema(Komunitas Ikan Laut), Alif(Ayo man kamu bisa man cepet3x), Handoko(Gak pijet maneh heee), Zaenal(The Monox API 05), Agung Pk(Insafiah wis tuek hee), Santo (Pria plg romantis se ITN), Puguh(Mantan pacar dokter heeee suwon guh), Aan(Tetap berjuang an),Agung Blitar(),Udin,Wachid(Suwon cah hee),Nanang(Kpn2x klo ke karangkates ak mampir yo),Wawan+Eli (calone pak lurah cepu rek),Piere(Kta nenekku kamu paling ganteng diantara temen2xku hee),Ruly(Makasih diajak maen bola),Tommy(Ayo tom semangat skripsinya kya bikin gol),Daud(Glend fredly ITN ),Aryo+Erma(Tambah romantis aja),Andi cepu(Makasih atas ilmunya & boleh copy isi hardiskmu bagus2x loh heee),Dodik (Jangan Ngeblok aja sekali2x berbaur ma yg laen),Sigit kal(Wah arek males iki),Andik gresik(Melon Ndut) ,Choirul(Tetp rajin rul),Ali(Wis dadi ta fotone),Edwin(Soto Lamongan),Agung ponorogo(Moga2x kincirnya cpt kelar),Aris(Jombang tok ndek ITN heee),SGY(Gimana pak presiden mash PP Malang-Blitar??????),Widya(Tetep gondrong sampek tua),Wiky(Sorry duluan ya wik),Yasin(Arek pesantren gaul),Samsi+Samsul(twin brother yg rukun),Mardiono+Emi(Kompakan Lulus bareng ya),Handil(Ukuran sepatumu big size ),Dimas(jek Rajin nang Kantin) ,Yusak(Aremania tulen),Dedy mlg(Pengusaha HP) ,Nyoman(Moga2x Bali selalu damai),Dito(Ayo to ajok suwe2x ng ITN),Anton Mlg(Filmnya uda ditransfer blm????)

Juga buat Semua anak-anak Elektro '00 semua yang telah mambantu baik secara moril dan spriritual yang gak bisa aku sebutin satu-persatu Makasih.....Makasih.....Makasih banyak



my email for you

Sobatku

yan-id@plasa.com / tida\_itn@yahoo.com

## ABSTRAKSI

### ANALISIS REKONFIGURASI JARINGAN DISTRIBUSI PRIMER 20 kV UNTUK MEMINIMALISASI RUGI-RUGI SECARA *MULTI-OBJECTIVE* DENGAN METODE *GENETIC ALGORITHM*

( Yan Aditya P, Nim. 00.12052, Teknik Elektro/T.Energi Listrik S-1)  
( Dosen Pembimbing : Ir. H. Taufik Hidayat, MT )

**Kata Kunci :** *Jaringan distribusi, Rekonfigurasi, Fungsi multi-objective, Minimalisasi rugi-rugi, Genetic Algorithm.*

Sistem distribusi merupakan bagian dari sistem tenaga listrik yang berfungsi untuk menyalurkan dan mendistribusikan tenaga listrik ke pusat-pusat beban atau konsumen. Permasalahan yang harus dihadapi jaringan distribusi adalah bagaimana menyalurkan daya dengan baik pada saat tertentu maupun di masa yang akan datang. Kelemahan sistem distribusi yang hampir semuanya menggunakan sistem radial yakni sulitnya memenuhi aspek teknis. Hal ini karena lokasi beban dan variasi dari kerapatan beban yang menyebabkan tingginya rugi-rugi daya dan jatuh tegangan pada saat proses penyalurannya.

Dalam skripsi ini mempresentasikan cara yang efektif dalam rekonfigurasi jaringan untuk meminimalisasi rugi-rugi daya dan juga memperbaiki profil tegangan pada saluran jaringan distribusi. Menggunakan metode *Genetic Algorithm* untuk memecahkan masalah secara *multi-objective* masalah optimasi pada sistem distribusi radial. Dengan pengaturan keadaan *switch Normally Open / Normally Closed* yang optimal akan diperoleh rugi-rugi yang paling kecil (*Loss Minimum*).

Dari hasil rekonfigurasi jaringan pada penyulang-penyulang keluaran dari Trafo II Gardu Induk Blimbing Malang menggunakan metode *Genetic Algorithm* mampu meminimalisasi rugi-rugi daya sebesar 0,126 MW dan 0,192 MVAR.

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PERSETUJUAN .....	ii
ABSTRAKSI .....	iii
KATA PENGANTAR .....	iv
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xii
DAFTAR GRAFIK .....	xiii
<b>BAB I    PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Pembahasan .....	2
1.4. Batasan Masalah .....	3
1.5. Metode Penelitian .....	3
1.6. Sistematika Penulisan .....	4
1.7. Kontribusi .....	5
<b>BAB II    SISTEM DISTRIBUSI TENAGA LISTRIK DAN TEORI DASAR</b>	
<b><i>GENETIC ALGORITHM</i></b> .....	<b>6</b>
2.1. Sistem Tenaga Listrik .....	6
2.2 Sistem Distribusi .....	7
2.3. Struktur Jaringan Distribusi Tenaga Listrik .....	8
2.3.1. Sistem Jaringan Distribusi Radial .....	8

2.3.2. Sistem Jaringan Distribusi Loop .....	10
2.3.3. Sistem Jaringan Distribusi Mesh .....	11
2.4. Beban Sistem Distribusi .....	11
2.5. Daya Sistem Distribusi .....	12
2.5.1. Daya Aktif .....	12
2.5.2. Daya Reaktif.....	12
2.5.3. Daya Semu .....	12
2.6. Rugi-rugi jaringan .....	13
2.7. Meminimalisasi Rugi-rugi pada Sistem Distribusi .....	14
2.8. Teori Dasar <i>Genetic Algorithm</i> (GA).....	16
2.8.1. Tiga Golongan Teknik Optimasi <i>Genetic Algorithm</i> .....	16
2.8.2. Istilah Dasar <i>Genetic Algorithm</i> .....	17
2.8.3. Parameter <i>Genetic Algorithm</i> (GA).....	17
2.8.4. Fungsi Evaluasi .....	19
2.8.5. Fungsi Objective yang Digunakan pada <i>Genetic</i> <i>Algorithm</i> .....	20
2.8.6. Proses Kerja <i>Genetic Algorithm</i> .....	21
2.8.6.1. Pengkodean .....	21
2.8.6.2. Operator Genetik .....	22
2.8.6.3. Seleksi ( <i>selection</i> ).....	22
2.8.6.3.1. Seeksi Roda Roulette .....	23
2.8.6.4. Pindah Silang ( <i>Crossover</i> ) .....	23
2.8.6.5. Mutasi ( <i>Mutation</i> ) .....	24



<b>BAB III</b>	<b>ANALISIS REKONFIGURASI JARINGAN</b>	25
3.1.	Analisa Aliran Daya	25
3.2.	Pendekatan Studi Aliran Daya	25
3.3.	Sistem per-Unit	26
3.4.	Klasifikasi Bus	27
3.4.1.	Bus Beban atau P-Q Bus	28
3.4.2.	Bus Generator atau P-V Bus	28
3.4.3.	Bus Referensi atau V- $\delta$ Bus	28
3.5.	Metode Newton Raphson	28
3.5.1	Prosedur Aliran Daya Newton Raphson	32
3.6.	Metode Rekonfigurasi Jaringan	33
3.6.1.	Rekonfigurasi Jaringan menggunakan Metode <i>Genetic Algorithm</i>	34
3.7.	Penerapan Metode <i>Genetic Algorithm</i>	34
3.7.1.	Skema Pengkodean	34
3.7.2.	Inisialisasi	35
3.7.3.	Seleksi <i>Elitism</i>	35
<b>BAB IV</b>	<b>ANALISIS DAN HASIL PROGRAM REKONFIGURASI JARINGAN</b>	36
4.1.	Program Metode Rekonfigurasi Jaringan	36
4.2.	Algoritma program	36
4.2.1.	<i>Floew Chart</i> Program	37
4.2.2.	<i>Floew Chart</i> Perhitungan Aliran Daya Metode Newton Rapshon	38

4.2.3. <i>Floew Chart Genetic Algorithm</i> .....	39
4.3. Validasi Program .....	40
4.3.1. Hasil Validasi IEEE menggunakan Metode <i>Genetic Algorithm</i> .....	43
4.4. Data Jaringan .....	44
4.5. Analisis Perhitungan Aliran Daya Sebelum Rekonfigurasi .....	46
4.6. Analisa Rekonfigurasi Jaringan .....	48
4.6.1. Analisis Perhitungan Rekonfigurasi Jaringan .....	48
4.7. Analisis Perbandingan Hasil Perhitungan Sebelum dan Sesudah Rekonfigurasi menggunakan metode <i>Genetic Algorithm</i> .....	52
4.7.1. Analisis Hasil Perhitungan Tegangan dan Sudut Fasa Tegangan .....	52
4.7.2. Perbandingan Tingkat Rugi-rugi Pada Saluran .....	54
4.8. Hasil dan Analisa Hasil .....	55
4.8.1. Tampilan Program .....	55
<b>BAB V KESIMPULAN DAN SARAN</b> .....	60
5.1 Kesimpulan .....	60
5.2 Saran .....	61
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2-1. Skema Penyaluran Energi Listrik .....	6
Gambar 2-2. Jaringan Distribusi Tegangan Menengah (JTM), Jaringan Tegangan Rendah (JTR) dan sambungan Rumah ke Pelanggan .....	7
Gambar 2-3. Bentuk Sederhana dari Sistem Distribusi Radial .....	9
Gambar 2-4. Jaringan Tegangan Menengah Sistem Distribusi Radial .....	10
Gambar 2-5. Jaringan Menengah Struktur Loop .....	10
Gambar 2-6. Jaringan Tegangan Menengah Struktur Mesh .....	11
Gambar 2-7. Rugi-rugi Pada Bagian Sistem Tenaga Listrik .....	13
Gambar 2-8. Contoh Kondisi Ganggguan Pada Sistem Distribusi Radial .....	15
Gambar 2-9. Jumlah Generasi .....	18
Gambar 2-10. Jumlah Populasi .....	18
Gambar 2-11. Skema Pengkodean .....	21
Gambar 2-12. Operator Genetik.....	22
Gambar 2-13. Roulette Wheel .....	23
Gambar 2-14. Ilustrasi Operasi Crossover .....	24
Gambar 2-15. Ilustrasi Operasi Mutasi .....	24
Gambar 3-1. Diagram Skematik Tipe (a) <i>NC</i> dan (b) <i>NO switch</i> .....	33
Gambar 3-2. Jaringan Distribusi.....	34
Gambar 4-1. Struktur Jaringan Rekonfigurasi Test Sistem .....	40

Gambar 4-2. Konfigurasi Jaringan Radial dari Single line Diagram G.I Blimbing .....	44
Gambar 4-3. Konfigurasi Jaringan Mesh .....	48
Gambar 4-4. Konfigurasi Jaringan Radial Baru Setelah Rekonfigurasi .....	50
Gambar 4-5. Tampilan Menu Utama Program .....	55
Gambar 4-6. Tampilan Input Data General G.I Blimbing .....	55
Gambar 4-7. Tampilan Data Saluran .....	56
Gambar 4-8. Tampilan Data Pembebanan .....	56
Gambar 4-9. Tampilan Hasil Perhitungan Aliran Daya (Sebelum Rekonfigurasi) .....	57
Gambar 4-10. Tampilan Hasil Perhitungan Aliran Daya (Sebelum Rekonfigurasi) .....	57
Gambar 4-11. Tampilan Parameter GA .....	58
Gambar 4-12. Tampilan Hasil Perhitungan Aliran Daya (Setelah Rekonfigurasi) ..	58
Gambar 4-13. Tampilan Hasil Perhitungan Aliran Daya (Setelah Rekonfigurasi) ..	59
Gambar 4-14. Tampilan Hasil Kombinasi Optimal Switching, Best Fitness, $V_{drop}$ rata-rata, dan rugi daya aktif (Setelah Rekonfigurasi) .....	59

## DAFTAR TABEL

Tabel 2-1.	Istilah Genetic Algorithm .....	17
Tabel 3-1.	Skema Pengkodean dari contoh Gambar 3-2 .....	35
Tabel 4-1.	Data Pembebanan Jurnal IEEE.....	41
Tabel 4-2.	Data Saluran Jurnal IEEE .....	42
Tabel 4-3.	Perbandingan Data Jurnal Dengan Hasil Program .....	43
Tabel 4-4.	Data Hasil Perhitungan Daya dan Impedansi Saluran Jaringan Distribusi G.I BLimbing .....	45
Tabel 4-5.	Hasil Perhitungan Aliran Daya Sebelum Rekonfigurasi .....	46
Tabel 4-6.	Hasil Perhitungan Aliran Daya Sebelum Rekonfigurasi .....	47
Tabel 4-7.	Hasil Konmbinasi Optimal Switch.....	49
Tabel 4-8.	Hasil Perhitungan Aliran Daya Setelah Rkonfigurasi .....	51
Tabel 4-9.	Hasil Perhitungan Aliran Daya Setelah Rekonfigurasi .....	51
Tabel 4-10.	Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Saluran.....	53

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Sistem distribusi merupakan bagian dari sistem tenaga listrik yang berfungsi untuk menyalurkan dan mendistribusikan tenaga listrik ke pusat-pusat beban atau konsumen. Permasalahan pada jaringan distribusi yang harus dihadapi adalah bagaimana menyalurkan daya dengan baik pada saat tertentu maupun dimasa yang akan datang. Kelemahan sistem distribusi yang hampir semuanya menggunakan sistem radial yakni sulitnya memenuhi aspek teknis. Hal ini karena lokasi beban dan variasi dari kerapatan beban yang menyebabkan tingginya rugi-rugi daya dan jatuh tegangan pada saat proses penyalurannya.

Usaha yang dilakukan untuk meminimalisasi rugi-rugi daya dan memperbaiki profil tegangan pada saluran jaringan sistem distribusi salah satunya adalah dengan melakukan rekonfigurasi jaringan distribusi dengan pengaturan keadaan *switch* (*Normally open / Normally closed*) pada tiap-tiap saluran.

Sebelum melakukan rekonfigurasi jaringan distribusi terlebih dahulu dilakukan proses analisa aliran daya. Hasil analisa aliran daya memberikan informasi mengenai besarnya tegangan pada tiap-tiap saluran, besarnya daya pada tiap-tiap saluran dan untuk mengetahui rugi-rugi daya yang terjadi pada pengoperasian normal sehingga perhitungan akan dilakukan secara bersama-sama untuk mendapat hasil yang sesuai setelah dievaluasi kedalam sistem distribusi.

Dalam skripsi ini mencoba untuk menganalisis rekonfigurasi jaringan distribusi dengan menggunakan metode *Genetic Algorithm* untuk memecahkan

secara *multi-objective* masalah optimasi. Metode *Genetic Algorithm* ini dapat menentukan kombinasi *switch* yang optimal untuk meminimalisasi rugi-rugi daya dan memperbaiki profil tegangan pada saluran jaringan sistem distribusi.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang diatas rekonfigurasi jaringan distribusi dengan pengaturan keadaan *switch* (*Normally open / Normally closed*) dengan simulasi program rekonfigurasi jaringan menggunakan metode *Genetic Algorithm* maka akan diperoleh kombinasi *switch* yang paling optimal untuk meminimalisasi rugi-rugi daya dan dapat memperbaiki profil tegangan pada jaringan sistem distribusi.

Dari permasalahan diatas, maka dalam skripsi ini mengambil tema:

**“ANALISIS REKONFIGURASI JARINGAN DISTRIBUSI PRIMER 20 kV  
UNTUK MEMINIMALISASI RUGI-RUGI SECARA *MULTI-OBJECTIVE*  
DENGAN METODE *GENETIC ALGORITHM*”**

## 1.3. Tujuan Pembahasan

Tujuan dari penulisan skripsi ini adalah :

1. Mengaplikasikan metode *Genetic Algorithm* dalam rekonfigurasi jaringan untuk menentukan kombinasi *switch* yang paling optimal pada jaringan sistem distribusi.
2. Meminimalisasi rugi-rugi daya pada saluran jaringan sistem distribusi.
3. Meningkatkan profil tegangan pada saluran jaringan sistem distribusi.

#### 1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan, maka pembahasan skripsi ini dibatasi hal-hal berikut :

1. Hanya membahas rekonfigurasi jaringan distribusi untuk meminimalisasi rugi-rugi daya dan memperbaiki profil tegangan.
2. Beban diasumsikan sebagai beban tiga fasa seimbang.
3. Tidak membahas jenis dan penyebab terjadinya gangguan.
4. Tidak membahas aliran daya secara mendetail.
5. Tidak membahas pengontrolan switching yang dilakukan.
6. Power factor diasumsikan 0,85.
7. Analisa dilakukan hanya sebatas pengkajian beban suatu penyulang yang telah ada guna meminimalisasi rugi-rugi daya dan memperbaiki profil tegangan.
8. Data dan acuan diambil dari PT. PLN (Persero) Distribusi Jawa Timur pada Trafo II G.I Blimbing Malang.

#### 1.5. Metode Penelitian

1. Studi literature : Pengumpulan referensi untuk pembahasan pada skripsi ini yaitu dengan mempelajari teori yang terkait dengan pembahasan melalui literatur yang sesuai.
2. Pengambilan Data: Memperoleh data sebagai acuan dalam melengkapi parameter-parameter yang digunakan untuk menganalisa masalah.
3. Analisa Data : Menganalisa data yang diperoleh dengan metode *Genetic Algorithm* yang disimulasikan dengan program komputer.



## 1.6. Sistematika Penulisan

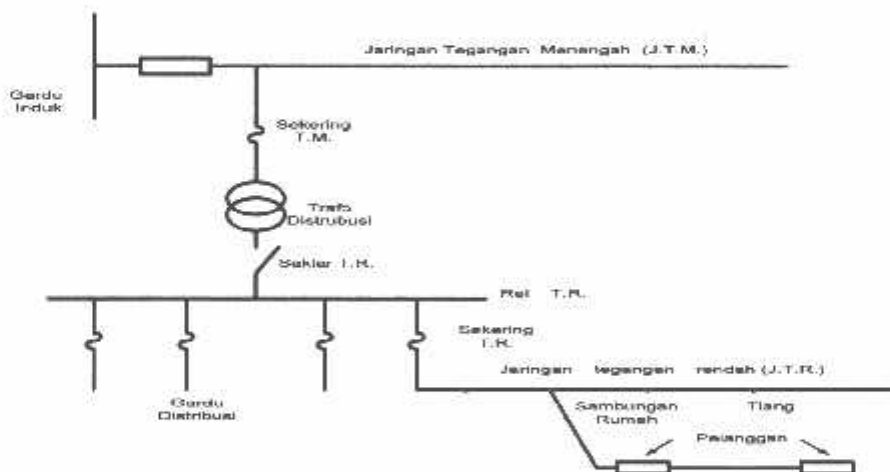
Sistematika penulisan skripsi ini adalah sebagai berikut :

- Bab I :       Pendahuluan
- Berisikan latar belakang, rumusan masalah, tujuan pembahasan, batasan masalah, metode penelitian, sistematika penulisan dan kontribusi penelitian.
- Bab II :       Sistem Distribusi Tenaga Listrik dan Teori Dasar *Genetic Algorithm*.
- Kajian pustaka mengenai sistem jaringan distribusi, macam-macam beban, macam-macam daya, rugi-rugi jaringan dan teori dasar metode *Genetic Algorithm*.
- Bab III :      Rekonfigurasi Jaringan
- Membahas analisis aliran daya, metode rekonfigurasi jaringan dan aplikasi penggunaan metode *Genetic Algorithm* pada rekonfigurasi jaringan.
- Bab IV:       Analisis dan Hasil Program Rekonfigurasi Jaringan.
- Berisi tentang program simulasi dan analisis yang terdiri dari prosedur program simulasi, hasil program simulasi dan analisis.
- Bab V :       Kesimpulan dan Saran
- Kesimpulan yang dapat diambil dari hasil pembahasan pada bab-bab sebelumnya dan saran-saran untuk pengembangan lebih lanjut.

3. Sistem Distribusi dan Beban ; berperan sebagai distributor energi ke konsumen yang memerlukan energi tersebut.

## 2.2. Sistem Distribusi <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 mendistribusikan 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 tegangan 220/380 Volt, yang merupakan tegangan siap pakai untuk peralatan-peralatan rumah tangga.

Dengan demikian maka sistem distribusi tenaga listrik dapat diklasifikasian berdasarkan tingkat tegangan distribusinya yaitu :

### ***1. Sistem Distribusi Primer***

Jaringan Distribusi Primer merupakan sistem tenaga listrik yang menyalurkan energi listrik antara Gardu Induk Distribusi sampai ke Gardu Distribusi dengan tegangan kerja 20 kV atau 6kV.

### ***2. Sistem Distribusi Sekunder***

Tingkat tegangan yang digunakan pada sistem distribusi sekunder adalah tegangan rendah 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>**

- a. Struktur Jaringan Radial.
- b. Struktur Jaringan *Loop* ( Lingkaran ).
- c. Struktur Jaringan *Grid* Atau *Mesh* ( Anyaman ).

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

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

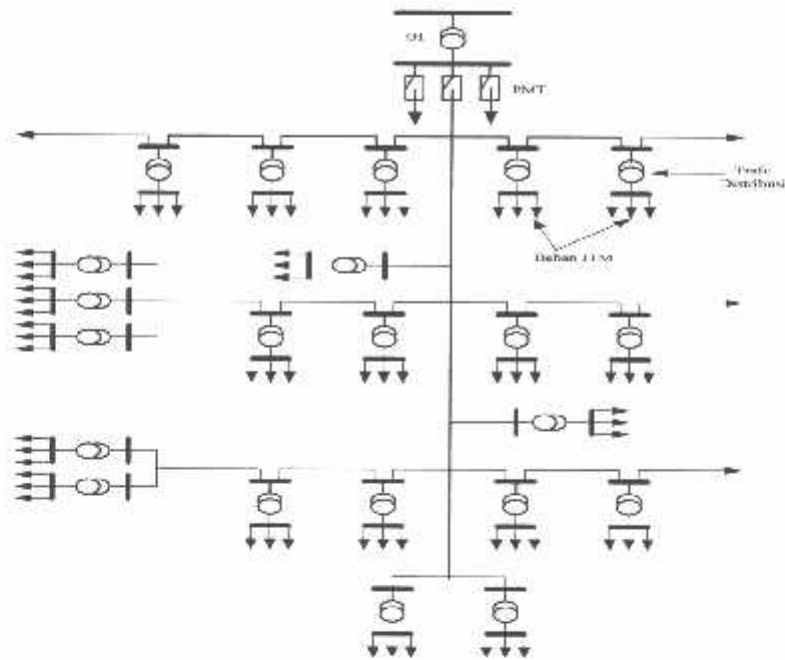
struktur jaringan ini tidak ada alternatif pasokan tenaga listrik, karena itu tingkat keandalannya relatif rendah.



**Gambar 2-3**  
**Bentuk sederhana dari sistem distribusi radial** [2]

Radial ganda adalah langkah dalam usaha meningkatkan keandalan jaringan, hal ini apabila rute dari sirkuitnya berlainan satu sama lain. Langkah lain untuk mempertinggi tingkat keandalan dari struktur radial adalah mengupayakan pasokan daya tidak hanya dari satu arah, walaupun pada pengoperasiannya dilaksanakan secara radial.

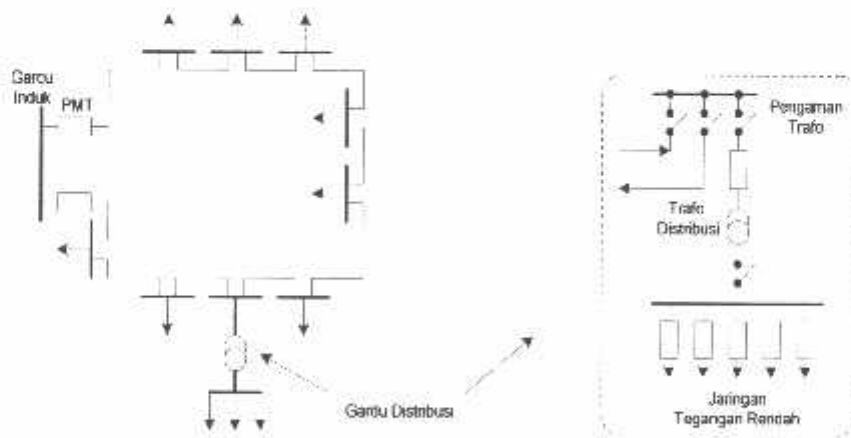
Bentuk yang paling umum digunakan pada sistem distribusi radial adalah seperti pada gambar 2-4. Dapat dilihat bahwa sebuah penyulang memasok sejumlah gardu distribusi. Jika terjadi gangguan pada jaringan tegangan menengahnya, maka pemutus beban yang ada di gardu induk akan membuka, hal ini menyebabkan semua gardu distribusinya akan mengalami pemadaman, maka pada penyulang dipasang peralatan pemisah seperti pelebur.



**Gambar 2-4**  
**Jaringan Tegangan Menengah Sistem Distribusi Radial <sup>[2]</sup>**

### 2.3.2. Sistem Jaringan Distribusi Loop.

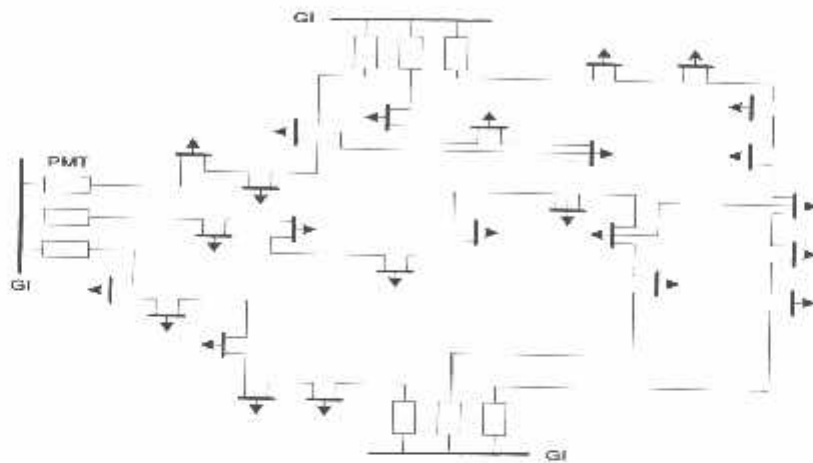
Sistem ini disebut jaringan distribusi loop karena saluran primer yang menyalurkan daya sepanjang daerah beban yang dilayani, membentuk suatu rangkaian loop.



**Gambar 2-5**  
**Jaringan Tegangan Menengah Struktur Loop <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 2-6  
Jaringan Tegangan Menengah Struktur Mesh <sup>[2]</sup>

### 2.4. Beban Sistem Distribusi <sup>[4]</sup>

Secara garis besar beban dapat diklasifikasikan menjadi tiga, yaitu :

#### 1. Beban Perumahan ( Rumah Tangga )

Beban perumahan pada umumnya berupa penerangan, alat-alat elektronik rumah tangga dan lain-lain.

#### 2. Beban Komersial

Beban komersial pada umumnya berupa penerangan toko, reklame dan lain-lain.

#### 3. Beban Industri

Beban industri diklasifikasikan menjadi 2 yaitu skala besar dan kecil.

## 2.7. Meminimalisasi rugi-rugi pada sistem distribusi <sup>18)</sup>

Seperti telah disinggung sebelumnya bahwa pada kebanyakan sistem distribusi dalam penyaluran tenaga listrik umumnya dioperasikan secara radial.

Namun demikian, sistem radial tersebut dioperasikan dengan beberapa “*improvisasi*”. Biasanya terdapat beberapa *tie switch* yang tersedia untuk saling interkoneksi bagi beberapa bus beban (*load bus*) baik dalam satu *feeder* maupun berlainan *feeder*.

Kondisi improvisasi tersebut sangat diperlukan demi memenuhi persyaratan sebagai sistem yang cukup baik, terutama yaitu kontinuitas pelayanan. Hal ini dapat dijelaskan dengan melihat gambar 2-8.

Suatu sistem penyaluran distribusi radial dari gardu induk mempunyai tiga buah *feeder* ( penyulang ) yaitu *feeder* A, B dan C. *Feeder* A mempunyai bus-bus beban a1, a2, ..., a9. *Feeder* B mempunyai bus-bus beban b1, b2, ..., b10. *Feeder* C mempunyai bus-bus beban c1, c2, ..., c9. Jika pada *feeder* A terdapat gangguan pada saluran antara bus a3 dan a5 sehingga sampai menyebabkan putusnya saluran maka bus a5, a6, ... a9 akan mengalami pemadaman total. Oleh karena itu bus-bus beban yang mengalami pemadaman total tersebut harus *disuply* dari *feeder* lain ( bisa juga dari bus a4 ). Tanda (---) menunjukkan bahwa pada saluran itu terdapat switch ( *Normally Open Switch* ). Dari gambar dapat dilihat bahwa NO switch terdapat antara bus a6 pada *feeder* A dengan bus a4 pada *Feeder* B. Jika pada saat gangguan terjadi sehingga bus a3 dan a5 *open* maka NO switch a6-b4 dapat ditutup ( *close* ) sehingga dengan demikian bus-bus pada *feeder* A yang mengalami pemadaman total dapat diatasi. Hal yang sama juga dapat dilaksanakan antara *feeder* B dengan *feeder* C dengan NO switch b9-c8.

## 2.8. Teori Dasar *Genetic Algorithm* ( GA )<sup>[6] [7]</sup>.

*Genetic Algorithm* adalah suatu algoritma yang meniru prinsip evolusi alam sebagai metoda untuk memecahkan masalah optimasi parameter. Prinsip yang mendasari *Genetic Algorithm* pertama kali dikembangkan oleh John Holland pada tahun 1962. Teori *Genetic Algorithm* (GA) dikembangkan menyerupai proses yang terjadi di alam raya, yaitu seleksi alam (*Darwinism*) dan evolusi genetika, dimana individu yang lebih kuat memiliki kemungkinan untuk menjadi pemenang dan mempunyai kesempatan hidup yang lebih besar di dalam lingkungan yang kompetitif.

*Genetic Algorithm* bekerja dengan populasi kromosom/string, dan melakukan proses pencarian nilai optimal secara paralel. Dengan menggunakan operator genetik *Genetic Algorithm* akan melakukan rekombinasi antar individu. Elemen dasar yang diproses algoritma genetik adalah string (kromosom) dengan panjang tertentu yang tersusun dari rangkaian substring (gen), dan biasanya merupakan kode biner (0,1). *Genetic Algorithm* merupakan prosedur iteratif, bekerja dengan suatu kumpulan string sebagai kandidat solusi. Setiap langkah iterasi menghasilkan generasi baru. Selama proses evolusi genetika, kromosom yang lebih sehat memiliki kecenderungan menghasilkan keturunan (*offspring*) yang sehat pula, dan kromosom yang sehat diharapkan menghasilkan keturunan yang lebih banyak serta mungkin dapat bertahan pada generasi selanjutnya.

### 2.8.1. Tiga Golongan Teknik Optimasi *Genetic Algorithm*<sup>[7]</sup>

1. Berbasis Kalkulus.
2. Enumeratif.
3. Pencarian acak terarah (*guide random search*).



### 2.8.2. Istilah dalam *Genetic Algorithm* <sup>[7]</sup>

Seperti sudah dijelaskan sebelumnya, bahwa algoritma ini berdasarkan pada mekanisme genetika yang ada pada proses alami dan sistem buatan. Sehingga istilah-istilah yang digunakan pada algoritma ini adalah gabungan dari dua disiplin ilmu, yaitu ilmu biologi dan ilmu komputer, tabel 2-1 menjelaskan mengenai istilah-istilah yang digunakan dalam GA.

**TABEL 2-1**  
**Istilah *Genetic Algorithm* <sup>[7]</sup>**

<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

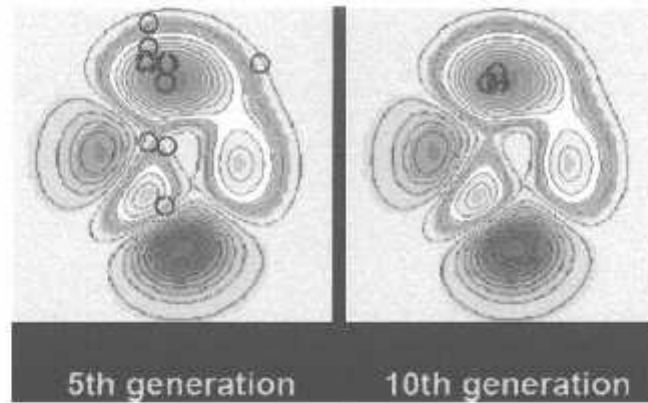
### 2.8.3. Parameter *Genetic Algorithm* ( GA ) <sup>[7]</sup>

Terdapat beberapa parameter yang digunakan dalam *Genetic Algorithm*. Parameter tersebut digunakan untuk melihat kompleksitas dari *Genetic Algorithm*. Parameter yang digunakan tersebut adalah :

- Jumlah Generasi ( *Max\_Gen* )

Merupakan jumlah iterasi (Perulangan dalam melakukan rekombinasi dan seleksi). Semakin besar jumlah generasi maka semakin optimum solusi yang mungkin dicapai, tetapi memerlukan waktu komputasi yang lama. Sebaliknya jika jumlah generasi sedikit maka akan terjadi *local optimum*

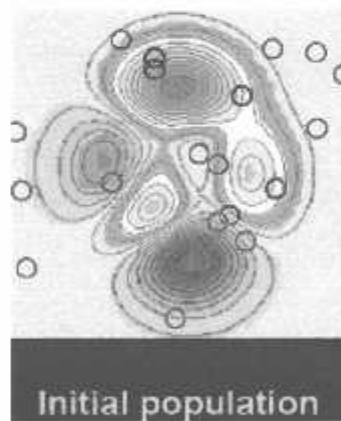
{keadaan dimana kromosom dengan nilai fitness terbaik akan mendominasi populasi sehingga terjadi konvergensi awal}.



**Gambar 2-9**  
**Jumlah Generasi <sup>[10]</sup>**

- Ukuran Populasi ( *Pop\_Size* )

Jumlah individu yang terdapat didalam suatu populasi yang merupakan kromosom atau string. Ukuran populasi mempengaruhi unjuk kerja dan keefektifan dari GA. GA dengan populasi kecil akan memiliki unjuk kerja yang buruk karena populasi tidak menyediakan cukup materi untuk memenuhi ruang permasalahan. Populasi yang besar dibutuhkan untuk mempresentasikan keseluruhan ruang permasalahan. Ukuran populasi yang sering digunakan berkisar antara 50 – 100.



**Gambar 2-10**  
**Jumlah Populasi <sup>[10]</sup>**  
18

- Probabilitas Mutasi ( $P_m$ )

Presentase dari jumlah total gen yang akan mengalami mutasi. Suatu nilai yang menentukan operasi Mutasi dalam rekombinasi GA. Jika nilai  $P_m$  terlalu rendah akan mengakibatkan gen yang berpotensi tidak dicoba untuk dapat dimutasikan, sebaliknya jika  $P_m$  terlalu tinggi maka Offspring akan kehilangan kemiripan induknya. Biasanya digunakan berkisar 0,001-0,02.

- Probabilitas Crossover ( $P_c$ )

Suatu nilai yang menentukan dalam operasi Crossover dalam rekombinasi GA. Rasio dari Offspring yang dihasilkan dalam setiap generasi didalam suatu populasi. Jika  $P_c$  semakin besar maka semakin cepat struktur kromosom baru terbentuk dalam populasi, sebaliknya jika  $P_c$  semakin kecil maka akan menghalangi pencarian solusi bagi permasalahan. Biasanya nilai  $P_c$  sering digunakan berkisar antara 0,6-0,8.

#### 2.8.4. Fungsi Evaluasi <sup>[61][7]</sup>

Fungsi evaluasi juga merupakan masalah yang penting dalam GA. Fungsi evaluasi yang baik harus mampu memberikan nilai fitness yang sesuai dengan kinerja kromosom.

Ada 2 hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu : evaluasi fungsi fitness (fungsi tujuan) dan konversi fungsi objective kedalam fungsi fitness.

**2.8.5. Fungsi objective yang digunakan pada Genetic Algorithm<sup>(1)</sup>.**

**Fungsi Objective**

**(a). Minimalisasi total rugi saluran:**

Min

$$TL_t = \sum_{j=1}^m (I_j^p)^2 \cdot r_j^p + (I_j^n)^2 \cdot r_j^n \dots\dots\dots(2.1)$$

dimana:

$TL_t$  = Total rugi-rugi saluran

$I_j^p$  dan  $r_j^p$  adalah arus dan resistansi dari fasa p dari segmen feeder j secara berurutan.  $I_j^n$  dan  $r_j^n$  adalah arus dan resistansi netral dari segmen cabang j.

**(b) Minimalisasi jatuh tegangan rata-rata:**

Min

$$AV_d = \frac{1}{n} \sum_{k=1}^n VD_k \dots\dots\dots(2.2)$$

dimana :

$AV_d$  = Jatuh tegangan rata-rata

n adalah jumlah total dari titik beban feeder objek, dan

$$VD_k = \left| \frac{V_{nominal} - V_k^p}{V_{nominal}} \right| \times 100\% \dots\dots\dots(2.3)$$

$V_{nominal}$  menampilkan tegangan nominal,  $V_k^p$  menandakan besarnya tegangan tiap saluran pada titik beban, dan  $VD_k$  menandakan rata-rata jatuh tegangan tiap saluran pada titik beban.

➔ **Multi-objective pemecahan masalah optimasi**

Dengan mengkombinasikan persamaan (2.1) dan (2.2), masalah optimasi dapat dihubungkan sebagai masalah optimasi multi-objective sebagai berikut :

Minimalisasi :

$$f = ( TL_t + AV_d )$$

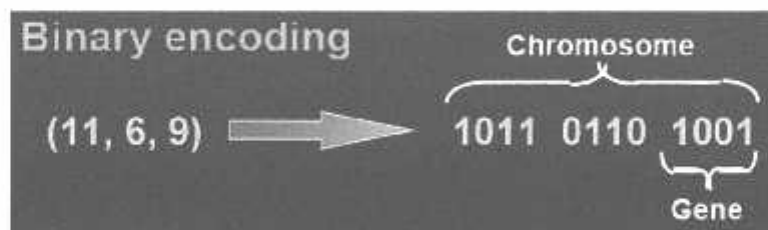
Dimana :  $TL_t$  = Total rugi-rugi daya pada saluran

$AV_d$  = Jatuh Tegangan Rata – rata

**2.8.6. Proses Kerja *Genetic Algorithm*** <sup>[7]</sup>

**2.8.6.1. Pengkodean**

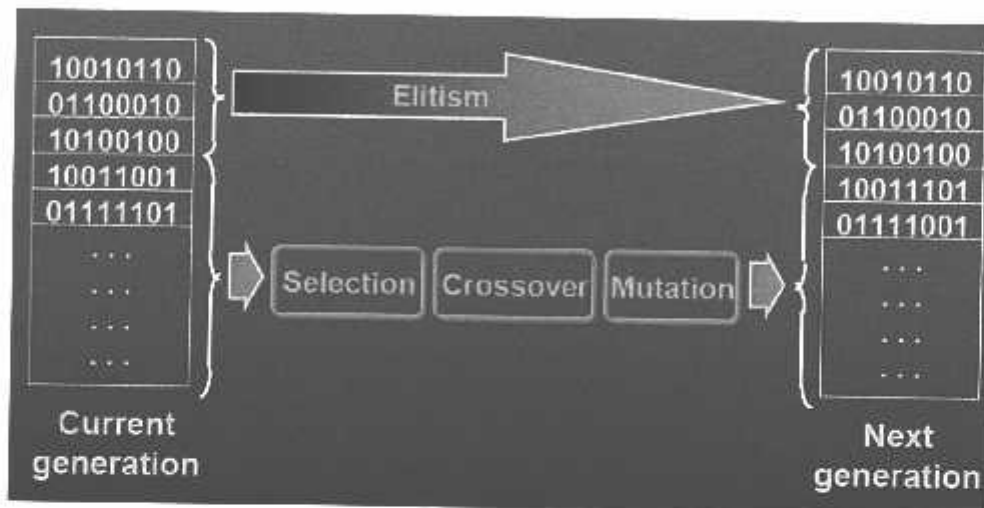
Langkah pertama yang dilakukan dalam penggunaan *Genetic Algorithm* adalah melakukan pengkodean atau representasi terhadap permasalahan yang akan dilakukan, dimana dibentuk oleh serangkaian kromosom. Setiap elemen dalam kromosom ini adalah variabel string yang disebut gen. Selanjutnya beberapa kromosom dibentuk dan berkumpul membentuk populasi. Populasi inilah awal bagi *Genetic Algorithm* untuk awal melakukan pencarian.



**Gambar 2-11**  
**Skema Pengkodean** <sup>[10]</sup>

### 2.8.6.2. Operator Genetik

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



Gambar 2-12

Operator Genetik <sup>[10]</sup>

### 2.8.6.3. Seleksi (*Selection*)

Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu-individu terpilih tersebut. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai fitness. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai fitness inilah yang nantinya akan dipergunakan pada tahap-tahap seleksi berikutnya. Salah satu metode seleksi adalah menggunakan *Roda Roulette*.

## BAB III

### ANALISIS REKONFIGURASI JARINGAN

#### 3.1. Analisa Aliran Daya <sup>[4]</sup>

Dengan semakin kompleknya problem di dalam sistem tenaga listrik, sebagai akibat dari meningkatnya permintaan konsumen, bertambahnya jumlah saluran transmisi dan distribusi, maka perlu adanya studi aliran daya dalam analisa sistem. Mempelajari aliran daya ini dilakukan untuk menentukan :

- Aliran daya aktif dan daya reaktif pada cabang-cabang rangkaian.
- Tidak adanya rangkaian yang mempunyai beban lebih dari tegangan busbar dalam batas-batas yang diterima.
- Pengaruh penambahan atau perubahan pada suatu sistem.
- Pengaruh hilangnya hubungan dalam keadaan darurat.
- Kondisi optimum pembebanan sistem.
- Kehilangan daya optimum sistem.

#### 3.2. Pendekatan Studi Aliran Daya <sup>[5]</sup>

Di dalam pengoperasian sistem tenaga listrik, parameter-parameter listrik yang perlu diperhatikan schubungan dengan analisa aliran daya adalah besarnya magnitude tegangan  $|V|$ , sudut fasa tegangan  $\theta$ , daya nyata  $P$  dan daya reaktif  $Q$ . Daya nyata  $P$  mempunyai ketergantungan yang kuat dengan sudut fasa tegangan  $\theta$  dan daya reaktif  $Q$  mempunyai ketergantungan yang kuat dengan besarnya magnitude tegangan  $|V|$ . Bila  $P$  dan  $Q$  berubah, maka  $\theta$  dan  $|V|$  berubah pula demikian sebaliknya.

Di dalam analisa aliran daya terdapat 3 jenis variabel :

1. Variabel bebas, misalnya  $|V|$  dan  $\theta$  pada bus beban atau  $\theta$  dan  $Q$  pada bus generator.
2. Variabel tidak bebas, misalnya  $P$  dan  $|V|$  pada bus generator.
3. Variabel yang tidak dapat diatur, misalnya kebutuhan konsumen.

### 3.3. Sistem Per-Unit <sup>(2)</sup>

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

$$\text{Besaran per-unit} = \frac{\text{besaran sebenarnya}}{\text{besaran dasar dengan kuantitas yang sama}} \quad \dots\dots(3.1)$$

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

- Untuk sistem 1 fasa

$$\text{Arus dasar, A} = \frac{\text{Dasar kVA}_{1\phi}}{\text{Tegangan dasar, kV}_{LN}} \quad \dots\dots(3.2)$$

$$\text{Impedansi dasar} = \frac{\text{Tegangan dasar, V}_{LN}}{\text{Arus dasar}} \quad \dots\dots(3.3)$$

$$\text{Impedansi dasar} = \frac{(\text{Tegangan dasar, kV}_{LN})^2 \times 1000}{\text{Dasar kVA}_{1\phi}} \quad \dots\dots(3.4)$$

$$\text{Impedansi dasar} = \frac{(\text{Tegangan dasar, kV}_{LN})^2}{\text{Dasar MVA}_{1\phi}} \quad \dots\dots(3.5)$$



- Untuk sistem 3 fasa

$$\text{Arus dasar, } A = \frac{\text{kVA}_{3\phi} \text{ dasar}}{\sqrt{3} \times \text{tegangan dasar, kV}_{LL}} \dots\dots\dots(3.6)$$

$$\text{Impedansi dasar} = \frac{(\text{Tegangan dasar, kV}_{LL})^2 \times 1000}{\text{kVA}_{3\phi} \text{ dasar}} \dots\dots(3.7)$$

$$\text{Impedansi dasar} = \frac{(\text{Tegangan dasar, kV}_{LL})^2}{\text{MVA}_{3\phi} \text{ dasar}} \dots\dots\dots(3.8)$$

### 3.4. Klasifikasi Bus <sup>[3]</sup>

Pada setiap simpul (rel atau bus) terdapat parameter-parameter sebagai berikut :

4. Daya nyata, dinyatakan dengan P satuannya Megawatt (MW).
5. Daya reaktif, dinyatakan dengan Q satuannya Mega Volt Ampere Reaktif (MVAR).
6. Besar (magnitude) tegangan mempunyai symbol  $|V|$  dengan satuan kiloVolt (kV).
7. Sudut fasa tegangan mempunyai symbol  $\delta$  dengan satuan radian.

Dalam analisis aliran daya, pada setiap busnya perlu diketahui 2 parameter dari keseluruhan 4 parameter yang diperhitungkan. Dengan melihat kedua parameter yang diketahui, setiap bus dalam suatu sistem dapat diklasifikasikan menjadi tiga yaitu :

1. Bus beban atau *load bus*.
2. Bus generator.
3. Bus referensi atau *slack bus*.

### 3.4.1. Bus Beban atau P-Q Bus

Pada bus ini terhubung beban-beban yang permintaan daya aktif dan daya reaktif jelas diketahui, sedangkan tegangan  $|V|$  dan sudut fasa tegangan  $\delta$  merupakan dua besaran yang akan dihitung nilainya.

### 3.4.2. Bus Generator atau P-V Bus

Bus ini terdapat generator-generator yang nilai tegangan dan daya aktifnya diketahui, sedangkan besarnya dan daya reaktif dan sudut fasa tegangan dihitung.

### 3.4.3. Bus Referensi atau V- $\delta$ Bus

Pada bus ini nilai daya aktif dan daya reaktif dibiarkan mengambang atau tidak diketahui, hal ini dikarenakan daya yang dikirim kepada sistem oleh generator tidak dapat dipastikan terlebih dahulu. Besarnya daya aktif dan daya reaktif selain ditentukan oleh besarnya beban juga ditentukan oleh besarnya daya yang hilang atau rugi-rugi pada saluran, nilainya hanya dapat ditentukan pada akhir perhitungan. Pada bus ini nilai tegangan masing-masing telah ditetapkan, yaitu sebesar 1 pu dan 0 derajat.

## 3.5. Metode *Newton Raphson* <sup>[5]</sup>

Secara matematis persamaan aliran daya dapat diselesaikan dengan menggunakan koordinat rectangular, koordinat polar atau bentuk *hybrid*. Dalam pembahasan skripsi ini memakai persamaan aliran daya *Newton Raphson* yang menggunakan koordinat polar.

Persamaan daya aktif dan reaktif pada bus  $i$  adalah :

$$P_i = \sum_{k=1}^n |V_i V_k Y_{ik}| \cos(\delta_i - \delta_k - \theta_{ik}) \dots\dots\dots(3.9)$$

$$Q_i = \sum_{k=1}^n |V_i V_k Y_{ik}| \sin(\delta_i - \delta_k - \theta_{ik}) \dots\dots\dots(3.10)$$

Kedua persamaan (3.1) dan (3.2) di atas akan menghasilkan suatu kumpulan persamaan yang tidak linier untuk setiap simpul sistem tenaga listrik.

Untuk mengetahui magnitudo tegangan ( $|V|$ ) dan sudut fasa ( $\delta$ ) disetiap simpul dapat diselesaikan dengan menggunakan persamaan di atas yang dilinierkan dengan metode *Newton Raphson* dibawah ini :

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix} \dots\dots\dots(3.11)$$

dimana :

$\Delta P$  : Selisih injeksi netto daya aktif dengan jumlah aliran daya aktif tiap saluran yang menghubungkan simpul dengan  $|V|$  yang didapat dari perhitungan iterasi ke- $k$ .

$\Delta Q$  : Selisih injeksi netto daya reaktif dengan jumlah 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 magnitudo tegangan

H, L, M, N merupakan elemen-elemen off diagonal dan diagonal dari submatrik Jacobian yang dibentuk dengan mendefinisikan persamaan (3.9) dan (3.10).

Dimana :

$$H_{ik} = \frac{\partial P_i}{\partial \delta_k} \quad N_{ik} = \frac{\partial P_i}{\partial |V_k|}$$

$$M_{ik} = \frac{\partial Q_i}{\partial \delta_k} \quad L_{ik} = \frac{\partial Q_i}{\partial |V_k|}$$

Adapun rumus dari elemen matrik Jacobian adalah :

Untuk H :

$$\frac{\partial P_i}{\partial \delta_k} = |V_i V_k Y_{ik}| \sin(\delta_i - \delta_k - \theta_{ik}) \quad i \neq k \quad \dots\dots\dots(3.12)$$

$$\frac{\partial P_i}{\partial \delta_k} = - \sum_{\substack{k=1 \\ k \neq i}}^n |V_i V_k Y_{ik}| \sin(\delta_i - \delta_k - \theta_{ik}) \quad \dots\dots\dots(3.13)$$

Untuk N :

$$\frac{\partial P_i}{\partial |V_k|} = |V_i V_k Y_{ik}| \cos(\delta_i - \delta_k - \theta_{ik}) \quad i \neq k \quad \dots\dots\dots(3.14)$$

$$\frac{\partial P_i}{\partial |V_k|} = 2|V_i Y_{ii}| \cos \theta_{ii} + \sum_{\substack{k=1 \\ k \neq i}}^n |V_i Y_{ik}| \cos(\delta_i - \delta_k - \theta_{ik}) \quad \dots\dots(3.15)$$

Untuk M :

$$\frac{\partial Q_i}{\partial \delta_k} = |V_i V_k Y_{ik}| \cos(\delta_i - \delta_k - \theta_{ik}) \quad i \neq k \quad \dots\dots\dots(3.16)$$

$$\frac{\partial Q_i}{\partial \delta_k} = - \sum_{\substack{k=1 \\ k \neq i}}^n |V_i V_k Y_{ik}| \cos(\delta_i - \delta_k - \theta_{ik}) \quad \dots\dots\dots(3.17)$$

Untuk L :

$$\frac{\partial Q_i}{\partial |V_k|} = |V_i V_k Y_{ik}| \sin(\delta_i - \delta_k - \theta_{ik}) \quad i \neq k \quad \dots\dots\dots(3.18)$$

$$\frac{\partial Q_i}{\partial |V_k|} = 2|V_i Y_{ii}| \sin \theta_{ii} + \sum_{\substack{k=1 \\ k \neq i}}^n |V_i Y_{ik}| \sin(\delta_i - \delta_k - \theta_{ik}) \quad \dots\dots(3.19)$$

Untuk menghitung selisih daya, maka mula-mula ditentukan harga awal tegangan simpul dan sudut fasanya, kemudian daya aktif dan reaktif dihitung dengan menggunakan persamaan (3.9) dan (3.10). Selisih daya antara daya yang telah ditentukan dengan daya hasil perhitungan ini merupakan perubahan daya yang terjadi pada simpul.

Magnitude tegangan dan sudut fasa yang diasumsikan ( $|V|$ ) dan ( $\delta$ ) serta selisih daya yang dihitung ( $\Delta P$  dan  $\Delta Q$ ) digunakan untuk memperoleh elemen-elemen matrik Jacobian.

Persamaan (3.15) 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 :

$$|V^{k+1}| = |V|^k + \Delta |V|^k \quad \dots\dots\dots(3.20)$$

$$\Delta^{k+1} = \delta^k + \Delta \delta^k \quad \dots\dots\dots(3.21)$$

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

### 3.5.1. Prosedur Aliran Daya *Newton Raphson* <sup>[5]</sup>

1. Tentukan nilai-nilai  $P_i$  (*ditentukan*) dan  $Q_i$  (*ditetapkan*) yang mengalir kedalam 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 Jacobian dengan menggunakan nilai-nilai perkiraan atau yang ditentukan dari besar dan sudut tegangan dalam persamaan untuk turunan parsial yang ditentukan dengan differensiasi persamaan (3.9) dan (3.11).
4. Balikan Jacobian itu dan hitung koreksi-koreksi tegangan  $\Delta\delta_k$  dan  $\Delta|V_k|$  pada setiap rel.
5. Hitung nilai baru dari  $\delta_k$  dan  $|V_k|$  dengan menambahkan  $\Delta\delta_k$  dan  $\Delta|V_k|$  pada nilai sebelumnya.
6. Kembalilah ke langkah 1 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.

### 3.6. Metode Rekonfigurasi Jaringan

Proses mengatur konfigurasi awal dari jaringan (*initial reconfigure*) menjadi suatu konfigurasi optimum akhir (*optimal reconfigure*), sehingga dari konfigurasi yang terakhir ini diperoleh kerugian daya dan jatuh tegangan pada sistem yang paling kecil (*Loss Minimum*). Meskipun jaringan distribusi dioperasikan menggunakan sistem radial, namun besarnya sistem tersebut dikombinasikan kedalam beberapa tipe. Salah satu tipe yang digunakan adalah model jaringan distribusi radial dengan kombinasi *Normally Open switch* dan *Normally Closed switch*.

Rekonfigurasi jaringan distribusi dapat dilakukan dengan mengatur kondisi dari *switch – switch* ini dalam keadaan *on* atau *off*. *Switch* yang terdapat dalam jaringan distribusi terdapat dua macam :



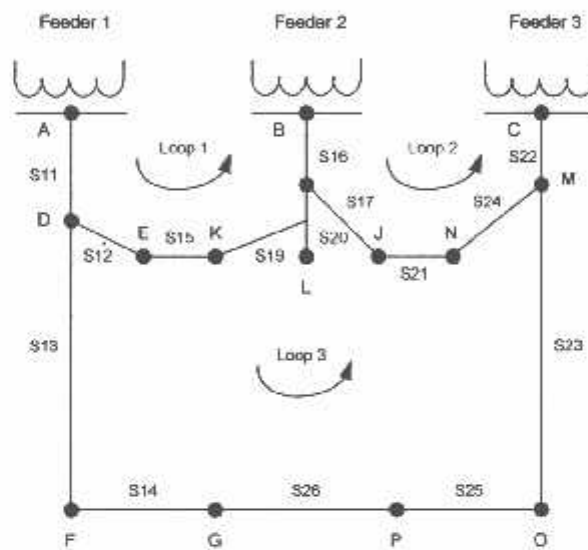
Gambar 3-1

Diagram skematik Tipe (a) NC dan (b) NO Switch

Jadi rekonfigurasi jaringan dalam sistem distribusi tenaga listrik adalah proses mengganti struktur topologi dari jaringan dengan mengubah status *open / closed* dari *NO / NC switch* sehingga dari proses ini bisa didapatkan aliran daya yang optimal pada sistem distribusi yang menyebabkan kerugian daya dan jatuh tegangan (*Loss Minimum*).

### 3.6.1. Rekonfigurasi Jaringan menggunakan metode *Genetic Algorithm* <sup>[11]</sup>

Gambar 3-2 menunjukkan contoh jaringan radial yang terdiri dari 3 feeder dengan 13 *switch normally closed (NC)* dan 3 *switch normally open (NO)* yaitu s12, s21 dan s26. Notasi (YZ) digunakan untuk menunjukkan operasi pembukaan *switch* Y dan penutupan *switch* Z. Misal operasi (s12, s15) mentransfer energi untuk beban 5 dari feeder 1 ke feeder 2 dengan membuka bagian *switch* s15 dan menutup *switch* s12.



Gambar 3-2  
Jaringan Distribusi <sup>[11]</sup>

### 3.7. Penerapan metode *Genetic Algorithm*

*Genetic Algorithm* adalah algoritma penelitian yang didasarkan pada mekanisme seleksi dan genetik yang bekerja dengan prosedur iteratif.

#### 3.7.1. Skema Pengkodean

Skema pengkodean adalah suatu cara untuk menentukan posisi *switch open* atau *closed* pada cabang tertentu pada konfigurasi jaringan radial, dengan memberikan kode biner (0 dan 1). Jika posisi *open (off)* diberikan kode biner 0,



jika posisi *closed (on)* diberikan kode biner 1. Dari tabel 3-1 dinyatakan bahwa ‘o’ menunjukkan bahwa *switch* terbuka (*open*) dan ‘c’ menunjukkan bahwa *switch* tertutup (*closed*) dan ‘m’ menunjukkan jumlah total *loop*, dan sama dengan jumlah total *switch* terbuka (*open*). Dalam rekonfigurasi jaringan distribusi, jaringan radial yang sudah ada akan dimodifikasi menjadi jaringan Mesh. Untuk mempertahankan agar struktur jaringan tetap radial maka hanya satu *switch* saja yang dibuka pada masing-masing *loop*.

**Tabel 3-1**  
**Skema Pengkodean dari contoh gambar 3-2**

m	1					2					3										
s	11	12	15	16	16	17	21	24	22	13	14	26	25	23	24	21	17	18	19	15	12
a	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1
w	15					21					26										

m = no.*loop*, s = no.*switch*, a = status *switch*, w = skema pengkodean

### 3.7.2. Inisialisasi

Populasi awal dibangkitkan secara *random*, dimana individu atau kromosom yang dibangkitkan sesuai dengan jumlah populasi yang telah ditentukan. Populasi awal adalah  $X_i = 1, \dots, p$  dimana p adalah ukuran populasi.

### 3.7.3. Seleksi *elitsm*

Kromosom induk dan keturunan akan dirangking secara naik menurut nilai *fitness* didalam suatu populasi. Individu dari induk dan *offspring* dengan *fitness* terbaik dijaga sebagai induk untuk generasi berikutnya. Untuk menghasilkan individu berikutnya sepenuhnya digunakan pendekatan *roulette wheel*.

**BAB IV**

**ANALISA DAN HASIL PROGRAM**

**REKONFIGURASI JARINGAN DISTRIBUSI**

**4.1. Program Metode Rekonfigurasi Jaringan**

Untuk pemecahan masalah rekonfigurasi jaringan distribusi digunakan bantuan program komputer. Program komputer ini menggunakan menggunakan bahasa pemrograman *Borland Delphi* versi 7.0 yang memiliki bahasa pemrograman terstruktur sehingga relatif mudah untuk dipelajari dan digunakan untuk mempercepat proses perhitungan yang membutuhkan ketelitian tinggi dan sering melibatkan iterasi yang membutuhkan waktu yang lama bila dikerjakan manual.

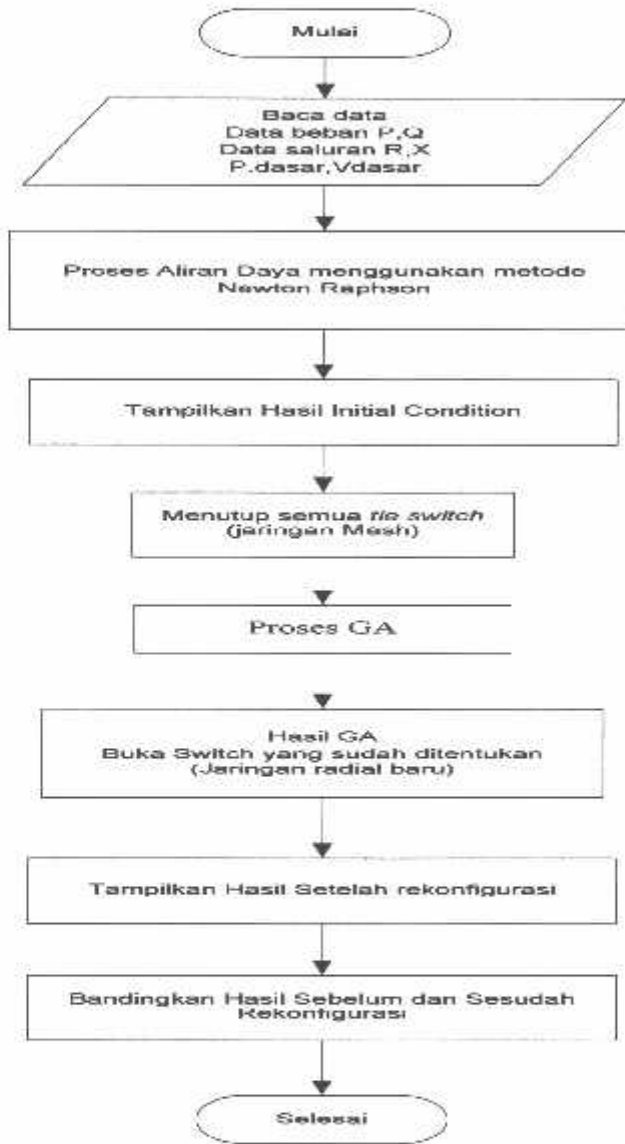
**4.2. Algoritma Program**

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

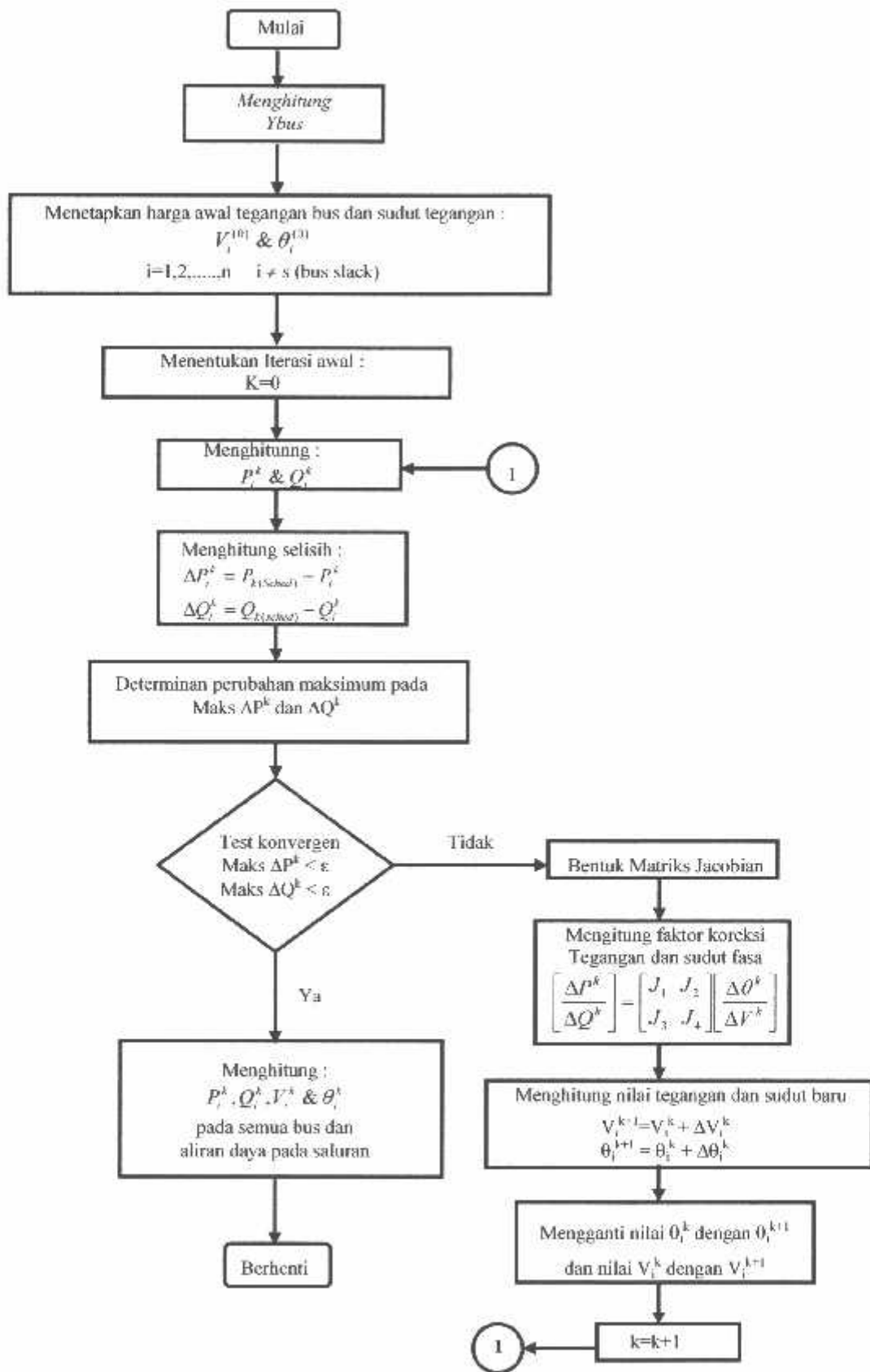
1. Masukkan data-data jaringan distribusi PLN.
2. Menjalankan proses *load flow* menggunakan metode *Newton Rhapson*.
3. Tampilkan hasil *initial condition*.
4. Menutup semua *tie switch* pada jaringan dan mengubah menjadi jaringan Mesh.
5. Proses *Genetic Algorithm*.
6. Hasil GA kombinasi *switch* yang sudah ditentukan (jaringan radial baru).

7. Menampilkan hasil minimalisasi rugi-rugi *multi-objective* GA.
8. Bandingkan hasil sebelum rekonfigurasi dan sesudah rekonfigurasi
9. Selesai

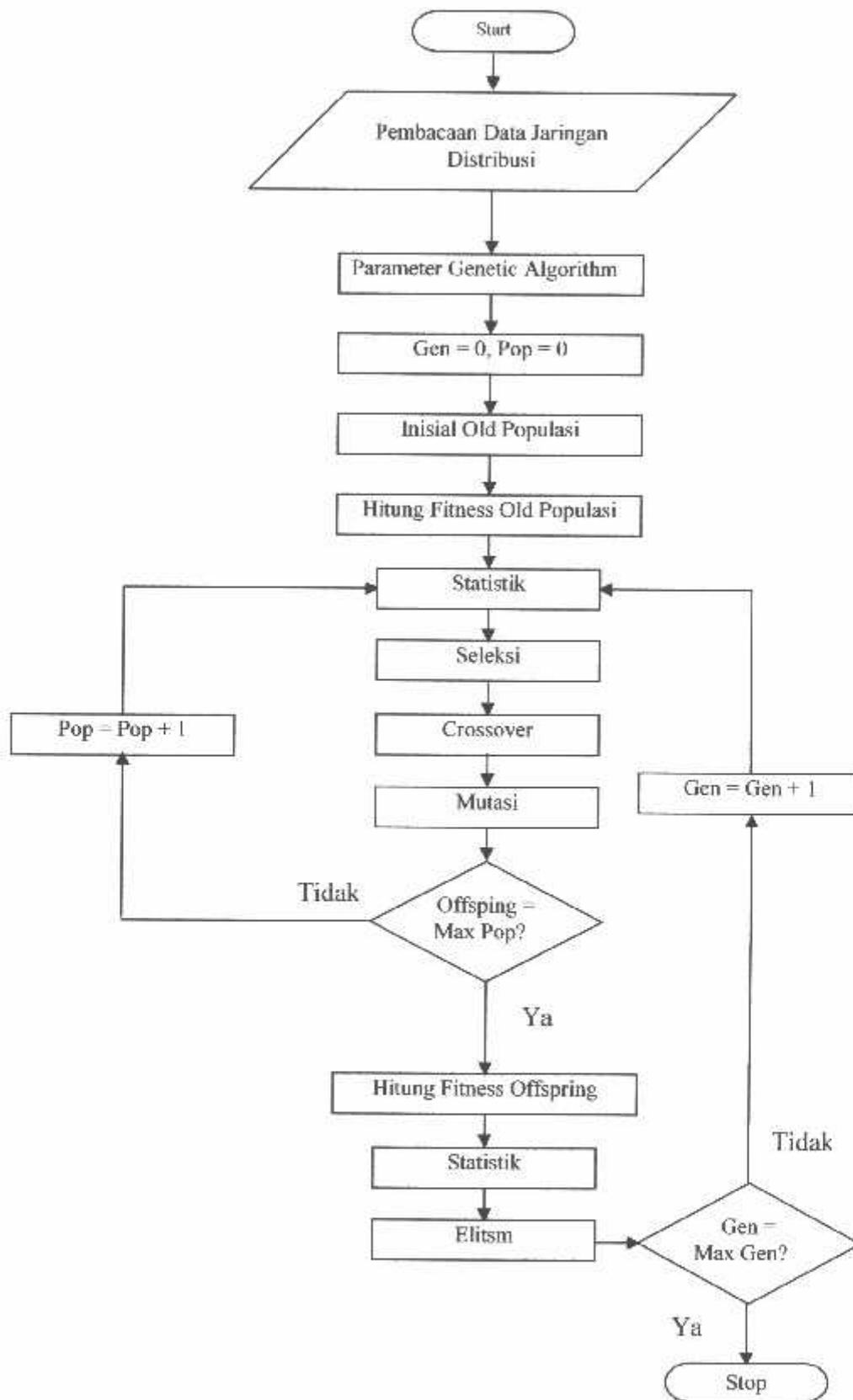
#### 4.2.1. FlowChart Program



4.2.2. FlowChart Perhitungan Aliran Daya Metode Newton Rapshon .

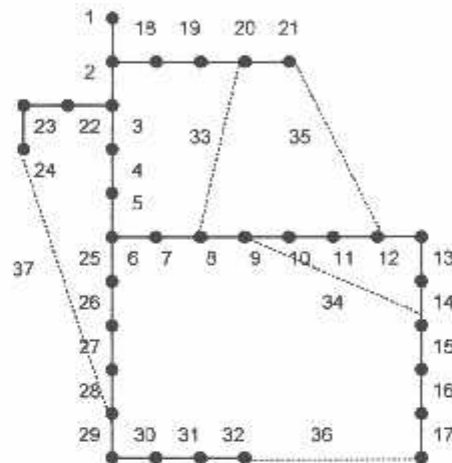


#### 4.2.3. FlowChart Genetic Algorithm



#### 4.3. Validasi Program.

Program rekonfigurasi jaringan dengan menggunakan metode *Genetic Algorithm* telah diuji validasinya dengan data dari jurnal A.Moussa, M.El-Gammal, E.N.Abdallah, A.I.Attia "A *Genetic Based Algorithm For Loss Reduction in Distribution System*" *Research and Energi Conservation sector in Alexandria Electricity Distribution company (AEDC), Alexandria, Egypt.* <sup>[1]</sup>



Gambar 4-1

Struktur Jaringan Rekonfigurasi Test Sistem <sup>[1]</sup>

➔ Data teknis dari jurnal adalah sebagai berikut <sup>[1]</sup>:

- Tegangan Dasar : 23 kV
- Daya Dasar : 100 kVA

➔ Data masukan parameter GA <sup>[1]</sup>:

- Jumlah Generasi : 100
- Jumlah Populasi : 10
- Probabilitas *Crossover* (Pc) : 0,98
- Probabilitas Mutasi (Pm) : 0,05
- Jumlah Loop : 5
- Jumlah *switch* : 37

**Tabel 4-1**  
**Data Pembebanan jurnal IEEE <sup>(1)</sup>**

<b>Bus</b>	<b>P (kW)</b>	<b>Q (kVAR)</b>
1	100	60
2	90	40
3	120	80
4	60	30
5	60	20
6	200	100
7	200	100
8	60	20
9	60	20
10	45	30
11	60	35
12	60	35
13	120	80
14	60	10
15	60	20
16	60	20
17	90	40
18	90	40
19	90	40
20	90	40
21	90	40
22	90	40
23	420	200
24	420	0.2
25	60	25
26	60	25
27	60	20
28	120	70
29	200	600
30	150	70
31	210	100
32	60	40

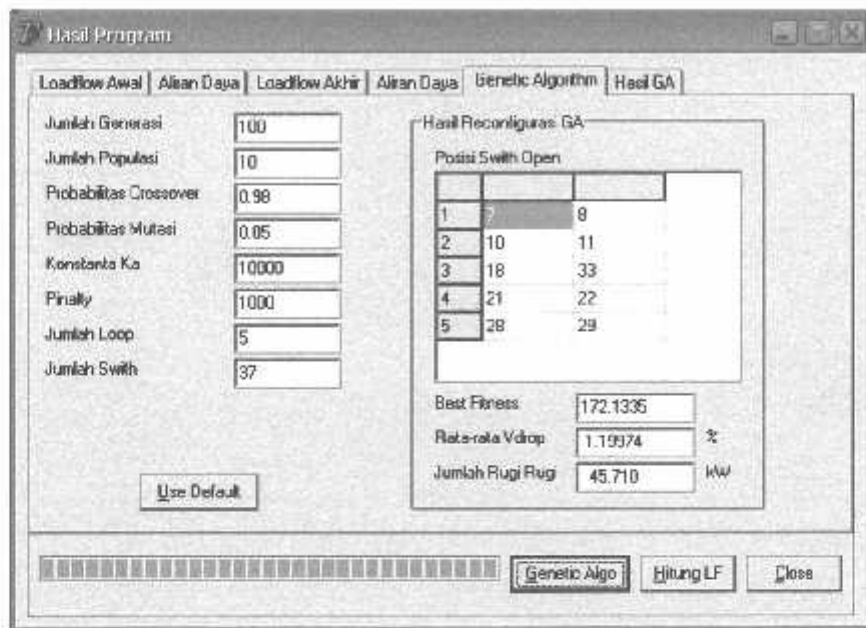
**Tabel 4-2**  
**Data Saluran jurnal IEEE <sup>(1)</sup>**

Saluran	Bus Pangkal	Bus Ujung	R (Ohm)	X (Ohm)
1	0	1	0,0922	0,0470
2	1	2	0,4930	0,2511
3	2	3	0,3660	0,1864
4	3	4	0,3811	0,1941
5	4	5	0,8190	0,7070
6	5	6	0,1872	0,6188
7	6	7	0,7114	0,2351
8	7	8	1,0300	0,7400
9	8	9	1,0440	0,7400
10	9	10	0,1966	0,0650
11	10	11	0,3744	0,1238
12	11	12	1,4680	1,0550
13	12	13	0,5416	0,7129
14	13	14	0,5910	0,5260
15	14	15	0,7463	0,5450
16	15	16	1,2890	1,7210
17	16	17	0,7320	0,5740
18	1	18	0,1640	0,1565
19	18	19	1,5042	1,3554
20	19	20	0,4095	0,4784
21	20	21	0,7089	0,9373
22	2	22	0,4512	0,3083
23	22	23	0,8980	0,7091
24	23	24	0,8960	0,7011
25	5	25	0,2030	0,1034
26	25	26	0,2842	0,1447
27	26	27	1,0590	0,9337
28	27	28	0,8042	0,7006
29	28	29	0,5075	0,2585
30	29	30	0,9744	0,9630
31	30	31	0,3105	0,3619
32	31	32	0,3410	0,5320
33*	7	20	2,0000	2,0000
34*	8	14	2,0000	2,0000
35*	11	21	2,0000	0,5000
36*	17	32	0,5000	0,5000
37*	24	28	0,5000	0,5000

**Keterangan :** \* Pada saluran tersebut terdapat *tie switch*.



### 4.3.1. Hasil Validasi IEEE menggunakan metode *Genetic Algorithm*.



Perbandingan hasil data jurnal dengan hasil perhitungan program setelah Rekonfigurasi pada jurnal selengkapnya dapat dilihat pada table 4-3 dibawah ini dengan *error* maksimal adalah  $TL_s = 0,078 \%$  dan  $AV_d = 0,22 \%$  .

**Tabel 4-3**  
Perbandingan data jurnal dengan hasil program

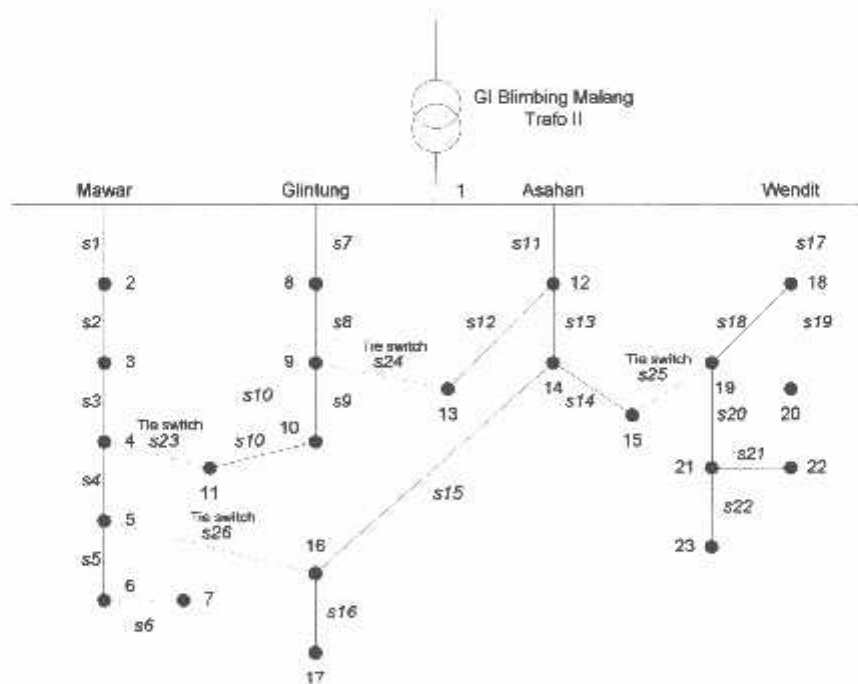
Data jurnal (pu)			Data hasil perhitungan program (pu)		
$TL_s$ (kW)	$AV_d$ (%)	Posisi switch open	$TL_s$ (kW)	$AV_d$ (%)	Posisi switch open
Total rugi-rugi saluran	Jatuhtegangan rata-rata		Total rugi-rugi saluran	Jatuhtegangan rata-rata	
45,746	1,197	7,10,36,21,28	45,710	1,19974	7,10,36,21,28

#### 4.4. Data Jaringan

Dalam skripsi ini data yang digunakan adalah penyulang-penyulang yang keluar dari Trafo II GI Blimbing Malang. Kondisi awal jaringan (*initial condition*) terlihat pada gambar 4-2. Alasan dari pemilihan data tersebut karena *output* dari Trafo II GI Blimbing Malang menyuplai daerah yang kepadatan bebannya bervariasi.

➔ Data teknis dari Trafo II G.I Blimbing Malang adalah sebagai berikut :

- Tegangan Dasar : 20 kV
- Daya Dasar : 30 MVA
- Penghantar SUTM 20 kV : AAAC 150 mm<sup>2</sup>
- Impedansi Saluran :  $0,1162 + j 0,3305 \Omega / \text{km}$
- $\text{Cos } \phi$  : 0,8



**Ket :** ● = *Node bus* tiap seksi (Data trafo distribusi unit jaringan malang mei 2005).  
S = Saluran (tiap-tiap saluran terdapat *switch*), - - - pada saluran tersebut terdapat *tie switch*

**Gambar 4-2**

**Konfigurasi Jaringan Radial dari *Single Line Diagram* G.I Blimbing Malang**

Data beban untuk masing-masing penyulang dapat dilihat pada lampiran dalam tabel A-1. sampai tabel A-7. Dari lampiran, daya P (MW) dan Q (MVAR) dari bus 1-2 Penyulang Mawar dapat dihitung sebagai berikut :

\* Untuk bus 1-2 :

$$P = 2286,80 \cdot \cos \varphi \rightarrow P = 2286,80 \cdot 0,8 = 1,8294 \text{ MW}$$

$$Q = 2286,80 \cdot \sin \varphi \rightarrow Q = 2286,80 \cdot 0,6 = 1,3721 \text{ MVAR}$$

Dengan cara menghitung yang sama, maka beban untuk masing-masing bus akan diperoleh hasil seperti pada tabel 4-4. berikut :

**Tabel 4-4**  
**Data Hasil Perhitungan Daya dan Impedansi Saluran**  
**Jaringan Distribusi G.I Blimbing Malang**

BUS PANGKAL	BUS UJUNG	P BUS UJUNG (MW)	Q BUS UJUNG (MVAR)	JARAK (km)	R (Ω)	X (Ω)
1	2	1,8294	1,3721	5,616	1,2142	1,8560
2	3	0,5962	0,4472	4,085	0,8832	1,3501
3	4	0,8966	0,6725	5,04	1,0896	1,6657
4	5	0,7369	0,5527	3,987	0,8425	1,2880
5	6	0,4358	0,3269	0,677	0,1464	0,2238
6	7	0,5231	0,3923	1,678	0,3628	0,5546
7	8	0,4802	0,3601	1,381	0,2986	0,4564
8	9	0,6159	0,4619	1,194	0,2581	0,3946
9	10	0,2826	0,2120	1,016	0,2197	0,3358
10	11	0,0576	0,0432	1,492	0,3226	0,4931
11	12	0,3656	0,2742	2,474	0,5349	0,8177
12	13	0,0556	0,0417	3,443	0,7444	1,1379
13	14	0,6274	0,4705	1,646	0,3559	0,5440
14	15	0,0593	0,0445	1,183	0,2558	0,3910
15	16	0,5703	0,4277	2,745	0,5935	0,9072
16	17	0,5961	0,4471	0,131	0,0283	0,0433
1	18	0,0634	0,0475	1,782	0,3853	0,5890
18	19	0,1138	0,0853	3,031	0,6553	1,0017
18	20	0,2201	0,1651	7,639	1,6516	2,5247
19	21	0,2493	0,1870	1,549	0,3349	0,5119
21	22	0,3670	0,2753	1,476	0,3191	0,4878
21	23	0,5581	0,4186	4,689	1,0138	1,5497
4	11	-	-	0,207	0,0448	0,0684
9	13	-	-	0,173	0,0374	0,0572
15	19	-	-	1,114	0,2408	0,3682
5	16	-	-	0,262	0,0566	0,0866

**Keterangan :** Bus 1 merupakan *slack bus*, bus 2 sampai dengan 23 merupakan bus beban.

#### 4.5. Analisa Perhitungan Aliran Daya Sebelum Rekonfigurasi

Untuk mengetahui kondisi awal jaringan (*initial condition*) dilakukan proses *load flow* menggunakan metode *Newton Raphson*. Data yang ada terlebih dahulu dirubah kedalam satuan per-unit hal ini dilakukan untuk mempermudah proses perhitungan. Dengan memasukkan data dari table 4-4 ke dalam program simulasi, maka akan diperoleh hasil seperti yang terdapat pada tabel 4-5 dan 4-6 dibawah ini :

**Tabel 4-5**  
**Hasil Perhitungan Aliran Daya Sebelum Rekonfigurasi**

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
1	1	0	10,544	8,113	0	0
2	0,96516	-0,70402	-	-	1,829	1,372
3	0,94887	-1,04699	-	-	0,596	0,447
4	0,93248	-1,40309	-	-	0,897	0,672
5	0,92418	-1,58802	-	-	0,737	0,553
6	0,92336	-1,60637	-	-	0,436	0,327
7	0,92226	-1,63123	-	-	0,523	0,392
8	0,99769	-0,04793	-	-	0,480	0,360
9	0,99636	-0,07563	-	-	0,616	0,462
10	0,99596	-0,08403	-	-	0,283	0,212
11	0,99586	-0,08612	-	-	0,058	0,043
12	0,99339	-0,13660	-	-	0,366	0,274
13	0,99317	-0,14128	-	-	0,056	0,042
14	0,98980	-0,21140	-	-	0,627	0,471
15	0,98972	-0,21312	-	-	0,059	0,045
16	0,98603	-0,29050	-	-	0,570	0,428
17	0,98594	-0,29244	-	-	0,596	0,447
18	0,99671	-0,06777	-	-	0,063	0,048
19	0,99213	-0,16297	-	-	0,114	0,085
20	0,99475	-0,10865	-	-	0,220	0,165
21	0,98999	-0,20762	-	-	0,249	0,187
22	0,98936	-0,22095	-	-	0,367	0,275
23	0,98691	-0,27220	-	-	0,558	0,419

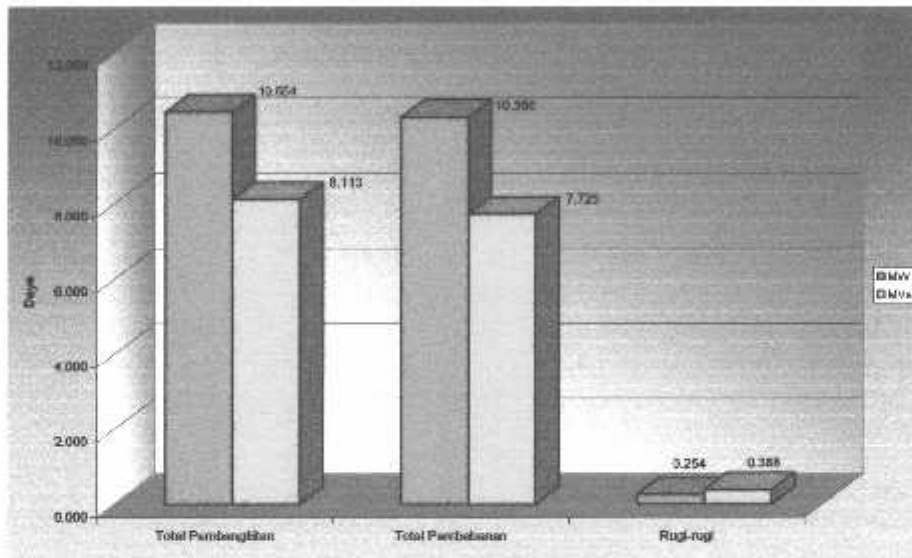
Tabel 4-6

Hasil Perhitungan Aliran Daya ( Sebelum Rekonfigurasi )

Total Pembangkitan		Total Pembebanan		Total Rugi Saluran	
P (MW)	Q (MVAR)	P (MW)	Q (MVAR)	P (MW)	Q (MVAR)
10,554	8,113	10,300	7,725	0,254	0,388

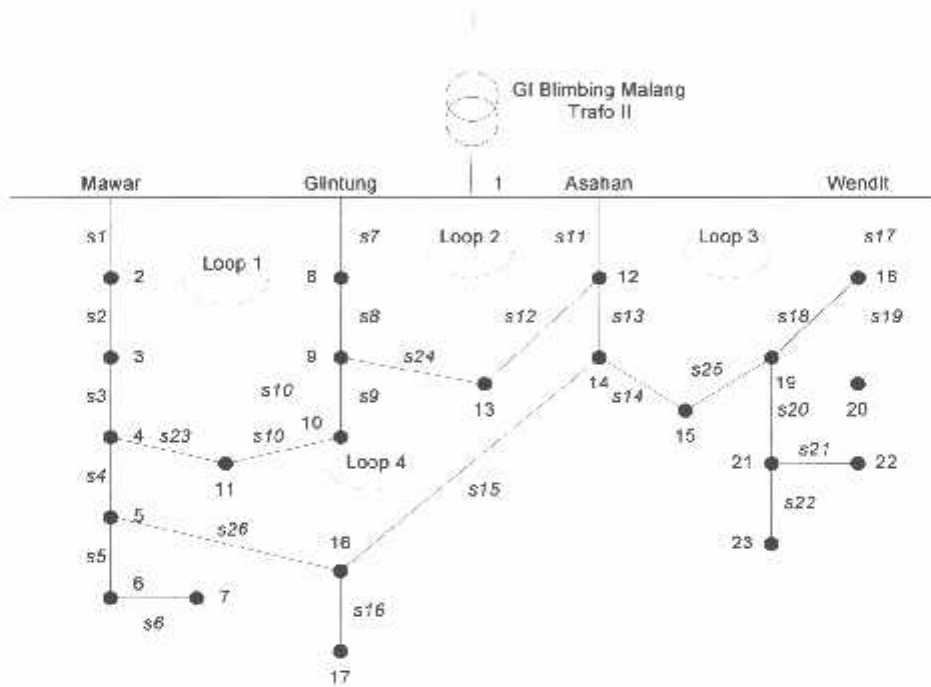
Grafik 4-1

Grafik Hasil Perhitungan Aliran Daya Sebelum Rekonfigurasi



#### 4.6. Analisa Rekonfigurasi Jaringan

Dalam analisis rekonfigurasi jaringan semua *tie switch* (*normally open*) yang saling interkoneksi pada beberapa bus beban (*load bus*) baik dalam satu *feeder* maupun berlainan *feeder* ditutup (*normally closed*) sehingga jaringan radial seperti pada gambar 4-2 berubah menjadi jaringan Mesh seperti terlihat pada gambar 4-3 dalam prosesnya disebut tahap (*stage*). Dari penutupan *tie switch* terdapat 4 *loop*.



Gambar 4-3  
Konfigurasi Jaringan Mesh

##### 4.6.1. Analisis Perhitungan Rekonfigurasi Jaringan

Langkah-langkah analisa perhitungan adalah sebagai berikut :

###### 1. Masukan Awal

Dengan memasukkan data pada tabel 4-4 ke dalam program simulasi rekonfigurasi *Genetic Algorithm*.

## 2. Memasukkan Parameter *Genetic Algorithm*

Memasukkan parameter *Genetic Algorithm* yang akan diolah oleh program simulasi rekonfigurasi *Genetic Algorithm* untuk memperoleh hasil yang diinginkan.

→ Data masukan parameter GA ini berupa :

- Jumlah generasi : 100
- Jumlah Populasi : 10
- Probabilitas Crossover ( $P_c$ ) : 0.98
- Probabilitas Mutasi ( $P_m$ ) : 0.05
- Jumlah Loop : 4
- Jumlah switch : 26

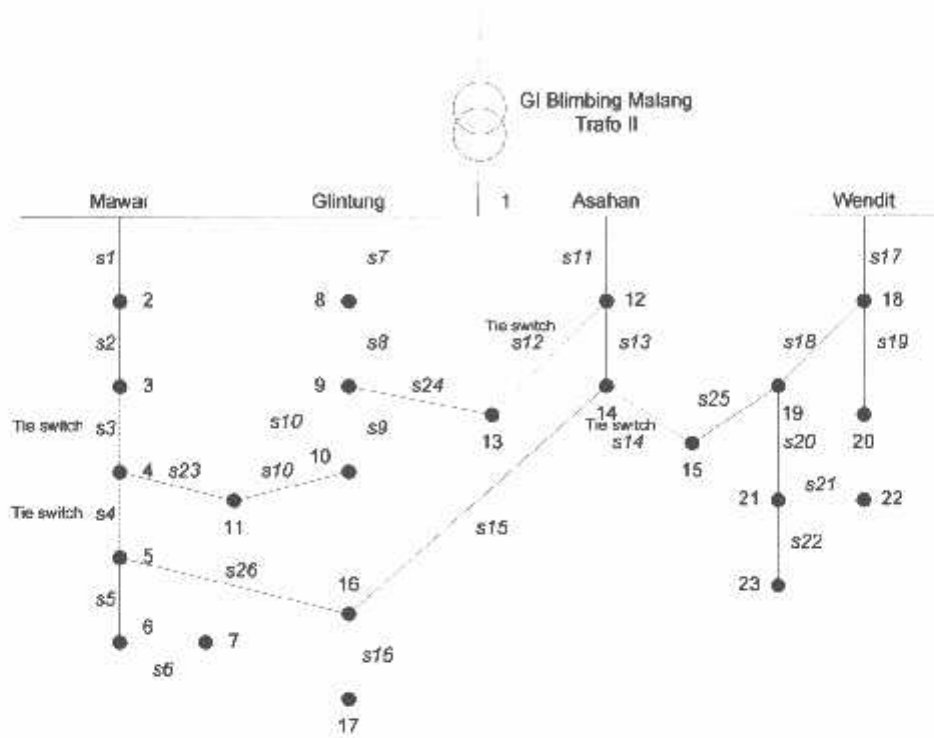
Dengan memasukkan data jaringan distribusi pada program simulasi rekonfigurasi jaringan untuk menghitung aliran daya dan mencari konfigurasi jaringan yang baru dengan metode *Genetic Algorithm*. Sesuai dengan hasil rekonfigurasi pada tabel 4-7 didapat kondisi *switch 3*, *switch 4*, *switch 12* dan *switch 14* dalam status terbuka (*Normally Open*), sehingga jaringan mesh yang telah ada akan berubah menjadi konfigurasi jaringan radial *open loop* baru.

**Tabel 4-7**  
**Hasil Kombinasi Optimal *Switch***

No.	Bus Pangkal	Bus Ujung	Posisi <i>switch open</i>
1.	3	4	3
2.	4	5	4
3.	12	13	12
4.	14	15	14

Konfigurasi jaringan radial *open loop* baru dapat dilihat pada gambar 4-4

berikut :



**Gambar 4-4**  
**Konfigurasi Jaringan Radial Setelah Rekonfigurasi**

Konfigurasi jaringan radial *open loop* setelah dilakukan rekonfigurasi tersebut merupakan konfigurasi jaringan radial *open loop* terbaik dari berbagai kemungkinan konfigurasi jaringan yang ada dengan menggunakan parameter *Genetic Algorithm* yang berbeda dengan parameter yang telah digunakan sebelumnya. Hasil minimalisasi rugi-rugi daya yang diperoleh merupakan minimalisasi rugi-rugi daya terbaik dari minimalisasi rugi-rugi daya yang ada. Kombinasi optimal switching didapat dengan durasi perhitungan sekitar 0,16 detik. Hasil aliran daya setelah proses rekonfigurasi dapat dilihat dalam tabel 4-8.



Tabel 4-8

## Hasil Perhitungan Aliran Daya Setelah Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs (pu)	Sudut (pu)	MW	MVAR	MW	MVAR
1	1	0	10,428	7,921	0	0
2	0,98391	-0,33381	-	-	1,829	1,372
3	0,98103	-0,39465	-	-	0,596	0,447
4	0,99012	-0,20561	-	-	0,897	0,672
5	0,97184	-0,58514	-	-	0,737	0,553
6	0,97107	-0,60173	-	-	0,436	0,327
7	0,97002	-0,62420	-	-	0,523	0,392
8	0,99614	-0,07984	-	-	0,480	0,360
9	0,99347	-0,13536	-	-	0,616	0,462
10	0,99200	-0,16611	-	-	0,283	0,212
11	0,99033	-0,20105	-	-	0,058	0,043
12	0,98862	-0,23266	-	-	0,366	0,274
13	0,99346	-0,13560	-	-	0,056	0,042
14	0,98176	-0,37535	-	-	0,627	0,471
15	0,99171	-0,17158	-	-	0,059	0,045
16	0,97237	-0,57379	-	-	0,570	0,428
17	0,97228	-0,57579	-	-	0,596	0,447
18	0,99659	-0,07033	-	-	0,063	0,048
19	0,99179	-0,16996	-	-	0,114	0,085
20	0,99463	-0,11123	-	-	0,220	0,165
21	0,98965	-0,21464	-	-	0,249	0,187
22	0,98902	-0,22798	-	-	0,367	0,275
23	0,98658	-0,27926	-	-	0,558	0,419

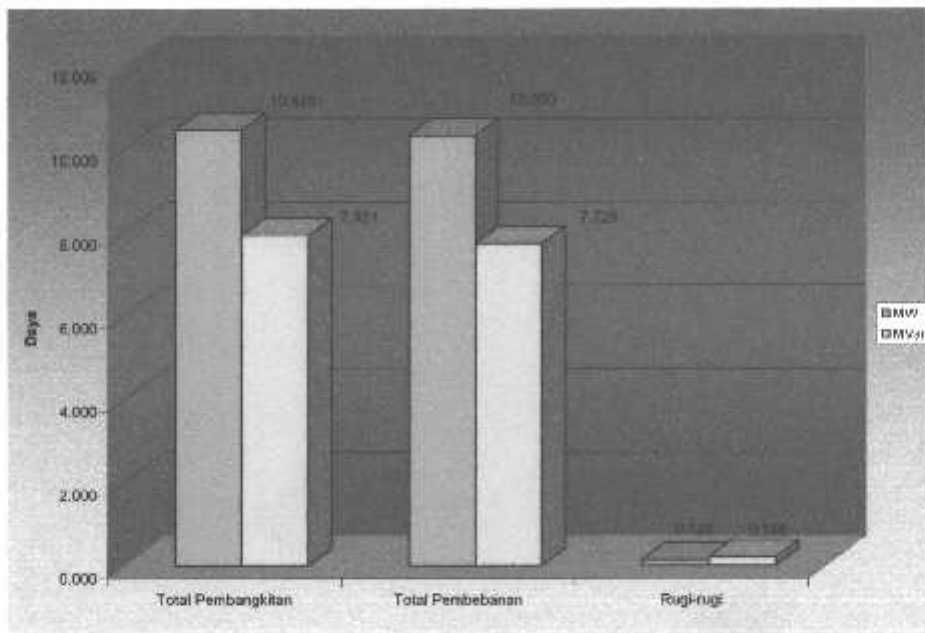
Tabel 4-9

## Hasil Perhitungan Aliran daya (Setelah Rekonfigurasi)

Total Pembangkitan		Total Pembebanan		Total Rugi Saluran	
P (MW)	Q (MVAR)	P (MW)	Q (MVAR)	P (MW)	Q (MVAR)
10,428	7,921	10,300	7,725	0,128	0,196

**Grafik 4-2**

**Grafik Hasil Perhitungan Aliran Daya Setelah Rekonfigurasi**



#### **4.7. Analisis Perbandingan Hasil Perhitungan Sebelum dan Setelah Rekonfigurasi Menggunakan Metode *Genetic Algorithm*.**

##### **4.7.1. Analisis Hasil Perhitungan Tegangan Dan Sudut Fasa Tegangan**

Pada proses perhitungan, semua data yang dipergunakan dirubah dalam satuan per-unit. Dimana nilai dasar yang digunakan adalah 20 kV untuk tegangan dan 30 MVA untuk daya.

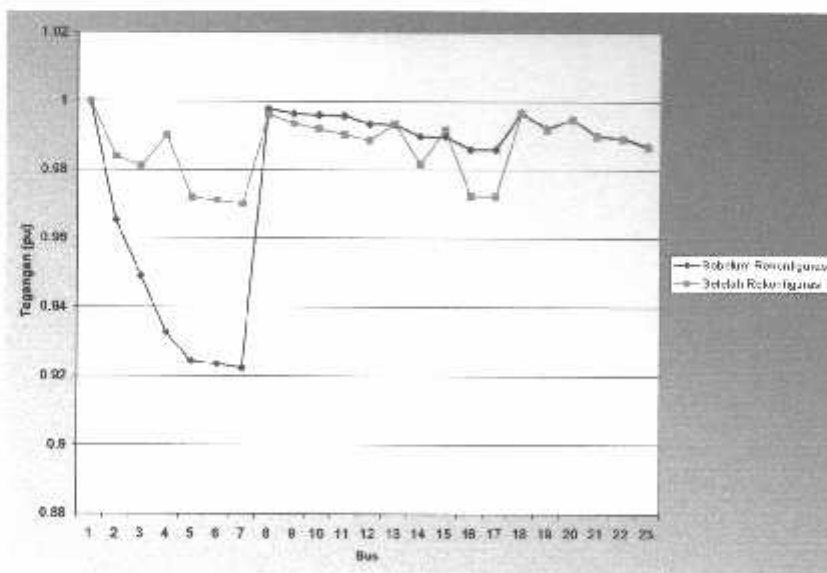
Dengan mengacu pada 4 penyulang GI Blimbing malang yang memiliki 23 bus (bus 1 sebagai referensi) dan 22 saluran, maka asumsi awal dari nilai  $V_p^{(0)} = 1$  pu dan  $\delta_p^{(0)} = 0^\circ$ . Berikut ini adalah tabel perbandingan tingkat tegangan dan sudut tegangan pada tiap bus.

**Tabel 4-10**  
**Perbandingan Tingkat Tegangan dan Sudut Tegangan Pada Tiap Bus**

Bus	Sebelum Rekonfigurasi		Setelah Rekonfigurasi	
	Abs (pu)	Sudut (pu)	Abs (pu)	Sudut (pu)
1	1	0	1	0
2	0,96516	-0,70402	0,98391	-0,33381
3	0,94887	-1,04699	0,98103	-0,39465
4	0,93248	-1,40309	0,99012	-0,20561
5	0,92418	-1,58802	0,97184	-0,58514
6	0,92336	-1,60637	0,97107	-0,60173
7	0,92226	-1,63123	0,97002	-0,62420
8	0,99769	-0,04793	0,99614	-0,07984
9	0,99636	-0,07563	0,99347	-0,13536
10	0,99596	-0,08403	0,99200	-0,16611
11	0,99586	-0,08612	0,99033	-0,20105
12	0,99339	-0,13660	0,98862	-0,23266
13	0,99317	-0,14128	0,99346	-0,13560
14	0,98980	-0,21140	0,98176	-0,37535
15	0,98972	-0,21312	0,99171	-0,17158
16	0,98603	-0,29050	0,97237	-0,57379
17	0,98594	-0,29244	0,97228	-0,57579
18	0,99671	-0,06777	0,99659	-0,07033
19	0,99213	-0,16297	0,99179	-0,16996
20	0,99475	-0,10865	0,99463	-0,11123
21	0,98999	-0,20762	0,98965	-0,21464
22	0,98936	-0,22095	0,98902	-0,22798
23	0,98691	-0,27220	0,98658	-0,27926

**Grafik 4-3**

**Grafik Perbandingan Tegangan Sebelum dan Sesudah Rekonfigurasi**

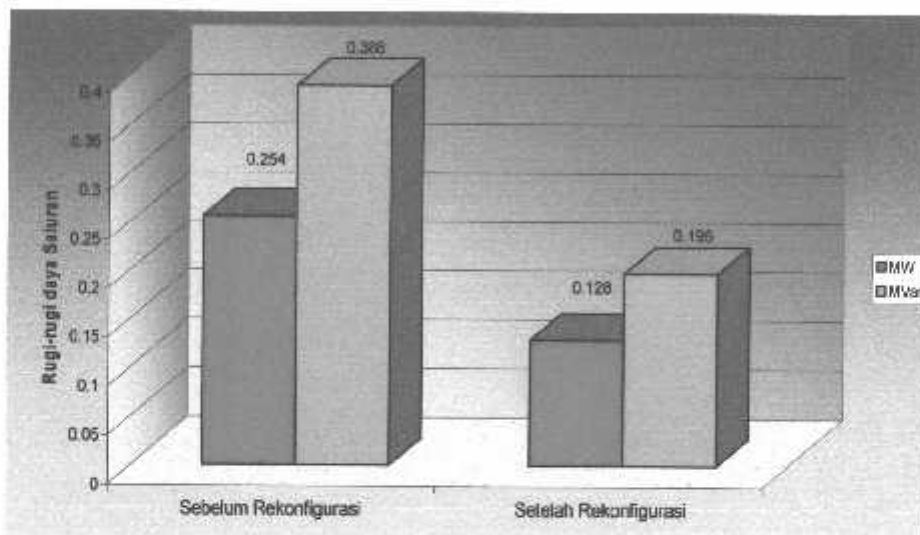


#### 4.7.2. Perbandingan Tingkat Rugi-rugi Daya Pada Saluran

Total Rugi-rugi daya saluran sebelum Rekonfigurasi adalah 0,254 MW dan 0,388 MVAR, sedangkan total rugi-rugi daya saluran setelah Rekonfigurasi adalah 0,128 MW dan 0,196 MVAR. Dari hasil perhitungan dapat dilihat bahwa selisih total rugi-rugi daya turun sebesar 0,126 MW dan 0,192 MVAR.

Grafik 4-4

Grafik Perbandingan Total Rugi Daya Sebelum dan Sesudah Rekonfigurasi

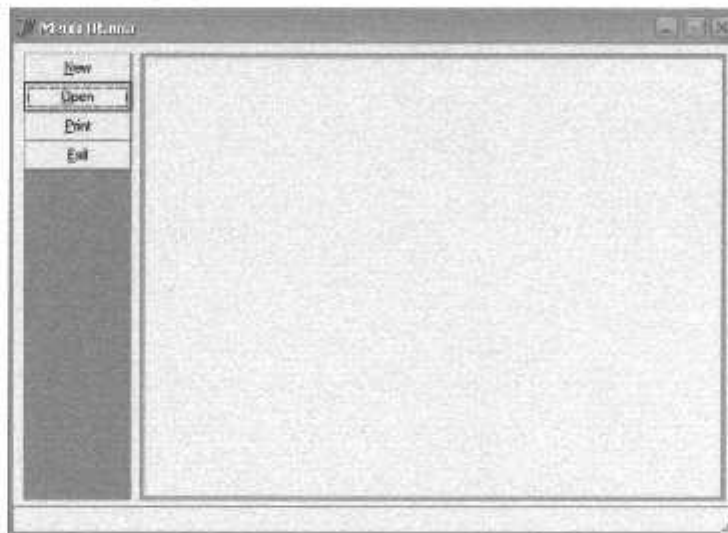


## 4.8. Hasil dan Analisa Hasil

### 4.8.1. Tampilan Program

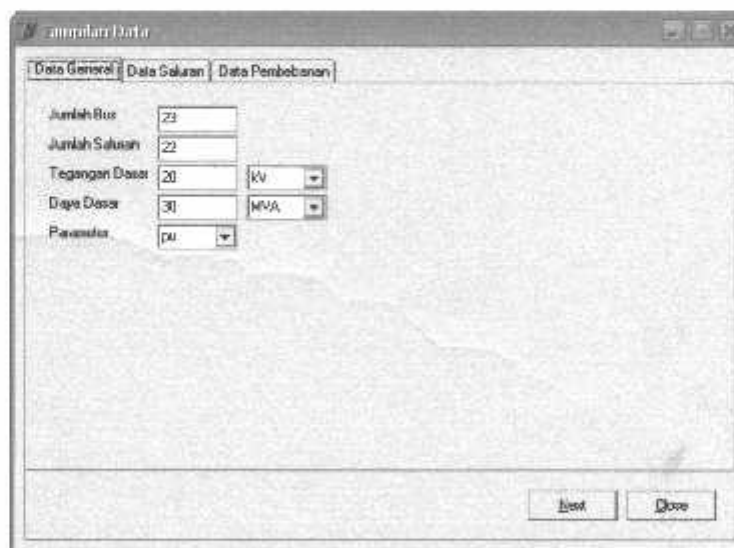
Program dalam skripsi ini dijalankan dengan menggunakan program Borland Delphi versi 7.0 dan diaplikasikan pada komputer menggunakan sistem operasi Windows XP, PC AMD Athlon XP 2,2 GHz, memori 256 Mb, Mengenai jalannya program ikuti prosedur program sebagai berikut :

1. Tampilan utama dari program.



Gambar 4-5 Tampilan Menu Utama Program

2. Tekan tombol *open* untuk membuka file yang sudah tersimpan.



Gambar 4-6 Tampilan Input Data General G.I Blimbing

3. Kemudian pilih tombol data saluran.

Tampilan Data							
Data General		Data Saluran		Data Pembebanan			
No	Dist	Ke	R (pu)	X (pu)	Lo (pu)	Ti	Kapasitas (A)
1	1	2	1.2142	1.896	0	0	
2	2	3	0.0032	1.3501	0	0	
3	3	4	1.0886	1.8687	0	0	
4	4	5	0.8425	1.288	0	0	
5	5	6	0.1464	0.2238	0	0	
6	6	7	0.3629	0.5546	0	0	
7	1	8	0.2586	0.4694	0	0	
8	8	9	0.2581	0.2948	0	0	
9	9	10	0.2197	0.3358	0	0	
10	10	11	0.3225	0.4561	0	0	
11	1	12	0.5343	0.9177	0	0	
12	12	13	0.7444	1.1379	0	0	
13	12	14	0.2653	0.544	0	0	
14	14	15	0.2663	0.381	0	0	
15	14	16	0.5935	0.3072	0	0	

Gambar 4-7 Data Saluran

4. Kemudian pilih tombol data pembebanan.

Tampilan Data								
Data General		Data Saluran		Data Pembebanan				
No	Abs V (pu)	Sud V (deg)	Pgen (MW)	Qgen (MVAR)	Pload (MW)	Qload (MVAR)	Cap	Typ Bus
1	1	0	0	0	0	0	0	1
2	1	0	0	0	1.0294	1.3721	0	3
3	1	0	0	0	0.9362	0.4472	0	3
4	1	0	0	0	0.8966	0.6724	0	3
5	1	0	0	0	0.7369	0.6527	0	3
6	1	0	0	0	0.4398	0.3268	0	3
7	1	0	0	0	0.9231	0.3923	0	3
8	1	0	0	0	0.4902	0.3601	0	3
9	1	0	0	0	0.6158	0.4619	0	3
10	1	0	0	0	0.2826	0.212	0	3
11	1	0	0	0	0.0570	0.0432	0	3
12	1	0	0	0	0.3826	0.2742	0	3
13	1	0	0	0	0.0566	0.0417	0	3
14	1	0	0	0	0.6274	0.4755	0	3
15	1	0	0	0	0.0593	0.0445	0	3

Gambar 4-8 Data Pembebanan

5. Tekan tombol hitung *load flow* dengan metode *Newton Rapshon* untuk melihat hasil perhitungan *loadflow* pada kondisi awal.

Bus	Abs V (pu)	Sud V (deg)	Pgen (MW)	Dgen (MW)	Flood (MVAR)	Gload (MVAR)
1	1.0000	0.0000	10.554	8.113	0.000	0.000
2	0.96516	-0.70402	0.000	0.000	1.825	1.372
3	0.94887	-1.24639	0.000	0.000	0.596	0.447
4	0.93249	-1.40309	0.000	0.000	0.867	0.672
5	0.92410	-1.58802	0.000	0.000	0.737	0.553
6	0.92336	-1.58637	0.000	0.000	0.436	0.327
7	0.92226	-1.63123	0.000	0.000	0.520	0.392
8	0.91769	-0.04793	0.000	0.000	0.480	0.360
9	0.89636	-0.07563	0.000	0.000	0.616	0.462
10	0.99696	0.88403	0.000	0.000	0.293	0.212
11	0.99696	-0.08612	0.000	0.000	0.058	0.043
12	0.99338	-0.13660	0.000	0.000	0.366	0.274
13	0.99317	0.14128	0.000	0.000	0.075	0.047

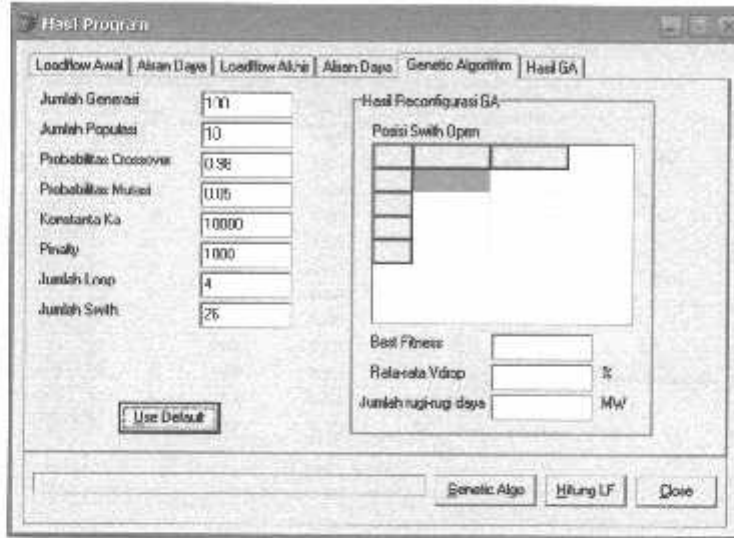
Gambar 4-9 Hasil Perhitungan Aliran Daya ( Sebelum Rekonfigurasi )

6. Pilih tombol aliran daya untuk melihat hasil perhitungan *loadflow* pada kondisi awal.

No	Dari	Ke	P (MW/rt)	Q (MVAR)	Anz
1	1	2	5.237	4.059	261
2	1	8	1.440	1.082	71
3	1	12	2.296	1.735	112
4	1	13	1.583	1.197	75
5	2	1	-6.103	-3.883	-261
6	2	3	3.274	2.522	167
7	3	2	-3.233	-2.460	-167
8	3	4	2.637	2.003	136
9	4	3	-2.604	-1.962	-136
10	4	5	1.707	1.290	89
11	5	4	-1.695	-1.273	-89
12	5	6	0.960	0.720	50

Gambar 4-10 Hasil Perhitungan Aliran Daya ( Sebelum Rekonfigurasi )

7. Kemudian pilih tombol *Genetic Algorithm* dan *use default* untuk melihat parameter *Genetic Algorithm*.



**Gambar 4-11 Parameter GA**

8. Kemudian tekan tombol *Genetic Algorithm* untuk diproses lalu pilih tombol *Load Flow* akhir untuk mengetahui kondisi akhir jaringan.

Bus	Abs V (pu)	Sud V (deg)	Pgen (MW)	Qgen (MW)	Pload (MVAR)	Qload (MVAR)
1	1.0000	0.0000	10.429	7.921	0.000	0.000
2	0.98391	-1.33081	0.000	0.000	1.629	1.372
3	0.98703	-0.39435	0.000	0.000	0.596	0.447
4	0.99012	-0.20561	0.000	0.000	0.697	0.672
5	0.97184	-0.58514	0.000	0.000	0.737	0.553
6	0.97107	-0.60173	0.000	0.000	0.436	0.327
7	0.97002	-0.62420	0.000	0.000	0.523	0.392
8	0.98614	-0.07904	0.000	3.000	0.490	0.360
9	0.99347	-0.13636	0.000	3.000	0.616	0.462
10	0.99200	0.16611	0.000	0.000	0.293	0.272
11	0.99033	0.20105	0.000	0.000	0.059	0.043
12	0.98862	0.23265	0.000	0.000	0.366	0.274
13	0.99346	-0.13660	0.000	0.000	0.056	0.042

**Gambar 4-12 Hasil Perhitungan Aliran Daya ( Setelah Rekonfigurasi )**



#### 4.9. Hasil Perhitungan Aliran Daya Pada Kondisi Awal

Berikut ini adalah tabel hasil perhitungan aliran daya sebelum rekonfigurasi (kondisi awal).

##### 1. Penyulang Mawar

Tabel 4-11  
Hasil Perhitungan Aliran Daya Tiap Bus Sebelum Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
1	1	0	10,544	8,113	0	0
2	0,96516	-0,70402	-	-	1,829	1,372
3	0,94887	-1,04699	-	-	0,596	0,447
4	0,93248	-1,40309	-	-	0,897	0,672
5	0,92418	-1,58802	-	-	0,737	0,553
6	0,92336	-1,60637	-	-	0,436	0,327
7	0,92226	-1,63123	-	-	0,523	0,392

##### 2. Penyulang Glintung

Tabel 4-12  
Hasil Perhitungan Aliran Daya Tiap Bus Sebelum Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
8	0,99769	-0,04793	-	-	0,480	0,360
9	0,99636	-0,07563	-	-	0,616	0,462
10	0,99596	-0,08403	-	-	0,283	0,212
11	0,99586	-0,08612	-	-	0,058	0,043

##### 3. Penyulang Asahan

Tabel 4-13  
Hasil Perhitungan Aliran Daya Tiap Bus Sebelum Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
12	0,99339	-0,13660	-	-	0,366	0,274
13	0,99317	-0,14128	-	-	0,056	0,042
14	0,98980	-0,21140	-	-	0,627	0,471
15	0,98972	-0,21312	-	-	0,059	0,045
16	0,98603	-0,29050	-	-	0,570	0,428
17	0,98594	-0,29244	-	-	0,596	0,447

#### 4. Penyulang Wendit

**Tabel 4-14**  
**Hasil Perhitungan Aliran Daya Tiap Bus Sebelum Rekonfigurasi**

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
18	0,99671	-0,06777	-	-	0,063	0,048
19	0,99213	-0,16297	-	-	0,114	0,085
20	0,99475	-0,10865	-	-	0,220	0,165
21	0,98999	-0,20762	-	-	0,249	0,187
22	0,98936	-0,22095	-	-	0,367	0,275
23	0,98691	-0,27220	-	-	0,558	0,419

**Tabel 4-15**  
**Hasil Total Perhitungan Aliran Daya Sebelum Rekonfigurasi**

Total Pembangkitan		Total Pembebanan		Total Rugi Saluran	
P (MW)	Q (MVAR)	P (MW)	Q (MVAR)	P (MW)	Q (MVAR)
10,554	8,113	10,300	7,725	0,254	0,388

#### 4.10. Hasil Perhitungan Aliran Daya Setelah Rekonfigurasi

Berikut ini adalah tabel hasil perhitungan aliran daya setelah rekonfigurasi.

##### 1. Penyulang Mawar

**Tabel 4-16**  
**Hasil Perhitungan Aliran Daya Tiap Bus Setelah Rekonfigurasi**

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
1	1	0	10,428	7,921	0	0
2	0,98391	-0,33381	-	-	1,829	1,372
3	0,98103	-0,39465	-	-	0,596	0,447
4	0,99012	-0,20561	-	-	0,897	0,672
5	0,97184	-0,58514	-	-	0,737	0,553
6	0,97107	-0,60173	-	-	0,436	0,327
7	0,97002	-0,62420	-	-	0,523	0,392

## 2. Penyulang Glitung

Tabel 4-17  
Hasil Perhitungan Aliran Daya Tiap Bus Setelah Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
8	0,99614	-0,07984	-	-	0,480	0,360
9	0,99347	-0,13536	-	-	0,616	0,462
10	0,99200	-0,16611	-	-	0,283	0,212
11	0,99033	-0,20105	-	-	0,058	0,043

## 3. Penyulang Asahan

Tabel 4-18  
Hasil Perhitungan Aliran Daya Tiap Bus Setelah Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
12	0,98862	-0,23266	-	-	0,366	0,274
13	0,99346	-0,13560	-	-	0,056	0,042
14	0,98176	-0,37535	-	-	0,627	0,471
15	0,99171	-0,17158	-	-	0,059	0,045
16	0,97237	-0,57379	-	-	0,570	0,428
17	0,97228	-0,57579	-	-	0,596	0,447

## 4. Penyulang Wendit

Tabel 4-19  
Hasil Perhitungan Aliran Daya Tiap Bus Setelah Rekonfigurasi

Bus	Tegangan		Daya Pembangkitan		Beban	
	Abs(pu)	Sudut(pu)	MW	MVAR	MW	MVAR
18	0,99659	-0,07033	-	-	0,063	0,048
19	0,99179	-0,16996	-	-	0,114	0,085
20	0,99463	-0,11123	-	-	0,220	0,165
21	0,98965	-0,21464	-	-	0,249	0,187
22	0,98902	-0,22798	-	-	0,367	0,275
23	0,98658	-0,27926	-	-	0,558	0,419

**Tabel 4-20**  
**Hasil Total Perhitungan Aliran Daya Setelah Rekonfigurasi**

Total Pembangkitan		Total Pembebanan		Total Rugi Saluran	
P (MW)	Q (MVAR)	P (MW)	Q (MVAR)	P (MW)	Q (MVAR)
10,428	7,921	10,300	7,725	0,128	0,196

**4.11. Hasil Perbandingan Perhitungan Aliran Daya Sebelum dan Sesudah Rekonfigurasi**

Berikut ini adalah tabel perbandingan hasil total perhitungan aliran daya sebelum dan sesudah rekonfigurasi.

**Tabel 4-21**  
**Perbandingan Hasil Perhitungan Aliran Daya Sebelum dan Sesudah Rekonfigurasi**

Penyulang	Pembangkitan		Pembebanan		Rugi-rugi	
	Sebelum (MW)	Sesudah (MW)	Sebelum (MW)	Sesudah (MW)	Sebelum (MW)	Sesudah (MW)
Mawar Glintang Asahan Wendit	10,554	10,428	10,300	10,300	0,254	0,128

**DATA TRAF0 DISTRIBUSI  
UNIT JARINGAN MALANG**

**SEKSI : MAPPING**

**BULAN : MEI 2005**



**PT. PLN (PERSERO)**  
**AREA PELAYANAN DAN JARINGAN MALANG**  
**JL. JEND. BASUKI RACHMAD 100 MALANG**

**Tabel A-1**  
**Data Beban Tiap Seksi Penyulang Mawar**

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	186	150	53,87	80,81
	764	160	80,43	124,06
	588	160	65,84	105,34
	424	200	69,53	139,04
	270	150	103,52	155,29
	775	250	77,53	124,06
	744	160	80,43	128,68
	306	160	61,05	78,67
	370	150	0,00	0,00
	455	250	53,92	134,81
	369	150	71,61	107,42
	425	100	39,53	39,53
	605	100	50,5	50,5
	542	150	52,46	78,59
2	305	200	61,05	97,67
	230	160	69,87	139,74
	543	150	77,61	116,41
	741	160	45,38	72,6
	358	150	0,00	0,00
	38	160	54,84	87,75
	237	250	92,43	231,08
3	240	160	52,32	83,71
	22	150	97,12	145,67
	323	150	46,94	70,41
	150	100	65,48	65,48
	369	150	71,61	107,42
	777	250	59,04	147,61
	53	150	90,74	136,1
	265	150	53,86	80,78
	570	160	55,76	89,21
	100	100	23,98	23,98
	412	160	14,44	23,10
122	150	98,33	147,49	
4	615	250	66,42	166,06
	585	200	68,69	137,39
	284	150	62,77	94,16
	737	250	60,91	152,28
	920	50	77,09	38,54
	193	250	81,66	204,16
	253	150	85,68	128,52
5	706	100	7,45	7,45
	15	200	51,7	129,25
	21	200	60,54	121,08
	244	160	52,32	83,71
	36	250	81,33	203,32
6	133	150	0,00	0,00
	584	160	74,06	118,5
	230	160	69,87	139,74
	110	150	0,00	0,00

	358	150	0,00	0,00
	231	160	76,36	122,17
	32	100	74,99	74,99
	335	75	0,00	0,00
	235	200	35,86	71,72
	315	150	84,51	126,77
	729	160	0,00	0,00

**Tabel A-2**  
**Data Beban Tiap Seksi Penyulang Glantung**

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	196	250	69,47	138,94
	996	200	0,00	0,00
	575	160	51,03	51,03
	141	200	71,76	143,52
	93	250	46,22	46,22
	281	160	60,49	96,78
	51	250	49,50	123,75
2	148	250	49,5	123,36
	210	400	0,00	0,00
	16	250	67,94	169,84
	211	150	73,58	110,28
	40	250	85,76	214,4
	398	160	25,55	40,98
	153	160	69,35	110,96
3	165	200	70,3	140,61
	107	100	45,9	45,9
	762	100	0,00	0,00
	432	160	0,00	0,00
	142	250	66,71	166,77
	449	100	0,00	0,00
4	860	200	36,03	72,05
	114	160	0,00	0,00

**Tabel A-3**  
**Data Beban Tiap Seksi Penyulang Asahan**

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	267	100	80,24	109,44
	125	160	137,56	206,34
	124	100	33,99	67,98
	128	160	19,95	19,95
	757	630	0,00	0,00
	623	200	0,00	0,00
	622	100	0,00	0,00
2	735	160	33,34	53,33
	285	25	4,8	1,2
	752	200	34,2	68,3

3	582	160	0,00	0,00
	667	160	0,00	0,00
	354	160	85,46	128,18
	815	630	0,00	0,00
	234	100	66,83	66,83
	314	150	85,17	127,75
	3	200	59,63	119,25
	906	160	0,00	0,00
	354	160	85,46	128,18
	2	200	78,59	147,17
	910	100	0,00	0,00
	234	100	66,83	66,83
	624	630	0,00	0,00
	4	279	250	37,07
528		800	0,00	0,00
5	668	250	75	187,5
	865	200	75	150
	654	100	23	36,9
	88	100	75	187,5
	621	100	12,6	20,1
	660	160	6,8	10,9
	445	630	75	120
6	924	200	0,00	0,00
	726	160	29,56	47,84
	735	160	33,34	53,33
	913	100	0,00	0,00
	960	160	0,00	0,00
	372	250	102,24	264,02
	373	160	62,93	100,72
	697	160	93,05	150,22
	434	160	0,00	0,00
	694	160	0,00	0,00
	767	160	0,00	0,00
	734	160	8,90	13,39
	735	160	33,34	53,33
	875	160	31,24	40,59
	938	200	12,35	21,74
	939	100	0,00	0,00
899	1250	0,00	0,00	

**Tabel A-4**  
**Data Beban Tiap Seksi Penyulang Wendit**

Seksi	No. Trafo	Nominal (kVA)	Persen Beban (%)	Beban Trafo (kVA)
1	224	200	49,50	79,20
2	606	160	43,24	64,86
	28	160	49,35	77,38
3	763	160	126,26	202,02
	844	200	36,54	73,08
4	215	160	43,96	87,91
	577	250	89,50	223,74
5	727	160	73,84	118,14
	185	150	67,61	101,42



	254	150	64,08	0,00
	428	100	57,98	57,98
	469	630	0,00	0,00
	705	50	15,77	7,88
	714	160	34,53	55,25
	715	160	73,84	118,14
6	854	100	43,39	43,99
	537	200	60,06	120,12
	699	100	0,00	0,00
	955	200	18,09	36,19
	956	100	88,66	88,66
	822	100	45,15	90,31
	768	160	37,91	60,65
	769	160	63,85	102,15
	844	200	36,54	73,08
	864	200	27,50	55,00
	974	160	17,17	27,47

**Sumber : Data Trafo Distribusi Unit Jaringan Malang**

**PT. PLN ( Persero )**

**Seksi : Mapping**

**Bulan : Mei 2005**

**Tabel A-5**  
**Penghantar AAAC**

Penampang Nominal (mm <sup>2</sup> )	Jari-Jari (mm)	Urut	GMR (mm)	Impedansi Urutan Positif (Ω/km)	Impedansi Urutan Nol (Ω/km)
16	2,2563	7	1,6380	2,0161 + j0,4036	2,1641 + j1,6911
25	2,8203	7	2,0475	1,2903 - j0,3895	1,4384 + j1,6770
35	3,3371	7	2,4227	0,9217 - j0,3790	1,0697 + j1,6665
50	3,9886	7	2,8957	0,6452 + j0,3678	0,7932 + j1,6553
70	4,7193	7	3,4262	0,4608 + j0,3572	0,6088 + j1,6447
95	5,4979	19	4,1674	0,3396 + j0,3449	0,4876 + j1,6324
120	6,1791	19	4,6837	0,2688 + j0,3376	0,4168 + j1,6251
150	6,9084	19	5,2368	0,2162 + j0,3305	0,3631 + j1,6180
185	7,6722	19	5,8155	0,1744 + j0,3239	0,3224 + j1,6114
240	8,7386	19	6,6238	0,1344 + j0,3158	0,2824 + j1,6033

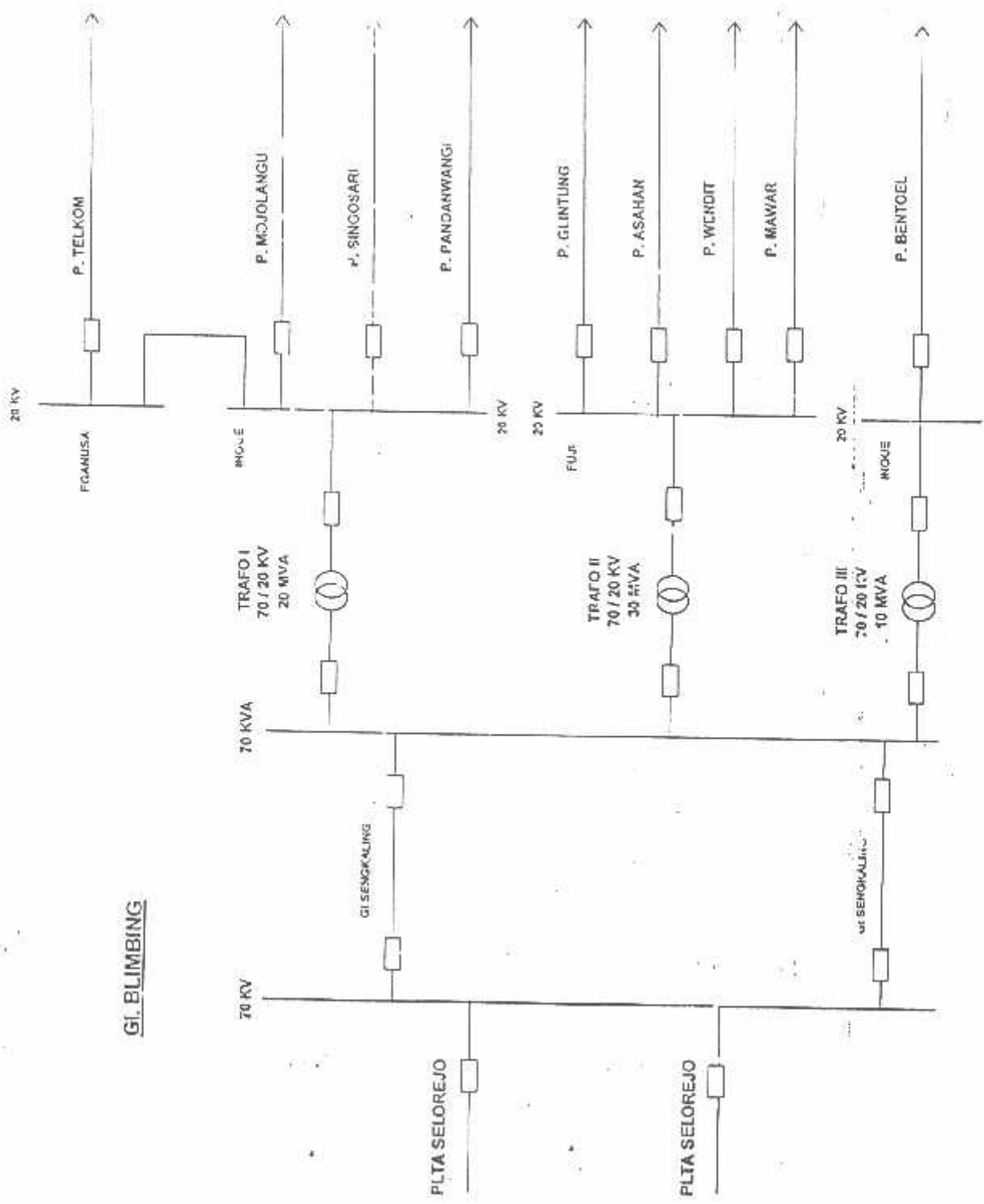
**Tabel A-6**  
**KHA Penghantar Campuran Aluminium Teltanjang (AAAC)**

Luas Penampang (mm <sup>2</sup> )	KHA terus menerus (A)
16	105
25	135
35	170
50 (7 kawat)	210
50 (19 kawat)	210
70	255
95	320
120	365
150	425
185	490
240	585
300	670
400	810
500	930
630	1075
800	1255
1000	1450

# SINGLE LINE DIAGRAM



GI. BLIMBING

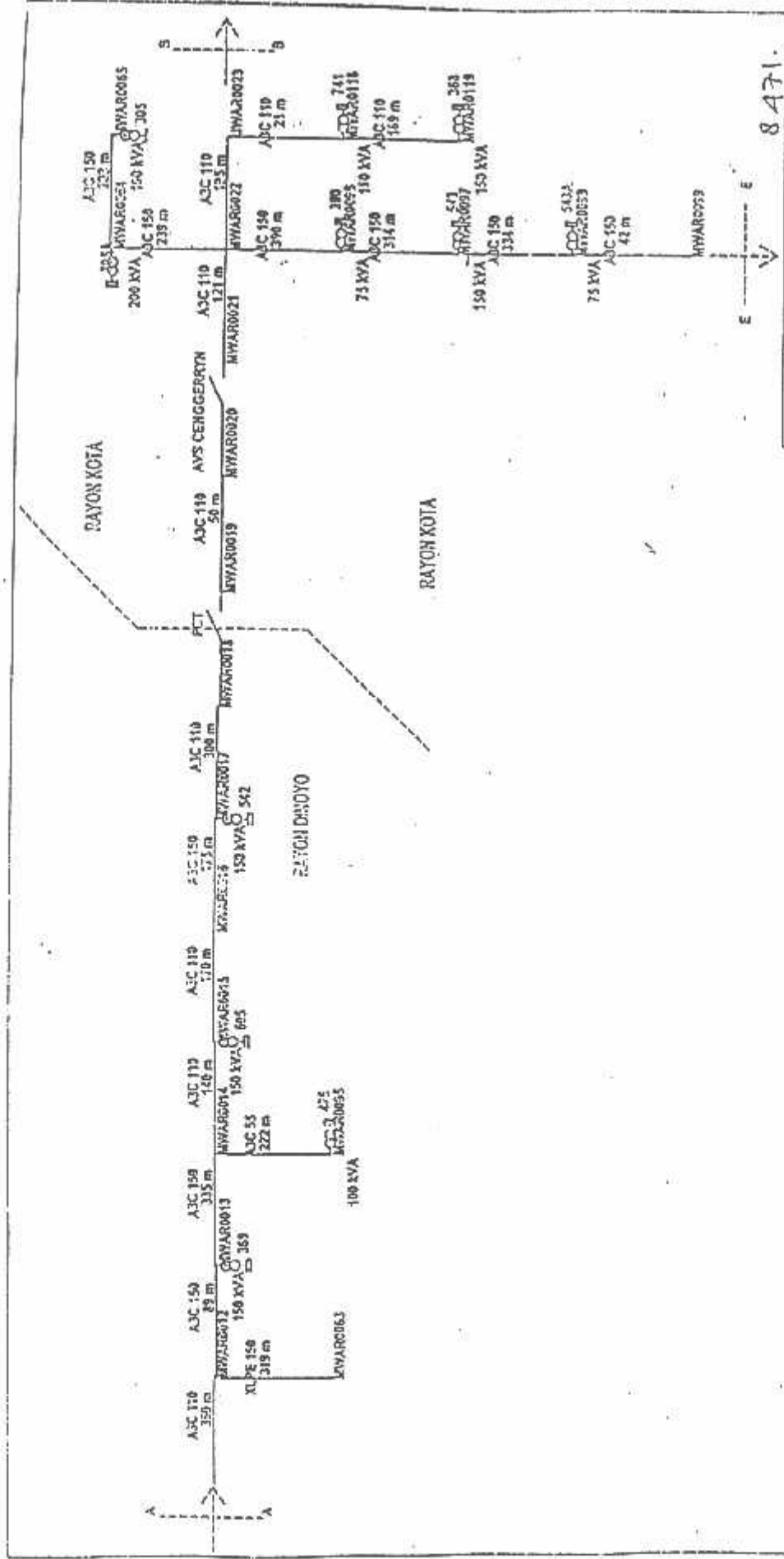



- ⊗ : TRANSFORMATOR
- : PEMUTUS
- : PENTULANE

		PT. PLN (PERSERO)	
		DISTRIBUSI JATIM	
		APRI MALARANG	
DIS. ARKIBAR	DIS. PERIKSAA	DIS. SETUJUT	
DPDST	DJUMADLO	ARIEFH	

SINGLE LINE DIAGRAM  
GARDU INDUK BLIMBING

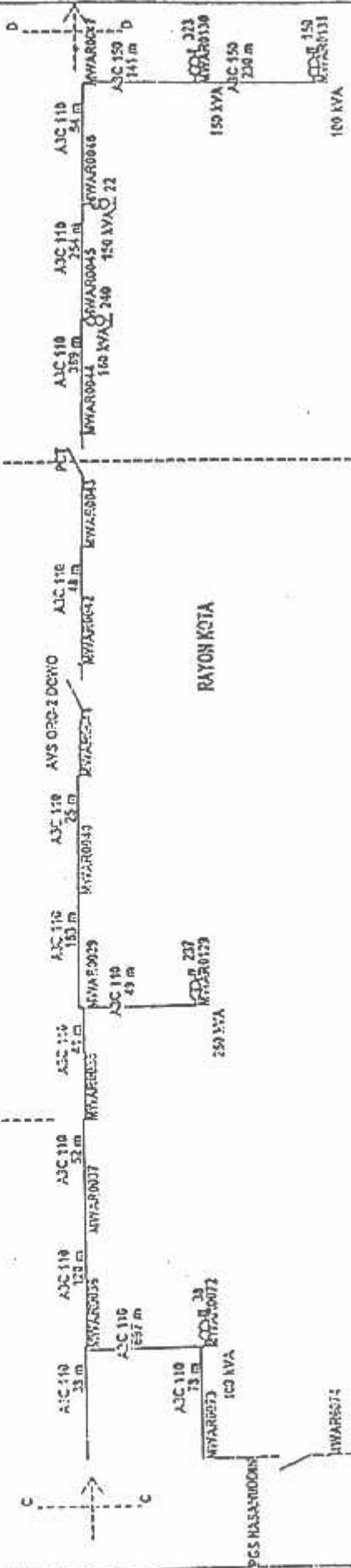




	
<b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b>	
Project Name: PENJURUAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PEMBANGUNAN MWAR	
Design By :	Drawing No.:
Drawn By :	
Approved By :	4162311001-L-11-05-MWAR-013-02-0
Checked By :	
Date :	SEPTEMBER, 2000 (Rev. 0)

F  
F

RAYON DINOYO

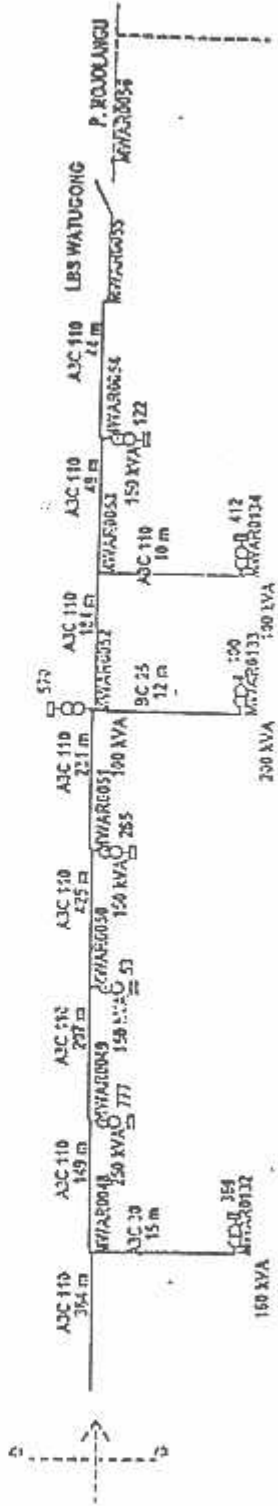


15.477




PT. PLN (PERSERO)  
JASAJENJIRING

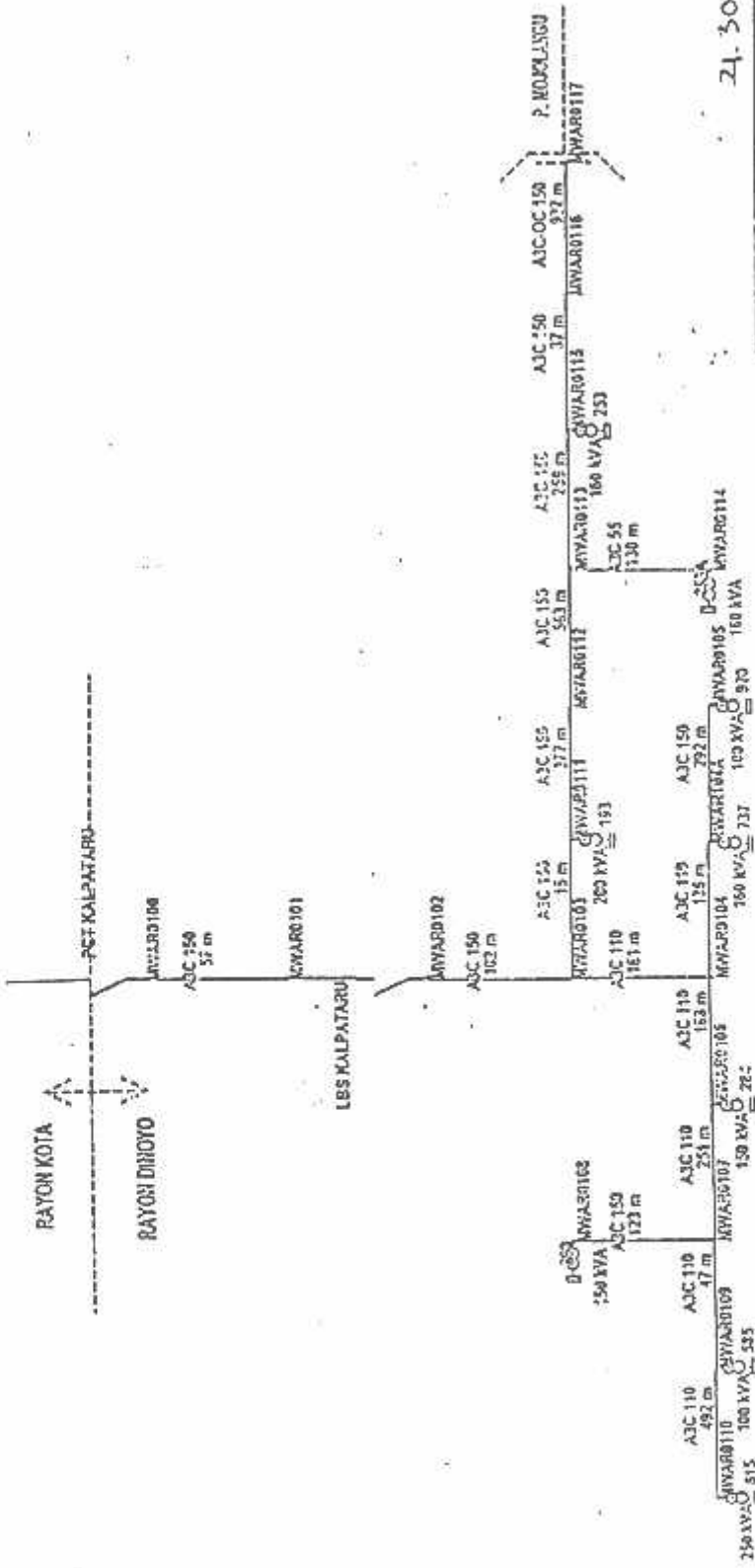
Project Name	
PT. PLN (PERSERO) DISTIBUSI JATIM - CABANG HALANG	
Title	
SINGLE LINE DIAGRAM PEMULANG MAYAR	
Design By :	Drawing No.:
Drawn By :	
Approved By :	K182-3100-L-31-05-HWAR-013-D-0
Checked By :	
Date	: SEPTEMBER, 2000 (Rev.: 0)



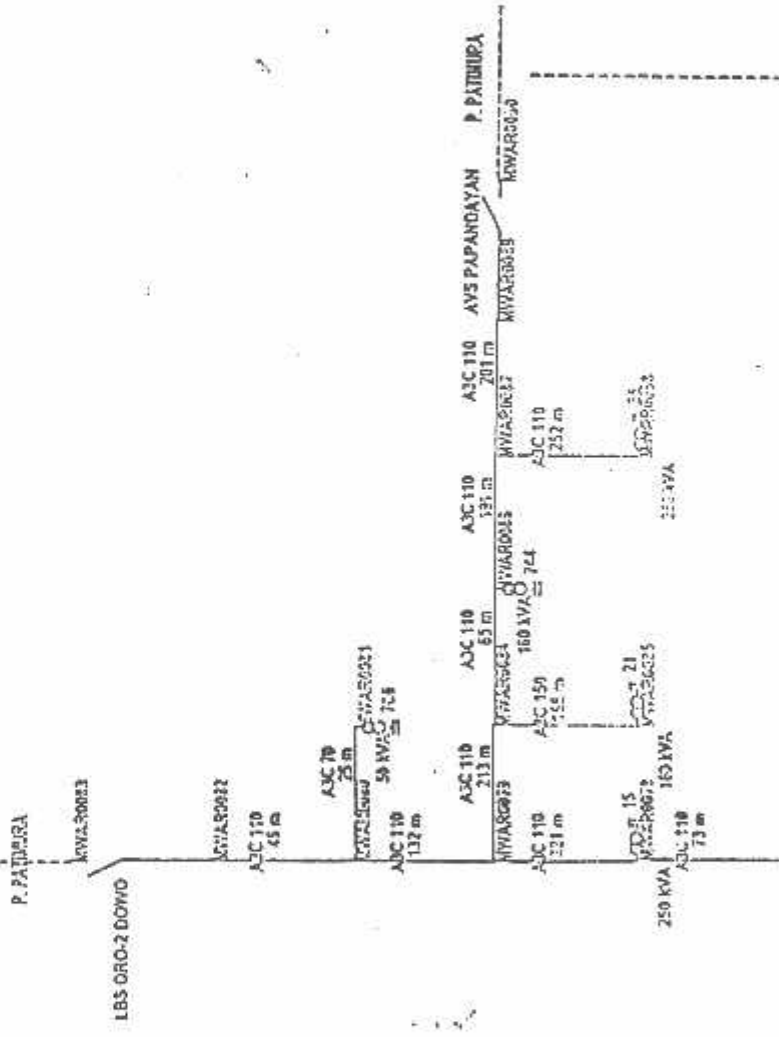
17.167

	
<b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b>	
Project Name:	PENDURIAN SUSUT RWA
Project Name:	PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG
Title:	SINGLE LINE DIAGRAM PENYULANG MAWAR
Design By :	
Drawn By :	
Approved By :	
Checked By :	
Date :	SEPTEMBER, 2000
Rev. :	0






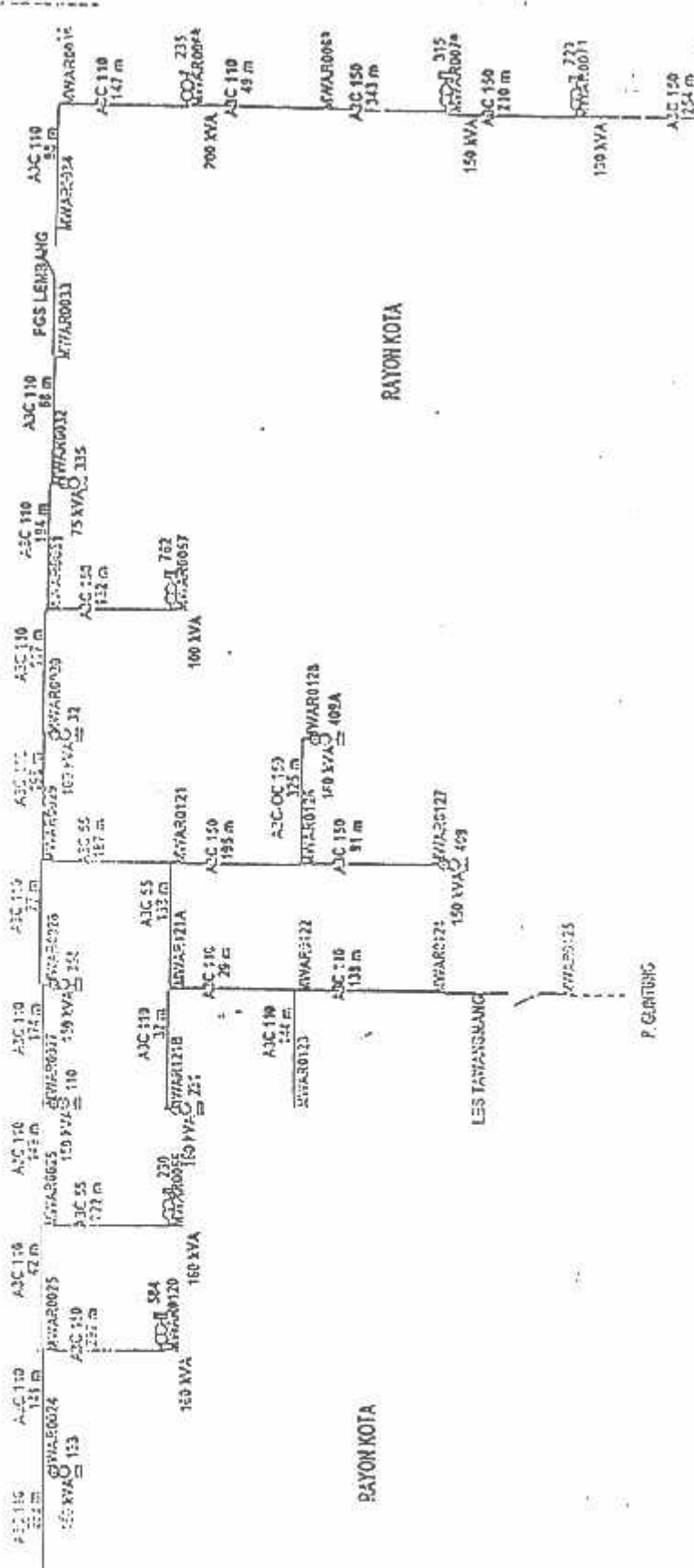
<b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b>	
Project Name:	PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CANGKUNG MALANG
Title:	SINGLE LINE DIAGRAM PENYULANG NAWAR
Design By :	Drawing No.:
Drawn By :	
Approved By :	4102-3100-L-51-05-MWAS-315-05-0
Checked By :	
Date :	SEPTEMBER, 2000 Rev.   0



RAYON DINOYO



	
<b>PT. PLN (PERSERO) JASA ENJINIRING</b>	
Project Name: <b>PENURUAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG</b>	
Title: <b>SINGLE LINE DIAGRAM PENYULANG MAVIAR 22, 98</b>	
Design By :	Drawing No.:
Drawn By :	
Approved By:	4162-3160-L-51-05-MWWS-015-02-0
Checked By :	
Date :	SEPTEMBER, 2000 (Rev.: 0)



12.553

**PT. PLN (PERSERO):**  
**JASA ENJINIRING**

**Project Name:** PENURUNAN SUSUT WH  
**PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG 1401**

**Title:** SINGLE LINE DIAGRAM PERMULANG SAWAR

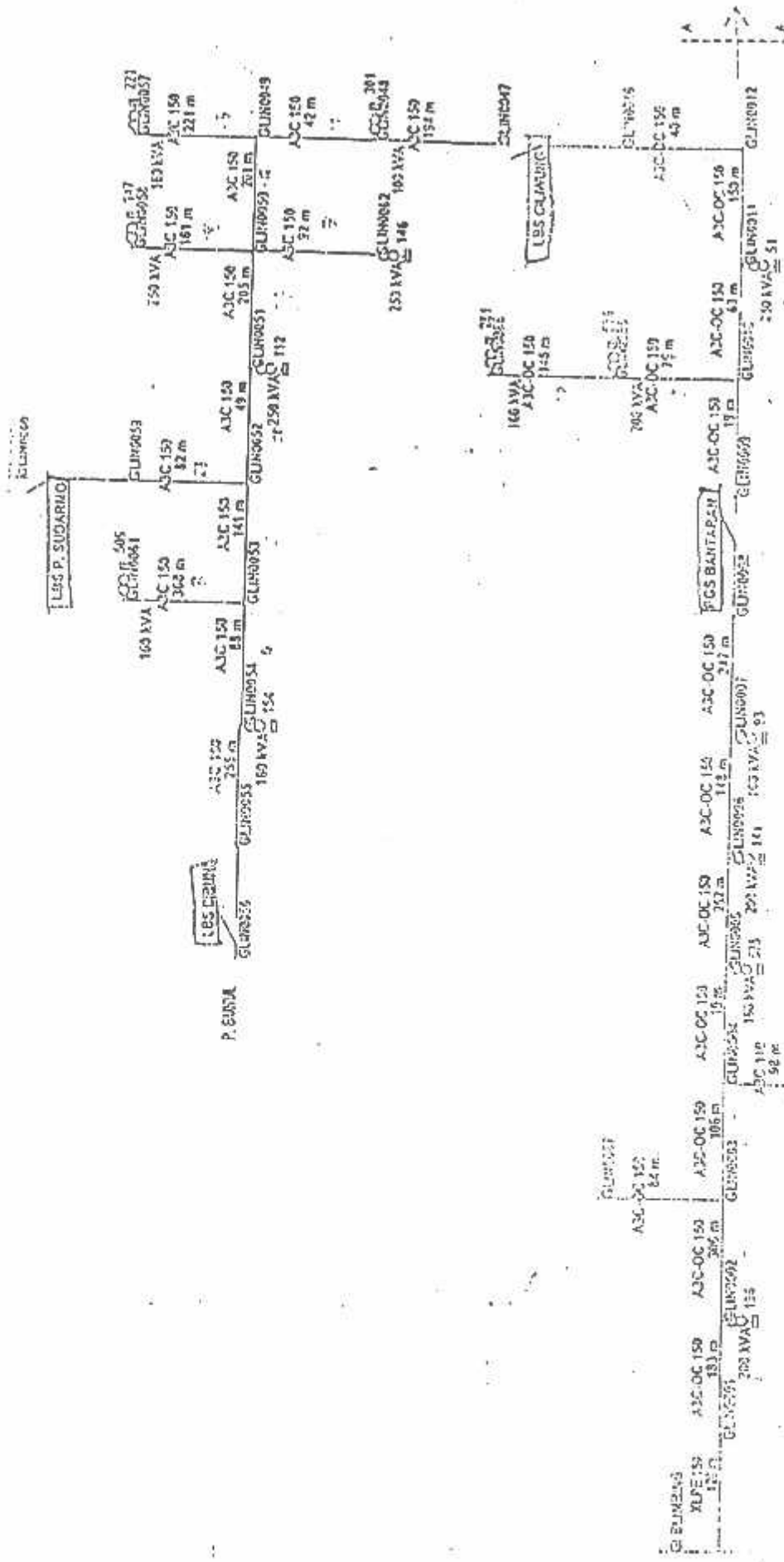
**Design By:** \_\_\_\_\_ **Design No.:** \_\_\_\_\_

**Drawn By:** \_\_\_\_\_

**Approved By:** \_\_\_\_\_ **Approved Date:** 11/03/2011

**Checked By:** \_\_\_\_\_

**Date:** SEPTEMBER, 2010 **Rev:** 0 1



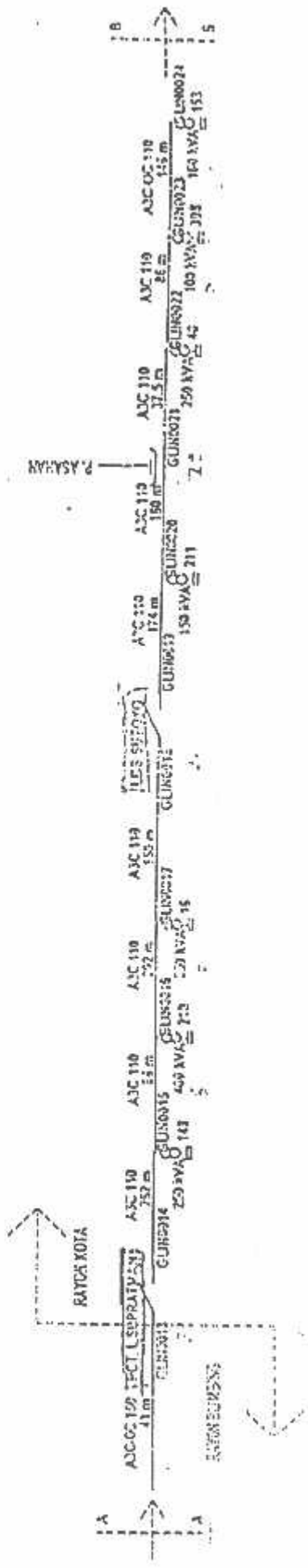
**PT. FLN (PERSERO)  
JASA ENJINIRING**


Project Name:  
 PENYULUHAN SUSUT KWH  
 PT. PULI (PERSERO) DISTRIBUSI JATIM - CARANG MAI A17  
 144

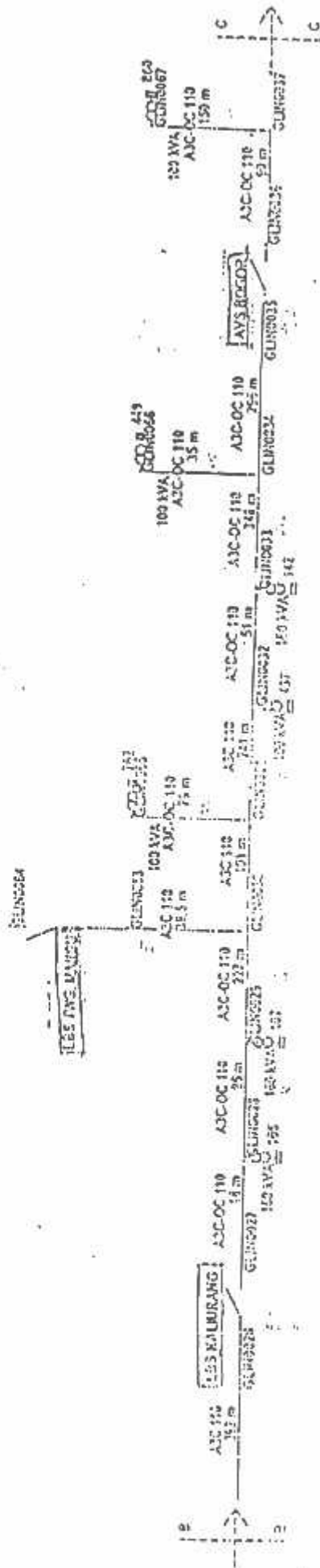
SINGLE LINE DIAGRAM PERYULUANG GUPITUNG

Checked By:  
 Approved By:  
 Checked By:

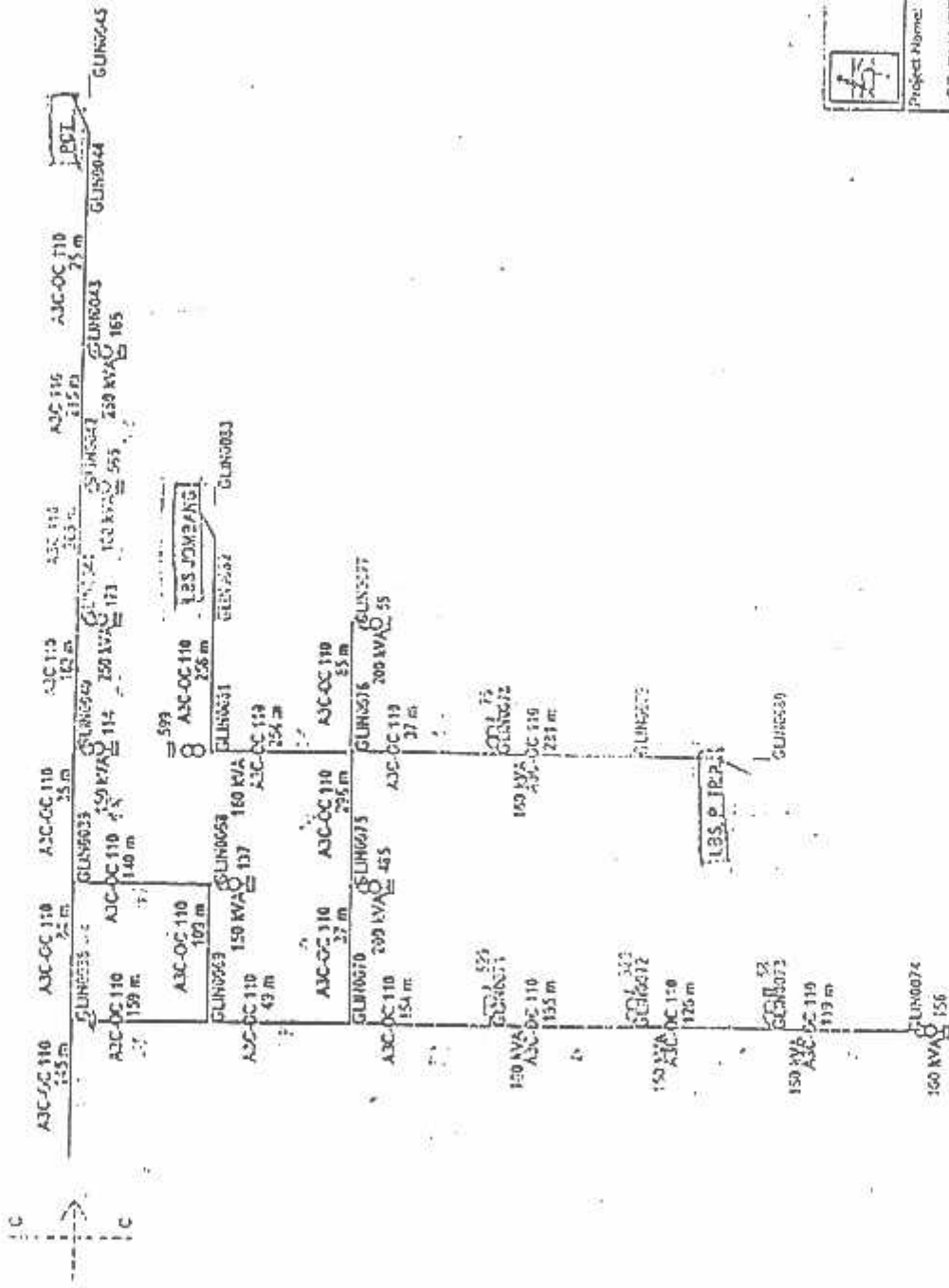
SEPTEMBER 2009



 <b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b>	
Project Name: <b>PENURUAN SUGUT WWH</b>	
Project: <b>PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG</b>	
Title: <b>SINGLE LINE DIAGRAM PENYULANG GUNUNG</b>	
Design By:	Drawing No:
Drawn By:	
Approved By:	4162-0166 1 51-026 02-016 02-0
Checked By:	
Date: <b>SEPTEMBER, 2000</b>	Rev: <b>0</b>



<b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b>	
Project Name:	PENUPUAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG
Tab:	SINGLE LIHE DIAGRAM PENYULANG GUNTUR 03
Design By:	.....
Drawn By:	.....
Approved By:	.....
Checked By:	.....
Date:	SEPTEMBER, 2000 Rev. 0



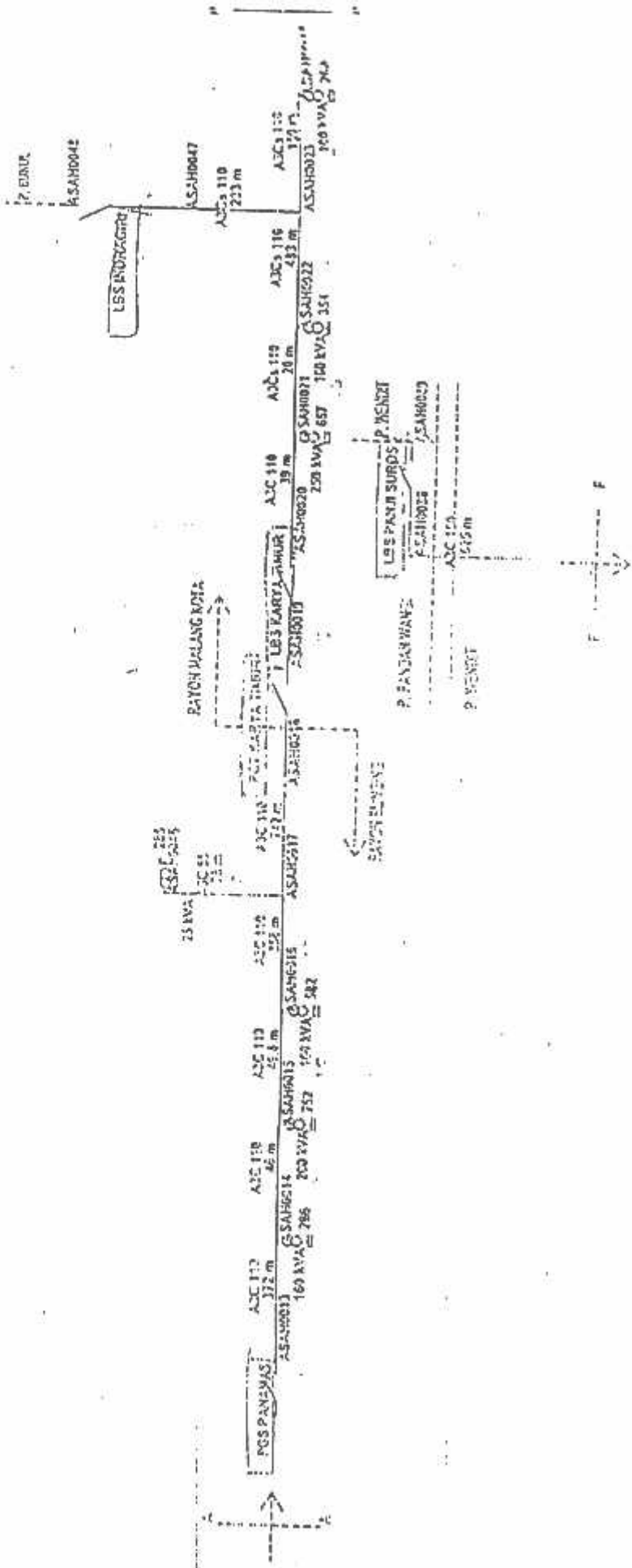
**PT. PLN (PERSERO)  
JASA ENJINIRING**

Project Name: **PENURUHAN SUSUT KWH  
PT. PLN (PERSERO) DISTRIBUSI JATIM - CADANGAN 10/2011**

Title: **SINGLE LINE DIAGRAM PERULANGAN GUNTUR**

Design By: \_\_\_\_\_  
 Drawn By: \_\_\_\_\_  
 Approved By: \_\_\_\_\_  
 Checked By: \_\_\_\_\_

Date: **SEPTEMBER 2011**

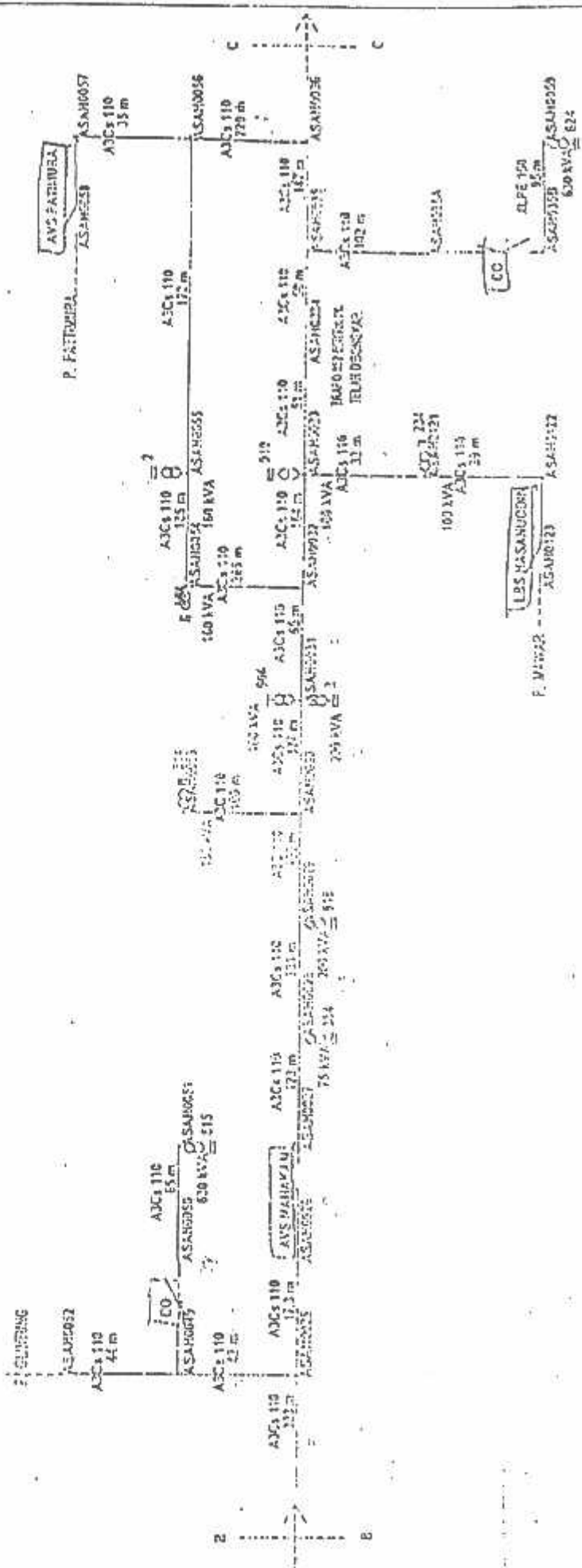


**PT. PLN (PERSERO)  
JASA ENJINIRING**

Project Name	PENURUAN SUSUT KVA PT. PLN (PERSERO) DISTRIBUTI JAWA BARAT
Design By	.....
Approved By	.....
Checked By	.....
Date	SEPTEMBER, 2000

Page 1





**PT. PLN (PERSERO)**  
**JASA ENJINIRING**

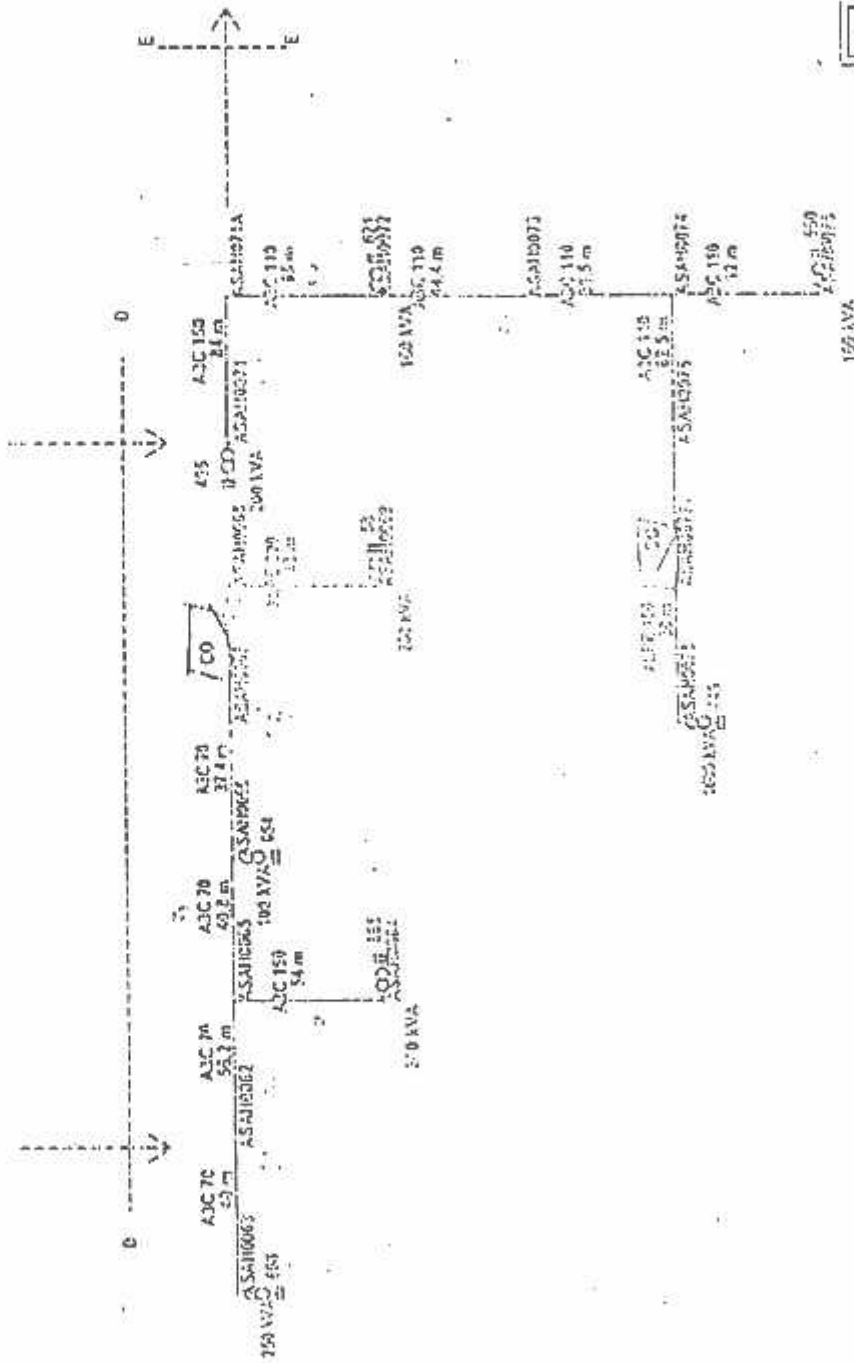
PT. PURBUNAH SUGUT WHT  
 PT. PULIPERSON DIGITRECUJ JATIM - CABANG MALANG

**SINGLE LINE DIAGRAM PERULANGAN ASATUWI**

Desain By: [Signature]  
 Drawn By: [Signature]  
 Checked By: [Signature]  
 Date: [Signature]

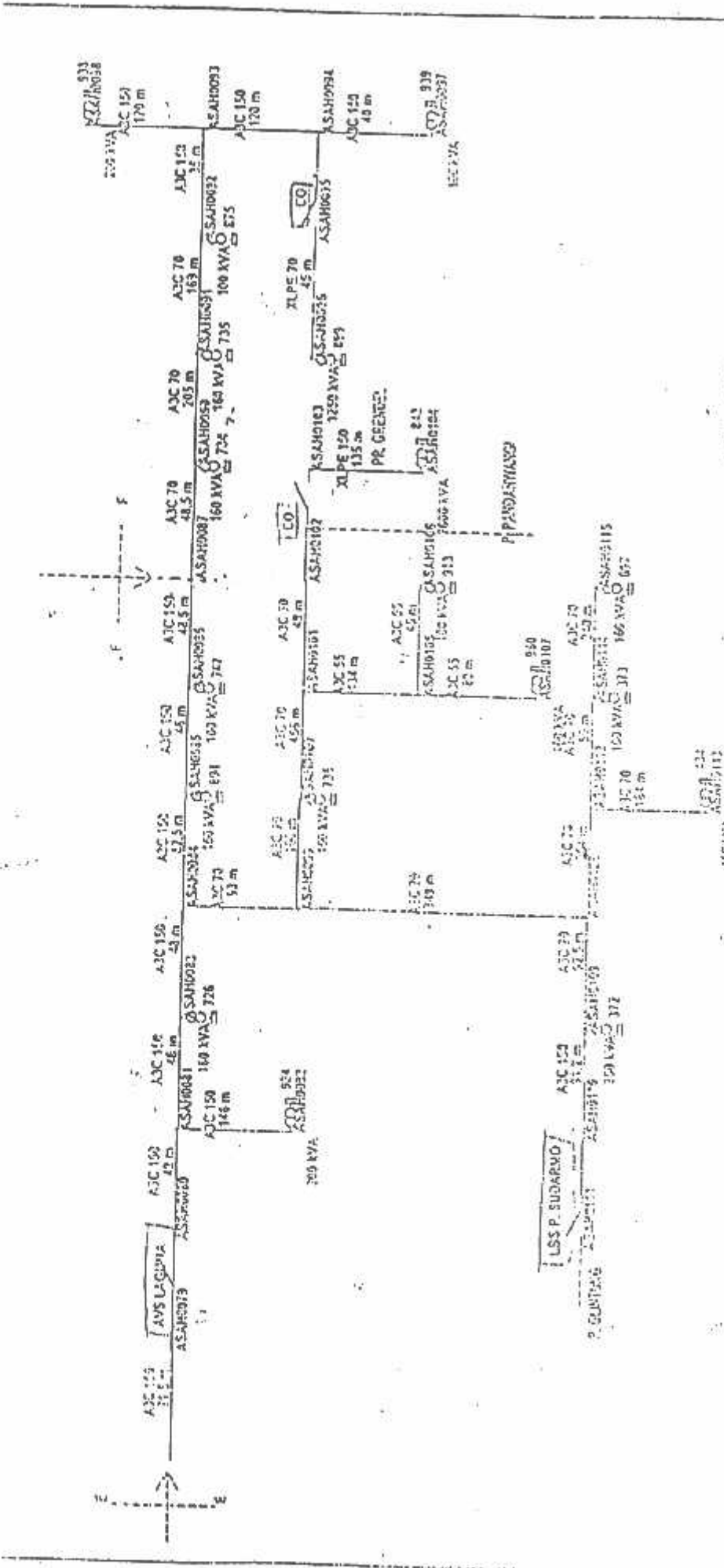


 <b>PT. PLN (PERSERO)</b> <b>JASA ENGINEERING</b>	
PENUNJANG SUSUT XMM PT. PLN (PERSERO) DISTRIKUSI JATIM - CADANGAN 12/17/19	
Title: SINGLE LINE DIAGRAM PENYULANG ASAHAR	
Drawn By:	Drawing No:
Check By:	Approved By:
Date:	400031004-SHDS-ASAHAR/19/19
Scale:	SHEET No. 1/1

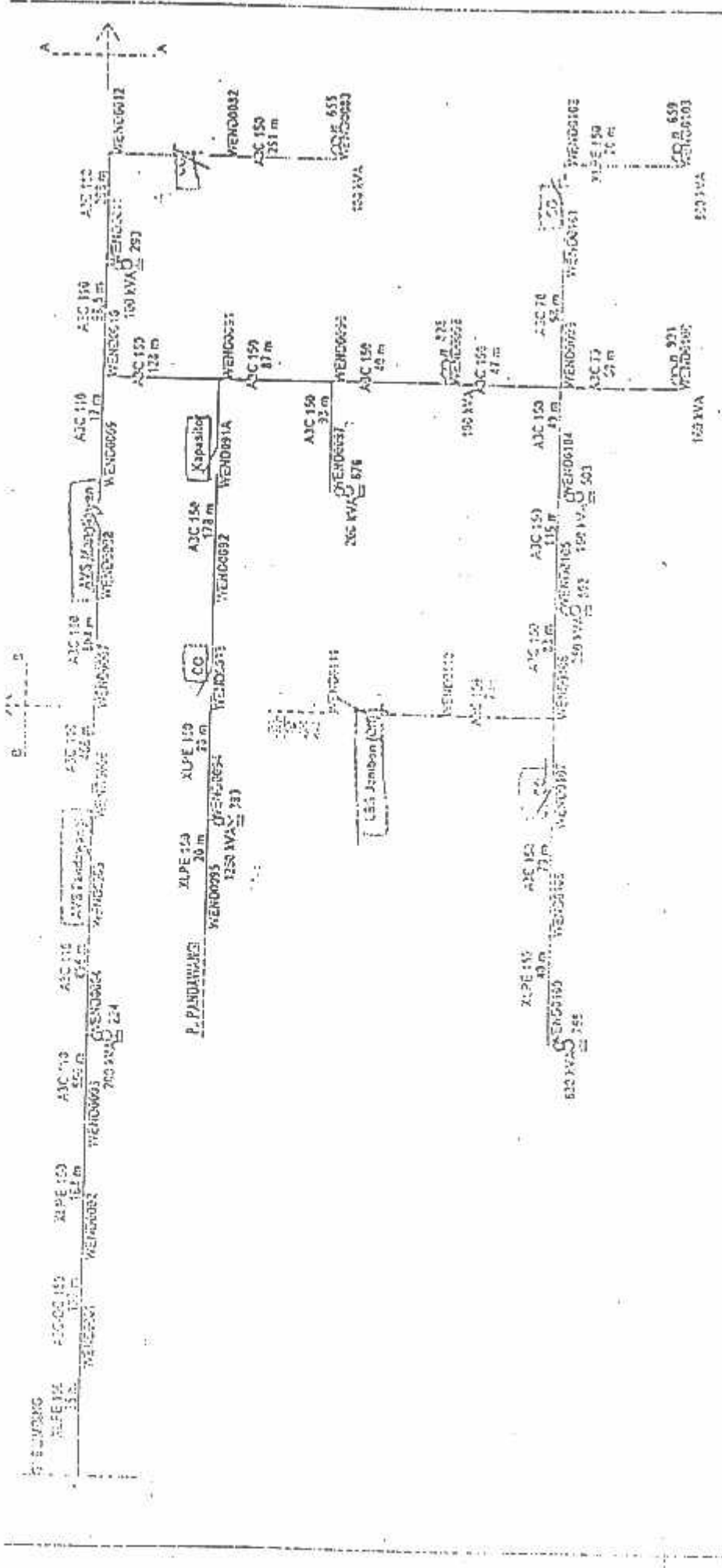


**PT. PLN (PERSERO)  
JASA ENJINIRING**

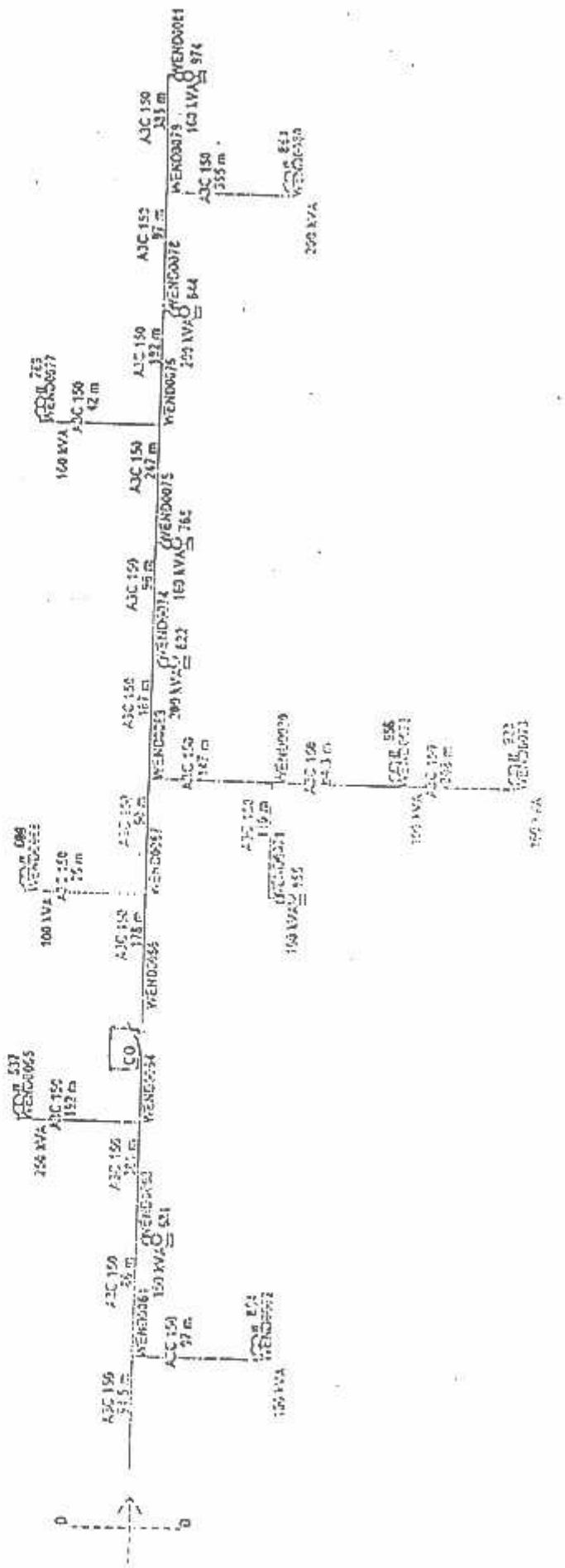
Project Name	PENUNJANG SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALA		
Title	SINGLE LINE DIAGRAM PENYULANG ASAHAN		
Designer	Drawing No		
Drawn By	11623100-L-51-OS-AS/R/03		
Approved By	Date		
Checked By	SEPTEMBER 2000 (Rev: 0)		



<b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b>	
Project Name: PENURUAN GUGUT AWAS PT. PLN (PERSERO) DISTRIKSI JATIM - CABANG MALANG	
Drawn By: Daryono	Approved By: Daryono
Checked By:	Date: SEPTEMBER 2000

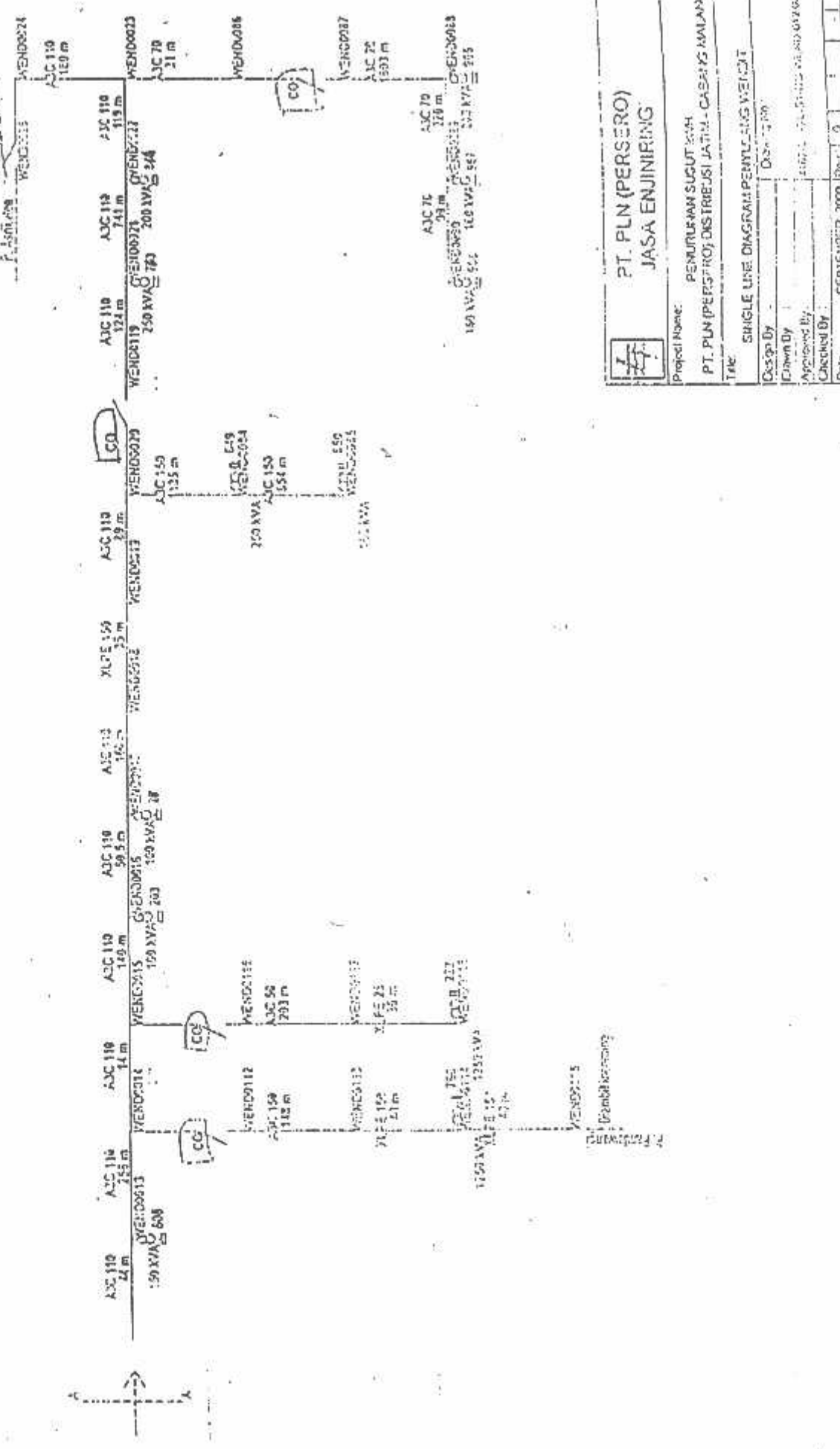



<b>PT. PLN (PERSERO) JASA ENJINIRING</b>	
Project Name:	PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIKULI JATI - CADANG KEJURANG
Type:	SINGLE LINE DIAGRAM PEMULANGAN KEMDIT
Drawn By:	[Signature]
Checked By:	[Signature]
Date:	SEPTEMBER 2020

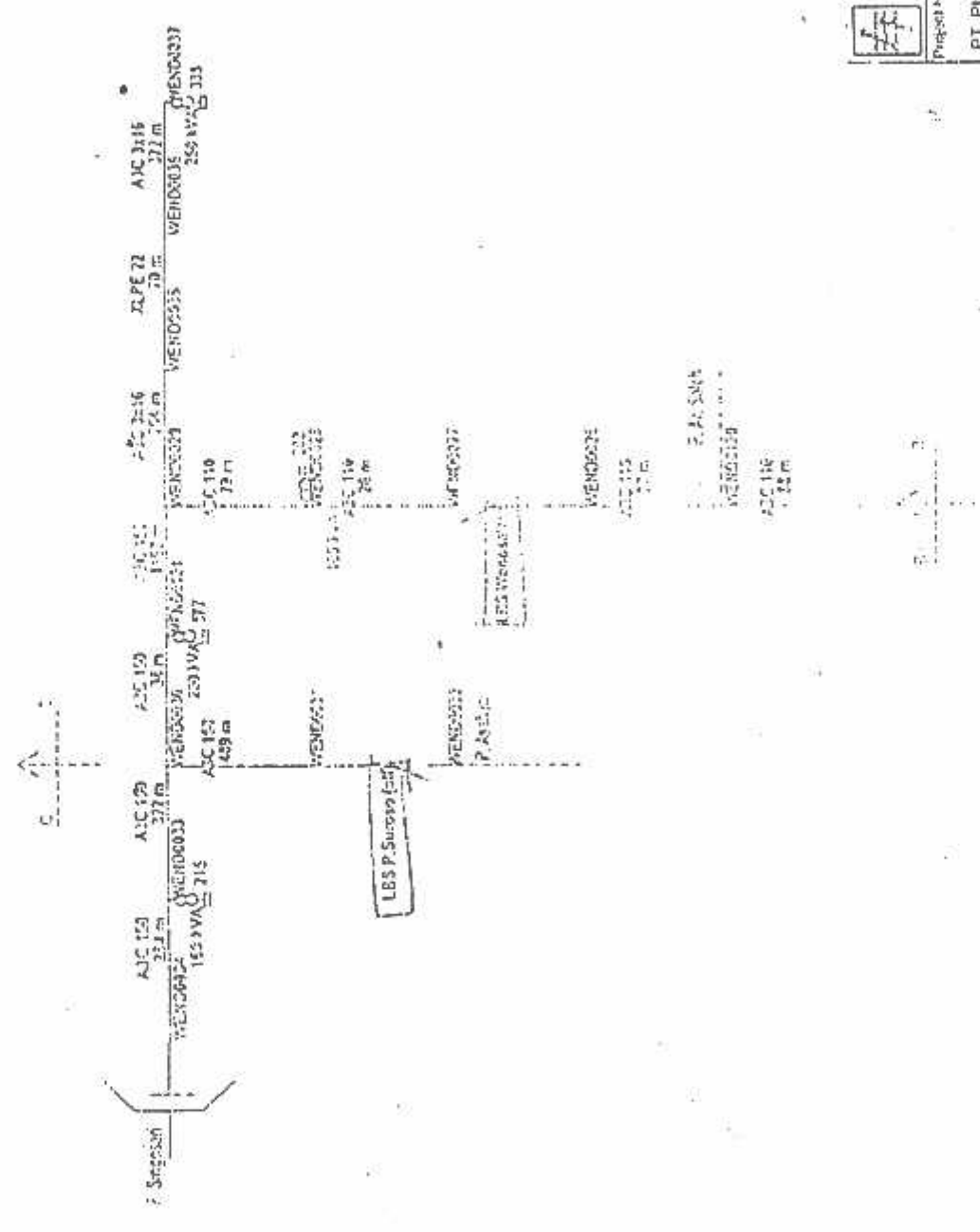


<b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b>	
Project Name: PENYUPAAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MPLANG	
168 SINGLE LINE DIAGRAM PENYUPAAN WENDOT	
Drawn By: IK Swa Dy	02/08/2010
Approved By:	
Checked By:	
Scale:	1:1
Sheet No.:	01

LES EMBELIM



 <p><b>PT. PLN (PERSERO)</b> <b>JASA ENJINIRING</b></p>	
Project Name:	PEMURUHAN SUGUT SWH PT. PLN (PERSERO) DISTRIK JATI M - CASANG MALANG
Title:	SINGLE LINE DIAGRAM PENYULING VEICENT
Design By:	Danang DW
Drawn By:	
Approved By:	
Checked By:	
Date:	SEPTEMBER, 2009



**PT. PLN (PERSERO)  
JASA ENJINIRING**

Project Name

PERURAHAN SUGUT 5000  
PT. PLN (PERSERO) DISTRIKUSI JATIM - CALAKA

Scale

SINGLE LINE DIAGRAM PENYALURAN

Drawn By

.....  
Drawing No

Approved By

.....  
4122 3100 65113

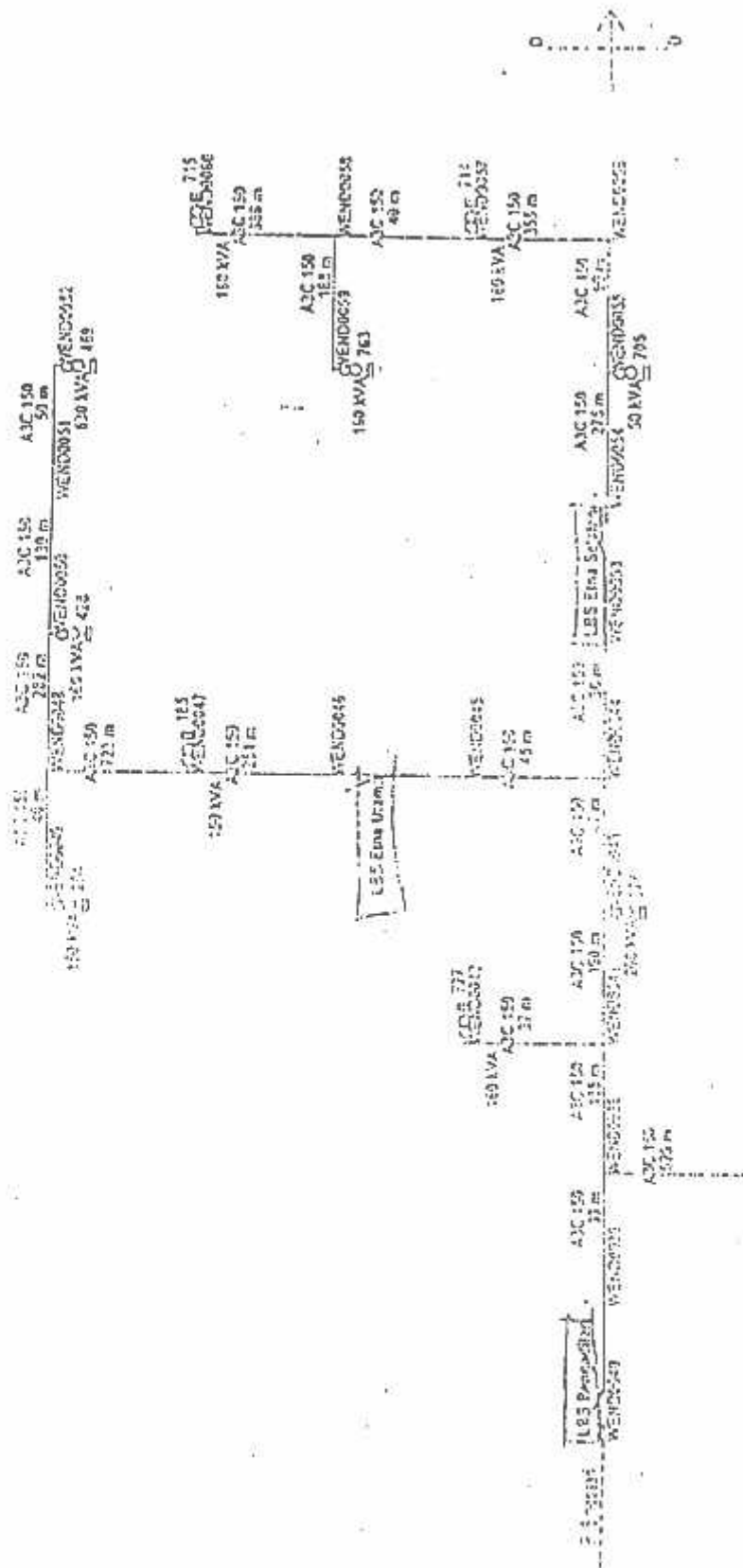
Checked By

.....

Date

SEPTEMBER 2000 (Rev. 1.0)





**PT. PLN (PERSERO)  
JASA ENJINIRING**

Project Name:	PERUBAHAN SUSUT KAWI PT. PLN (PERSERO) DISTRIKUSI JATIM - CADANGAN: BINA 717
Title:	SINGLE LINE DIAGRAM PENYULANG V.1.1.1
Drawn By:	Grading Ho
Approved By:	
Checked By:	
Date:	SEPTEMBER 2000 (Rev 1)

# LISTING PROGRAM

---

```

unit Utama;

interface

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

type
  TfrmUtama = class(TForm)
    pnlbtn: TPanel;
    btnNew: TButton;
    btnOpen: TButton;
    btnPrint: TButton;
    btnExit: TButton;
    OpenDialog1: TOpenDialog;
    Panel1: TPanel;
    StatusBar1: TStatusBar;
    procedure btnExitClick(Sender: TObject);
    procedure btnNewClick(Sender: TObject);
    procedure btnOpenClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmUtama: TfrmUtama;

implementation

uses Input, Komplex, TypeData;

{$R *.dfm}

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

procedure TfrmUtama.btnNewClick(Sender: TObject);
begin
  frmInput.Caption:='Input Data';
  frmInput.btnSave.Caption:='&Save';
  frmInput.cmbKonstV.Text:='kV';
  frmInput.cmbKonstP.Text:='MVA';
  frmInput.cmbKonstParam.Text:='pu';

```

---

```

frmInput.Show;
end;

procedure TfrmUtama.btnOpenClick(Sender: TObject);
var NamaFile:string;
    output:TextFile;
    i,j,Typ,dari,ke:integer;
    Cap,absV,sudV,Pgen,Qgen,Pload,Qload:double;
    R,X,Ic,Tr:double;
begin
    try
        if OpenFileDialog1.Execute then
            begin
                NamaFile:=OpenDialog1.FileName;
                AssignFile(output,NamaFile);
                Reset(output);
                Readln(output,Nbus);
                Readln(output,Nsal);
                Readln(output,Vbase);
                Readln(output,KonstV);
                Readln(output,Pbase);
                Readln(output,KonstP);
                Readln(output,param);
                Zbase:=sqr(Vbase*KonstV)/(Pbase*KonstP);
                lbase:=(Vbase*KonstV)/Zbase;
                frmInput.edtNbus.Text:=IntToStr(Nbus);
                frmInput.edtNsal.Text:=IntToStr(Nsal);
                frmInput.edtVbase.Text:=FloatToStr(Vbase);
                if KonstV=1 then
                    begin
                        frmInput.cmbKonstV.Text:='V';
                    end
                else if KonstV=1000 then
                    begin
                        frmInput.cmbKonstV.Text:='kV';
                    end
                else if KonstV=1000000 then
                    begin
                        frmInput.cmbKonstV.Text:='MV';
                    end;
                frmInput.edtPbase.Text:=FloatToStr(Pbase);
                if KonstP=1 then
                    begin
                        frmInput.cmbKonstP.Text:='VA';
                    end
                else if KonstP=1000 then
                    begin
                        frmInput.cmbKonstP.Text:='kVA';
                    end
                end;
            end;
        end;
    end;
end;

```

---

```

end
else if KonstP=1000000 then
begin
  frmInput.cmbKonstP.Text:='MVA';
end;
if param=1 then
begin
  frmInput.cmbKonstParam.Text:='pu';
end
else if param=2 then
begin
  frmInput.cmbKonstParam.Text:='ohm';
end;
SetLength(Va,Nbus+1);
SetLength(Sga,Nbus+1);
SetLength(SLa,Nbus+1);
SetLength(Capa,Nbus+1);
SetLength(Typa,Nbus+1);
SetLength(Za,Nbus+1,Nbus+1);
SetLength(Lca,Nbus+1,Nbus+1);
SetLength(Tra,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
  Readln(output,absV,sudV,Pgen,Qgen,Pload,Qload,Cap,Typ);
  Va[i]:=TKomplex.Create(absV,sudV);
  Sga[i]:=TKomplex.Create(Pgen/Pbase,Qgen/Pbase);
  SLa[i]:=TKomplex.Create(Pload/Pbase,Qload/Pbase);
  Capa[i]:=cap;
  Typa[i]:=Typ;
  frmInput.fgBus.Cells[0,i]:=IntToStr(i);
  frmInput.fgBus.Cells[1,i]:=FloatToStr(absV);
  frmInput.fgBus.Cells[2,i]:=FloatToStr(sudV);
  frmInput.fgBus.Cells[3,i]:=FloatToStr(Pgen);
  frmInput.fgBus.Cells[4,i]:=FloatToStr(Qgen);
  frmInput.fgBus.Cells[5,i]:=FloatToStr(Pload);
  frmInput.fgBus.Cells[6,i]:=FloatToStr(Qload);
  frmInput.fgBus.Cells[7,i]:=FloatToStr(Cap);
  frmInput.fgBus.Cells[8,i]:=IntToStr(Typ);
end;
for i:=1 to Nbus do
begin
  for j:=1 to Nbus do
  begin
    Za[i,j]:=TKomplex.Create;
  end;
end;
for i:=1 to Nsal do
begin

```

---

```

Readln(output,dari,ke,R,X,Lc,Tr);
if param=1 then
begin
  Za[dari,ke].Assign(R,X);
end
else if param=2 then
begin
  Za[dari,ke].Assign(R/Zbase,X/Zbase);
end;
Lca[dari,ke]:=Lc;
Tra[dari,ke]:=Tr;
frmInput.fgSaluran.Cells[0,i]:=IntToStr(i);
frmInput.fgSaluran.Cells[1,i]:=IntToStr(dari);
frmInput.fgSaluran.Cells[2,i]:=IntToStr(ke);
frmInput.fgSaluran.Cells[3,i]:=FloatToStr(R);
frmInput.fgSaluran.Cells[4,i]:=FloatToStr(X);
frmInput.fgSaluran.Cells[5,i]:=FloatToStr(Lc);
frmInput.fgSaluran.Cells[6,i]:=FloatToStr(Tr);
end;
CloseFile(output);
frmInput.Caption:='Tampilan Data';
frmInput.btnSave.Caption:='&Next';
frmInput.Show;
end;
except
  MessageDlg('File Corrupt atau Error Program!',mtWarning,[mbOK],0);
end;
end;

end.

```

---

unit Input;

interface

uses

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

type

```
TfrmInput = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  Panel1: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  edtNbus: TEdit;
  edtNsal: TEdit;
  fgBus: TStringGrid;
  fgSaluran: TStringGrid;
  btnClose: TButton;
  btnSave: TButton;
  label3: TLabel;
  Label4: TLabel;
  edtVbase: TEdit;
  Label5: TLabel;
  cmbKonstV: TComboBox;
  cmbKonstP: TComboBox;
  cmbKonstParam: TComboBox;
  edtPhase: TEdit;
  SaveDialog1: TSaveDialog;
  procedure btnCloseClick(Sender: TObject);
  procedure edtNbusChange(Sender: TObject);
  procedure edtNsalChange(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure cmbKonstPChange(Sender: TObject);
  procedure cmbKonstParamChange(Sender: TObject);
  procedure btnSaveClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

frmInput: TfrmInput;

---

implementation

uses Hasil, TypeData;

{\$R \*.dfm}

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

```
procedure TfrmInput.edtNbusChange(Sender: TObject);
var i:integer;
begin
  if edtNbus.Text="" then
  begin
    fgBus.RowCount:=2;
  end
  else
  begin
    fgBus.RowCount:=StrToInt(edtNbus.Text)+1;
    for i:=1 to StrToInt(edtNbus.Text) do
    begin
      fgBus.Cells[0,i]:=IntToStr(i);
      fgBus.Cells[1,i]:=IntToStr(1);
      fgBus.Cells[2,i]:=IntToStr(0);
    end;
  end;
end;
```

```
procedure TfrmInput.edtNsalChange(Sender: TObject);
var i:integer;
begin
  if edtNsal.Text="" then
  begin
    fgSaluran.RowCount:=2;
  end
  else
  begin
    fgSaluran.RowCount:=StrToInt(edtNsal.Text)+1;
    for i:=1 to StrToInt(edtNsal.Text) do
    begin
      fgSaluran.Cells[0,i]:=IntToStr(i);
    end;
  end;
end;
```

```
procedure TfrmInput.FormCreate(Sender: TObject);
```

---



```

begin
  fgBus.Cells[0,0]:='No';
  fgBus.Cells[1,0]:='Abs V (pu)';
  fgBus.Cells[2,0]:='Sud V (deg)';
  fgBus.Cells[3,0]:='Pgen (MW)';
  fgBus.Cells[4,0]:='Qgen (MVAR)';
  fgBus.Cells[5,0]:='Pload (MW)';
  fgBus.Cells[6,0]:='Qload (MVAR)';
  fgBus.Cells[7,0]:='Cap';
  fgBus.Cells[8,0]:='Typ Bus';
  fgSaluran.Cells[0,0]:='No';
  fgSaluran.Cells[1,0]:='Dari';
  fgSaluran.Cells[2,0]:='Ke';
  fgSaluran.Cells[3,0]:='R (pu)';
  fgSaluran.Cells[4,0]:='X (pu)';
  fgSaluran.Cells[5,0]:='Lc (pu)';
  fgSaluran.Cells[6,0]:='Tr';
  fgSaluran.Cells[7,0]:='Kapasitas (A)';
end;

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

procedure TfrmInput.cmbKonstParamChange(Sender: TObject);
begin
  if cmbKonstParam.Text='pu' then

```

---

```

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

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

```

---

```

begin
  param:=1;
end
else if cmbKonstParam.Text='ohm' then
begin
  param:=2;
end;
if btnSave.Caption='&Save' then
begin
  try
    if SaveDialog1.Execute then
    begin
      Nbus:=StrToInt(edtNbus.Text);
      Nsal:=StrToInt(edtNsal.Text);
      Vbase:=StrToFloat(edtVbase.Text);
      Pbase:=StrToFloat(edtPbase.Text);
      NamaFile:=SaveDialog1.FileName;
      AssignFile(input,NamaFile+'.txt');
      Rewrite(input);
      Writeln(input,Nbus);
      Writeln(input,Nsal);
      Writeln(input,Vbase:6:2);
      Writeln(input,KonstV:7:0);
      Writeln(input,Pbase:6:2);
      Writeln(input,KonstP:7:0);
      Writeln(input,param);
      for i:=1 to Nbus do
      begin
        absV:=StrToFloat(fgBus.Cells[1,i]);
        sudV:=StrToFloat(fgBus.Cells[2,i]);
        Pgen:=StrToFloat(fgBus.Cells[3,i]);
        Qgen:=StrToFloat(fgBus.Cells[4,i]);
        Pload:=StrToFloat(fgBus.Cells[5,i]);
        Qload:=StrToFloat(fgBus.Cells[6,i]);
        Cap:=StrToFloat(fgBus.Cells[7,i]);
        Typ:=StrToInt(fgBus.Cells[8,i]);
        Writeln(input,absV:7:5,',',sudV:7:5,',',Pgen:9:3,',',Qgen:9:3,
          ',',Pload:9:3,',',Qload:9:3,',',Cap:7:5,',',Typ);
      end;
      for i:=1 to Nsal do
      begin
        dari:=StrToInt(fgSaluran.Cells[1,i]);
        kc:=StrToInt(fgSaluran.Cells[2,i]);
        R:=StrToFloat(fgSaluran.Cells[3,i]);
        X:=StrToFloat(fgSaluran.Cells[4,i]);
        Lc:=StrToFloat(fgSaluran.Cells[5,i]);
        Tr:=StrToFloat(fgSaluran.Cells[6,i]);
        Writeln(input,dari,',',kc,',',R:7:5,',',X:7:5,',',

```

---

```
        Lc:7:5,' ',Tr:7:5);
    end;
    CloseFile(input);
    MessageDlg('File berhasil disimpan!',mtInformation,[mbOK],0);
    end;
except
    MessageDlg('Tolong dicek angka-angkanya kembali!',mtWarning,[mbOK],0);
    end;
end
else if btnSave.Caption='&Next' then
begin
    frmHasil.Show;
    frmHasil.fgLoadflow.RowCount:=Nbus+1;
    frmHasil.fgAliranDaya.RowCount:=Nsal*2+1;
    end;
end;

end.
```

---

```

unit GeneticAlgorithm;

interface

uses Komplex;

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

implementation

//constructor
constructor TGA.Create;
begin
  inherited Create;

```

---

```
maxgen:=1;
popsize:=1;
Nparam:=1;
ka:=1;
end;

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

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

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

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

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

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

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

function TGA.GetKa:double;
begin
  result:=ka;
end;

procedure TGA.SetMaxgen(dMaxgen:integer);
```

---

```

begin
  maxgen:=dMaxgen;
end;

procedure TGA.SetPopsiz(dPopsiz:integer);
begin
  popsize:=dpopsiz;
end;

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

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

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

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

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

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

//data processing
function TGA.GetFlip(const param:double):boolean;
var rand:double;
begin
  rand:=random;
  if rand<=param then
  begin
    result:=true;
  end
end

```

---

```

end;

var fita:TFitness;

implementation

{ TFitness }

function TFitness.BuatMatrikJalur(const dX: Arr2): Arr2;
var i,j:integer;
begin
  SetLength(result, FNbus+1, FNbus+1);
  for i:=1 to FNbus do
  begin
    for j:=1 to FNbus do
    begin
      result[i,j]:=0;
    end;
  end;
  for i:=1 to FNbus do
  begin
    for j:=1 to FNbus do
    begin
      if dX[i,j]<>0 then
      begin
        result[i,j]:=1;
        result[j,i]:=1;
      end;
    end;
  end;
end;

function TFitness.CekTegVio(const dBeban: Arr2): integer;
var i:integer;
    absV:double;
begin
  result:=0;
  for i:=1 to FNbus do
  begin
    absV:=sqrt(sqrt(dBeban[i,1])+sqrt(dBeban[i,2]));
    if absV<0.90 then result:=result+1;
    if absV>1.05 then result:=result+1;
  end;
end;

function TFitness.FindTegVio(const dBeban:Arr2):double;
var i,count:integer;
    Vs,absV,sumVdrop:double;

```

---



```

begin
  sumVdrop:=0;
  Vs:=1.0;
  count:=0;
  for i:-1 to FNbus do
  begin
    absV:=sqrt(sqr(dBeban[i,1])+sqr(dBeban[i,2]));
    if dBeban[i,7]=3 then
    begin
      inc(count);
      sumVdrop:=sumVdrop+(Vs-absV);
    end;
  end;
  result:=sumVdrop/count;
end;

constructor TFitness.Create(const dBeban,dR,dX,dLc,dTr:Arr2;
  const dSwith:1SwithArr);
var i,j:integer;
begin
  inherited Create;
  FNbus:=high(dBeban);
  FNSwith:=high(dSwith);
  SetLength(FBeban,FNbus+1,9);
  SetLength(FR,FNbus+1,FNbus+1);
  SetLength(FX,FNbus+1,FNbus+1);
  SetLength(FLc,FNbus+1,FNbus+1);
  SetLength(FTr,FNbus+1,FNbus+1);
  for i:=1 to FNbus do
  begin
    FBeban[i,1]:=dBeban[i,1];
    FBeban[i,2]:=dBeban[i,2];
    FBeban[i,3]:=dBeban[i,3];
    FBeban[i,4]:=dBeban[i,4];
    FBeban[i,5]:=dBeban[i,5];
    FBeban[i,6]:=dBeban[i,6];
    FBeban[i,7]:=-dBeban[i,7];
    FBeban[i,8]:=dBeban[i,8];
    for j:=1 to FNbus do
    begin
      FR[i,j]:=dR[i,j];
      FX[i,j]:=dX[i,j];
      FLc[i,j]:=dLc[i,j];
      FTr[i,j]:=dTr[i,j];
    end;
  end;
  SetLength(FSwitha,FNSwith+1);
  for i:=1 to FNSwith do

```

---

```

begin
  FSwitha[i].dari:=dSwith[i].dari;
  FSwitha[i].ke:=dSwith[i].ke;
end;
end;

constructor TFitness.Create;
begin
  inherited Create;
end;

destructor TFitness.Destroy;
begin
  inherited Destroy;
end;

function TFitness.doCalcFitness(const dChrom:TChromBin1):double;
var i,j,ia,ja,ite,TegVio:integer;

jmlGenR,jmlGenX,jmlLoadR,jmlLoadX,jmlLossR,jmlLossX,avgVdrop:double;
  Ra,Xa,AlirR,AlirX,ArusR,ArusX,mJalur:Arr2;
  cek:boolean;
begin
  result:=0;
  SetLength(Ra,FNbus+1,FNbus+1);
  SetLength(Xa,FNbus+1,FNbus+1);
  for i:=1 to FNbus do
  begin
    for j:=1 to FNbus do
    begin
      Ra[i,j]:=FR[i,j];
      Xa[i,j]:=FX[i,j];
    end;
  end;
  for i:=1 to FNSwith do
  begin
    if dChrom[i]=true then
    begin
      ia:=FSwitha[i].dari;
      ja:=FSwitha[i].ke;
      Xa[ia,ja]:=0;
      Ra[ia,ja]:=0;
    end;
  end;
  mJalur:=BuatMatrikJalur(Xa);
  cek:=TesJaringan(mJalur);
  if cek=true then
  begin

```

---

```

NewtonRaphson(FNbus,Ra,Xa,FLc,FTr,FPbase,ite,FBeban,AlirR,AlirX,
ArusR,ArusX,jmlGenR,jmlGenX,jmlLoadR,jmlLoadX,jmlLossR,jmlLossX);
if ite<15 then
begin
  TegVio:=CekTegVio(FBeban);
  AvgVdrop:=FindTegVio(FBeban);
  result:=jmlLossR+FPinalty*TegVio+FPinalty*AvgVdrop;//pin aliran daya
dan pin kap trafo
end;
end;
end;

```

```

function TFitness.GetBeban: Arr2;
var i,j:integer;
begin
  SetLength(result,FNbus+1,9);
  for i:=1 to FNbus do
  begin
    for j:=1 to 8 do
    begin
      result[i,j]:=FBeban[i,j];
    end;
  end;
end;

```

```

function TFitness.GetLc: Arr2;
var i,j:integer;
begin
  SetLength(result,FNbus+1,FNbus+1);
  for i:=1 to FNbus do
  begin
    for j:=1 to FNbus do
    begin
      result[i,j]:=FLc[i,j];
    end;
  end;
end;

```

```

function TFitness.GetR: Arr2;
var i,j:integer;
begin
  SetLength(result,FNbus+1,FNbus+1);
  for i:=1 to FNbus do
  begin
    for j:=1 to FNbus do
    begin
      result[i,j]:=FR[i,j];
    end;
  end;
end;

```

---

```
end;  
end;
```

```
function TFitness.GetSwit: TSwitArr;  
var i:integer;  
begin  
  SetLength(result,FNSwit+1);  
  for i:=1 to FNSwit do  
    begin  
      result[i].dari:=FSwit[i].dari;  
      result[i].ke:=FSwit[i].ke;  
    end;  
end;
```

```
function TFitness.GetTr: Arr2;  
var i,j:integer;  
begin  
  SetLength(result,FNbus+1, FNbus+1);  
  for i:=1 to FNbus do  
    begin  
      for j:=1 to FNbus do  
        begin  
          result[i,j]:=FTr[i,j];  
        end;  
      end;  
    end;  
end;
```

```
function TFitness.GetX: Arr2;  
var i,j:integer;  
begin  
  SetLength(result, FNbus+1, FNbus+1);  
  for i:=1 to FNbus do  
    begin  
      for j:=1 to FNbus do  
        begin  
          result[i,j]:=-FX[i,j];  
        end;  
      end;  
    end;  
end;
```

```
procedure TFitness.SetBeban(const dBeban: Arr2);  
var i:integer;  
begin  
  SetLength(FBeban, FNbus+1, 9);  
  for i:=1 to FNbus do  
    begin  
      FBeban[i,1]:=dBeban[i,1];  
      FBeban[i,2]:=dBeban[i,2];  
    end;  
end;
```

---

```
    FBeban[i,3]:=dBeban[i,3];
    FBeban[i,4]:=-dBeban[i,4];
    FBeban[i,5]:=-dBeban[i,5];
    FBeban[i,6]:=dBeban[i,6];
    FBeban[i,7]:=dBeban[i,7];
    FBeban[i,8]:=dBeban[i,8];
end;
end;
```

```
procedure TFitness.SetLc(const dLc: Arr2);
var i,j:integer;
begin
    SetLength(FLc, FNbus+1, FNbus+1);
    for i:=1 to FNbus do
        begin
            for j:=1 to FNbus do
                begin
                    FLc[i,j]:=dLc[i,j];
                end;
            end;
        end;
end;
```

```
procedure TFitness.SetNbus(const dNbus: integer);
begin
    FNbus:=dNbus;
end;
```

```
procedure TFitness.SetNSwith(const dNSwith: integer);
begin
    FNSwith:=dNSwith;
end;
```

```
procedure TFitness.SetPbase(const dPbase: double);
begin
    FPbase:=dPbase;
end;
```

```
procedure TFitness.SetPinalty(const dPinalty: double);
begin
    FPinalty:=dPinalty;
end;
```

```
procedure TFitness.SetR(const dR: Arr2);
var i,j:integer;
begin
    SetLength(FR, FNbus+1, FNbus+1);
    for i:=1 to FNbus do
        begin
```

```
    for j:=1 to FNbus do
    begin
        FR[i,j]:=-dR[i,j];
    end;
end;
end;
```

```
procedure TFitness.SetSwith(const dSwith: TSwithArr);
var i:integer;
begin
    FNswith:=high(dSwith);
    SetLength(FSwitha,FNswith+1);
    for i:=1 to FNswith do
    begin
        FSwitha[i].dari:=dSwith[i].dari;
        FSwitha[i].ke:=dSwith[i].ke;
    end;
end;
```

```
procedure TFitness.SetTr(const dTr: Arr2);
var i,j:integer;
begin
    SetLength(FTr,FNbus+1,FNbus+1);
    for i:=1 to FNbus do
    begin
        for j:=1 to FNbus do
        begin
            FTr[i,j]:=dTr[i,j];
        end;
    end;
end;
```

```
procedure TFitness.SetX(const dX: Arr2);
var i,j:integer;
begin
    SetLength(FX,FNbus+1,FNbus+1);
    for i:=1 to FNbus do
    begin
        for j:=1 to FNbus do
        begin
            FX[i,j]:=dX[i,j];
        end;
    end;
end;
end.
```

---

unit Hasil;

interface

uses

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

type

```
TfrmHasil = class(TForm)
  PageControl1: TPageControl;
  TabSheet2: TTabSheet;
  fgLoadflow: TStringGrid;
  TabSheet3: TTabSheet;
  fgAliranDaya: TStringGrid;
  GroupBox2: TGroupBox;
  lblSumGen1: TLabel;
  lblSumLoad1: TLabel;
  lblSumLoss1: TLabel;
  lblSumGen: TLabel;
  lblSumLoad: TLabel;
  lblSumLoss: TLabel;
  Label7: TLabel;
  lblIterasi1: TLabel;
  Label2: TLabel;
  edtSumGen: TEdit;
  edtSumLoad: TEdit;
  edtSumLoss: TEdit;
  edtTime: TEdit;
  edtIterasi: TEdit;
  TabSheet4: TTabSheet;
  fgLoadflow2: TStringGrid;
  TabSheet5: TTabSheet;
  fgAliranDaya2: TStringGrid;
  GroupBox1: TGroupBox;
  Label1: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  lblSumGen2: TLabel;
  lblSumLoad2: TLabel;
  lblSumLoss2: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label11: TLabel;
  edtSumGen2: TEdit;
  edtSumLoad2: TEdit;
  edtSumLoss2: TEdit;
```

---

```
edtTime2: TEdit;
edtIterasi2: TEdit;
Panel1: TPanel;
pbIterasi: TProgressBar;
btnGA: TButton;
btnHitungNR: TButton;
btnClose: TButton;
TabSheet7: TTabSheet;
Label8: TLabel;
edtmaxgen: TEdit;
edtpopsize: TEdit;
edtpcross: TEdit;
edtpmutasi: TEdit;
edtKa: TEdit;
edtNLoop: TEdit;
edtNSwith: TEdit;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
lblNLoop: TLabel;
lblNSwith: TLabel;
btnUseDefault: TButton;
edtPinalty: TEdit;
lblPinalty: TLabel;
GroupBox3: TGroupBox;
fgHasilGA: TStringGrid;
Label5: TLabel;
TabSheet1: TTabSheet;
PageControl2: TPageControl;
TabSheet6: TTabSheet;
TabSheet8: TTabSheet;
fgGA: TStringGrid;
Chart1: TChart;
Series1: TLineSeries;
Series2: TLineSeries;
Series3: TLineSeries;
Label6: TLabel;
Label16: TLabel;
edtBestFitness: TEdit;
edtAvgdV: TEdit;
Label17: TLabel;
Label18: TLabel;
edtJmlLoss: TEdit;
Label19: TLabel;
procedure btnCloseClick(Sender: TObject);
procedure btnHitungNRClick(Sender: TObject);
procedure FormActivate(Sender: TObject);
```

---



```

    procedure btnUseDefaultClick(Sender: TObject);
    procedure btnGAClick(Sender: TObject);
    procedure edtNLoopChange(Sender: TObject);
    procedure FormCreate(Sender: TObject);
private
    { Private declarations }
    function getAvgTeg(const rV:Arr1):double;
public
    { Public declarations }
end;

var
    frmHasil: TfrmHasil;

implementation

uses TypeData, LFNRPolar, Recursive, TypDatGA, Fitness, IGA;

{$R *.dfm}

var Vb,Sgb:CArr1;
    dBeb,dR,dX,dLc,dTr:Arr2;

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

procedure TfrmHasil.btnHitungNRClick(Sender: TObject);
var i,j,baris:integer;
    iterasi:double;
    pi,absV,sudV,sgr,sgx,slr,slx,slor,slox:double;
    SumGen,SumLoad,SumLoss,CPbase:TKomplex;
    Mulai,Selesai,Selang:TDateTime;
    jam,menit,detik,mdetik:word;
    AlirSb,IsI.b:CArr2;
    ite:integer;
    R,X,Lc,Tr,Beb,AlirR,AlirX,ArusR,ArusX:Arr2;
begin
    pi:=4*arctan(1);
    SetLength(dBeb,Nbus+1,9);
    SetLength(dR,Nbus+1,Nbus+1);
    SetLength(dX,Nbus+1,Nbus-1);
    SetLength(dLc,Nbus+1,Nbus+1);
    SetLength(dTr,Nbus+1,Nbus+1);
    SetLength(AlirR,Nbus+1,Nbus+1);
    SetLength(AlirX,Nbus+1,Nbus+1);
    for i:=1 to Nbus do

```

---

```

begin
  dBcb[i,1]:=Va[i].xRe;
  dBeb[i,2]:=Va[i].xIm;
  dBeb[i,3]:=Sga[i].xRe*Pbase;
  dBeb[i,4]:=-Sga[i].xIm*Pbase;
  dBeb[i,5]:=-SLa[i].xRe*Pbase;
  dBeb[i,6]:=SLa[i].xIm*Pbase;
  dBeb[i,7]:=Typa[i];
  dBeb[i,8]:=Capa[i];
  for j:=1 to Nbus do
    begin
      dR[i,j]:=Za[i,j].xRe;
      dX[i,j]:=Za[i,j].xIm;
      dLc[i,j]:=Lca[i,j];
      dTr[i,j]:=Tra[i,j];
    end;
  end;
  SetLength(Beb,Nbus+1,9);
  SetLength(R,Nbus+1,Nbus+1);
  SetLength(X,Nbus+1,Nbus+1);
  SetLength(Lc,Nbus+1,Nbus+1);
  SetLength(Tr,Nbus+1,Nbus+1);
  SetLength(AlirR,Nbus+1,Nbus+1);
  SetLength(AlirX,Nbus+1,Nbus+1);
  for i:=1 to Nbus do
    begin
      Beb[i,1]:=dBeb[i,1];
      Beb[i,2]:=dBeb[i,2];
      Beb[i,3]:=dBeb[i,3];
      Beb[i,4]:=dBeb[i,4];
      Beb[i,5]:=dBeb[i,5];
      Beb[i,6]:=dBeb[i,6];
      Beb[i,7]:=dBeb[i,7];
      Beb[i,8]:=dBeb[i,8];
      for j:=1 to Nbus do
        begin
          R[i,j]:=dR[i,j];
          X[i,j]:=dX[i,j];
          Lc[i,j]:=dLc[i,j];
          Tr[i,j]:=dTr[i,j];
        end;
      end;
    end;
  mulai:=time;
  NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase,ite,Beb,AlirR,AlirX,ArusR,ArusX,
  sgr,sgx,slr,slx,slor,slox);
  selesai:=time;
  sumGen:=TKomplex.Create(sgr,sgx);
  sumLoad:=TKomplex.Create(slr,slx);

```

---

```

sumLoss:=TKomplex.Create(slor,slox);
SetLength(Vb,Nbus+1);
SetLength(Sgb,Nbus+1);
SetLength(AlirSb,Nbus+1,Nbus+1);
SetLength(IsLb,Nbus+1,Nbus+1);
for i:=1 to Nbus do
begin
  Vb[i]:=TKomplex.Create(Beb[i,1],Beb[i,2]);
  Sgb[i]:=TKomplex.Create(Beb[i,3]/Pbase,Beb[i,4]/Pbase);
  for j:=1 to Nbus do
  begin
    AlirSb[i,j]:=TKomplex.Create(AlirR[i,j]/Pbase,AlirX[i,j]/Pbase);
    IsLb[i,j]:=TKomplex.Create(ArusR[i,j],ArusX[i,j]);
  end;
end;
Selang:=Selesai-Mulai;
DecodeTime(Selang,jam,menit,detik,mdetik);
edtTime.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+'+
IntToStr(detik)+'+'+IntToStr(mdetik);
CPbase:=TKomplex.Create(Pbase,0);
for i:=1 to Nbus do
begin
  absV:=-Vb[i].GetAbs;
  sudV:=Vb[i].GetAngleRad*180/pi;
  fgLoadflow.Cells[1,i]:=RealToStr(absV,5);
  fgLoadflow.Cells[2,i]:=RealToStr(SudV,5);
  fgLoadflow.Cells[3,i]:=RealToStr(Sgb[i].xRe*Pbase,3);
  fgLoadflow.Cells[4,i]:=RealToStr(Sgb[i].xIm*Pbase,3);
  fgLoadflow.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
  fgLoadflow.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);
end;
baris:=0;
for i:=1 to Nbus do
begin
  for j:=1 to Nbus do
  begin
    if AlirSb[i,j].xIm <> 0 then
    begin
      baris:=baris+1;
      fgAliranDaya.Cells[0,baris]:=IntToStr(baris);
      fgAliranDaya.Cells[1,baris]:=IntToStr(i);
      fgAliranDaya.Cells[2,baris]:=IntToStr(j);
      fgAliranDaya.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
      Pbase,3);
      fgAliranDaya.Cells[4,baris]:=RealToStr(AlirSb[i,j].xIm*
      Pbase,3);
      fgAliranDaya.Cells[5,baris]:=-RealToStr(IsLb[i,j].xRe*
      Pbase,3);
    end;
  end;
end;

```

---

```

    fgAliranDaya.Cells[6,baris]:=RealToStr(IsLb[i,j].xlm*
    lbase,3);
end;
end;
end;
edtSumGen.Text:=SumGen.toStringJ(3);
edtSumLoad.Text:=SumLoad.toStringJ(3);
edtSumLoss.Text:=SumLoss.toStringJ(3);
edtIterasi.Text:=IntToStr(ite);
for i:=1 to Nbus do
begin
    Sgb[i].Free;
    for j:=1 to Nbus do
    begin
        AlirSb[i,j].Free;
        IsLb[i,j].Free;
    end;
end;
end;
end;

```

```

procedure TfrmHasil.FormActivate(Sender: TObject);
var i:integer;
begin
    fgLoadflow.RowCount:=Nbus+1;
    fgAliranDaya.RowCount:=Nsal+1;
    fgLoadflow.Cells[0,0]:='Bus';
    fgLoadflow.Cells[1,0]:='Abs V (pu)';
    fgLoadflow.Cells[2,0]:='Sud V (deg)';
    fgLoadflow2.RowCount:=Nbus+1;
    fgAliranDaya2.RowCount:=Nsal+1;
    fgLoadflow2.Cells[0,0]:='Bus';
    fgLoadflow2.Cells[1,0]:='Abs V (pu)';
    fgLoadflow2.Cells[2,0]:='Sud V (deg)';
    if KonstP=1 then
    begin
        fgLoadflow.Cells[3,0]:='Pgen (W)';
        fgLoadflow.Cells[4,0]:='Qgen (W)';
        fgLoadflow.Cells[5,0]:='Pload (VAR)';
        fgLoadflow.Cells[6,0]:='Qload (VAR)';
        fgAliranDaya.Cells[3,0]:='P (Watt)';
        fgAliranDaya.Cells[4,0]:='Q (VAR)';
        lblSumGen.Caption:='VA';
        lblSumLoad.Caption:='VA';
        lblSumLoss.Caption:='VA';
        fgLoadflow2.Cells[3,0]:='Pgen (W)';
        fgLoadflow2.Cells[4,0]:='Qgen (W)';
        fgLoadflow2.Cells[5,0]:='Pload (VAR)';
        fgLoadflow2.Cells[6,0]:='Qload (VAR)';
    end;
end;

```

---

```

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

```

---

```

fgAliranDaya.Cells[0,0]:='No';
fgAliranDaya.Cells[1,0]:='Dari';
fgAliranDaya.Cells[2,0]:='Ke';
fgAliranDaya.Cells[5,0]:='Arus re (A)';
fgAliranDaya.Cells[6,0]:='Arus im (A)';
fgAliranDaya2.Cells[0,0]:='No';
fgAliranDaya2.Cells[1,0]:='Dari';
fgAliranDaya2.Cells[2,0]:='Ke';
fgAliranDaya2.Cells[5,0]:='Arus re (A)';
fgAliranDaya2.Cells[6,0]:='Arus im (A)';
for i:=1 to Nbus do
begin
  fgLoadflow.Cells[0,i]:=IntToStr(i);
  fgLoadflow2.Cells[0,i]:=IntToStr(i);
end;
for i:=1 to Nsal*2 do
begin
  fgAliranDaya.Cells[0,i]:=IntToStr(i);
  fgAliranDaya2.Cells[0,i]:=IntToStr(i);
end;
end;
procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
  edtmaxgen.Text:='100';
  edtpopsize.Text:='10';
  edtpcross.Text:='0.98';
  edtpmutasi.Text:='0.05';
  edtKa.Text:='10000';
  edtNLoop.Text:='4';
  edtNSwith.Text:=IntToStr(Nsal);
  edtPinalty.Text:='1000';
  btnGA.Enabled:=true;
end;

function TfrmHasil.getAvgLeg(const rV:Array of double):double;
var i,rows:integer;
    sumV:double;
begin
  rows:=high(rV);
  sumV:=0;
  for i:=1 to rows do
  begin
    sumV:=sumV+abs(1.0-rV[i]);
  end;
  result:=sumV/rows;
end;

procedure TfrmHasil.btnGAClick(Sender: TObject);

```

---

```

var swith:TSwithArr;
  i,j,ia,ja,sa,ita,baris:integer;
  R,X,Lc,Tr,Beb,AlirR,AlirX,ArusR,ArusX:Arr2;
  gas:TIGA;
  chromHasil:TChromBin1;
  pi,absV,sudV,sgr,sgx,slr,slx,slor,slox,fitness,avgdV:double;
  SumGen,SumLoad,SumLoss,CPbase:TKomplex;
  Mulai,Selesai,Selang:TDateTime;
  jam,menit,detik,mdetik:word;
  AlirSb,IsLb:CArr2;
  min,avg,max,Vc:Arr1;
begin
  SetLength(swith,StrToInt(edtNSwith.Text)+1);
  sa:=0;
  for i:=1 to Nbus do
  begin
    for j:=1 to Nbus do
    begin
      if Za[i,j].xIm<0 then
      begin
        sa:=sa+1;
        swith[sa].dari:=i;
        swith[sa].ke:=j;
      end;
    end;
  end;
  SetLength(Beb,Nbus+1,9);
  SetLength(R,Nbus+1,Nbus+1);
  SetLength(X,Nbus+1,Nbus+1);
  SetLength(Lc,Nbus+1,Nbus+1);
  SetLength(Tr,Nbus+1,Nbus+1);
  SetLength(AlirR,Nbus+1,Nbus+1);
  SetLength(AlirX,Nbus+1,Nbus+1);
  for i:=1 to Nbus do
  begin
    Beb[i,1]:=dBeb[i,1];
    Beb[i,2]:=dBeb[i,2];
    Beb[i,3]:=dBeb[i,3];
    Beb[i,4]:=dBeb[i,4];
    Beb[i,5]:=dBeb[i,5];
    Beb[i,6]:=dBeb[i,6];
    Beb[i,7]:=dBeb[i,7];
    Beb[i,8]:=dBeb[i,8];
    for j:=1 to Nbus do
    begin
      R[i,j]:=dR[i,j];
      X[i,j]:=dX[i,j];
      Lc[i,j]:=Lca[i,j];
    end;
  end;
end;

```

---

```

    Tr[i,j]:=Tra[i,j];
end;
end;
fita:=TFitness.Create(Beb,R,X,Lc,Tr,Swith);
fita.Pbase:=Pbase;
fita.Pinalty:=StrToFloat(edtPinalty.Text);
gas:=TIGA.Create;
gas.gaMaxgen:=StrToInt(edtMaxGen.Text);
pbIterasi.Max:=gas.gaMaxgen;
gas.gaPopsiz:=StrToInt(edtpopsiz.Text);
gas.gaPcross:=StrToFloat(edtpcross.Text);
gas.gaPmutat:=StrToFloat(edtpmutasi.Text);
gas.gaKa:=StrToFloat(edtKa.Text);
gas.gaLchrom:=StrToInt(edtNSwith.Text);
gas.gaPflip:=StrToFloat(edtNLoop.Text)/StrToFloat(edtNSwith.Text);
gas.gaNLoop:=StrToInt(edtNLoop.Text);
chromHasil:=gas.gaChromHasil;
fitness:=gas.gaFitBest;
edtBestFitness.Text:=RealToStr(fitness,4);
min:=gas.gaMin;
avg:=gas.gaAvg;
max:=gas.gaMax;
fgGA.RowCount:=gas.gaMaxgen+1;
Series1.Clear;
Series2.Clear;
Series3.Clear;
for i:=1 to gas.gaMaxgen do
begin
    fgGA.Cells[0,i]:=IntToStr(i);
    fgGA.Cells[1,i]:=RealToStr(min[i],3);
    fgGA.Cells[2,i]:=RealToStr(avg[i],3);
    fgGA.Cells[3,i]:=RealToStr(max[i],3);
    Series1.Add(min[i],IntToStr(i));
    Series2.Add(avg[i],IntToStr(i));
    Series3.Add(max[i],IntToStr(i));
end;
sa:=0;
fgHasilGA.RowCount:=gas.gaNLoop+1;
for i:=1 to high(chromHasil) do
begin
    if chromHasil[i]=true then
    begin
        sa:=sa+1;
        ia:=Swith[i].dari;
        ja:=Swith[i].ke;
        fgHasilGA.Cells[0,sa]:=IntToStr(sa);
        fgHasilGA.Cells[1,sa]:=IntToStr(ia);
        fgHasilGA.Cells[2,sa]:=IntToStr(ja);
    end;
end;

```

---



```

R[ia,ja]:=0;
X[ia,ja]:=0;
end;
end;
mulai:=time;
NewtonRaphson(Nbus,R,X,Lc,Tr,Pbase,ite,Beb,AlirR,AlirX,ArusR,ArusX,
sgr,sgx,slr,slx,slor,slox);
selesai:=time;
sumGen:=TKomplex.Create(sgr,sgx);
sumLoad:=TKomplex.Create(slr,slx);
sumLoss:=TKomplex.Create(slor,slox);
SetLength(Vb,Nbus+1);
SetLength(Sgb,Nbus+1);
SetLength(AlirSb,Nbus+1,Nbus+1);
SetLength(IsLb,Nbus+1,Nbus+1);
SetLength(Vc,Nbus+1);
pi:=4*arctan(1);
for i:=1 to Nbus do
begin
Vb[i]:=TKomplex.Create(Beb[i,1],Beb[i,2]);
Vc[i]:=Vb[i].GetAbs;
Sgb[i]:=TKomplex.Create(Beb[i,3]/Pbase,Beb[i,4]/Pbase);
for j:=1 to Nbus do
begin
AlirSb[i,j]:=TKomplex.Create(AlirR[i,j]/Pbase,AlirX[i,j]/Pbase);
IsLb[i,j]:=TKomplex.Create(ArusR[i,j],ArusX[i,j]);
end;
end;
Selang:=Selesai-Mulai;
DecodeTime(Selang,jam,menit,detik,mdetik);
edtTime2.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
IntToStr(detik)+':'+IntToStr(mdetik);
CPbase:=TKomplex.Create(Pbase,0);
avgdV:=getAvgTeg(Vc);
edtAvgdV.Text:=RealToStr(avgdV*100,5);
for i:=1 to Nbus do
begin
absV:=Vb[i].GetAbs;
sudV:=Vb[i].GetAngleRad*180/pi;
fgLoadflow2.Cells[1,i]:=RealToStr(absV,5);
fgLoadflow2.Cells[2,i]:=RealToStr(SudV,5);
fgLoadflow2.Cells[3,i]:=RealToStr(Sgb[i].xRe*Pbase,3);
fgLoadflow2.Cells[4,i]:=RealToStr(Sgb[i].xIm*Pbase,3);
fgLoadflow2.Cells[5,i]:=RealToStr(SLa[i].xRe*Pbase,3);
fgLoadflow2.Cells[6,i]:=RealToStr(SLa[i].xIm*Pbase,3);
end;
baris:=0;
for i:=1 to Nbus do

```

```

begin
  for j:=1 to Nbus do
    begin
      if AlirSb[i,j].xIm<>0 then
        begin
          baris:=baris+1;
          fgAliranDaya2.Cells[0,baris]:=IntToStr(baris);
          fgAliranDaya2.Cells[1,baris]:=IntToStr(i);
          fgAliranDaya2.Cells[2,baris]:=IntToStr(j);
          fgAliranDaya2.Cells[3,baris]:=RealToStr(AlirSb[i,j].xRe*
            Pbase,3);
          fgAliranDaya2.Cells[4,baris]:=-RealToStr(AlirSb[i,j].xIm*
            Pbase,3);
          fgAliranDaya2.Cells[5,baris]:=RealToStr(Isl.b[i,j].xRe*
            Ibase,3);
          fgAliranDaya2.Cells[6,baris]:=RealToStr(IsLb[i,j].xIm*
            Ibase,3);
        end;
      end;
    end;
  edtSumGen2.Text:=SumGen.toStringJ(3);
  edtSumLoad2.Text:=SumLoad.toStringJ(3);
  edtSumLoss2.Text:=SumLoss.toStringJ(3);
  edtIterasi2.Text:=IntToStr(ite);
  edtJmlLoss.Text:=RealToStr(SumLoss.xRe,3);
  gas.Free;
  fita.Free;
end;
procedure TfrmHasil.edtNLoopChange(Sender: TObject);
begin
  if edtNLoop.Text="" then
    begin
      fgHasilGA.RowCount:=2;
    end
  else
    begin
      fgHasilGA.RowCount:=StrToInt(edtNLoop.Text)+1;
    end;
end;

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
  fgGA.Cells[0,0]:='Gen';
  fgGA.Cells[1,0]:='-Min';
  fgGA.Cells[2,0]:='Avg';
  fgGA.Cells[3,0]:='Max';
end;
end.

```

---



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

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

PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

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

Malang, 1 Juni 2005

Nomor : ITN-456/I.SKP /2/05  
Lampiran : satu lembar  
Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. **Ir. H. TAUFIK HIDAYAT, MT**  
Dosen Institut Teknologi Nasional  
di -  
Malang

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

Nama : YAN ADITYA P  
Nim : 0012052  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro  
Konsentrasi : T. Energi Listrik (S-1)

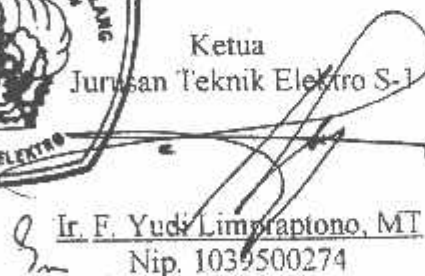
Dengan ini pembimbingan skripsi tersebut kami serahkan sepenuhnya kepada saudara/I selama masa waktu **6 (enam) bulan** terhitung mulai tanggal:

23 Mei 2005 s/d 23 Nop. 2005

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



Ketua  
Jurusan Teknik Elektro S-1

  
Ir. F. Yudi Limpraptono, MT  
Nip. 1039500274

Tindakan :

1. Mahasiswa yang bersangkutan
2. Arsip.

Form. S-4a



### FORMULIR BIMBINGAN SKRIPSI

Nama : Yan. Aditya. P  
Nim : 00.12.052  
Masa Bimbingan : 23 Mei 2005 s/d 23 November 2005  
Judul Skripsi : MENGURANGI RUGI - RUGI DENGAN MENGGUNAKAN METODE GENETIC ALGORITHM PADA SISTEM DISTRIBUSI 20 kV DI GARDU INDUK BLIMBING MALANG

No.	Tanggal	Uraian	Parap Pembimbing
1.	13/05/05	Selesaikan Bab IV, lengkapi referensi yang diacu (journal)	[Signature]
2.	30/05/05	- Uraikan abstrak direktori listrik - uraian. Cutrip menggaris milih	[Signature]
3.	19/08/05	- gambar 4-2 data input dari GI. Lengkap data luputannya.	[Signature]
4.	3/08/05	1) Kesimpulan & tujuan hasil analisis 2) Kesimpulan munculkan angka dari hasil analisis.	[Signature]
5.	7/09/05	Tambahkan uraian gambar 10 dan single line diagram yang di lampirkan berupa data pendahuluan saja, beri batasan pada caption.	[Signature]
7.	15/09/05	Ace untuk mengilustrasikan semua ini	[Signature]
8.			
9.			
10.			

Malang, 27/09/2005  
Dosen Pembimbing,

[Signature]  
Ir. H. Taufik Hidayat, MT

Form.S-4b



T. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

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

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

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karangjo, Km 2 Telp. (0341) 417638 Fax. (0341) 417634 Malang

Nomor : ITN-1476/III.TA/2/2005  
Lampiran : -  
Perihal : Survey / Permintaan Data

Malang, 2 Juli 2005

Kepada : Yth Pimpinan  
PT. PLN ( Persero )  
Distribusi Jatim APJ Malang  
Jl. Basuki Rachmat  
di-  
Malang

Bersama ini dengan hormat kami mohon kebijaksanaan Bapak agar mahasiswa kami dari Fakultas Teknologi Industri. Jurusan Teknik Elektro S-1 Konsentrasi Teknik Energi Listrik dapat diijinkan untuk melaksanakan survey untuk mendapatkan data-data guna penyusunan Tugas Akhir dengan Judul :

**MENGURANGI RUGI-RUGI DENGAN MENGGUNAKAN METODE GENETIC ALGORITHM PADA SISTEM DISTRIBUSI 20 KV DI GARDU INDUK BLIMBING MALANG.**

Mahasiswa tersebut adalah :

1. Yan Aditya P Nim : 00 12 052

Adapun lama Survey adalah : 2 ( Dua ) Minggu  
Demikian agar maklum dan atas perhatian serta bantuanya kami ucapkan terima kasih.

Dekan  
Fakultas Teknologi Industri  
Yakil Dekan I  
  
Ir. I Made Wartana, MT  
Nip. 131 991 182

Tembusan disampaikan kepada Yth :

1. Ketua Jurusan
2. Arsip

Form.SK-2



**PT. PLN ( PERSERO )**

DISTRIBUSI JAWA TIMUR

AREA PELAYANAN & JARINGAN MALANG

No. 0341 - 326034 ( Hunting )  
Kode Pos 211

Facsimile : 0341 - 362046

Nomor : 974/330/APJ – MLG/2005  
Surat Sdr.No : ITN-1476/III.TA/2/2005  
Tempiran : -  
Perihal : Survey/ Permintaan Data

Malang , 11 Agustus 2005

Kepada  
**ITN (Institut Teknologi Nasional)**  
Jl. Bendungan Sigura-gura No.2  
Di  
Malang

Menjawab surat Saudara nomor ITN-1476/III.TA/2/2005 pada tanggal 02 Juli 2005 perihal tersebut diatas, dengan ini kami beritahukan bahwa pada prinsipnya kami tidak keberatan / mengizinkan mahasiswa saudara atas nama :

**1. YAN ADITYA P      00 12 052**

Untuk melaksanakan Survey di UJ Malang PT. PLN (Persero) Area Pelayanan dan Jaringan Malang mulai tanggal 08 Agustus 2005 sampai dengan 19 Agustus 2005 dengan catatan PLN hanya memberikan data yang **TIDAK BERSIFAT RAHASIA**.

Sebelumnya diminta agar mahasiswa tersebut mengisi surat pernyataan, dilengkapi dengan pas photo ukuran 3 x 4 Cm di SDM & Administrasi PT. PLN (Persero) Area Pelayanan dan Jaringan Malang.

Demikian agar menjadikan maklum.



Tembusan :  
Manajer UJ Malang



## PERSETUJUAN PERBAIKAN SKRIPSI

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

Hari : Senin  
Tanggal : 3 Oktober 2005

Telah dilakukan perbaikan skripsi oleh :

1. Nama : Yan Aditya Perdana
2. NIM : 00.12.052
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : ANALISIS REKONFIGURASI JARINGAN DISTRIBUSI PRIMER 20 kV UNTUK MEMINIMALISASI RUGI-RUGI SECARA *MULTI-OBJECTIVE* DENGAN METODE *GENETIC ALGORITHM*

Perbaikan meliputi :

No	Materi Perbaikan	Ket
1.	Tampilkan beban tiap-tiap penyulang sebelum dan sesudah rekonfigurasi untuk mengetahui apakah hasil setelah rekonfigurasi jaringan masih memenuhi KHA saluran	<i>as</i>

### Anggota Penguji

Ir. Choirul Saleh, MT  
Penguji Pertama

Ir. Yusuf Ismail Nakhoda, MT  
Penguji Kedua

### Dosen Pembimbing

Ir. H. Taufik Hidayat, MT



## LEMBAR BIMBINGAN SKRIPSI

1. Nama : Yan Aditya Perdana
2. NIM : 00.12.052
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : ANALISIS REKONFIGURASI  
JARINGAN DISTRIBUSI PRIMER  
20 kV UNTUK MEMINIMALISASI  
RUGI-RUGI SECARA *MULTI-  
OBJECTIVE* DENGAN METODE  
*GENETIC ALGORITHM*
6. Tanggal Mengajukan Skripsi : 23 Mei 2005
7. Tanggal Menyelesaikan Skripsi : 3 Oktober 2005
8. Dosen Pembimbing : Ir. H. Taufik Hidayat, MT
10. Telah dievaluasi dengan nilai : 83 ( delapan puluh tiga)

Mengetahui  
Ketua Jurusan Teknik Elektro



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

Diperiksa dan Disetujui  
Dosen Pembimbing



Ir. H. Taufik Hidayat, MT  
NIP. 101.870.00151