

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



SKRIPSI

**OPTIMALISASI PEMILIHAN KONDUKTOR PADA SALURAN
DISTRIBUSI PRIMER 20 KV PENYULANG PUJON MENGGUNAKAN
METODE *EVOLUTIONARY PROGRAMMING***

Disusun oleh :

Nama : Agus Hariyanto

Nim : 00.12.028

April 2005

LEMBAR PERSETUJUAN

**OPTIMALISASI PEMILIHAN KONDUKTOR PADA SALURAN
DISTRIBUSI PRIMER 20 kV PENYULANG PUJON MENGGUNAKAN
METODE *EVOLUTIONARY PROGRAMMING***

SKRIPSI

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

Disusun Oleh :


**AGUS HARIYANTO
NIM. 00.12.028**

**Mengetahui,
Ketua Jurusan Teknik Elektro**



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

**Menyetujui,
Dosen Pembimbing**



Ir. Taufik Hidayat, MT
NIP. 108700151

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

OPTIMALISASI PEMILIHAN KONDUKTOR PADA SALURAN DISTRIBUSI PRIMER 20 kV PENYULANG PUJON MENGGUNAKAN METODE *EVOLUTIONARY PROGRAMMING*

(Agus Hariyanto, Nim. 00.12.028, Teknik Elektro S-1/Teknik Energi Listrik)
(Dosen Pembimbing : Ir. Taufik Hidayat, MT)

Kata Kunci : Sistem Distribusi Tenaga Listrik, Optimalisasi Pemilihan Konduktor,
Evolutionary Programming

Perkembangan yang semakin pesat dalam bidang ilmu pengetahuan di berbagai sektor kehidupan mengakibatkan kebutuhan akan energi listrik dari tahun ke tahun mengalami peningkatan. Pada sistem tenaga listrik, khususnya pada sistem distribusi dituntut mempunyai nilai ekonomis, andal, dan berkualitas dalam menyalurkan energi listrik.

Dalam mendisain sistem distribusi dibutuhkan beberapa langkah perencanaan, dan perhitungan yang tepat. Seleksi konduktor untuk mendesain dan meningkatkan saluran distribusi adalah bagian penting dari proses perencanaan.

Dengan menggunakan metode *Evolutionary Programming* dalam memilih konduktor pada saluran distribusi, ini dapat mengoptimalkan penyaluran daya listrik untuk mengurangi rugi-rugi daya pada jaringan sistem distribusi.

Analisis dari hasil perhitungan Optimasi Pemilihan Konduktor Pada Saluran Distribusi Primer 20 kV Penyulang Pujon memberikan hasil pengurangan biaya investasi konduktor dari Rp 1.433.625.930,- menjadi Rp 1.376.826.823 atau terjadi optimasi sebesar 3,96 %, rugi-rugi daya dari $132.625 + j185.404$ kVA menjadi $121.471 + j173.736$ kVA., atau terjadi optimasi sebesar 8,41 %.

KATA PENGANTAR

Dengan rahmat Allah SWT, dan mengucapkan syukur kehadirat-Nya atas karunia yang dilimpahkan kepada saya sehingga dapat menyelesaikan skripsi yang berjudul "OPTIMALISASI PEMILIHAN LUAS PENAMPANG KONDUKTOR AAAC PADA SALURAN DISTRIBUSI PRIMER 20 kV PENYULANG PUJON MENGGUNAKAN METODE *EVOLUTIONARY PROGRAMMING* "

Skripsi ini bertujuan untuk memenuhi kurikulum akademik yang harus ditempuh oleh setiap mahasiswa ITN Malang dalam menempuh sekaligus mengakhiri pendidikan pada jenjang S-1 pada jurusan Teknik Elektro program studi Energi Listrik.

Saya menyadari bahwa skripsi ini masih jauh dari kesempurnaan, karena itu saran dan kritik membangun sangat saya harapkan.

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

1. Bapak Dr. Ir. Abraham Lomi, MSEE, selaku Rektor ITN Malang.
2. Bapak Ir. F Yudi Limpraptono, MT, selaku Kajur Teknik Elektro ITN Malang.
3. Bapak Ir. Taufik Hidayat, MT, selaku Dosen Pembimbing.
4. Bapak Ir. Widodo, selaku pembimbing lapangan di PT. PLN (Persero) Distribusi Jawa Timur Area Malang UP&J-Batu.

Akhirnya saya mengharapkan skripsi ini berguna dan bermanfaat bagi rekan mahasiswa khususnya jurusan Teknik Elektro.

Malang, Maret 2005

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
DAFTAR GRAFIK.....	xii
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah.....	2
1.5. Metodologi Pembahasan	3
1.6. Sistematika Penulisan	3
1.7. Kontribusi Penelitian.....	4
BAB II SISTEM DISTRIBUSI TENAGA LISTRIK	
2.1. Sistem Tenaga Listrik	5
2.2. Sistem Distribusi	5
2.2.1. Sistem Distribusi Primer	6
2.2.2. Sistem Distribusi Sekunder	7

2.3.	Pola Dasar Struktur Jaringan Tegangan Menengah	7
2.3.1.	Sistem Jaringan Distribusi Tipe Radial	7
2.3.2.	Sistem Jaringan Distribusi Tipe <i>Loop</i>	9
2.3.3.	Sistem Jaringan Distribusi Tipe <i>Mesh/Grid</i>	10
2.4.	Konstruksi Jaringan Sistem Distribusi	11
2.4.1.	Konduktor Pada Sistem Distribusi	12
2.4.2.	Jenis Konduktor Saluran Udara Pada Sistem Distribusi Primer	13
2.5.	Daya Sistem Distribusi.....	17
2.5.1.	Daya Nyata (<i>Real Power</i>)	17
2.5.2.	Daya Reaktif (<i>Reaktif Power</i>)	17
2.5.3.	Daya Komplek (<i>Complex Power</i>).....	17
2.6.	Beban Sistem Distribusi	19
2.7.	Rugi-rugi Daya Sistem Distribusi	19
BAB III	OPTIMALISASI PEMILIHAN KONDUKTOR MENGGUNAKAN METODE <i>EVOLUTIONARY PROGRAMMING</i>	
3.1.	Analisa Aliran Daya Jaringan	20
3.1.1.	Metode <i>Newton Rapshon</i>	20
3.1.2.	Algoritma Aliran Daya <i>Newton Rapshon</i>	24
3.2.	<i>Evolutionary Programming</i>	25
3.2.1.	Parameter <i>Evolutionary Programming</i>	27
3.2.2.	Mekanisme <i>Evolutionary Programming</i>	28
3.3.	Formulasi Masalah Pemilihan Konduktor	31

3.4. Adaptasi <i>Evolutionary Programming</i> Ke Masalah Pemilihan Konduktor	31
3.4.1. Representasi Solusi.....	32
3.4.2. Inisialisasi.....	32
3.4.3. <i>Fitness</i> Calon Solusi	32
3.4.4. Menghasilkan Solusi Baru Dengan Mutasi	33
3.4.5. Seleksi Individu dengan Kompetisi.....	34

BAB IV ANALISA HASIL MENGGUNAKAN METODE EVOLUTIONARY PROGRAMMING

4.1. Program Komputer Optimal Pemilihan Konduktor Menggunakan Metode <i>Evolutionary Programming</i>	35
4.1.1. Algoritma Program	35
4.1.2. Algoritma Program Pemecahan Masalah Pemilihan Konduktor Menggunakan Metode <i>Evolutionary Programming</i>	36
4.1.3. Algoritma Program <i>Fitness</i>	37
4.1.4. Flowchart Program Pemecahan masalah Secara Umum	38
4.1.5. Flowchart Pemecahan Masalah Pemilihan Konduktor Menggunakan Metode <i>Evolutionary Programming</i>	39
4.1.6. Flowchart Program <i>Fitness</i>	40
4.2. Data Perhitungan.....	41
4.2.1. Data Saluran	42

4.2.2. Data Pembebanan Penyulang Pujon	43
4.3. Analisa Perhitungan	43
4.4. Tampilan Program	44
4.5. Hasil Perhitungan	50

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan	56
5.2. Saran	57

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar :	Hal :
2-1. Skema Penyaluran system distribusi Tenaga Listrik	6
2-2. Sistem Distribusi Primer Tipe Radial	8
2-3. Sistem Distribusi Primer Tipe <i>Loop</i>	9
2-4. Sistem Distribusi Primer Tipe <i>Mesh/Grid</i>	10
3-1. Mutasi Gaussian Dari Induk Menghasilkan Anak	30
4-1. Flowchart Pemecahan Masalah Secara Umum	38
4-2. Flowchart Program Pemecahan Masalah Pemilihan Konduktor Menggunakan Metode <i>Evolutionary Programming</i>	39
4-3. Algoritma Program <i>Fitness</i>	40
4-4. Tampilan Utama Program	44
4-5. Tampilan Inputan Data	44
4-6. Tampilan Data <i>Bus</i>	45
4-7. Tampilan Data Saluran	45
4-8. Tampilan Data <i>Cable</i>	46
4-9. Tampilan Hasil Aliran Daya Sebelum Pemilihan Konduktor Dengan Metode <i>Evolutionary Programming</i>	46
4-10. Tampilan Parameter <i>Evolutionary Programming</i>	47
4-11. Tampilan Hasil <i>Evolutionary Programming</i> Dalam pemilihan Konduktor	47
4-12. Tampilan Hasil Aliran Daya Sesudah Pemilihan Konduktor	

	Dengan Metode <i>Evolutionary Programming</i>	48
4-13	Tampilan Total Rugi-rugi Saluran Sesudah Pemilihan Konduktor Dengan Metode <i>Evolutionary Programming</i>	49

DAFTAR TABEL

Tabel:	Hal :
2-1. Konstruksi Konduktor Udara Tembaga Telanjang (BBC).....	14
2-2. Konstruksi Konduktor Udara Alumunium Telanjang (AAC).....	15
3-3. Konstruksi Konduktor Udara Alumunium Campuran Telanjang (AAAC).....	16
4-1. Data Konduktor Penyulang Pujon.....	42
4-2. Data Saluran Penyulang Pujon.....	42
4-3. Daftar harga Konduktor AAAC	42
4-4. Data Pembebanan Penyulang Pujon.....	43
4-5. Kondisi Tegangan dan sudut Fasa Tegangan Tiap <i>Node</i> Penyulang Pujon Sebelum Optimasi.....	50
4-6. Besarnya Rugi-rugi Daya Tiap Saluran Penyulang Pujon Sebelum Optimasi	50
4-7. Besarnya Aliran Daya Tiap Saluran Penyulang Pujon Sebelum Optimasi	51
4-8. Kondisi Tegangan dan sudut Fasa Tegangan Tiap <i>Node</i> Penyulang Pujon Setelah Optimasi	52
4-9. Besarnya Rugi-rugi Daya Tiap Saluran Penyulang Pujon Sesudah Optimasi	53
4-10. Besarnya Aliran Daya Tiap Saluran Penyulang Pujon Sesudah Optimasi	53

DAFTAR GRAFIK

Grafik :	Hal :
4-1. Tegangan Tiap-tiap <i>Node</i> Terhadap Asumsi Tegangan Awal Sebelum Dan Sesudah Optimasi	54
4-2. Rugi Daya Sebelum Dan sesudah Optimasi	54

BAB I PENDAHULUAN

1.1. Latar Belakang^[1]

Perkembangan yang semakin pesat dalam bidang ilmu pengetahuan di berbagai sektor kehidupan mengakibatkan kebutuhan akan energi listrik dari tahun ke tahun mengalami peningkatan. Pada sistem tenaga listrik, khususnya pada sistem distribusi dituntut mempunyai nilai ekonomis, andal, dan berkualitas dalam menyalurkan energi listrik .

Dalam mendisain sistem distribusi dibutuhkan beberapa langkah perencanaan, dan perhitungan yang tepat. Pemilihan konduktor untuk mendisain dan *mengupgrade* sistem distribusi merupakan bagian – bagian penting dalam proses perencanaan. Seleksi konduktor untuk mendesain dan meningkatkan saluran distribusi adalah bagian penting dari proses perencanaan.

Dengan menggunakan metode *Evolutionary Programming* dalam memilih konduktor pada saluran distribusi, ini dapat mngoptimalkan penyaluran daya listrik untuk mengurangi rugi-rugi daya pada jaringan sistem distribusi.

1.2. Rumusan Masalah

Suatu sistem distribusi dalam menyalurkan daya listrik mengingatkan rugi-rugi yang seminimal mungkin dalam menyalurkan energi listrik. Selain itu dapat menyalurkan daya listrik semaksimal mungkin. Sehingga dari hal tersebut permasalahan yang timbul adalah konduktor yang sesuai untuk penyulang Pujon?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menganalisa konduktor yang paling optimal bagi saluran distribusi primer 20 kV Penyulang Pujon.

1. Menganalisa biaya investasi modal yang optimal dalam pemilihan konduktor penyulang pujon menggunakan metode *Evolutionary Programming*.
2. Meminimalkan rugi-rugi daya pada saluran penyulang Pujon.
3. Dapat meningkatkan profil tegangan pada penyulang pujon

1.4. Batasan Masalah

Dalam pembahasan masalah ini agar tidak terlalu luas, maka diperlukan batasan masalah. Adapun batasan masalahnya adalah :

1. Saluran udara yang dibahas adalah saluran udara tegangan menengah 20 kV pada penyulang Pujon dari Gardu Induk Sengkaling.
 2. Pemilihan konduktor yang dibahas konduktor udara telanjang jenis AAAC
 3. Tidak membahas biaya investasi lain kecuali biaya investasi pemilihan konduktor pada penyulang Pujon.
 4. Tidak membahas secara mendetail metode *Newton Rapshon*
 5. Besarnya beban pada masing-masing fasa diasumsikan seimbang.
 6. Menggunakan metode *Evolutionary Programming* (EP) untuk memecahkan masalah optimalisasi pemilihan konduktor pada Penyulang Pujon.
-

1.5. Metodologi Pembahasan

Metodologi yang dipakai dalam pembahasan skripsi ini dilakukan dengan langkah-langkah sebagai berikut :

1. Studi kepustakaan mengenai hal-hal yang berhubungan dengan pembahasan masalah skripsi ini.
2. Studi lapangan untuk mendapatkan data parameter yang diperlukan dari obyek penelitian pada saluran distribusi primer penyulang pujan yang diperlukan dengan berpedoman pada teori yang diperoleh dari studi kepustakaan.
3. Analisa pemilihan konduktor menggunakan metode *Evolutionary Programming (EP)*.
4. Membuat evaluasi sehingga dapat disimpulkan hasil analisis yang diperoleh.

1.6. Sistematika Penulisan

Sistematika pembahasan dalam skripsi ini disusun sebagai berikut :

Bab I : Pendahuluan

Menguraikan latar belakang, tujuan penelitian, rumusan masalah, batasan masalah, metodologi, sistematika penulisan.

Bab II : Sistem Distribusi Tenaga listrik

Menguraikan pembahasan sistem distribusi tenaga listrik secara umum.

Bab III : Optimalisasi pemilihan konduktor menggunakan metode *Evolutionary Programming*.

Menguraikan teori dasar metode EP dan adaptasi EP ke permasalahan pemilihan konduktor

Bab IV : Analisa Hasil Menggunakan Metode *Evolutionary Programming*.

Menguraikan alur program hasil perhitungan pemilihan konduktor menggunakan metode EP.

Bab V : Kesimpulan dan saran

Berisikan tentang kesimpulan dan saran dari hasil analisa yang diperoleh.

1.7 .Kontribusi Masalah

Penelitian ini memberikan kontribusi dalam pengembangan jaringan distribusi terutama dalam pemilihan konduktor pada penyulang berdasarkan metode *Evolutionary Programming* (EP), sehingga nantinya diharapkan dapat memberikan manfaat dalam membangun atau *mengupgrade* sistem distribusi.

BAB II

SISTEM DISTRIBUSI TENAGA LISTRIK

2.1. Sistem Tenaga listrik⁽¹⁾

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

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

- 1.Sistem pembangkitan : berperan sebagai sumber daya tenaga listrik dan disebut juga sebagai prodaktor energi .
- 2.Sistem transmisi atau penyalur : berfungsi penyalur daya listrik secara besar-besaran dari pembangkit kebagian sistem distribusi
- 3.Sistem distribusi dan beban : berfungsi sebagai distributor energi kekonsumen yang memerlukan energi tersebut.

2.2 Sistem Distribusi

Jaringan distribusi berada pada akhir dari sistem tenaga listrik,perannya mendistribusikan tenaga listrik dari gardu induk atau pembangkit tenaga kekonsumen melalui gardu distribusi.

2.2.2 Sistem Distribusi Sekunder

Tingkat tegangan yang digunakan pada sistem distribusi sekunder adalah tegangan rendah 220/380 Volt, oleh karena itu sistem distribusi ini sering disebut dengan 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 Pola Dasar Struktur Jaringan Tegangan Menengah

Secara umum dapat dikatakan ada tiga struktur jaringan yang umum digunakan untuk menyalurkan dan mendistribusikan tenaga listrik yaitu:

1. Sistem distribusi primer Tipe radial
2. Sistem distribusi primer Tipe *loop*
3. Sistem distribusi primer Tipe *mesh/grid*

2.3.1 Sistem Jaringan Distribusi Tipe Radial

Sistem jaringan distribusi tipe radial merupakan struktur dasar dari jaringan distribusi dan mempunyai bentuk yang paling sederhana dan banyak digunakan di Indonesia. Dalam sistem radial, cara penyaluran tenaga listrik tersebut direntangkan secara radial dari gardu induk ke beban, selanjutnya disalurkan melalui percabangan-percabangan yang akan melayani beban yang ada. Sistem radial ini mempunyai beberapa keuntungan dan kerugian.

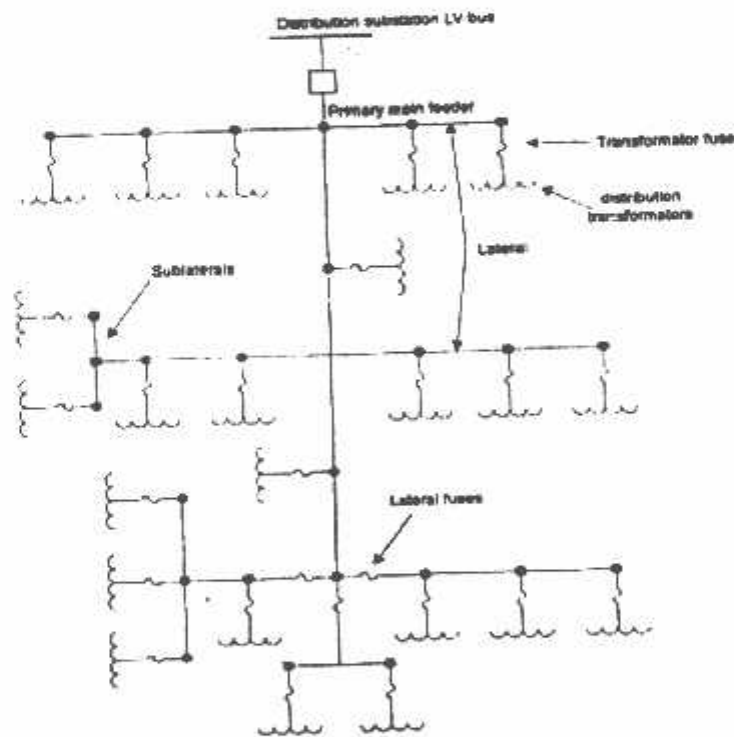
Keuntungan jaringan distribusi radial adalah:

1. Biaya investasi yang paling murah bila dibandingkan dengan sistem yang lainnya.
2. Bentuknya lebih sederhana dibandingkan dengan bentuk yang lainnya

Kerugian jaringan distribusi tipe radial adalah :

1. Kualitas pelayanannya kurang baik karena rugi-rugi daya yang terjadi pada saluran yang relative besar,
2. Kontinuitas pelayanan daya tidak terjamin karena antara sumber dan beban hanya ada satu saluran sehingga apabila terjadi gangguan maka semua dari sistem akan mengalami pemadaman total.

Gambar sistem distribusi jenis radial diperlihatkan pada gambar 2 – 2

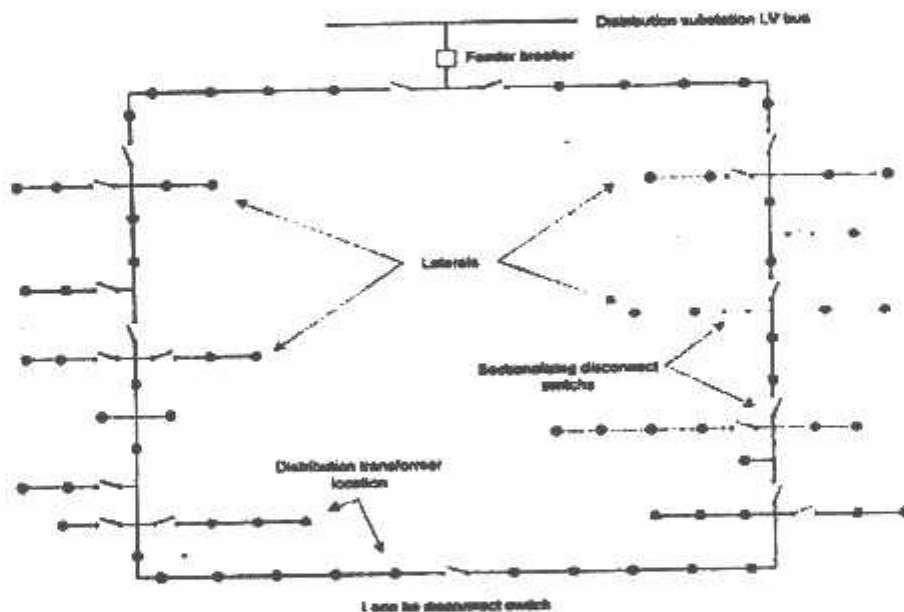


GAMBAR 2 – 2

Sistem Distribusi Primer Tipe Radial^[1]

2.3.2. Sistem Jaringan Distribusi Tipe *Loop*

Sistem distribusi jenis *loop* merupakan gabungan dari dua struktur jaringan radial, dimana pada ujung kedua jaringan dipasang sebuah pemutus (PMT/CB) atau pemisah (PMS/DS). Pada saat gangguan dapat diisolir, maka pemutus atau pemisah ditutup sehingga aliran daya listrik ke bagian yang tidak terkena gangguan tidak berhenti. Dalam kondisi normal, struktur jaringan *loop* ini merupakan dua struktur jaringan radial. Struktur jaringan ini mempunyai keandalan yang cukup, sehingga biaya pembangunannya lebih mahal dibandingkan dengan biaya pembangunan struktur radial.. Gambar sistem jaringan distribusi tipe *loop* dapat dilihat pada gambar 2 – 3.

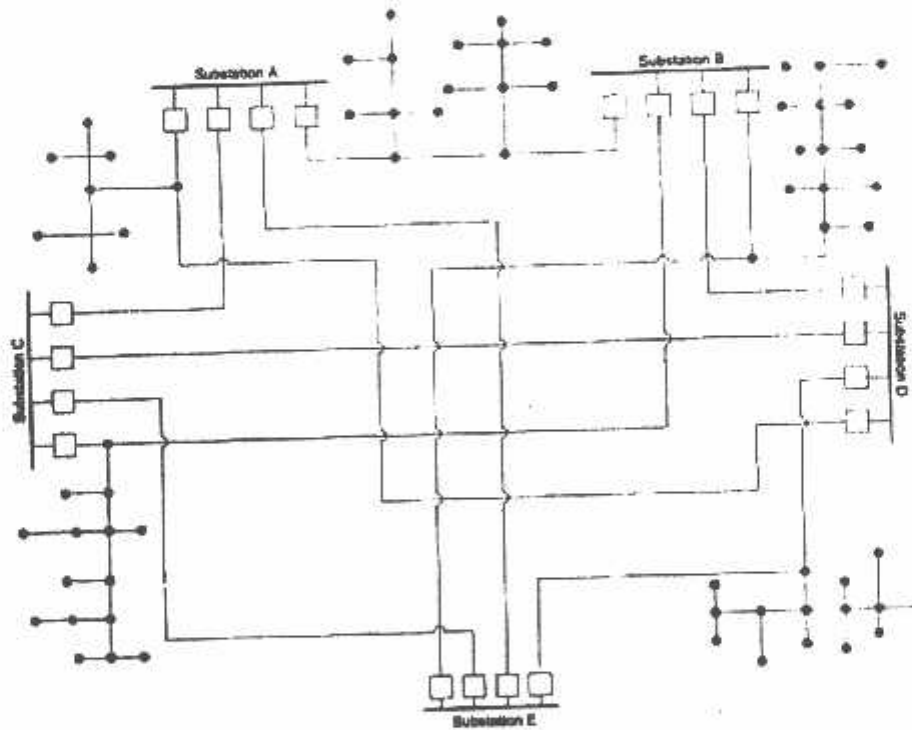


Gambar 2 – 3

Sistem Jaringan Distribusi Tipe *Loop*^[1]

2.3.3 Sistem Jaringan Distribusi Tipe *Mesh/Grid*

Sistem distribusi jenis *mesh/grid* merupakan salah satu sistem interkoneksi dari beberapa penyulang yang disuplai dari sejumlah gardu induk. Penyulang ini bias dilayani dari beberapa gardu induk secara langsung. Sistem ini bisa menyuplai beban dari beberapa arah. Umumnya rugi – rugi yang ditimbulkan relative lebih rendah jika dibandingkan dengan sistem radial. Keandalan dan kualitas pelayanan sangat tinggi jika dibandingkan sistem radial dan sistem *loop*. Walaupun demikian, sangat sulit untuk mendisain dan mengoperasikan sistem ini jika dibandingkan dengan sitem radial dan sistem loop. Sistem jaringan distribusi tipe *mesh/grid* diperlihatkan pada gambar 2 – 4.



Gambar 2-4

Sistem Jaringan Distribusi Tipe *Mesh/Grid*⁽¹⁾

2.4. Konstruksi Jaringan Distribusi

Jaringan sistem distribusi primer mendapat daya dari gardu induk. Cara penyaluran tenaga listrik pada distribusi primer dapat berupa saluran udara dan saluran bawah tanah tergantung pada sejumlah faktor yang sangat berlainan.

1. Sistem Saluran Udara (*Over Head*)

Saluran udara berfungsi menyalurkan daya listrik melalui kawat penghantar yang dipasang pada tiang listrik. Suhu kerja yang diinginkan pada saluran udara tergantung dari batas kerugian kekuatan akibat pelembekan (*Ancalling*). Arus maksimum yang direncanakan untuk suatu beban bagi setiap saluran udara tidak menimbulkan pemanasan. Beberapa keuntungan saluran udara :

- Lebih praktis dan mudah dalam pengerjaannya
- Investasi pembangunan saluran udara jauh lebih murah
- Sangat fleksibel terhadap perubahan beban dan jaring-jaring
- Bila terjadi gangguan, pencariannya mudah karena semuanya terlihat secara langsung .

2. Sistem Saluran Bawah Tanah. (*Under ground*)

Di kota-kota besar pendistribusian tenaga listrik dengan menggunakan sistem saluran udara jarang digunakan dengan alasan keamanan dan estetika.

Saluran distribusi dengan menggunakan kabel bawah tanah mempunyai beberapa keuntungan dibandingkan dengan sistem distribusi saluran udara antara lain :

- Sistem kabel bawah tanah tidak dipengaruhi oleh cuaca seperti hujan, badai, salju dan sebagainya. Pada saluran udara dipengaruhi oleh kondisi-kondisi diatas yang dapat berakibat terputusnya pelayanan tenaga listrik.
- Kabel bawah tanah tidak terganggu oleh binatang seperti burung yang berakibat putusnya konduktor karena benda yang jatuh diatasnya, begitu juga dengan petir. Sebaliknya gangguan tersebut dapat terjadi pada saluran udara.
- Pada daerah dengan kepadatan yang sangat tinggi, saluran udara kurang aman dan sangat potensial untuk mengakibatkan kecelakaan.
- Berkaitan dengan biaya, karena kemungkinan gangguan pada sistem kabel bawah tanah kecil, maka biaya perawatannya juga kecil, sebaliknya biaya perawatan untuk saluran udara besar karena kemungkinan gangguan besar.

Kerugian dari penggunaan kabel bawah tanah :

- Biaya untuk pengerjaan kabel bawah tanah adalah 2x sampai dengan 4 x lebih besar jika dibandingkan dengan biaya pengerjaan sistem saluran udara.
- Walaupun jarang terjadi gangguan pada sistem kabel bawah tanah, akan tetapi sekali terjadi gangguan akan lebih sulit untuk menentukan lokasi dan perbaikannya.

2.4.1 Konduktor Pada Sistem Distribusi

Konduktor berfungsi sebagai media untuk menyalurkan tenaga listrik dari suatu tempat ketempat lain. Material yang digunakan untuk konduktor pada

umumnya adalah tembaga, aluminium, atau campuran aluminium dengan logam lain, sedangkan baja hanya digunakan sebagai tulang kawat (penguat) Dalam pemilihan konduktor, hal yang perlu diperhatikan adalah :

- Memiliki rentang pembebanan yang ekonomis
- Memiliki kapasitas thermal yang cukup untuk menjaga keadaan beban yang tinggi.
- Optimal dalam biaya

Dalam menghitung tahanan konduktor dapat dilakukan dengan menggunakan persamaan :

$$R = \rho \frac{L}{A} \dots\dots\dots (2.1)$$

Dimana :

R = Tahanan konduktor (Ω)

L = Panjang Saluran (m)

A = Luas Penampang (mm^2)

ρ = Tahanan jenis penghantar ($\Omega - m$)

2.4.2. Jenis konduktor saluran udara Pada sistem distribusi primer

1. BCC (*Bare Copper Conductor*)

Bare copper conductor merupakan konduktor udara tembaga telanjang. Konduktor ini diregangkan pada isolator diantara tiang-tiang yang khusus untuk sistem distribusi Konduktor ini terbuat dari bahan tembaga elektrolit dengan kemurnian tidak kurang dari 99,90%.

Tabel 2-1
Konstruksi Konduktor Udara Tembaga Telanjang (BBC)^[8]

Luas Penampang Nominal (mm ²)	Luas Penampang Sebenarnya (mm ²)	Jumlah Kawat	Tahanan DC, 20°C maksimum (ohm/km)	Diameter Penghantar Nominal (mm)	Kapasitas arus maksimum (Amp)	Kuat Tarik Putus Penghantar (N)
6	6,16	1	1,8961	2,80	46	2428
10	10,02	7	1,2160	4,05	76	4049
16	15,89	7	1,0452	5,10	125	6421
25	24,25	7	0,5504	6,30	160	9698
35	34,36	7	0,3296	7,50	200	13545
50	48,36	19	0,2181	9,00	250	19281
95	93,27	19	0,1961	12,50	380	36767
110	103,35	19	0,1762	13,65	410	41444
120	117,10	19	0,1563	14,00	440	46121
150	147,10	37	0,1234	15,70	510	58649
185	181,60	37	0,1007	17,50	585	71587
240	242,50	61	0,0754	20,20	700	96685
300	299,43	61	0,0611	22,50	800	118023
400	400,10	61	0,0457	26,00	960	157719
500	499,10	61	0,0366	29,25	1110	194050

2. AAC (*All Aluminium Conductor*)

All Aluminium Conductor merupakan konduktor aluminium telanjang. Untuk saluran udara yang diregangkan pada isolator diantara tiang-tiang yang khusus untuk sistem distribusi. Konduktor ini terdiri dari kawat yang dipilin konstruksi konduktor AAC diperlihatkan pada tabel 2-2.

Tabel 2-2
Konstruksi Konduktor Udara Aluminium Telanjang (AAC)^[8]

Luas Penampang Nominal (mm ²)	Luas Penampang Sebenarnya (mm ²)	Jumlah Kawat	Tahanan DC, 20°C maksimum (ohm/km)	Diameter Penghantar Nominal (mm)	Berat Penghantar Kira-kira (kg/km)	Kapasitas arus maksimum (Amp)
16	16,84	7	1,0200	5,25	46	110
25	27,83	7	0,7290	6,75	76	145
35	34,36	7	0,3232	7,50	94	180
50	45,70	19	0,2286	9,00	135	255
50	58,07	7	0,2286	8,75	126	255
70	75,55	19	0,2148	11,25	208	270
95	93,27	19	0,2033	12,50	257	340
110	99,30	19	0,2013	12,75	272	350
120	112,85	19	0,1949	13,75	310	390
150	157,12	37	0,1825	16,25	434	455
185	181,63	37	0,1587	17,50	501	520
240	242,54	61	0,1191	20,25	670	625
300	299,44	61	0,0965	22,50	8277	710
400	431,18	61	0,0670	27,00	1195	855
500	506,04	61	0,0288	29,25	1398	990

3. AAAC (*All Aluminium Alloy Conductor*)

All Aluminium Alloy Conductor merupakan udara telanjang yang terbuat dari aluminium campuran. Konduktor ini memiliki kekuatan mekanis yang lebih tinggi dibandingkan AAC. Konduktor ini diregangkan pada isolator diantara tiang-tiang yang khusus untuk sistem distribusi. Konstruksi dari konduktor AAAC ini diperlihatkan pada tabel 2-3.

Tabel 2-3

Konstruksi Penghantar Udara Campuran Aluminium Telanjang (AAAC)^[8]

Luas Penampang Nominal (mm ²)	Luas Penampang Sebenarnya (mm ²)	Jumlah Kawat	Tahanan DC, 20°C maksimum (ohm/km)	Diameter Penghantar Nominal (mm)	Berat Penghantar Kira-kira (kg/km)	Kapasitas arus maksimum (Amp)
16	16,84	7	0,9551	5,25	46	105
25	27,83	7	0,5832	6,75	76	135
35	34,36	7	0,2127	7,50	94	170
50	49,48	7	0,2079	9,00	135	210
50	45,70	19	0,2079	8,75	126	210
70	75,55	19	0,2039	11,25	208	210
95	93,27	19	0,2013	12,50	256	255
100	99,30	7	0,2009	12,75	272	325
110	106,07	19	0,1968	13,25	291	345
120	112,85	19	0,1943	13,75	310	365
150	157,60	19	0,1909	16,25	406	425
185	181,60	37	0,1832	17,50	501	490
240	242,50	61	0,1392	20,00	657	585
300	299,40	61	0,1114	22,50	827	670
400	431,10	91	0,0776	27,00	1191	810
500	506,00	61	0,0664	29,25	1398	930

2.5. Daya Sistem Distribusi^[2]

2.5.1. Daya Nyata (*Real Power*)

Daya nyata dinyatakan dalam persamaan :

$$P = |V| |I| \cos \varphi \dots\dots\dots (2.2)$$

Daya nyata untuk beban tiga fasa seimbang :

$$P = \sqrt{3} |V_{j\text{ala jala}}| |I_{j\text{ala jala}}| \cos \varphi \dots\dots\dots (2.3)$$

2.5.2. Daya Reaktif (*Reaktif Power*)

Daya reaktif dinyatakan dalam persamaan :

$$P = |V| |I| \sin \varphi \dots\dots\dots (2.4)$$

Daya reaktif untuk beban 3 fasa seimbang :

$$P = \sqrt{3} |V_{j\text{ala jala}}| |I_{j\text{ala jala}}| \sin \varphi \dots\dots\dots (2.5)$$

2.5.3. Daya Komplek (*Complex Power*)

Daya kompleks dinyatakan dalam persamaan

$$S = |V| |I| \dots\dots\dots (2.6)$$

Daya kompleks untuk beban 3 fasa seimbang :

$$S = \sqrt{3} |V_{j\text{ala jala}}| |I_{j\text{ala jala}}| \dots\dots\dots (2.7)$$

Apabila fasor tegangan dan arus diketahui, untuk perhitungan daya nyata dan daya reaktif akan lebih mudah dalam bentuk kompleks. Apabila tegangan diantara kutub-kutub beban dan arus yang mengalir dalam beban dinyatakan dengan $V = |V| \angle \alpha$ dan $I = |I| \angle \beta$ maka hasil kali tegangan dengan konjugate arus adalah :

$$VI^* = |V| \angle \alpha \cdot |I| \angle -\beta \dots\dots\dots (2.8)$$

2.6. Beban Sistem Distribusi⁽¹⁾

Secara garis besar beban pada sistem distribusi dapat diklasifikasikan menjadi tiga, yaitu :

1. Beban Perumahan (Rumah Tangga)

Beban perumahan pada umumnya berupa penerangan, alat-alat rumah tangga

2. Beban Komersial

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

3. Beban industri

Beban industri pada umumnya berskala besar.

2.7. Rugi-Rugi Daya Saluran Distribusi

Rugi daya adalah rugi daya yang hilang dalam penyaluran daya elektrik rugi daya terdiri dari rugi daya aktif dan reaktif. Rugi-rugi ini dapat terjadi pada komponen-komponen umum pada sistem tenaga listrik seperti :

1. Rugi pada penyulang utama dan peralatan saluran

2. Rugi pada trafo distribusi

Dalam proses penyaluran energi listrik ke pelanggan terjadi rugi-rugi teknis (*losses*), yaitu rugi daya dan rugi energi, dimulai dari pembangkit, transmisi dan distribusi. Rugi-rugi teknis adalah pada penghantar saluran, adanya tahanan dari penghantar yang dialiri arus maka akan timbul rugi-rugi teknis (I^2R) pada jaringan tersebut. Rugi Rugi teknis dari jaringan tenaga listrik tergantung dari macam-macam pembebanan pada saluran tersebut

BAB III
OPTIMALISASI PEMILIHAN KONDUKTOR
MENGGUNAKAN METODE
EVOLUTIONARY PROGRAMMING

3.1. Analisa Aliran Daya Jaringan^[2]

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

Tujuan dari analisa daya adalah:

1. Untuk mengetahui keadaan tegangan pada setiap node dari sistem tenaga.
2. Untuk mengetahui besarnya daya yang mengalir pada setiap saluran dari struktur jaringan .
3. Untuk mengetahui besar rugi-rugi daya aktif dan daya reaktif pada setiap saluran.

3.1.1. Metode *Newton Raphson*

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

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

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

Injeksi daya pada simpul p adalah :

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

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

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

$$\begin{aligned} V_p^* &= |V_p| e^{-j\delta_p} \\ V_q &= |V_q| e^{j\delta_q} \\ Y_{pq}^* &= |Y_{pq}| e^{-j\theta_{pq}} \dots\dots\dots (3.3) \end{aligned}$$

Maka persamaan (3.1) dapat ditulis :

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

Dengan memisahkan bagian riil dan bagian imajiner maka diperoleh :

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

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

Kedua persamaan diatas akan menghasilkan suatu kumpulan persamaan serempak(simultan) yang tidak linier untuk setiap simpul sistem tenaga listrik. Untuk mengetahui magnitude tegangan (V) dan sudut fasa tegangan (δ) di setiap simpul dapat diselesaikan dengan persamaan (3.5) dan (3.6) yang di

linierkan dengan metode *Newton Rapson* yang dapat dilihat pada persamaan dibawah ini :

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

Dimana :

ΔP = Selisih injeksi bersih daya nyata dengan penjumlahan aliran daya nyata tiap saluran yang menghubungkan simpul dengan V yang didapat dari perhitungan iterasi ke-k

ΔQ = Selisih injeksi bersih daya reaktif dngan penjumlahan aliran daya reaktif tiap saluran yang menghubungkan simpul dengan V yang didapat dari perhitungan iterasi ke-k

$\Delta \delta$ = Vektor koreksi sudut fasa tegangan

$\Delta |V|$ = Vektor koreksi magnitude tegangan

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

Jaqobian yang dibentuk dengan mendefinisikan persamaan (3.5) dan (3.6),

dimana :

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

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

Adapun rumus dari elemen matriks jacobian adalah :

Untuk H :

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

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

Untuk N :

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

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

Untuk M :

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

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

Untuk L :

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

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

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

untuk turunan parsial yang ditentukan dengan differensial persamaan (3.5) dan (3.6).

4. Balikan jacobian itu dan hitung koreksi-koreksi tegangan $\Delta \delta_p$ dan $\Delta |V_p|$ pada setiap rel.
5. Hitung nilai baru dari δ_p dan $\Delta |V_p|$ dengan menambahkan $\Delta \delta_p$ dan $\Delta |V_p|$ pada nilai sebelumnya.
6. Kembali kelangkah pertama dan ulangi proses itu dengan menggunakan nilai untuk besar dan sudut tegangan yang ditentukan paling akhir sehingga semua nilai ΔP dan ΔQ atau semua nilai $\Delta \delta$ dan $\Delta |V|$ lebih kecil dari suatu indeks ketetapan yang telah dipilih.

3.2. *Evolutionary Programming*^{[3][4]}

Evolutionary Programming (EP) merupakan metode stokastik yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Metode ini didasarkan pada proses evolusi yang ada dalam mahluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam “siapa yang kuat, dia yang bertahan (*survive*)”. Dengan meniru proses ini, EP dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

EP adalah suatu metode strategi optimasi yang merupakan cabang dari *Evolutionary Computation* yang didalamnya terdiri dari Algoritma Genetika, *Genetic Programming*, *Evolutionary Strategies*, dan *Evolutionary Programming*. Perbedaan yang paling mendasar antara *Evolutionary Programming* dengan Algoritma Genetika adalah pada proses operasi. Dalam metode *Evolutionary*

Programming tidak menggunakan operasi *crossover* melainkan operasi *competition* (kompetisi).

EP ditemukan oleh Lawrence.J. Fogel dilandasi oleh sifat-sifat evolusi alam. Fogel percaya bahwa ini sangat cocok digabungkan dalam sebuah algoritma komputer, menghasilkan sebuah teknik penyelesaian untuk permasalahan-permasalahan yang sulit dengan langkah alami yaitu melalui evolusi. Fogel mulai bekerja dengan algoritma yang dibentuk dari string-string bilangan real yang disebut *kromosom*. Seperti halnya alam, metode ini menyelesaikan permasalahan-permasalahan dengan menemukan kromosom-kromosom yang baik dengan memanipulasi materi dan sifat (*gene*) kromosom-kromosom. Algoritma ini tidak mengetahui tipe permasalahan yang akan diselesaikan.

Sebelum EP dijalankan, maka sebuah kode yang sesuai (representasi) untuk persoalan harus dirancang. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pemakaian bilangan real (*floating point*) sebagai *allele* (nilai gen) memungkinkan penerapan operator EP yaitu proses produksi (*reproduction*), mutasi (*mutation*), dan kompetisi (*competition*) untuk menciptakan himpunan titik-titik solusi. Untuk memeriksa hasil optimasi, kita membutuhkan fungsi *fitness* yang menandakan gambaran hasil (*solution*) yang sudah dikodekan. Selama proses, induk harus digunakan untuk reproduksi, mutasi dan kompetisi untuk menciptakan keturunan.

3.2.1. Parameter *Evolutionary Programming*

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

1. Jumlah Generasi (*MAXGEN*)

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

2. Ukuran Populasi (*POPSIZE*)

Ukuran populasi mempengaruhi kinerja dan efektifitas dari EP. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan, sehingga pada umumnya kinerja EP menjadi buruk. Dalam hal ini dibutuhkan ruang yang lebih besar untuk mempersentasikan keseluruhan ruang permasalahan. Selain itu penggunaan populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah lokal.

3. Probabilitas Mutasi (*Pm*)

Mutasi digunakan untuk meningkatkan variasi populasi dan menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi $Pm \times POPSIZE \times N$, dimana N adalah panjang struktur/gen dalam satu individu. Probabilitas mutasi yang rendah akan menyebabkan gen-gen yang berpotensi

tidak dicoba. Dan sebaliknya, tingkat mutasi yang tinggi akan menyebabkan keturunan akan semakin mirip dengan induknya. Dalam EP mutasi menjalankan aturan penting yaitu :

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

4. Panjang Kromosom (*NVAR*)

Panjang kromosom berbeda-beda sesuai dengan model permasalahan. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom/string yang terdiri dari komponen genetik terkecil yaitu gen. Pengkodean memakai *string* bilangan *real*.

3.2.2. Mekanisme *Evolutionary Programming*

1. Pengkodean atau Representasi

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

Secara umum EP dibentuk oleh serangkaian kromosom yang ditandai dengan x_i ($i = 1, 2, \dots, N$). Setiap elemen dalam kromosom ini adalah variabel string yang disebut gen, berisi nilai-nilai atau allele. Variabel-variabel ini dapat dinyatakan dalam bentuk bilangan real (*floating point*).

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

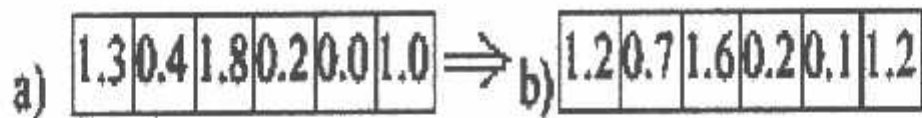
2. Fungsi *Fitness* (Fungsi Evaluasi)

Dalam EP, sebuah fungsi *fitness* $f(x)$ harus dirancang untuk masing-masing permasalahan yang akan diselesaikan. Dengan menggunakan kromosom tertentu, fungsi obyektif atau fungsi evaluasi akan mengevaluasi status masing-masing kromosom. Setiap gen x_i ($i = 1, 2, \dots, N$) dipergunakan untuk menghitung $f_k(x)$ ($k = 1, 2, \dots, \text{POPSIZE}$).

Pada permulaan optimasi, biasanya nilai *fitness* masing-masing individu masih mempunyai rentang yang lebar. Seiring dengan bertambah besar generasi, beberapa kromosom mendominasi populasi dan mengakibatkan rentang nilai *fitness* semakin kecil.

3. *Mutation* (Mutasi)

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam satu individu. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana $k = 1, 2, \dots, \text{NVAR}$ (panjang kromosom). Probabilitas mutasi (P_m) ditentukan dan digunakan untuk mengendalikan frekuensi operator mutasi. Apabila nilai random $r_k < P_m$ maka gen ke- k kromosom tersebut terpilih untuk mengalami mutasi. Proses mutasi dalam *Evolutionary Programming* sama dengan *Evolutionary Strategies* yaitu menggunakan operator *Gaussian mutation*, dimana setiap individu akan terpilih secara acak untuk mengalami mutasi berdasarkan nomor acak Gaussian untuk menciptakan individu baru (*offspring*).



Gambar 3.1.

Mutasi Gaussian Dari Induk (*parent*) a) Menghasilkan Anak (*offspring*) b)^[6]

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

4. *Competition* (Kompetisi)

Dalam tahap kompetisi, mekanisme seleksi dipakai untuk menghasilkan populasi baru dari populasi yang ada. Melalui penggunaan skema kompetisi setiap individu dalam populasi baik orang tua (*parent*) maupun anak (*offspring*) akan dikompetisi/bersaing satu dengan yang lainnya. Kompetisi setiap individu dengan lawannya didasarkan pada nilai *fitness* dari setiap individu tersebut. Agar optimal, solusi yang lebih pas atau lebih optimal seharusnya memiliki peluang seleksi yang lebih besar. Individu yang memenangkan dari kompetisi akan digunakan sebagai individu yang baru bagi pembangkitan selanjutnya.

3.3. Formulasi Masalah Pemilihan Konduktor

Masalah pemilihan konduktor berupaya mengoptimalkan besar biaya investasi dengan fungsi objektif f dan subyeknya berbagai batasan. Tujuan lainnya yaitu mencakup meminimalan kerugian daya dan pengoptimalan tingkat tegangan. Secara matematis dapat dirumuskan sebagai berikut :

$$\text{Min } f(x,u) \dots\dots\dots(3.8.)$$

Subjeknya:

$$g(x,u) = 0 \dots\dots\dots(3.9)$$

$$h(x,u) \leq 0 \dots\dots\dots(3.10)$$

Dimana u adalah vektor kontrol variabel yang terdiri dari daya aktif dan tegangan ; x adalah vektor variabel konduktor ; $f(x,u)$ adalah fungsi yang akan dioptimalkan; $g(x,u)$ adalah batasan aliran daya; dan $h(x,u)$ adalah batasan fungsi penalti.

3.4. Adaptasi *Evolutionary Programming* Ke Masalah *Optimal* Pemilihan konduktor.

Evolutionary Programming adalah suatu pada mekanisme seleksi alam. Individu dari sebuah populasi dikodekan secara real, Populasi pertama dibangkitkan secara random. Generasi baru dibuat dengan mengaplikasikan 3 operator berikut terhadap sebuah populasi yaitu : inialisasi, mutasi dan kompetisi dimana tingkat pengoptimalan setiap calon solusi atau individu diukur

dengan ketepatannya tergantung pada fungsi tujuan (*objective function*) permasalahan.

3.4.1. Representasi Solusi

Individu dalam populasi menunjukkan calon pada solusi Pemilihan konduktor. Elemen solusi terdiri dari daya aktif, daya reaktif, magnitudo tegangan, data saluran dan data konduktor.

3.4.2. Inisialisasi

Setiap variabel dari setiap individu dilakukan inisialisasi secara acak menggunakan distribusi nomor acak.

3.4.3. Fitness Calon Solusi^[5]

Setiap calon solusi menugaskan fungsi *Fitness* untuk mengukur ke optimalannya yang berkenaan dengan sasaran yang akan dioptimalkan. Dalam permasalahan pemilihan konduktor, fungsi *fitness* dari setiap individu *i* adalah:

$$f_i = \frac{M}{C_i + \sum_j VP_j + SQ} \dots\dots\dots(3.12.)$$

$$VP_j = \begin{cases} K_v (V_j - 10)^2 & \text{jika } V_j > V_j^{\max} \text{ atau } V_j < V_j^{\min} \\ 0 & \end{cases} \dots\dots\dots(3.13)$$

$$SQ = \begin{cases} K_q (w - w^{\max})^2 & \text{jika } w > w^{\max} \\ K_q (w - w^{\min})^2 & \text{jika } w < w^{\min} \\ 0 & \end{cases} \dots\dots\dots(3.14)$$

Dalam rumus di atas, *M* adalah kemungkinan biaya maksimum investasi sedang *C_i* adalah biaya investasi dari setiap individu *i*. *VP_j* adalah istilah hukuman/penalti pada konduktor atas pelanggaran batasan tegangan yang diset

sebelumnya V_j^{\min}, V_j^{\max} . SQ merepresentasikan hukuman/penalti pada konduktor untuk pelanggaran batas. K_v dan K_q adalah konstanta penalti/hukuman.

3.4.4. Menghasilkan Solusi Baru Dengan Mutasi^[7]

Populasi yang baru dari solusi pemilihan konduktor dihasilkan dari populasi yang ada melalui operator mutasi. Individu yang baru p_i' dihasilkan dari setiap individu p_i , dimana variabel j konduktor pada individu yang baru p_i' dihitung seperti:

$$x_{ji}' = x_{ji} + N(0, \sigma_{ji}^2) \dots\dots\dots(3.15.)$$

Dimana x_{ji}' menandakan nilai dari variabel j dalam p_i' . x_{ji} adalah dari variabel j dalam induk p_i dan $N(0, \sigma_{ji}^2)$ adalah nomor acak *Gaussian* dengan nilai tengah nol dan standart deviasi (simpangan baku) dari σ_{ji} . Cara untuk merancang σ_{ji} adalah:

$$\sigma_{ji}^2 = (x_j^{\max} - x_j^{\min}) \left((f_{\max} - f_i) / f_{\max} + a^r \right) \dots\dots\dots(3.16.)$$

Dimana f_i adalah fitness dari individu i ; f_{\max} adalah *fitness* maksimum di dalam populasi; x_j^{\max}, x_j^{\min} menandakan batasan atas dan bawah dari variabel j , a adalah konstanta angka positif; dan r adalah iterasi yang berulang – ulang. Istilah a^r menandakan suatu tingkat mutasi yang mana tergantung pada nilai a .

3.4.5. Seleksi Individu Dengan Kompetisi

Dalam tahap kompetisi, mekanisme seleksi dipakai untuk menghasilkan populasi baru dari populasi yang ada. Agar optimal, solusi yang lebih pas atau lebih optimal seharusnya memiliki peluang seleksi yang lebih besar. Teknik seleksi dipakai adalah skema *tournament* yang dapat dideskripsikan sebagai berikut :

k sebagai solusi induk (*parent*) p_i , seiring dengan anak (*offspring*) p_i terbentuk oleh mutasi, $i = 1, \dots, k$, masing – masing mengalami serangkaian turnamen n_j dengan lawan yang terseleksi secara acak. Setiap individu i diberi suatu skor S_i berdasarkan :

$$S_i = \sum_{j=1}^N n_j \dots\dots\dots(3.21.)$$

$$n_j = \begin{cases} 1 & \text{jika } f_i > f_r \\ 0 & \end{cases} \dots\dots\dots(3.22.)$$

Sedang f_i adalah *fitness* setiap individu i . Lawan r , terpilih acak dari setiap individu $2k$ berdasarkan $r = [2ku + 1]$. u adalah angka acak yang sama dalam interval $[0,1]$. k adalah solusi calon skor tertinggi diambil dari individu generasi selanjutnya.

BAB IV
ANALISA HASIL MENGGUNAKAN METODE
EVOLUTIONARY PROGRAMMING

4.1. Program Komputer Optimal Pemilihan Konduktor Menggunakan Metode *Evolutionary Programming*

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

Program komputer dalam skripsi ini dijalankan dengan menggunakan bahasa pemrograman *Borland Delphi* versi 7.0 dan diaplikasikan pada komputer pentium IV prosesor 1,8 GHz, dengan memori 128 Mb.

4.1.1. Algoritma Program Pemecahan Masalah Secara Umum

1. Memasukkan inputan data beban yang meliputi tegangan dasar (V), sudut fasa tegangan (δ), daya aktif (P), daya reaktif (Q), data impedansi saluran dan data konduktor
2. Lakukan proses *load flow Newton Raphson*.
3. Kemudian Mengecek apakah ada pelanggaran tegangan
 - Jika, Ya lanjutkan kelangkah 4
 - Jika, Tidak langsung kelangkah 5
4. Melakukan proses *Evolutionary Programming*
5. Cetak hasil

4.1.2. Algoritma Program Pemecahan Masalah Pemilihan Konduktor Menggunakan Metode *Evolutionary Programming* :

- 1 Memasukkan inputan data beban yang meliputi meliputi tegangan dasar (V), sudut fasa tegangan (δ), daya aktif (P), daya reaktif (Q) , data impedansi saluran dan data konduktor.
- 2 Menentukan parameter inputan EP yang meliputi jumlah populasi, maksimum generasi, nilai kemungkinan mutasi, dan panjang kromosom tiap-tiap individu.
- 3 Generasi = 0, Populasi = 0.
- 4 Melakukan inisialisasi calon solusi
- 5 Menghitung *fitness parent*
- 6 Melakukan proses *statistik*
- 7 Melakukan proses *mutasi*.
- 8 Apakah *offspring* yang diinginkan sudah terpenuhi (*max pop*)
- 9 Jika “ Tidak “ maka generasi = pop + 1, kembali ke langkah 7.
- 10 Jika “ Ya ” maka lanjutkan ke langkah 11
- 11 Menghitung *fitness* dari *offspring*.
- 12 Melakukan proses kompetisi *offspring & parent*.
- 13 Melakukan proses statistik.
- 14 Apakah generasi yang diinginkan sudah terpenuhi (*max Gen*).
- 15 Jika “Tidak “ maka generasi = gen + 1, kembali ke langkah 7.
- 16 Jika “ Ya “ maka perhitungan berhenti.

4.1.3. Algoritma Program *Fitness*

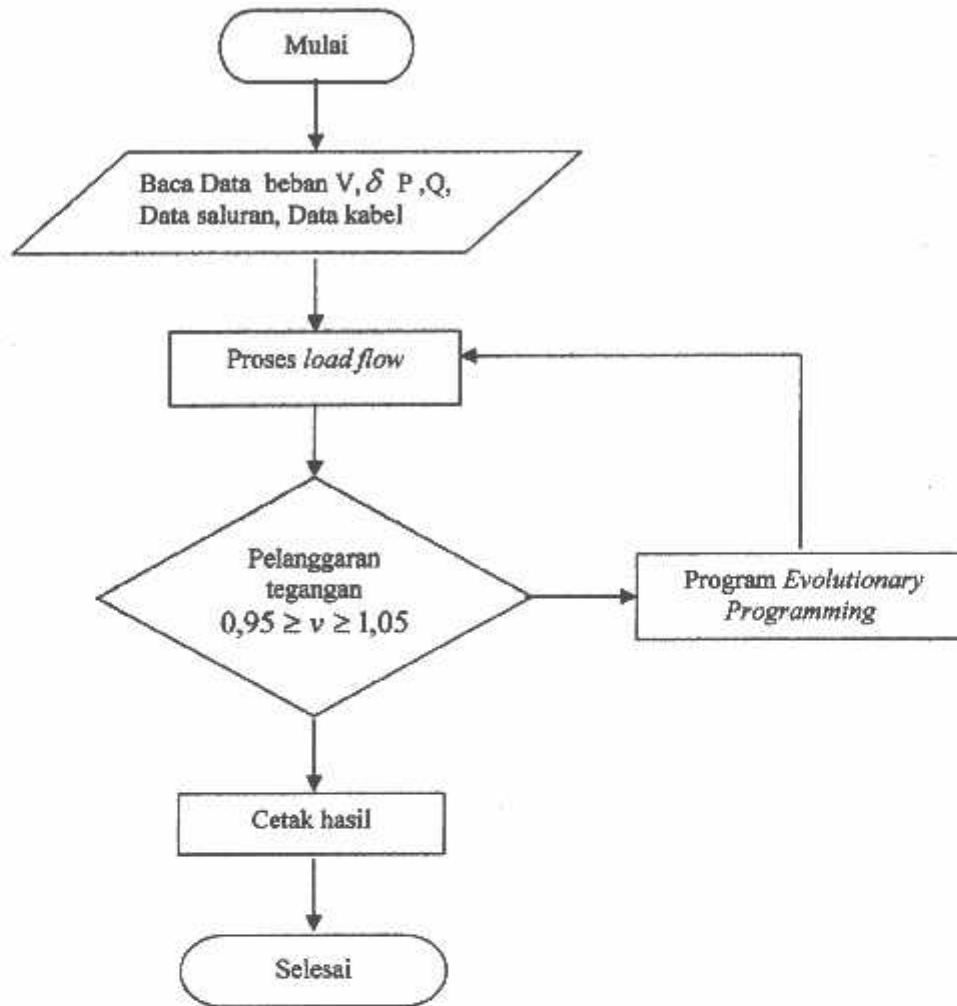
1. Memasukkan inputan data beban yang meliputi meliputi tegangan dasar (V), sudut fasa tegangan (δ), daya aktif (P), daya reaktif (Q) , data impedansi saluran dan data konduktor.
2. Melakukan pembacaan data kromosom.
3. Melakukan proses *Laod Flow*.
4. Hitung biaya rugi daya tahunan [$\Psi(jj,k) = Kp * LP(jj,k) * LP(jj,k) * T * Lsf$].
5. Hitung besar biaya investasi Konduktor [$\zeta(jj,k) = \alpha * A(k) * C * LEN(jj,k)$].

$$6. \text{ Cari pelanggaran batas tegangan} = \sqrt{\frac{\sum_{l=1}^{N/B} (V_{Ll} - V_{Ll,lim})^2}{N}}$$

$$7. \text{ Cari pelanggaran batas arus(pa)} = |I(jj,k)| < Cmax(k).$$

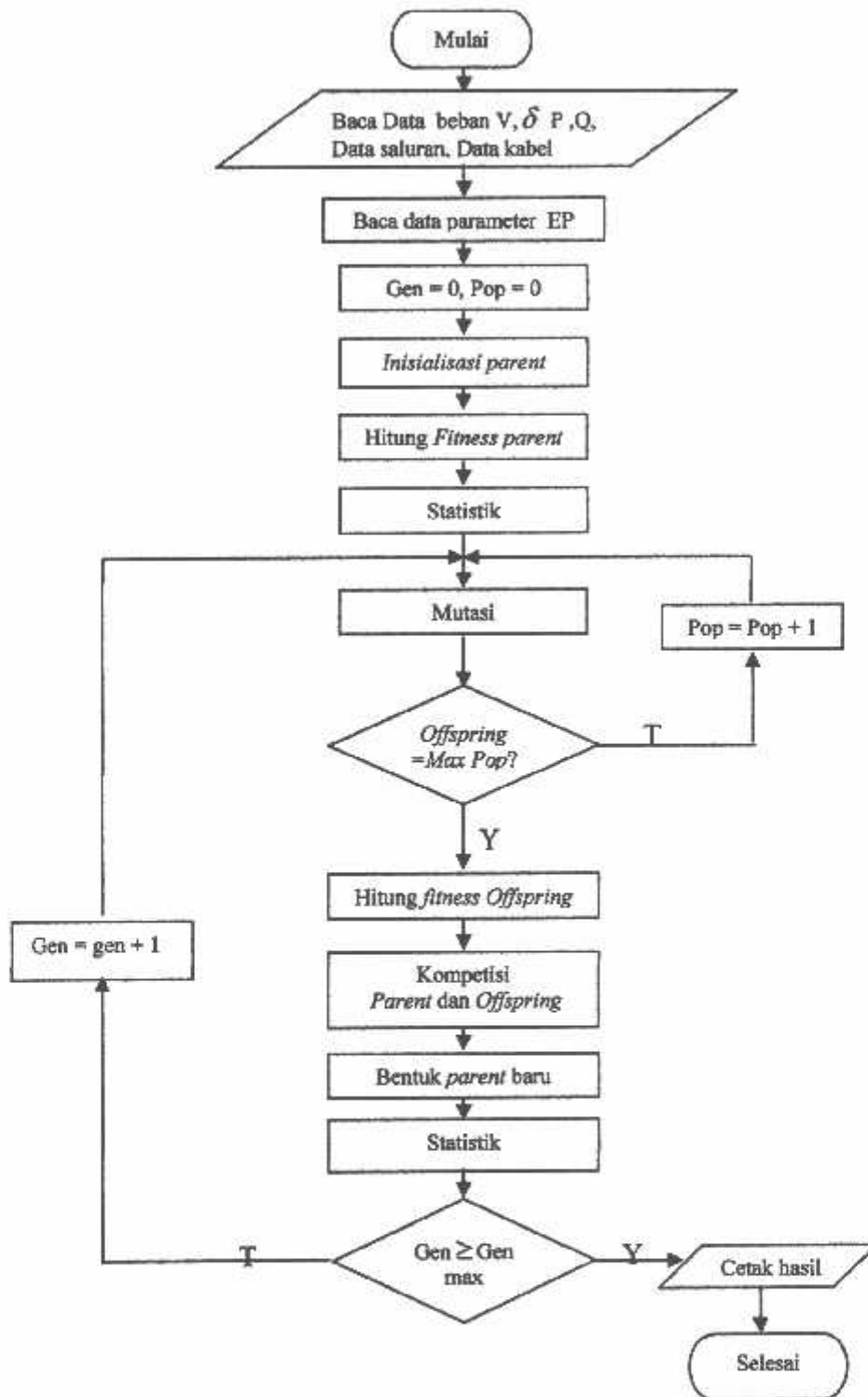
$$8. \text{ Hitung } Fitness = \Psi(jj,k) + \zeta(jj,k) + VDI + pa.$$

4.1.4. Flowchart Program Pemecahan Masalah Secara Umum

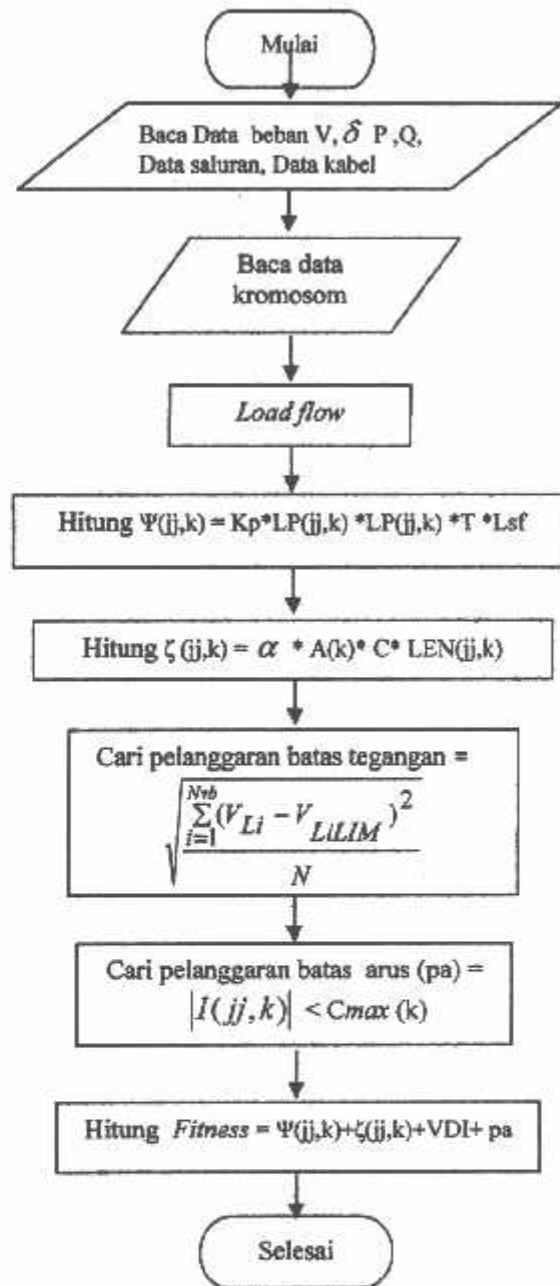


Gambar 4.1. Flowchart Pemecahan Masalah Secara Umum

4.1.5. *Flowchart Program Pemecahan Masalah Pemilihan Konduktor Menggunakan Metode Evolutionary Programming*



Gambar 4.2. *Flowchart Evolutionary Programming*

4.1.6. Flowchart Program *Fitness*Gambar 4.3. Flowchart Program *Fitness*

4.2.1. Data Saluran^[7]

Untuk menyalurkan tenaga listrik dari GI Sengkaling ke pelanggan, pada penyulang Pujon digunakan saluran jenis udara, dengan spesifikasi jenis konduktor yang digunakan adalah seperti pada tabel dibawah ini.

Tabel 4-1

Data Konduktor Penyulang Pujon 20 kV GI sengkaling^[7]

Jenis Konduktor	Penampang Nominal (mm ²)	Impedansi Saluran (Ω /km)	Kapasitas Arus Maksimum (Amp)
AAAC	35	0.2127 + j 0.3005	120
AAAC	50	0.2079 + j 0.2913	160
AAAC	70	0.2036 + j 0.2842	205
AAAC	110	0.1943 + j 0.2826	340
AAAC	150	0.1909 + j 0.2815	425

Tabel 4-1 Selengkapnya dapat dilihat pada lampiran

Tabel 4-2

Data Saluran Penyulang Pujon^[7]

Nomer Saluran	Dari Node	Ke Node	Panjang Saluran(m)	Impedansi Saluran	
				R (Ohm)	X (Ohm)
1	1	2	580.3	0.1255	0.1918
2	2	3	1432.6	0.1938	0.2963
3	3	4	863.9	0.1868	0.2855
4	4	5	85.4	0.0185	0.0282
5	4	6	683.3	0.1477	0.2258

Tabel 4-2 Selengkapnya dapat dilihat pada lampiran

Tabel 4-3

Daftar Harga Konduktor AAAC^[7]

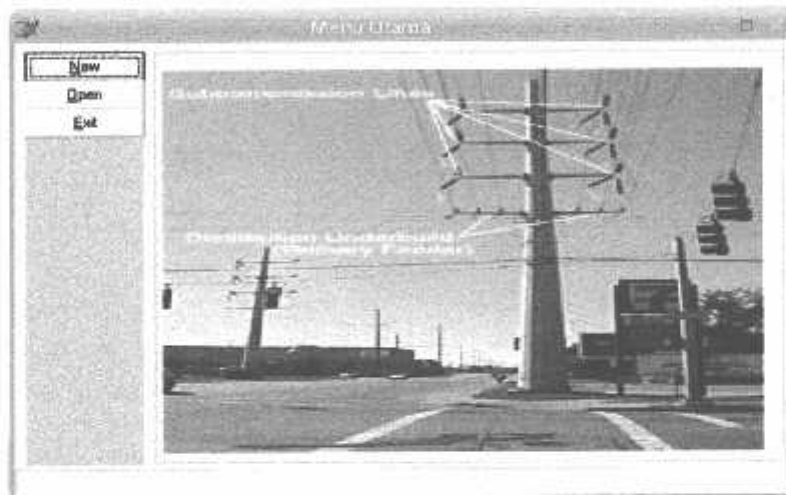
Jenis Konduktor	Penampang Nominal (mm ²)	Daftar harga Konduktor/m (Rp)	Daftar harga 3 Konduktor/m (Rp)
AAAC	35	3972	11.916
AAAC	50	4856	14.568
AAAC	70	6034	18.102
AAAC	110	8390	25.170
AAAC	150	10746	32.238

Tabel 4-3 Selengkapnya dapat dilihat pada lampiran

4.4. Tampilan Program^[8]

Program dalam skripsi ini dijalankan dengan menggunakan program *Borland Delphi versi 7.0* dan diaplikasikan pada komputer berprosesor Pentium IV 1800 MHz, dengan memori 128 Mb. Mengenai jalannya program ikuti prosedur program sebagai berikut :

1. Tampilan utama dari Program



Gambar 4-4. Tampilan Utama Program

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

Gambar 4-5. Tampilan Inputan Data

3. Kemudian tekan tombol data bus

Bus	absV (pu)	audV (deg)	Pg (kW)	Qg (KVAR)	PL (kW)	QL (KVAR)	Cap (ohm)	Type Bus
1	0	0	0	0	0	0	0	1
2	0	0	0	0	38.7	22.96	0	3
3	0	0	0	0	98.027	58.775	0	3
4	0	0	0	0	0	0	0	3
5	0	0	0	0	115.584	88.571	0	3
6	0	0	0	0	22.575	13.393	0	3
7	0	0	0	0	0	0	0	3
8	0	0	0	0	57.752	34.286	0	3
9	0	0	0	0	65.79	39.03	0	3
10	0	0	0	0	82.96	48.979	0	3
11	0	0	0	0	74.304	44.082	0	3
12	0	0	0	0	0	0	0	3
13	0	0	0	0	19.78	11.735	0	3
14	0	0	0	0	66.048	39.184	0	3

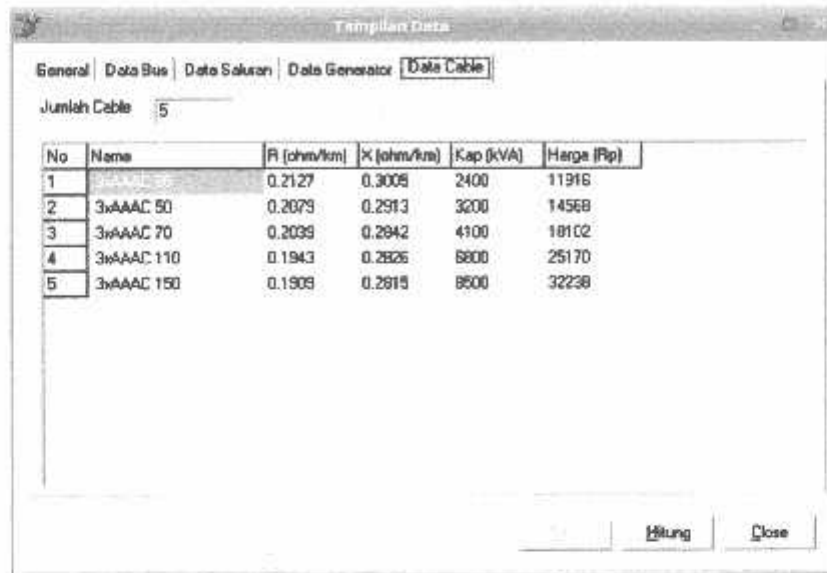
Gambar 4-6. Tampilan Data bus

4. Kemudian tekan tombol data saluran

No	Dari	Ka	R (ohm)	X (ohm)	Lc (ohm)	Tr	Tu	Su (deg)	Kap (kVA)
1	2	3	0.1255	0.1918	0	0	0	0	10000
2	2	3	0.3057	0.4735	0	0	0	0	10000
3	3	4	0.1863	0.2855	0	0	0	0	10000
4	4	5	0.0185	0.0282	0	0	0	0	10000
5	4	6	0.1477	0.2258	0	0	0	0	10000
6	6	7	0.4503	0.6884	0	0	0	0	10000
7	7	8	0.0334	0.051	0	0	0	0	10000
8	8	9	0.0696	0.137	0	0	0	0	10000
9	9	10	0.1338	0.2053	0	0	0	0	10000
10	10	11	0.1002	0.1532	0	0	0	0	10000
11	11	12	0.0977	0.1494	0	0	0	0	10000
12	12	13	0.308	0.4706	0	0	0	0	10000
13	12	14	0.2668	0.4078	0	0	0	0	10000
14	7	15	0.2691	0.4114	0	0	0	0	10000

Gambar 4-7. Tampilan Data Saluran

5. Tekan tombol data *cabl*



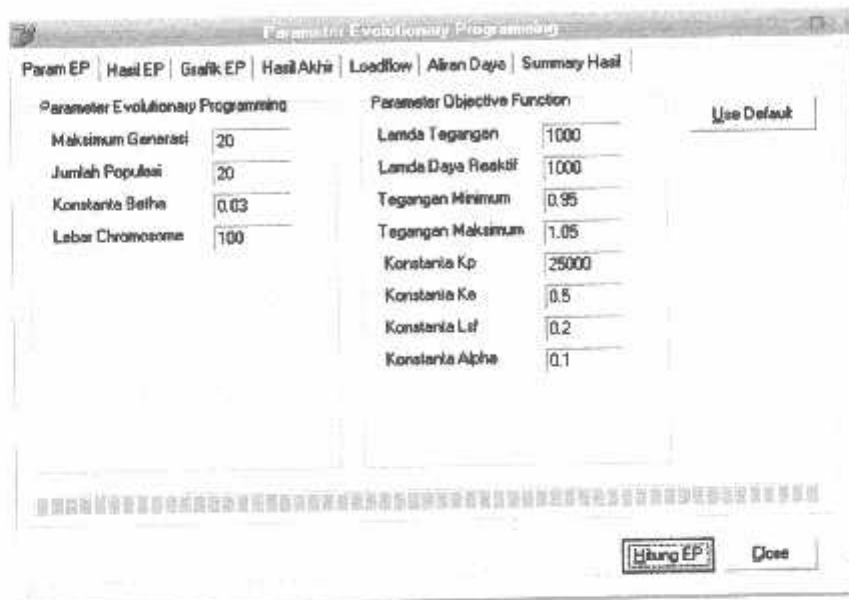
Gambar 4-8. Data *Cable*

6. Tekan tombol hitung maka dengan *newton rapshon* dapat melihat hasil perhitungan aliran daya pada kondisi awal.



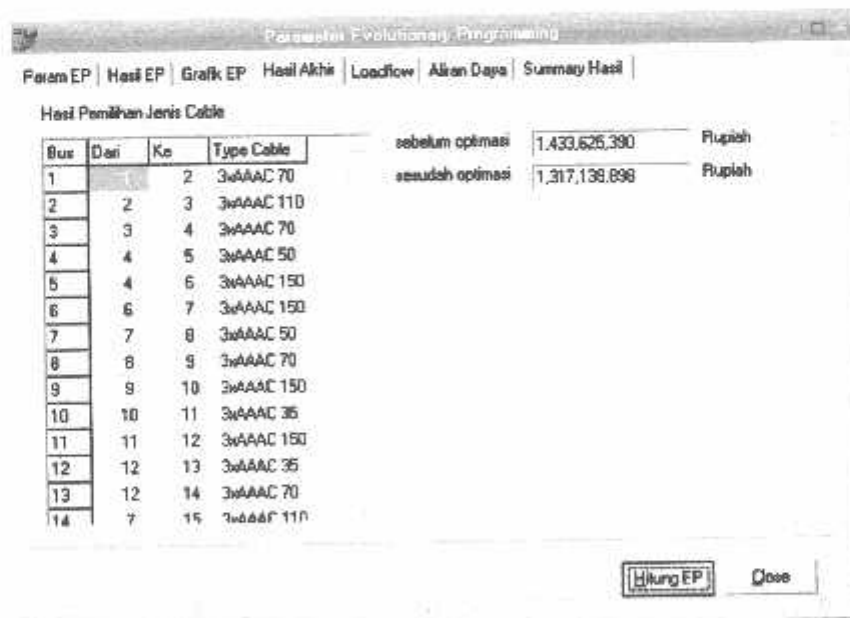
Gambar 4-9. Hasil Aliran Daya Sebelum Pemilihan Konduktor dengan Metode *Evolutionary Programming*.

7. Kemudian tekan *next* untuk melihat parameter *Evolutionary Programming* kemudian tekan tombol *Use Default*.



Gambar 4-10. Parameter *Evolutionary Programming*

8. Kemudian tekan tombol hitung EP.



Gambar 4-11. Hasil *Evolutionary Programming* Dalam Pemilihan Konduktor

9. Kemudian pilih tombol *Load Flow* dan tombol Aliran Daya

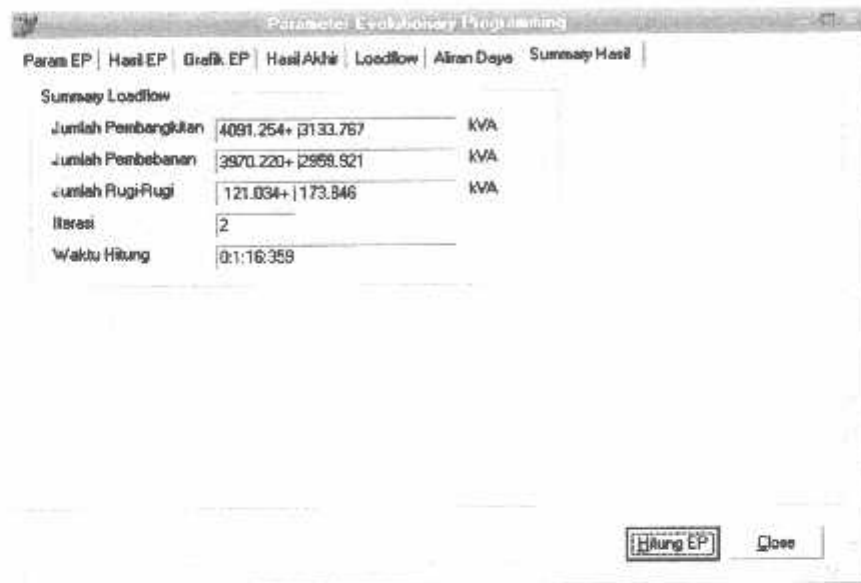
Bus	v(pu)	δ(deg)	Pg (kW)	Qg (kVAR)	PL (kW)	QL (kVAR)	Supe (pu)	Type Bus
1	1.0000	0.0000	4091.254	3133.757	0.000	0.000	0.000	1
2	0.98790	-0.00076	0.000	0.000	38.700	22.960	0.000	3
3	0.99153	-0.00272	0.000	0.000	98.027	58.775	0.000	3
4	0.98792	-0.00383	0.000	0.000	0.000	0.000	0.000	3
5	0.96791	-0.00383	0.000	0.000	115.584	68.571	0.000	3
6	0.96524	-0.00472	0.000	0.000	22.575	13.353	0.000	3
7	0.97712	-0.00746	0.000	0.000	0.000	0.000	0.000	3
8	0.97706	-0.00749	0.000	0.000	57.792	34.266	0.000	3
9	0.97594	-0.00754	0.000	0.000	85.790	36.030	0.000	3
10	0.97574	-0.00754	0.000	0.000	82.960	48.979	0.000	3
11	0.97666	-0.00767	0.000	0.000	74.304	44.062	0.000	3
12	0.97663	-0.00769	0.000	0.000	0.000	0.000	0.000	3
13	0.97680	-0.00770	0.000	0.000	19.780	11.736	0.000	3
14	0.97656	-0.00773	0.000	0.000	66.048	38.194	0.000	3
15	0.97264	-0.00890	0.000	0.000	0.000	0.000	0.000	3
16	0.97263	-0.00891	0.000	0.000	32.190	19.102	0.000	3

No	Dari	Ke	P (kW)	Q (kVAR)	Asus re (A)	Asus im (A)	Dari	Ke	P (kW)
1	2	3	4091.254	3133.757	204.503	156.588	2	1	-4093.390
2	2	3	4044.898	3099.856	202.624	155.536	3	2	-4026.536
3	3	4	3927.909	3014.696	197.638	152.993	4	3	-3916.529
4	4	5	115.585	68.572	5.837	3.493	5	4	-115.584
5	4	6	3800.944	2830.789	191.802	149.065	5	4	-3793.247
6	6	7	3770.672	2906.046	190.629	148.381	7	6	-3747.465
7	7	8	366.370	217.433	18.884	11.266	8	7	-366.365
8	8	9	308.563	183.126	15.720	8.489	8	8	-308.534
9	9	10	242.744	144.055	12.368	7.466	10	9	-242.708
10	10	11	160.149	95.023	8.161	4.927	11	10	-160.136
11	11	12	85.836	50.926	4.374	2.641	12	11	-85.832
12	12	13	19.780	11.736	1.008	0.609	13	12	-19.790
13	7	14	66.052	38.190	3.368	2.032	14	12	66.048
14	7	15	3381.095	2684.352	171.995	137.115	15	7	-3369.393
15	15	16	130.239	77.267	6.660	4.031	16	15	-130.238

Gambar 4-12. Hasil Aliran Daya Sesudah Pemilihan Konduktor Dengan Metode *Evolutionary Programming*.

10. Sesudah Pilih Tombol *Summary* Hasil Untuk Mengetahui Total Rugi-Rugi Saluran Sesudah Pemilihan Konduktor dengan Metode *Evolutionary Programming*.

10. Sesudah Pilih Tombol *Summary* Hasil Untuk Mengetahui Total Rugi-Rugi Saluran Sesudah Pemilihan Konduktor dengan Metode *Evolutionary Programming*.



Gambar 4-13. Total Rugi-Rugi Saluran Sesudah Pemilihan Konduktor Dengan Metode *Evolutionary Programming*.

4.5. Hasil Perhitungan

Dengan melakukan analisa dengan metode *Newton Raphson* maka akan didapatkan besarnya tegangan tiap-tiap *node*, besarnya rugi saluran tiap saluran, Dan aliran daya saluran seperti pada Tabel 4-5 sampai Tabel 4-8 :

Tabel 4-5

Kondisi Tegangan Dan Sudut Fasa Tegangan Tiap *Node* Penyulang Pujon Sebelum Optimasi.

Nomer <i>Node</i>	Tegangan (pu)	Sudut (deg)
1	1.00000	0.00000
2	1.99752	-0.00057
3	0.99146	-0.00196
4	0.98758	-0.00319
5	0.98757	-0.00319

Tabel 4-5 Selengkapanya dapat dilihat pada lampiran

Tabel 4-6

Besarnya Rugi-Rugi Daya Tiap Saluran Penyulang Pujon Sebelum Optimasi

Saluran	P (kW)	Q (kVAR)
1	8.385	10.142
2	20.336	24.525
3	11.72	61.658
4	0.001	0.001
5	8.774	12.225

Tabel 4-6 Selengkapanya dapat dilihat pada lampiran

Tabel 4-7

Besarnya Aliran Daya Tiap Saluran
Penyulang Pujon Sebelum optimasi.

Saluran	P (kW)	Q (kVAR)
1	4102.845	3145.325
2	4045.76	3112.222
3	3936.397	3028.922
4	115.585	68.572
5	3809.091	2943.691

Tabel 4-7 Selengkapnya dapat dilihat pada lampiran

Banyaknya Saluran = 100

Banyaknya *Node* = 101

Banyaknya Iterasi = 4

Total Pembangkitan = $4102.845 + j 3145.325$ kVA

Total Pembebanan = $3970.220 + j 2959.921$ kVA

Rugi Total Saluran = $132.625 + j 185.404$ kVA

Setelah dilakukan analisa dengan menggunakan metode *Evolutionary Programming*, maka diperoleh hasil yaitu pemilihan konduktor serta besar investasinya. Seperti terlihat pada gambar 4-14.

Parameter Evolutionary Programming

Param EP | Hasil EP | Grafik EP | Hasil Akhir | Loadflow | Aliran Daya | Summary Hasil

Hasil Pemilihan Jenis Cable

Bus	Dari	Ke	Type Cable
1		2	3xAAAC 70
2	2	3	3xAAAC 110
3	3	4	3xAAAC 70
4	4	5	3xAAAC 50
5	4	6	3xAAAC 150
6	6	7	3xAAAC 150
7	7	8	3xAAAC 50
8	8	9	3xAAAC 70
9	9	10	3xAAAC 150
10	10	11	3xAAAC 35
11	11	12	3xAAAC 150
12	12	13	3xAAAC 35
13	12	14	3xAAAC 70
14	7	15	3xAAAC 110

sebelum optimasi 1,433,625,390 Rupiah
 sesudah optimasi 1,317,138,898 Rupiah

Hitung EP Close

Gambar 4-14 Hasil Perhitungan *Evolutionary Programming*

Analisa pemilihan konduktor sebelum optimasi biaya investasi konduktor sebesar Rp 1.433.625.930 ,- Dan setelah dilakukan Optimasi biaya investasi Penyulang Pujon sebesar Rp 1.376.826.823,-

Setelah dilakukan pemilihan konduktor diperoleh perbaikan profil tegangan serta rugi – rugi saluran. Dapat dilihat pada table 4-8 sampai 4-10

Tabel 4-8

Kondisi Tegangan Dan Sudut Fasa Tegangan Tiap *Node* Penyulang Pujon Setelah Optimasi

Nomer <i>Node</i>	Tegangan (pu)	Sudut V (deg)
1	1.00000	0.00000
2	1.99750	-0.00076
3	0.99153	-0.00272
4	0.98792	-0.00383
5	0.98791	-0.00383

Tabel 4-8 Selengkapnya dapat dilihat pada lampiran

Tabel 4-9

Besar Rugi-Rugi daya Tiap saluran
Penyulang Pujon Sesudah Optimasi

Saluran	P (kW)	Q (kVAR)
1	7.857	10.951
2	18.164	26.419
3	10.982	15.225
4	0.001	0.001
5	7.698	11.352

Tabel 4-9 Selengkapnya dapat dilihat pada lampiran

Tabel 4-10

Besar Aliran Daya Tiap Saluran
Penyulang Pujon Sesudah Optimasi

Saluran	P (kW)	Q (kVAR)
1	4091.69	3133.66
2	4045.13	3099.75
3	3927.94	3014.55
4	115.585	68.572
5	3801.38	2930.67

Tabel 4-10 Selengkapnya dapat dilihat pada lampiran

Banyaknya saluran = 100

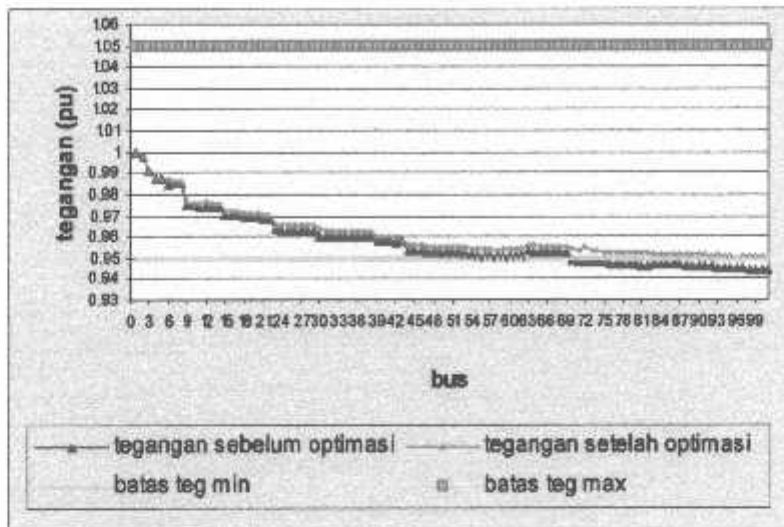
Banyaknya *Node* = 101

Banyaknya Iterasi = 2

Total Pembangkitan = $4091.691 + j 3133.657$ kVA

Total Pembebanan = $3970.220 + j 2959.921$ kVA

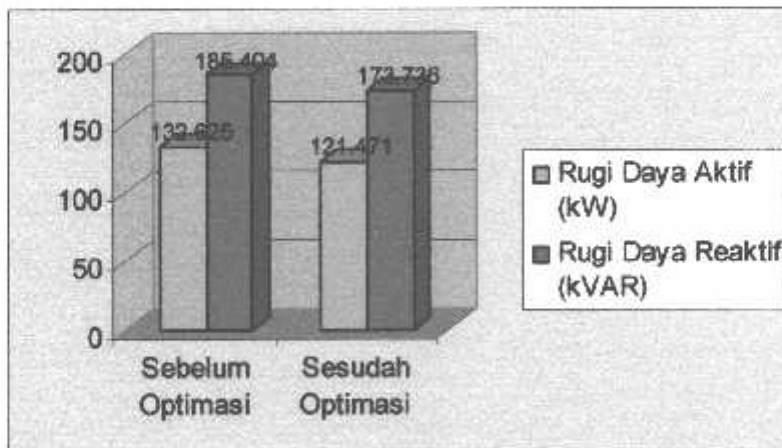
Rugi Total Saluran = $121.471 + j 173.736$ kVA



Grafik 4-1

Tegangan Tiap –Tiap *Node* Terhadap Asumsi Tegangan Awal Sebelum Dan Sesudah Optimasi

Dari grafik diatas dapat dilihat bahwa perubahan tegangan dari asumsi tegangan awal tiap-tiap *node* masih dalam batas-batas yang diijinkan, Dengan tegangan paling rendah terjadi pada *node* ke-101 yaitu 0.94418 pu atau 18883.6 kV, dan setelah di optimasi tegangan tersebut mengalami kenaikan menjadi 0.95025 atau 19.005 kV.



Grafik 4-2

Rugi Daya Sebelum Dan Sesudah Optimasi

Dari grafik diatas dapat dilihat perubahan rugi daya sebelum jaringan dioptimasi untuk daya aktif sebelum optimasi sebesar 132.625 kW setelah dikompensasi rugi daya aktif mengalami penurunan menjadi 121.471 kW, Untuk daya reaktif sebelum optimasi sebesar 185.404 kVAR, setelah optimasi mengalami penurunan menjadi 173.736 kVAR.

BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan analisa perhitungan Pemilihan Konduktor dengan menggunakan metode *Evolutionary Programming* (EP) maka dapat diambil kesimpulan sebagai berikut :

1. Penyelesaian permasalahan pemilihan konduktor dengan menggunakan metode EP pada penyulang pujon untuk Analisa pemilihan konduktor sebelum optimasi biaya investasi konduktor sebesar Rp 1.433.625.930 ,- Dan setelah dilakukan Optimasi biaya investasi Penyulang Pujon sebesar Rp 1.376.826.823,-
2. Rugi-rugi daya pada saluran berhasil dilakukan minimalisasi yaitu sebesar 11.154 kW dan 11.668 kVAR Dimana sebelum optimasi dihasilkan rugi-rugi daya sebesar 132.625 kW dan 185.404 kVAR, sedangkan perhitungan rugi-rugi daya setelah optimasi adalah 121.471 kW dan 173.736 kVAR.
3. Untuk tingkat tegangan didapatkan hasil bahwa dari output program pemilihan konduktor menggunakan metode *evolutionary programming* masih dalam range yang diijinkan yaitu antara -5 % sampai +5 % dari tegangan nominal 20 kV.

5.2. Saran

Penggunaan metode EP untuk menyelesaikan permasalahan pemilihan konduktor pada saluran distribusi masih perlu dikembangkan lagi ,agar dapat mereduksi kerugian daya yang hilang dengan biaya investasi yang minimal.

DAFTAR PUSTAKA

- [1] Hasan Basri " Sistem Distribusi Tenaga Listrik".
 - [2] William D. Stevenson JR, "Analisa Sistem Tenaga", Edisi keempat, Erlangga 1994, Jakarta
 - [3] Tram H.N and WEall D.L, "Optimal Conductor Selection In planning radial distribution systems Planning, IEEE Transaction on Power System", Vol. 3, No. 1, ,200-206. February 1988
 - [4] David Beasley,"*What's Evolutionary Programming (EP)*",June 2004
 - [5] Rakesh Ranjan*Ashvini Chaturvedi*Parmal Singh Solanki # D. Das, "*Optimal Conductor Selection of radial distribution systems Planning Using Evolutionary Programming*" Multi-media University, Malaka, IEEE 2003
 - [6] Talib S Hussian , "*An Introduction to Evolutionary Computation*" IEEE Trans on Power System, Vol 14, No 4, pp 1245-1250, Nov 1999.
 - [7] Data PT.PLN (Persero) Distribusi Jawa Timur Area Malang UP-J Batu
 - [8] SPLN 1981 PT.PLN (Persero)
 - [9] Program Komputer Bahasa Pemrograman *Borland Delphi* Versi 7.0
-

A. Data Saluran Penyulang Pujon.

Nomer Saluran	Dari Node	Ke Node	Panjang Saluran (m)	Impedansi Saluran		Nomer Saluran	Dari Node	Ke Node	Panjang Saluran (m)	Impedansi Saluran	
				R (Ohm)	X (Ohm)					R (Ohm)	X (Ohm)
1	1	2	580.3	0.1255	0.1918	52	51	53	940.35	0.2033	0.3108
2	2	3	1432.6	0.3097	0.4735	53	52	54	1715.3	0.7904	0.6127
3	3	4	863.9	0.1868	0.2855	54	54	55	1192.7	0.5496	0.426
4	4	5	85.4	0.0185	0.0282	55	55	56	1123.5	0.2429	0.3713
5	4	6	683.3	0.1477	0.2258	56	56	57	1353.3	0.2926	0.4473
6	6	7	2082.8	0.4503	0.6884	57	57	58	1067.1	0.2307	0.527
7	7	8	154.4	0.0334	0.051	58	56	59	396.5	0.0857	0.131
8	8	9	414.5	0.0896	0.137	59	59	60	296.8	0.0642	0.0981
9	9	10	896.5	0.1938	0.2963	60	59	61	1093.4	0.2364	0.3614
10	10	11	463.5	0.1002	0.1532	61	61	62	1162.6	0.2514	0.3842
11	11	12	452	0.0977	0.1494	62	44	63	497.1	0.1075	0.1643
12	12	13	1424.5	0.308	0.4708	63	63	64	570.4	0.1233	0.1885
13	12	14	1233	0.2666	0.4075	64	64	65	907	0.1961	0.2998
14	7	15	1244.8	0.2691	0.4114	65	65	66	1769.8	0.3826	0.5849
15	15	16	90.5	0.0196	0.0299	66	63	67	135.2	0.0292	0.0447
16	16	17	20.4	0.0044	0.0067	67	67	68	821.6	0.1776	0.2715
17	15	18	338	0.0731	0.1117	68	67	69	110.15	0.0238	0.0364
18	18	19	182.2	0.049	0.0615	69	69	70	891.3	0.1927	0.2946
19	18	20	402.8	0.1083	0.136	70	70	71	603.2	0.1304	0.1994
20	18	21	304.3	0.0658	0.1006	71	71	72	27.3	0.0059	0.009
21	21	22	293.7	0.079	0.0992	72	72	73	233.1	0.0504	0.077
22	21	23	1257.4	0.2719	0.4156	73	72	74	258.7	0.0559	0.0855
23	23	24	406.5	0.0879	0.1344	74	74	75	198.1	0.0428	0.0655
24	23	25	154.2	0.0995	0.0567	75	75	76	1538.1	0.3325	0.583
25	25	26	627.4	0.5783	0.2378	76	76	77	391.43	0.0895	0.1261
26	26	27	6.4	0.0059	0.0024	77	76	78	472	0.1021	0.156
27	26	28	204.2	0.1882	0.0774	78	78	79	449.6	0.0972	0.1486
28	28	29	1145.9	1.0562	0.4343	79	79	80	721.4	0.156	0.2383
29	23	30	868.2	0.1877	0.2869	80	80	81	940.5	0.2033	0.3108
30	30	31	141.2	0.0305	0.0467	81	81	82	1428.3	0.3088	0.4721
31	31	32	848.8	0.1835	0.2805	82	75	83	322.1	0.0696	0.1065
32	31	33	272.5	0.1758	0.1002	83	83	84	186.4	0.0403	0.0616
33	33	34	1127.1	0.7272	0.4145	84	83	85	98.5	0.0213	0.0326
34	34	35	102.2	0.0659	0.0376	85	85	86	243.9	0.0527	0.0806
35	34	36	496.35	0.1073	0.164	86	85	87	205.3	0.0444	0.0679
36	36	37	282.5	0.0611	0.0934	87	87	88	590.3	0.1276	0.1951
37	37	38	294.5	0.0637	0.0973	88	88	89	856.6	0.1852	0.2831
38	31	39	943.3	0.2039	0.3118	89	89	90	458.3	0.0991	0.1515
39	39	40	302.31	0.1951	0.1112	90	90	91	1162.2	0.2513	0.3841
40	39	41	548.11	0.1185	0.1812	91	89	92	207.4	0.0448	0.0686
41	41	42	105.9	0.0229	0.035	92	92	93	1061.5	0.2295	0.3508

33	T129	Jl.Flamboyan	83.936	49.805	Load	84	T195	KOP SAE	47.3	28.067	Load
34			0	0	Load	85			0	0	Load
35	T99	Jl.Flamboyan	55.9	33.17	Load	86	T62	KOP SAE	11.08	6.332	Load
36	T200	Ds.Tanbuh	1.72	1.021	Load	87	T33	Ds.Ngroto	131.15	77.821	Load
37	T201	Ds.Tanbuh	1.935	1.148	Load	88	T61	Ds.Ngroto	70.52	41.847	Load
38	T202	Ds.Tanbuh	1.29	0.766	Load	89			0	0	Load
39			0	0	Load	90	T64	Dk.Lebaksari	29.67	17.605	Load
40	T168	Bukit Pinus	4.73	2.87	Load	91	T147	Dk.Terong	15.91	9.441	Load
41	T174	Jl.Songgoriti	88.064	52.258	Load	92	T63	Dk.Mantung	33.54	19.902	Load
42	T175	Jl.Songgoriti	17.888	614	Load	93			0	0	Load
43	T16	Jl.Songgoriti	44.032	26.127	Load	94	T196	Dk.Bunder	32.68	19.392	Load
44			0	0	Load	95			0	0	Load
45	T57	Dk.Sebaluh	58.05	34.445	Load	96	T197	Dk.Ngehrong	15.48	9.185	Load
46	T153	Dk.Sebaluh	74.304	44.9	Load	97	T65	Ds.Ngaab	115.58	68.548	Load
47	T59	Dk.Maroon	67.94	40.314	Load	98	T66	Ds.Ngabab	47.3	28.067	Load
48	T87	Dk.Maroon	58.48	34.701	Load	99	T219	Ds.Manting	40.85	24.239	Load
49			0	0	Load	100	T76	Ds.Manting	37.84	22.453	Load
50	T112	Ds.Gunung Sari	10.105	5.996	Load	101	T77	Dk.Gerih	64.5	38.273	Load

C. Data konduktor

Data konduktor Penyulang Pujon
Sebelum Optimasi

Data konduktor Penyulang Pujon
Setelah Optimasi

Dari Node	Ke Node	Tipe Kond AAAC	Dari Node	Ke Node	Tipe Kond AAAC	Dari Node	Ke Node	Tipe Kondr AAAC	Dari Node	Ke Node	Tipe Kond AAAC
1	2	3x150	51	53	3x150	1	2	3x70	51	52	3x110
2	3	3x110	52	54	3x150	2	3	3x110	51	53	3x50
3	4	3x110	54	55	3x70	3	4	3x70	52	54	3x70
4	5	3x110	55	56	3x70	4	5	3x50	54	55	3x70
4	6	3x110	56	57	3x70	4	6	3x150	55	56	3x70
6	7	3x110	57	58	3x150	6	7	3x150	56	57	3x110
7	8	3x150	56	59	3x150	7	8	3x50	57	58	3x35
8	9	3x150	59	60	3x150	8	9	3x70	56	59	3x150
9	10	3x150	59	61	3x150	9	10	3x150	59	60	3x50
10	11	3x150	61	62	3x150	10	11	3x35	59	61	3x110
11	12	3x150	44	63	3x150	11	12	3x150	61	62	3x150
12	13	3x150	63	64	3x150	12	13	3x35	44	63	3x35
12	14	3x150	64	65	3x150	12	14	3x70	63	64	3x70
7	15	3x150	65	66	3x150	7	15	3x110	64	65	3x50
15	16	3x150	63	67	3x150	15	16	3x35	65	66	3x150
16	17	3x150	67	68	3x150	16	17	3x70	63	67	3x110
15	18	3x110	67	69	3x70	15	18	3x150	67	68	3x70
18	19	3x110	69	70	3x70	18	19	3x70	67	69	3x50
18	20	3x150	70	71	3x70	18	20	3x70	69	70	3x50
18	21	3x150	71	72	3x70	18	21	3x70	70	71	3x35

9	0.97502	-0.00752	60	0.95057	-0.01715	9	0.97664	-0.00754	60	0.95411	-0.01538
10	0.97479	-0.00764	61	0.95052	-0.01718	10	0.97644	-0.00764	61	0.95407	-0.01540
11	0.97471	-0.00768	62	0.95048	-0.01720	11	0.97637	-0.00767	62	0.95404	-0.01541
12	0.97467	-0.00770	63	0.9531	-0.01631	12	0.97634	-0.00769	63	0.956	-0.01451
13	0.97464	-0.00771	64	0.95301	-0.01625	13	0.97631	-0.00770	64	0.95592	-0.01455
14	0.97458	-0.00774	65	0.9529	-0.01641	14	0.97626	-0.00773	65	0.95582	-0.01460
15	0.97075	-0.00833	66	0.95288	-0.01642	15	0.97238	-0.00890	66	0.95579	-0.01461
16	0.97674	-0.00833	67	0.95288	-0.01642	16	0.97237	-0.00891	67	0.9558	-0.01461
17	0.97073	-0.00834	68	0.95286	-0.01643	17	0.97237	-0.00891	68	0.95579	-0.01461
18	0.9694	-0.00879	69	0.95271	-0.01651	18	0.97118	-0.00929	69	0.95565	-0.01468
19	0.96939	-0.00880	70	0.94844	-0.01556	19	0.97117	-0.00929	70	0.95443	-0.01525
20	0.96937	-0.00880	71	0.948	-0.01599	20	0.97115	-0.00929	71	0.95368	-0.01560
21	0.96823	-0.00919	72	0.94798	-0.01598	21	0.9701	-0.00959	72	0.95365	-0.01561
22	0.96821	-0.00920	73	0.94797	-0.01599	22	0.97009	-0.00959	73	0.95365	-0.01562
23	0.96346	-0.01081	74	0.94766	-0.01615	23	0.96594	-0.01082	74	0.95366	-0.01575
24	0.9634	-0.01083	75	0.94743	-0.01627	24	0.96599	-0.01084	75	0.95231	-0.01669
25	0.96343	-0.01080	76	0.94685	-0.01658	25	0.96594	-0.01082	76	0.95279	-0.01694
26	0.96307	-0.01075	77	0.94682	-0.01659	26	0.96599	-0.011088	77	0.95276	-0.01695
27	0.96307	-0.01075	78	0.94671	-0.01665	27	0.96586	-0.01088	78	0.95266	-0.01700
28	0.96307	-0.01075	79	0.94663	-0.01669	28	0.96586	-0.01088	79	0.95259	-0.01703
29	0.96269	-0.01069	80	0.94653	-0.01675	29	0.96586	-0.01094	80	0.9525	-0.01707
30	0.96054	-0.01173	81	0.94641	-0.01680	30	0.96574	-0.01157	81	0.9524	-0.01712
31	0.96007	-0.01189	82	0.94631	-0.01685	31	0.96345	-0.012169	82	0.95231	-0.01716
32	0.96005	-0.01190	83	0.94719	-0.01639	32	0.96303	-0.01170	83	0.95209	-0.01679
33	0.95998	-0.01188	84	0.94718	-0.01640	33	0.963	-0.01171	84	0.95209	-0.01680
34	0.95983	-0.01188	85	0.94681	-0.01695	34	0.96299	-0.01174	85	0.95203	-0.01682
35	0.95982	-0.01188	86	0.9468	-0.01695	35	0.96292	-0.01174	86	0.95203	-0.01682
36	0.95981	-0.01188	87	0.94666	-0.01702	36	0.96292	-0.01174	87	0.95191	-0.01688
37	0.95981	-0.01188	88	0.94634	-0.01719	37	0.96292	-0.01174	88	0.95161	-0.01702
38	0.95981	-0.01188	89	0.94593	-0.01740	38	0.96291	-0.01174	89	0.95124	-0.01719
39	0.95815	-0.01373	90	0.94591	-0.01741	39	0.96291	-0.01238	90	0.95122	-0.01720
40	0.95815	-0.01373	91	0.94589	-0.01742	40	0.96037	-0.01238	91	0.9512	-0.01721
41	0.9578	-0.01357	92	0.94585	-0.01744	41	0.96037	-0.01219	92	0.95118	-0.01723
42	0.95774	-0.01353	93	0.94544	-0.01766	42	0.96002	-0.01216	93	0.95116	-0.01741
43	0.95805	-0.01378	94	0.94534	-0.01766	43	0.95997	-0.01242	94	0.95114	-0.01743
44	0.95398	-0.01586	95	0.94516	-0.01765	44	0.96028	-0.01413	95	0.95106	-0.01746
45	0.95343	-0.01614	96	0.94514	-0.01765	45	0.95677	-0.01435	96	0.95068	-0.01747
46	0.95322	-0.01625	97	0.94493	-0.01765	46	0.95629	-0.01444	97	0.95068	-0.01751
47	0.95278	-0.01647	98	0.94442	-0.01764	47	0.9561	-0.01463	98	0.95059	-0.01761
48	0.95251	-0.01661	99	0.9444	-0.01765	48	0.9557	-0.01474	99	0.95036	-0.01762
49	0.95236	-0.01669	100	0.94422	-0.01765	49	0.95549	-0.01481	100	0.95035	-0.01766
50	0.95235	-0.0169	101	0.94418	-0.01765	50	0.95531	-0.01481	101	0.95027	-0.01767
51	0.95233	-0.01670				51	0.9553	-0.01482			

F. Besarnya Rugi-Rugi Daya Tiap Saluran Penyulang Pujon

Rugi-Rugi Daya Sebelum Optimasi

Rugi-Rugi Daya Sesudah Optimasi

Sal	Rugi-Rugi Daya Sebelum Optimasi		Sal	Rugi-Rugi Daya Sesudah Optimasi		Sal	Rugi-Rugi Daya Sesudah Optimasi		Sal	Rugi-Rugi Daya Sesudah Optimasi	
	P(kW)	Q(kVAR)		P(kW)	Q(kVAR)		P(kW)	Q(kVAR)		P(kW)	Q(kVAR)
1	8.385	10.142	51	0	0	1	7.857	10.951	51	0	0
2	20.336	24.525	52	0.09	0.136	2	18.164	26.419	52	0.083	0.116
3	11.72	16.658	53	0.175	0.136	3	10.982	15.225	53	0.079	0.11
4	0.001	0.001	54	0.039	0.057	4	0.001	0.001	54	0.094	0.073
5	8.774	12.225	55	0.019	19.973	5	7.598	11.352	55	0.02	0.031
6	26.459	40.45	56	0.006	0.007	6	24.79	34.554	56	0.006	0.008
7	0.016	0.02	57	0.001	0.002	7	0.018	0.022	57	0.001	0.02
8	0.03	0.046	58	0.002	0.003	8	0.027	0.04	58	0.002	0.003
9	0.041	0.062	59	0	0	9	0.036	0.053	59	0	0
10	0.009	0.014	60	0.002	0.003	10	0.008	0.012	60	0.002	0.003
11	0.002	0.004	61	0.01	0.001	11	0.002	0.003	61	0.001	0.001
12	0	0	62	0.946	1.375	12	0	0	62	1.062	1.623
13	0.004	0.006	63	0.009	0.013	13	0.004	0.005	63	0.009	0.014
14	13.111	15.172	64	0.009	0.013	14	11.498	16.955	64	0.01	0.015
15	0.001	19.104	65	0	0	15	0.001	0.02	65	0	0
16	0	0	66	0.235	0.313	16	0	0	66	0.24	0.367
17	3.316	5.068	67	0.32	0	17	3.105	4.328	67	0	0
18	0.001	0.001	68	0.169	0.246	18	0	0.001	68	0.19	0.29
19	0.002	0.002	69	0.254	1.771	19	0.001	0.02	69	6.191	1.968
20	2.779	4.248	70	0.744	1.037	20	2.653	3.717	70	0.796	1.218
21	0.001	0.001	71	0.031	0.044	21	0.001	0.001	71	0.031	0.005
22	11.101	16.977	72	0	0	22	10.175	14.347	72	0	0.001
23	0.005	0.008	73	0.259	0.361	23	0.005	0.007	73	0.277	0.423
24	0.002	0.01	74	1.667	2.324	24	0	0	74	0.177	0.271
25	0.188	0.036	75	1.0.139	0.202	25	0.02	0.028	75	0.152	0.238
26	0	0	76	0.002	0.002	26	0	0	76	0.002	0.02
27	0	0	77	0.028	0.04	27	0	0	77	0.031	0.047
28	0.049	0.02	78	0.009	0.013	28	0.011	0.016	78	0.01	0.016
29	5.926	9.058	79	0.009	0.013	29	5.285	7.688	79	0.009	0.014
30	0.95	1.455	80	0.008	0.012	30	0.908	1.272	80	0.01	0.014
31	0	0	81	0.005	0.004	31	0	0.001	81	0.005	0.007
32	0.013	0.008	82	0.112	0.164	32	0.005	0.06	82	0.127	0.194
33	0.01	0.006	83	0	0	33	0.003	0.005	83	0	0
34	0	0	84	0.029	0.044	34	0	0	84	0.05	0.515
35	0	0	85	0.072	0	35	0	0	85	0	0
36	0	0	86	0.063	39.911	36	0	0	86	0.068	0.103
37	0	0	87	0.117	0.163	37	0	0	87	0.122	0.188
38	1.08	8.633	88	0.123	0.171	38	5.285	7.366	88	0.132	0.201
39	0	0	89	0.001	0.001	39	0	0	89	0.001	0.001
40	0.147	0.225	90	0	0	40	0.166	0.227	90	0	0
41	0.023	0.036	91	0.024	0.035	41	0.021	0.031	91	0.025	0.039

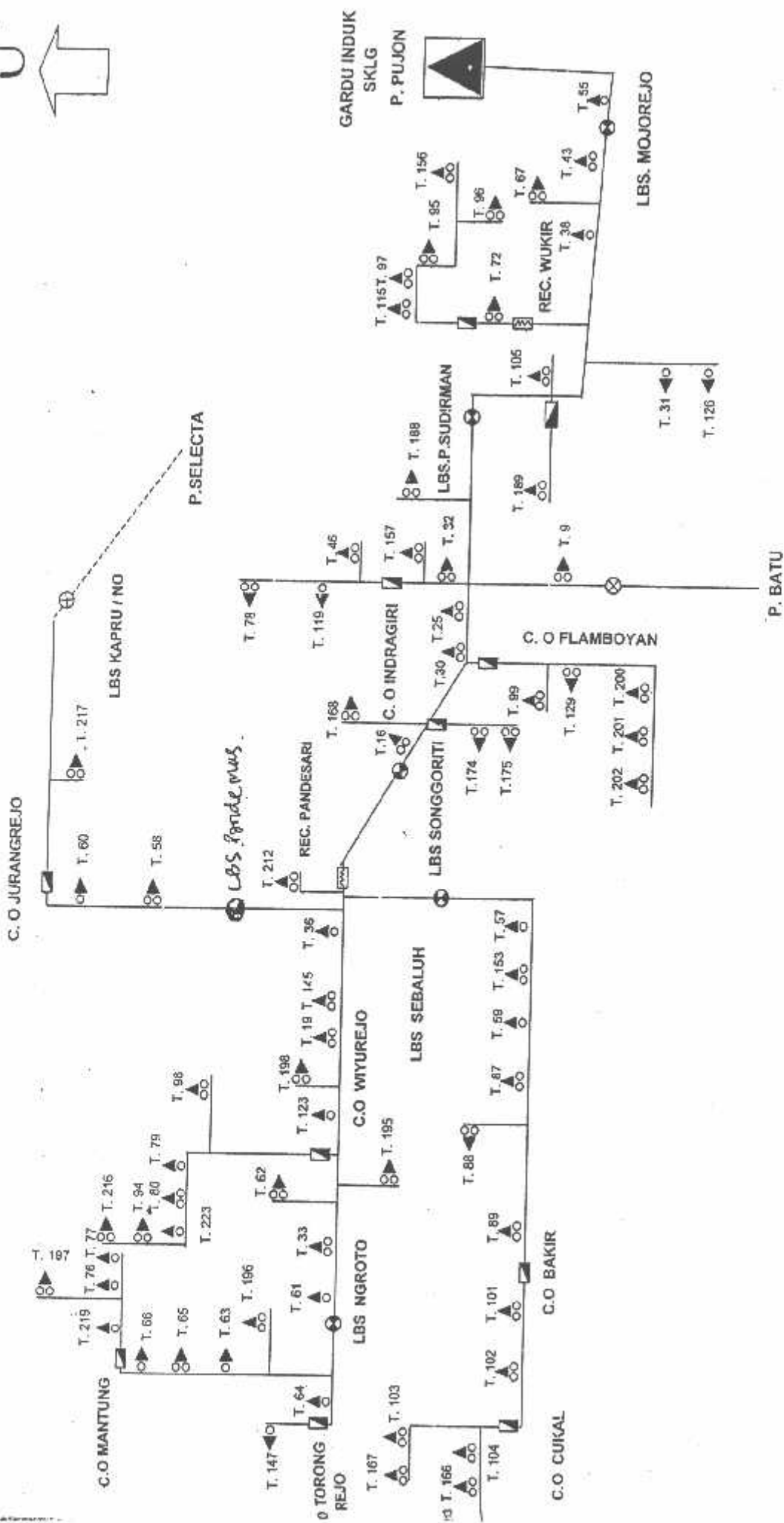
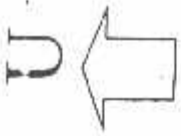
42	0.176	0.268	92	0.097	0.141	42	0.156	0.227	92	0.107	0.167
43	6.869	10.301	93	0.001	0.001	43	6.015	8.87	93	0.003	0.002
44	0.261	0.399	94	0.029	0.042	44	0.233	0.339	94	0.097	0.055
45	0.077	0.141	95	0	0	45	0.087	0.121	95	0	0
46	0.166	0.254	96	0.056	0.033	46	0.155	0.217	96	0.076	0.044
47	0.09	0.137	97	0.034	0.048	47	0.086	0.21	97	0.103	0.58
48	0.043	0.066	98	129	0.002	48	0.042	0.059	98	0.002	0.003
49	0	0	99	0.006	0.009	49	0	0	99	0.02	0.011
50	0.006	0.01	100	0.001	0.002	50	0.006	0.007	100	0.003	0.002

G. Besarnya Aliran Daya Tiap Saluran Penyulang Pujon

Besarnya Aliran Daya Sebelum Optimasi

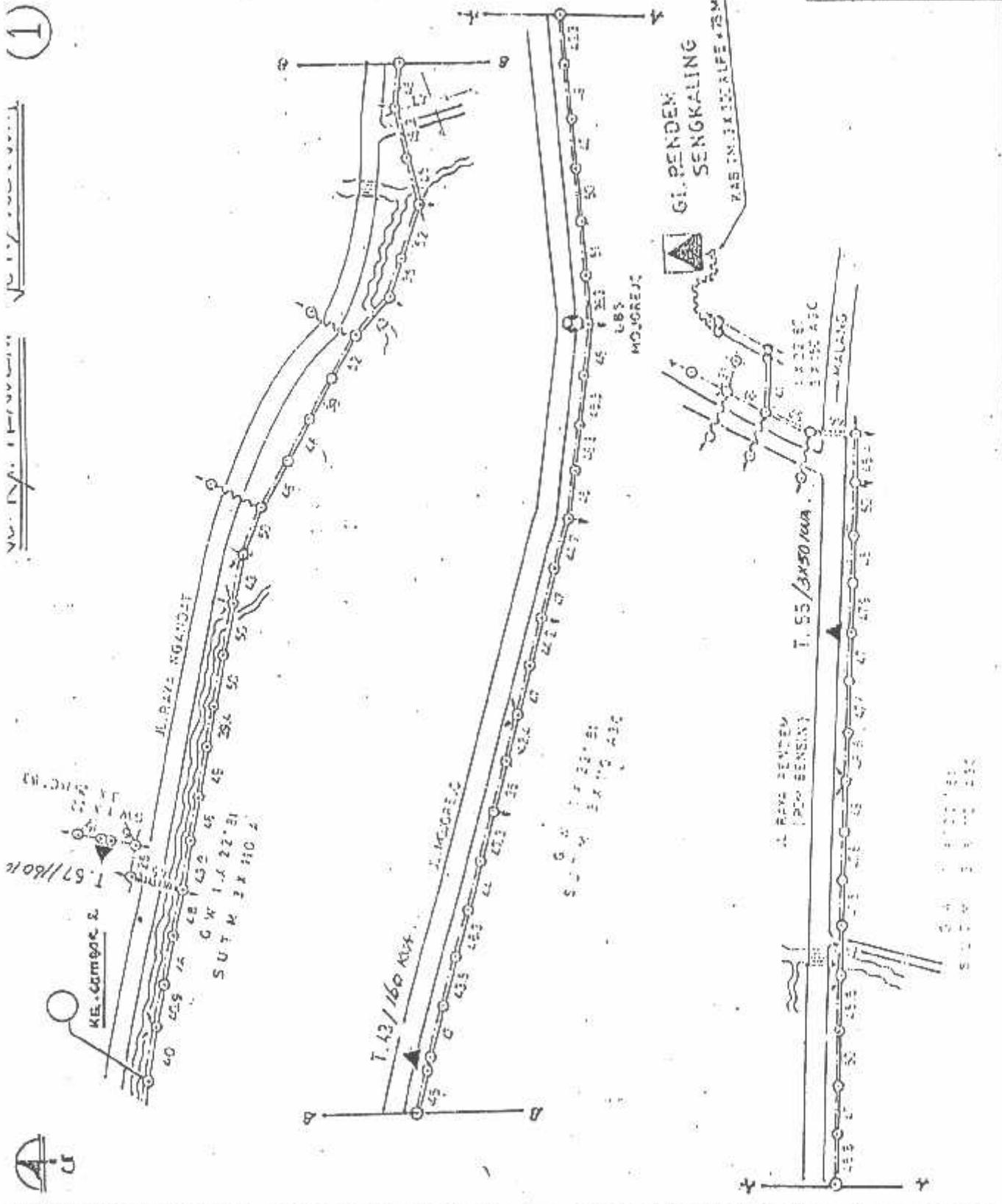
Besarnya Aliran Daya Sebelum Optimasi

Sal	P(kW)	Q(kVAR)	Sal	P(kW)	Q(kVAR)	Sal	P(kW)	Q(kVAR)	Sal	P(kW)	Q(kVAR)
1	4102.845	3145.325	51	30.1	17.861	1	4091.69	3133.657	51	30.1	47.861
2	4055.76	3112.222	52	343.101	203.749	2	4045.13	3099.746	52	342.94	203.68
3	3936.397	3028.922	53	244.111	144.928	3	3927.94	3014.552	53	243.96	144.879
4	115.585	68.572	54	213.836	126.931	4	115.585	68.572	54	213.78	126.908
5	3809.091	2943.691	55	150.102	89.096	5	3801.38	2930.474	55	150.1	69.084
6	3777.741	2918.071	56	70.527	41.855	6	3771.1	2905.29	56	70.527	41.854
7	366.377	217.449	57	36.981	21.945	7	366.368	217.433	57	36.981	21.945
8	308.569	183.143	58	79.555	47.21	8	308.561	183.125	58	79.555	47.209
9	242.749	144.067	59	32.25	19.136	9	242.742	144.054	59	32.25	19.136
10	160.148	95.026	60	47.303	28.07	10	160.146	95.021	60	47.303	28.07
11	85.835	50.93	61	29.241	17.351	11	85.834	50.528	61	29.241	17.351
12	19.78	11.736	62	1630.21	969.386	12	19.78	11.735	62	1626.2	970.004
13	66.0652	39.186	63	143.639	85.249	13	66.052	39.189	63	143.64	85.246
14	3384.903	2660.171	64	115.25	68.395	14	3379.94	2653.943	64	115.25	68.393
15	130.239	77.267	65	12.04	7.144	15	130.239	77.267	65	12.04	7.144
16	98.04	58.163	66	1485.51	882.513	16	98.04	58.163	66	1481.7	883.383
17	3241.552	2567.731	67	22.36	13.268	17	3238.21	2557.721	67	22.36	13.268
18	60.05	35.715	68	1462.91	868.877	18	60.2	35.715	68	1459.1	869.802
19	66.05	39.186	69	13338.7	795.106	19	66.049	39.186	69	1335.1	796.073
20	3111.986	2487.762	70	1274.2	758.436	20	3108.85	2480.493	70	1275.3	759.601
21	57.793	34.287	71	1192.22	709.048	21	57.793	34.287	71	1193.4	710.392
22	3051.413	2449.226	72	45.58	27.047	22	3048.85	2442.489	72	45.58	27.047
23	130.725	77.559	73	1146.61	681.995	23	130.725	77.558	73	1147.8	683.302
24	72.93	43.266	74	1047.01	622.632	24	72.928	43.265	74	1048.2	624.001
25	204.901	121.535	75	352.561	209.407	25	204.795	121.522	75	352.54	209.257
26	88.064	52.245	76	69.662	41.336	26	88.064	52.245	76	69.662	41.336
27	3.87	2.296	77	282.747	167.832	27	3.87	2.296	77	282.74	167.82
28	112.879	66.958	78	164.38	97.569	28	112.841	66.954	78	164.38	97.63
29	2631.755	2189.888	79	126.96	75.355	29	2630.23	2185.797	79	126.96	75.352
30	2606.479	2169.35	80	107.601	63.859	30	2605.6	2166.629	80	107.6	63.357
31	24.768	14.695	81	64.935	38.535	31	24.768	14.695	81	64.935	38.532



1

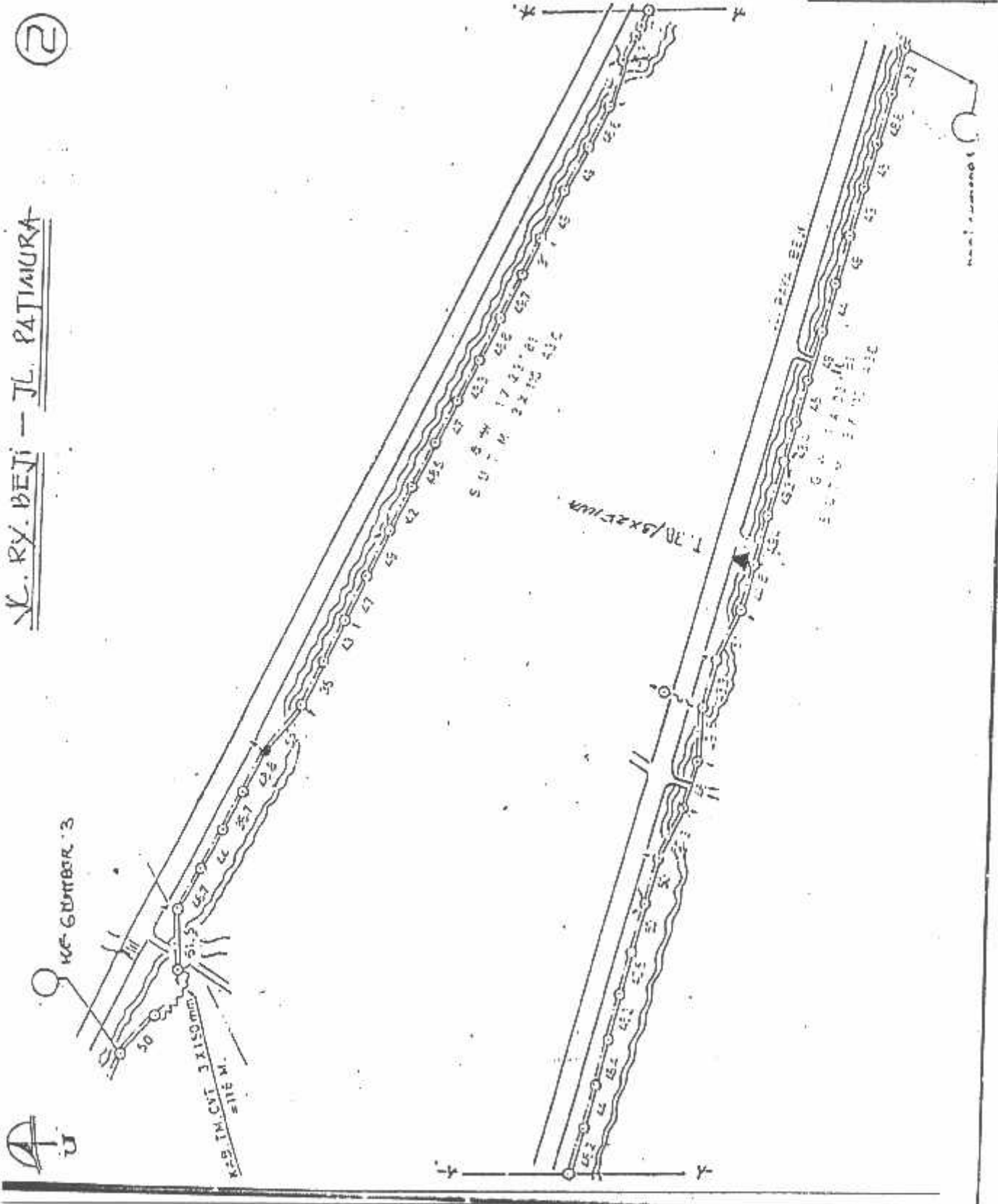
NO. T.P. 1111111111



No	LOKASI PENYULANG	LOKASI	POSISI
1	Pemang NUTM 3x/10 ABC	Kan	
2	TM1	Bh	
3	TM2	Bh	
4	TM3	Bh	
5	TM4	Bh	
6	TM5	Bh	
7	TM6	Bh	
8	TM7	Bh	
9	TM8	Bh	
10	TM9	Bh	
11	TM10	Bh	
12	TM11	Bh	
13	TM12	Bh	
14	TM13	Bh	
15	TM14	Bh	
16	TM15	Bh	
17	TM16	Bh	
18	TM17	Bh	
19	TM18	Bh	
20	TM19	Bh	
21	TM20	Bh	
22	TM21	Bh	
23	TM22	Bh	
24	TM23	Bh	
25	TM24	Bh	
26	TM25	Bh	
27	TM26	Bh	
28	TM27	Bh	
29	TM28	Bh	
30	TM29	Bh	
31	TM30	Bh	
32	TM31	Bh	
33	TM32	Bh	
34	TM33	Bh	
35	TM34	Bh	
36	TM35	Bh	
37	TM36	Bh	
38	TM37	Bh	
39	TM38	Bh	
40	TM39	Bh	
41	TM40	Bh	
42	TM41	Bh	
43	TM42	Bh	
44	TM43	Bh	
45	TM44	Bh	
46	TM45	Bh	
47	TM46	Bh	
48	TM47	Bh	
49	TM48	Bh	
50	TM49	Bh	
51	TM50	Bh	
52	TM51	Bh	
53	TM52	Bh	
54	TM53	Bh	
55	TM54	Bh	
56	TM55	Bh	
57	TM56	Bh	
58	TM57	Bh	
59	TM58	Bh	
60	TM59	Bh	
61	TM60	Bh	
62	TM61	Bh	
63	TM62	Bh	
64	TM63	Bh	
65	TM64	Bh	
66	TM65	Bh	
67	TM66	Bh	
68	TM67	Bh	
69	TM68	Bh	
70	TM69	Bh	
71	TM70	Bh	
72	TM71	Bh	
73	TM72	Bh	
74	TM73	Bh	
75	TM74	Bh	
76	TM75	Bh	
77	TM76	Bh	
78	TM77	Bh	
79	TM78	Bh	
80	TM79	Bh	
81	TM80	Bh	
82	TM81	Bh	
83	TM82	Bh	
84	TM83	Bh	
85	TM84	Bh	
86	TM85	Bh	
87	TM86	Bh	
88	TM87	Bh	
89	TM88	Bh	
90	TM89	Bh	
91	TM90	Bh	
92	TM91	Bh	
93	TM92	Bh	
94	TM93	Bh	
95	TM94	Bh	
96	TM95	Bh	
97	TM96	Bh	
98	TM97	Bh	
99	TM98	Bh	
100	TM99	Bh	
101	TM100	Bh	
102	TM101	Bh	
103	TM102	Bh	
104	TM103	Bh	
105	TM104	Bh	
106	TM105	Bh	
107	TM106	Bh	
108	TM107	Bh	
109	TM108	Bh	
110	TM109	Bh	
111	TM110	Bh	
112	TM111	Bh	
113	TM112	Bh	
114	TM113	Bh	
115	TM114	Bh	
116	TM115	Bh	
117	TM116	Bh	
118	TM117	Bh	
119	TM118	Bh	
120	TM119	Bh	
121	TM120	Bh	
122	TM121	Bh	
123	TM122	Bh	
124	TM123	Bh	
125	TM124	Bh	
126	TM125	Bh	
127	TM126	Bh	
128	TM127	Bh	
129	TM128	Bh	
130	TM129	Bh	
131	TM130	Bh	
132	TM131	Bh	
133	TM132	Bh	
134	TM133	Bh	
135	TM134	Bh	
136	TM135	Bh	
137	TM136	Bh	
138	TM137	Bh	
139	TM138	Bh	
140	TM139	Bh	
141	TM140	Bh	
142	TM141	Bh	
143	TM142	Bh	
144	TM143	Bh	
145	TM144	Bh	
146	TM145	Bh	
147	TM146	Bh	
148	TM147	Bh	
149	TM148	Bh	
150	TM149	Bh	
151	TM150	Bh	
152	TM151	Bh	
153	TM152	Bh	
154	TM153	Bh	
155	TM154	Bh	
156	TM155	Bh	
157	TM156	Bh	
158	TM157	Bh	
159	TM158	Bh	
160	TM159	Bh	
161	TM160	Bh	
162	TM161	Bh	
163	TM162	Bh	
164	TM163	Bh	
165	TM164	Bh	
166	TM165	Bh	
167	TM166	Bh	
168	TM167	Bh	
169	TM168	Bh	
170	TM169	Bh	
171	TM170	Bh	
172	TM171	Bh	
173	TM172	Bh	
174	TM173	Bh	
175	TM174	Bh	
176	TM175	Bh	
177	TM176	Bh	
178	TM177	Bh	
179	TM178	Bh	
180	TM179	Bh	
181	TM180	Bh	
182	TM181	Bh	
183	TM182	Bh	
184	TM183	Bh	
185	TM184	Bh	
186	TM185	Bh	
187	TM186	Bh	
188	TM187	Bh	
189	TM188	Bh	
190	TM189	Bh	
191	TM190	Bh	
192	TM191	Bh	
193	TM192	Bh	
194	TM193	Bh	
195	TM194	Bh	
196	TM195	Bh	
197	TM196	Bh	
198	TM197	Bh	
199	TM198	Bh	
200	TM199	Bh	
201	TM200	Bh	
202	TM201	Bh	
203	TM202	Bh	
204	TM203	Bh	
205	TM204	Bh	
206	TM205	Bh	
207	TM206	Bh	
208	TM207	Bh	
209	TM208	Bh	
210	TM209	Bh	
211	TM210	Bh	
212	TM211	Bh	
213	TM212	Bh	
214	TM213	Bh	
215	TM214	Bh	
216	TM215	Bh	
217	TM216	Bh	
218	TM217	Bh	
219	TM218	Bh	
220	TM219	Bh	
221	TM220	Bh	
222	TM221	Bh	
223	TM222	Bh	
224	TM223	Bh	
225	TM224	Bh	
226	TM225	Bh	
227	TM226	Bh	
228	TM227	Bh	
229	TM228	Bh	
230	TM229	Bh	
231	TM230	Bh	
232	TM231	Bh	
233	TM232	Bh	
234	TM233	Bh	
235	TM234	Bh	
236	TM235	Bh	
237	TM236	Bh	
238	TM237	Bh	
239	TM238	Bh	
240	TM239	Bh	
241	TM240	Bh	
242	TM241	Bh	
243	TM242	Bh	
244	TM243	Bh	
245	TM244	Bh	
246	TM245	Bh	
247	TM246	Bh	
248	TM247	Bh	
249	TM248	Bh	
250	TM249	Bh	
251	TM250	Bh	
252	TM251	Bh	
253	TM252	Bh	
254	TM253	Bh	
255	TM254	Bh	
256	TM255	Bh	
257	TM256	Bh	
258	TM257	Bh	
259	TM258	Bh	
260	TM259	Bh	
261	TM260	Bh	
262	TM261	Bh	
263	TM262	Bh	
264	TM263	Bh	
265	TM264	Bh	
266	TM265	Bh	
267	TM266	Bh	
268	TM267	Bh	
269	TM268	Bh	
270	TM269	Bh	
271	TM270	Bh	
272	TM271	Bh	
273	TM272	Bh	
274	TM273	Bh	
275	TM274	Bh	
276	TM275	Bh	
277	TM276	Bh	
278	TM277	Bh	
279	TM278	Bh	
280	TM279	Bh	
281	TM280	Bh	
282	TM281	Bh	
283	TM282	Bh	
284	TM283	Bh	
285	TM284	Bh	
286	TM285	Bh	
287	TM286	Bh	
288	TM287	Bh	
289	TM288	Bh	
290	TM289	Bh	
291	TM290	Bh	
292	TM291	Bh	
293	TM292	Bh	
294	TM293	Bh	
295	TM294	Bh	
296	TM295	Bh	
297	TM296	Bh	
298	TM297	Bh	
299	TM298	Bh	
300	TM299	Bh	
301	TM300	Bh	
302	TM301	Bh	
303	TM302	Bh	
304	TM303	Bh	
305	TM304	Bh	
306	TM305	Bh	
307	TM306	Bh	
308	TM307	Bh	
309	TM308	Bh	
310	TM309	Bh	
311	TM310	Bh	
312	TM311	Bh	
313	TM312	Bh	
314	TM313	Bh	
315	TM314	Bh	
316	TM315	Bh	
317	TM316	Bh	
318	TM317	Bh	
319	TM318	Bh	
320	TM319	Bh	
321	TM320	Bh	
322	TM321	Bh	
323	TM322	Bh	
324	TM323	Bh	
325	TM324	Bh	
326	TM325	Bh	
327	TM326	Bh	
328	TM327	Bh	
329	TM328	Bh	
330	TM329	Bh	
331	TM330	Bh	
332	TM331	Bh	
333	TM332	Bh	
334	TM333	Bh	
335	TM334	Bh	
336	TM335	Bh	
337	TM336	Bh	
338	TM337	Bh	
339	TM338	Bh	
340	TM339	Bh	
341	TM340	Bh	
342	TM341	Bh	
343	TM342	Bh	
344	TM343	Bh	
345	TM344	Bh	
346	TM345	Bh	
347	TM346	Bh	
348	TM347	Bh	
349	TM348	Bh	
350	TM349	Bh	
351	TM350	Bh	
352	TM351	Bh	
353	TM352	Bh	
354	TM353	Bh	
355	TM354	Bh	
356	TM355	Bh	
357	TM356	Bh	
358	TM357	Bh	
359	TM358	Bh	
360	TM359	Bh	
361	TM360	Bh	
362	TM361	Bh	
363	TM362	Bh	
364	TM363	Bh	
365	TM364	Bh	
366	TM365	Bh	
367	TM366	Bh	
368	TM367	Bh	
369	TM368	Bh	
370	TM369	Bh	
371	TM370	Bh	
372	TM371	Bh	
373	TM372	Bh	
374	TM373	Bh	
375	TM374	Bh	
376	TM375	Bh	
377	TM376	Bh	
378	TM377	Bh	
379	TM378	Bh	
380			

K. RY. BEJI - JL. PATIMURA

(2)



HASIL INSPEKSI JARINGAN		
No	LOKASI PERUSAHAAN	REVISI/REKAM
1	Panjang UTM 3300 ABC	Km 2/20
2	TM 1	Bh
3	TM 2	Bh
4	TM 4	Bh
5	TM 3	Bh
6	TM 8	Bh
7	TM 10	Bh
8	TM 11	Bh
9	TM 12	Bh
10	TM 14	Bh
11	TM 15	Bh
12	TM TYPE POLE	Bh
13	CONTRA MAS	Bh
14	SCBOOR	Bh
15	BRICK SCHOOL	Bh
16	Jumlah Tiang beton TM	Bh
17	Jumlah Tiang besi TM	Bh
18	Jumlah Gerdak	Bh
19	Jumlah kva. Trafo	Km
20	Jumlah Kapasitor	Bh
21	Jumlah PCU	Bh
22	Jumlah CO	Bh
23	Jumlah AVS	Bh
24	Jumlah LRS	Bh
No.	KADANG PERALAN	EDUK. KADANG PERALAN
1	Rampung - rampas pakek	A Kms 5
2	Pragnanisa Tiang TM	B Bh 5/1
3	Pragnanisa Tiang	C Bh
4	Pragnanisa Conaktor	D Kms
5	Pragnanisa Conaktor	E Kms
6	Tiang sabin/tropos	F Bh 2
7	Pragnanisa Pasak	G Bh
8	Pondasi Tiang/GTT	H Bh
9	Pragnanisa Gerdak Wkr	I Kms
10	Pragnanisa COLEKAVS	J Bh
11	Pragnanisa Pragnanisa I.B.S.I	K Bh
12	Pragnanisa Lintang Conaktor	L Bh
13	Pragnanisa Isoklar	M Bh
14	Pragnanisa Arator	N Bh
15	Pragnanisa COLEKAVS	O Bh
16	Pragnanisa Con. Nkr	P Bh

PT. PLN (PERSERO)
 DISTRIBUSI JATIM - AP&J MALANG
 UNIT PELAYANAN & JARINGAN DATU

PENYULANG : PUJON

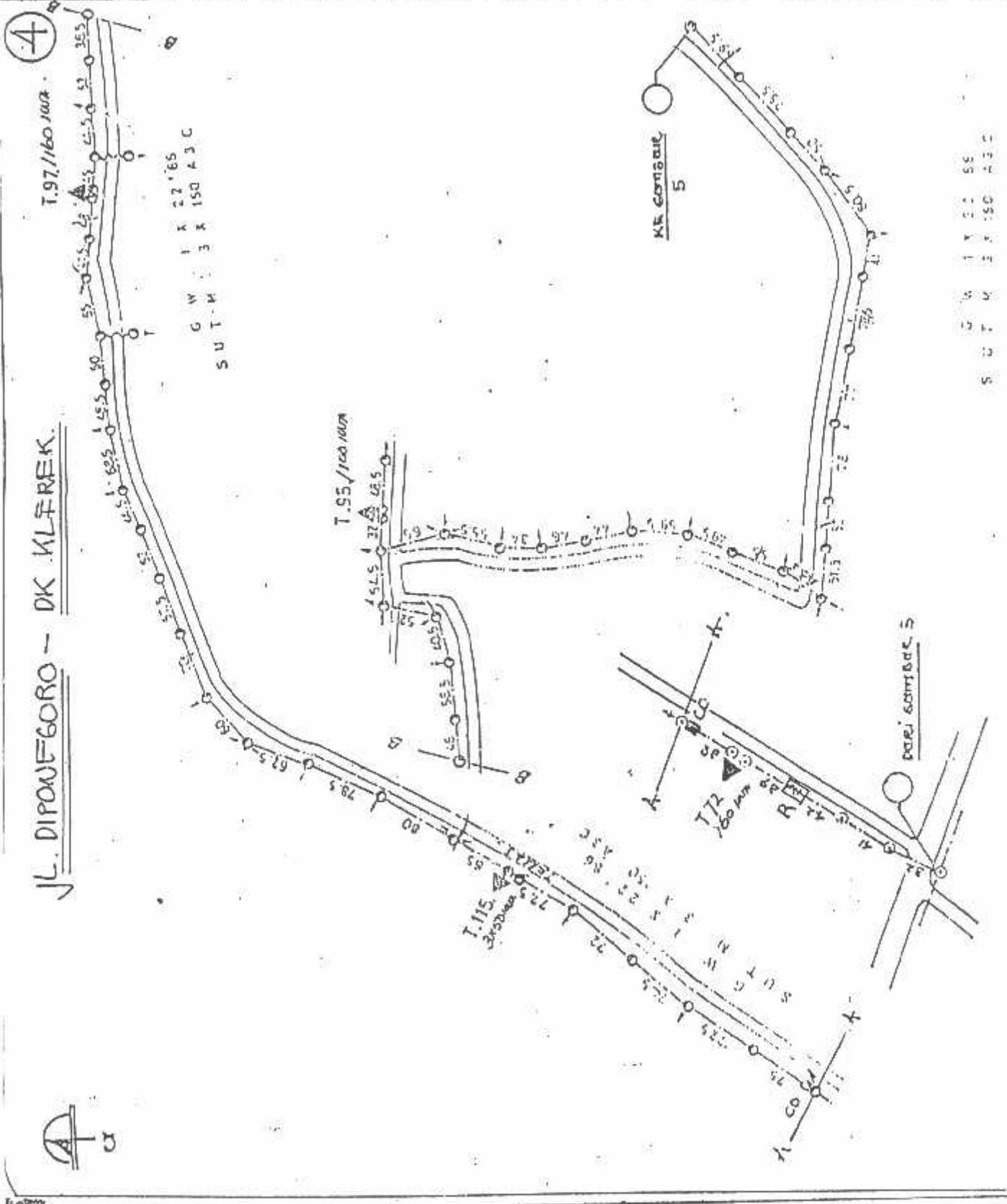
pelaksana	pengawas	diketahui
WIDODO	SUJOKO	MANAGER

HASIL INSPEKSI JARINGAN			
No	KADANG MERULANG	REKAM	EMILAY
1	Projeksi STTM 3X/50 A3C	Kad	3079,5
2	TM1	B4	20
3	TM2	B4	18
4	TM3	B4	1
5	TM4	B4	1
6	TM5	B4	1
7	TM10	B4	3
8	TM11	B4	
9	TM12	B4	
10	TM14	B4	
11	TM15	B4	
12	TM TYPE POLE	B4	
13	CONTRA MAS	B4	
14	ACBOOR	B4	20
15	DRUCK SCBOOR	B4	6
16	Jumlah Tiang beton TM	B4	52
17	Jumlah Tiang besi TM	B4	6
18	Jumlah Gardu	B4	9
19	Jumlah Tra Trofo	Kw	570
20	Jumlah Koperator	B4	
21	Jumlah PCT	B4	
22	Jumlah CO	B4	15
23	Jumlah AVS / DOC	B4	1
24	Jumlah LBS	B4	
No	MASUK PERAWAN	KODE	REKAM/REKAM
1	Resep - resep pokok	A	Kms
2	Prosesasi Tiang TM	B	Bh
3	Prosesasi Tiang	C	Bh
4	Prosesasi Conduktor	D	Kms
5	Prosesasi Conduktor	E	Kms
6	Tata alir/trafo	F	Bh
7	Pemeliharaan Piel	G	Bh
8	Pemeliharaan THT/GTT	H	Bh
9	Pemeliharaan Ground Wire	I	Kms
10	Pemeliharaan COLB/ASNS	J	Bh
11	Pemeliharaan Pemakaian LBS	K	Bh
12	Pemeliharaan Spring Conector	L	Bh
13	Pemeliharaan Isolat	M	Bh
14	Pemeliharaan Arteri	N	Bh
15	Pemeliharaan COLB/ASNS	O	Bh
16	Pemeliharaan Cross Arm	P	Bh

PT. PLN (PERSERO)
DISTRIBUSI JATIM - APAL MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG : PUJON

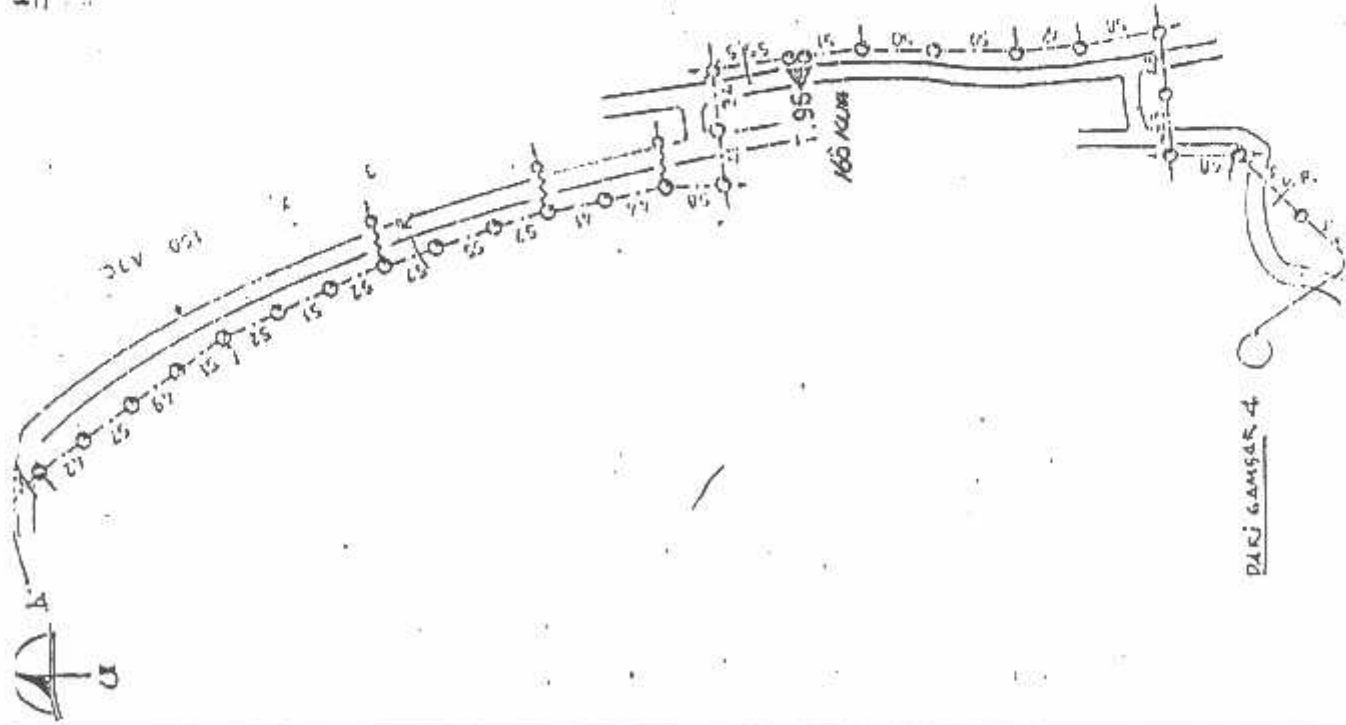
pelaksana pengawas diketahui
WIDDODO SUJOKO MANAGER



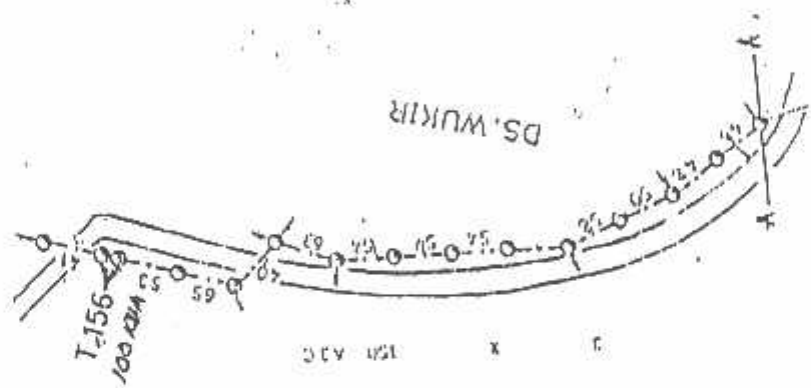
S U T M 3 X 1 5 0 A 3 C

PA. ULLEKEN - DK. GUNGUNG WUKIR

(5)



DIK. GAMSAR. 4



DS. WUKIR

1 x 22 86
1 x 150 230
1 x 150 230

No	KADANGAN PERTUKARAN	SATUAN	JMLAH
1	Panjang SUTK 3x150/45C	Km	2056,5
2	TM 1	Bh	20
3	TM 2	Bh	13
4	TM 4	Bh	/
5	TM 5	BA	/
6	TM 8	BA	/
7	TM 10	BA	/
8	TKG 1	RE	/
9	TM 12	BA	/
10	TM 14	BA	/
11	TM 15	BA	/
12	TM TYPE POLE	BA	/
13	CONTRA MAS	BA	3
14	SEBOR	BA	20
15	DRUCK SCBOR	Bh	6
16	Jumlah Tiang besi TM	BA	53
17	Jumlah Tiang besi TM	BA	2
18	Jumlah Gardu	Bh	200
19	Jumlah Era Temb	BA	/
20	Jumlah Kapasitor	BA	/
21	Jumlah PCT	BA	6
22	Jumlah CO	Bh	/
23	Jumlah AVS	Bh	/
24	Jumlah LBS	BA	/

No	MACAM PERALATAN	KODE	JMLAH	KETERANGAN
1	Rampung - rampas pelbon	A	Kms	
2	Pengaturan Tiang TM	B	Bh	
3	Pengaturan Tiang	C	Bh	
4	Pengaturan Conductor	D	Kms	
5	Pengaturan Conductor	E	Kms	
6	Tiang saringan/bropas	F	Bh	
7	Pemeliharaan Pond	G	Bh	
8	Pondasi Tiang/GTT	H	Bh	
9	Pembongkaran Ground Wire	I	Kms	
10	Pemasangan COLEKSAVVA	J	Bh	
11	Pemasangan pemastian LBS	K	Bh	
12	Pemasangan Lintap Conductor	L	Bh	
13	Pemasangan Isolator	M	Bh	
14	Pengaturan Arator	N	Bh	
15	Pengaturan COLEKSAVVA	O	Bh	
16	Pengaturan Cross Arm	P	Bh	
17				

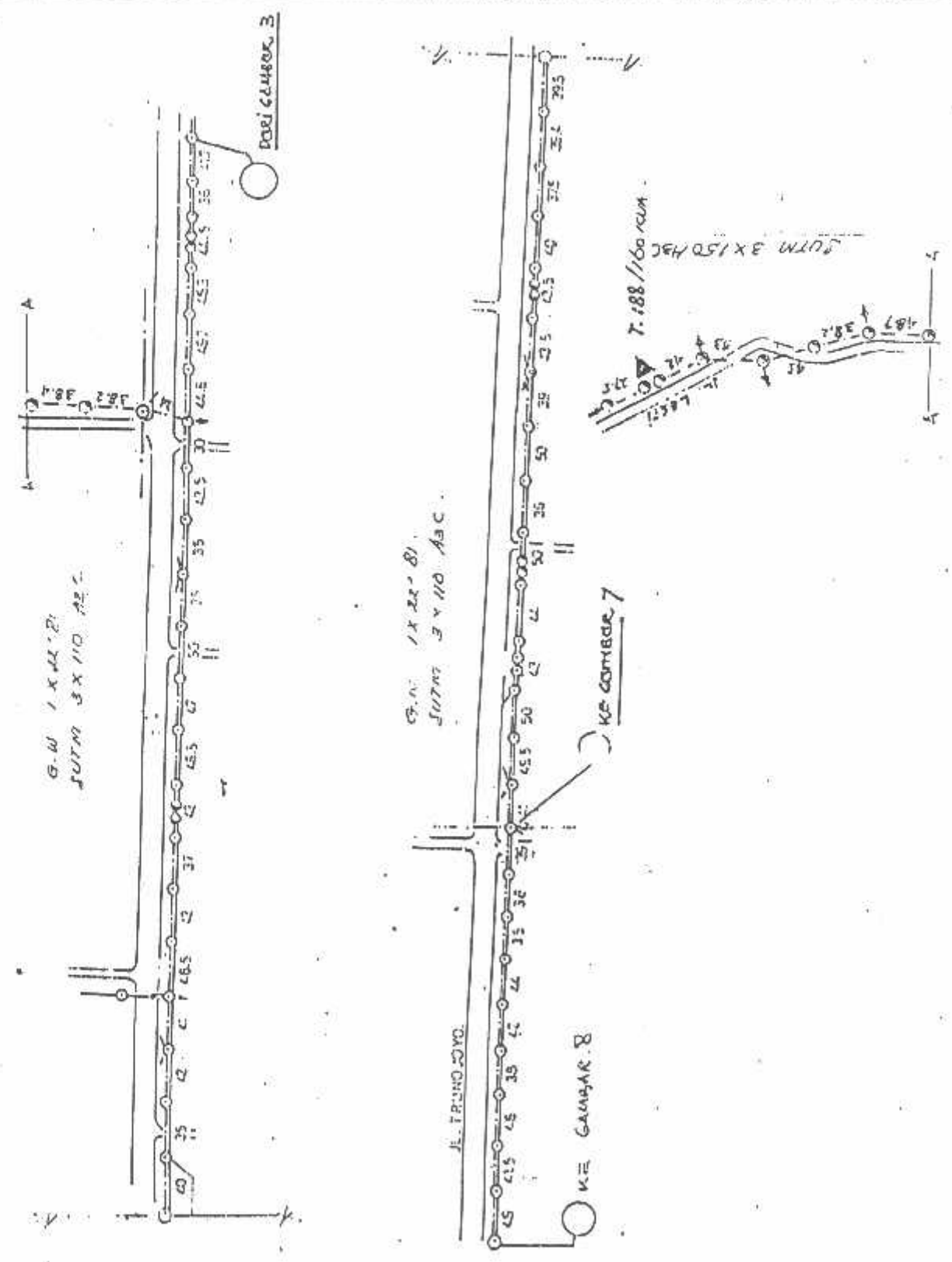
PT. PLN (PERSERO)
DISTRIBUSI JATIM - AP&J MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG: PUJON

pelaksana	pengawas	diketahui
WIDODO	SUJOKO	MANAGER

PT. SUKSES JAYA - J. KUMBUKA

(5)



No.	NAMA PERALAN	JANTAN	JANGAN
1	Tegang SUTM 3x110 A.S.C.	Kms	189,17
2	TM1	Bh	
3	TM2	Bh	3
4	TM3	Bh	
5	TM4	Bh	
6	TM5	Bh	
7	TM6	Bh	
8	TM7	Bh	
9	TM8	Bh	
10	TM9	Bh	
11	TM10	Bh	356
12	TM TYPE POLE	Bh	
13	CONTRA MAS	Bh	
14	SCROOR	Bh	
15	DRUCE SCROOR	Bh	
16	Jumlah Tiang beton TM	Bh	8
17	Jumlah Tiang besi TM	Bh	
18	Jumlah Gardu	Bh	
19	Jumlah Pan Tinfo	Kms	160
20	Jumlah Kapasitor	Bh	
21	Jumlah PCT	Bh	
22	Jumlah CO	Bh	
23	Jumlah AVS	Bh	
24	Jumlah LBS	Bh	6
No. NAMA PERALAN			
1	Balapan - tempat pohon	A	Kms
2	Pegangan Tiang TM	B	Bh
3	Pasokan Tiang	C	Bh
4	Pegangan Conductor	D	Kms
5	Pegangan Conductor	E	Kms
6	Tiang tiang/Tempat	F	Bh
7	Pegangan Pond	G	Bh
8	Pondan Tiang/GTT	H	Bh
9	Pembesaran Ground Wire	I	Kms
10	Pemasangan COLBANSYS	J	Bh
11	Pemasangan LBS	K	Bh
12	Pemasangan Linyer Conductor	L	Bh
13	Pemasangan Isolator	M	Bh
14	Pemasangan Anker	N	Bh
15	Pemasangan COLBANSYS	O	Bh
16	Pemasangan Chose Area	P	Bh

PT. PLN (PERSERO)
DISTRIBUSI JATIM - AP&J MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG : PUJON

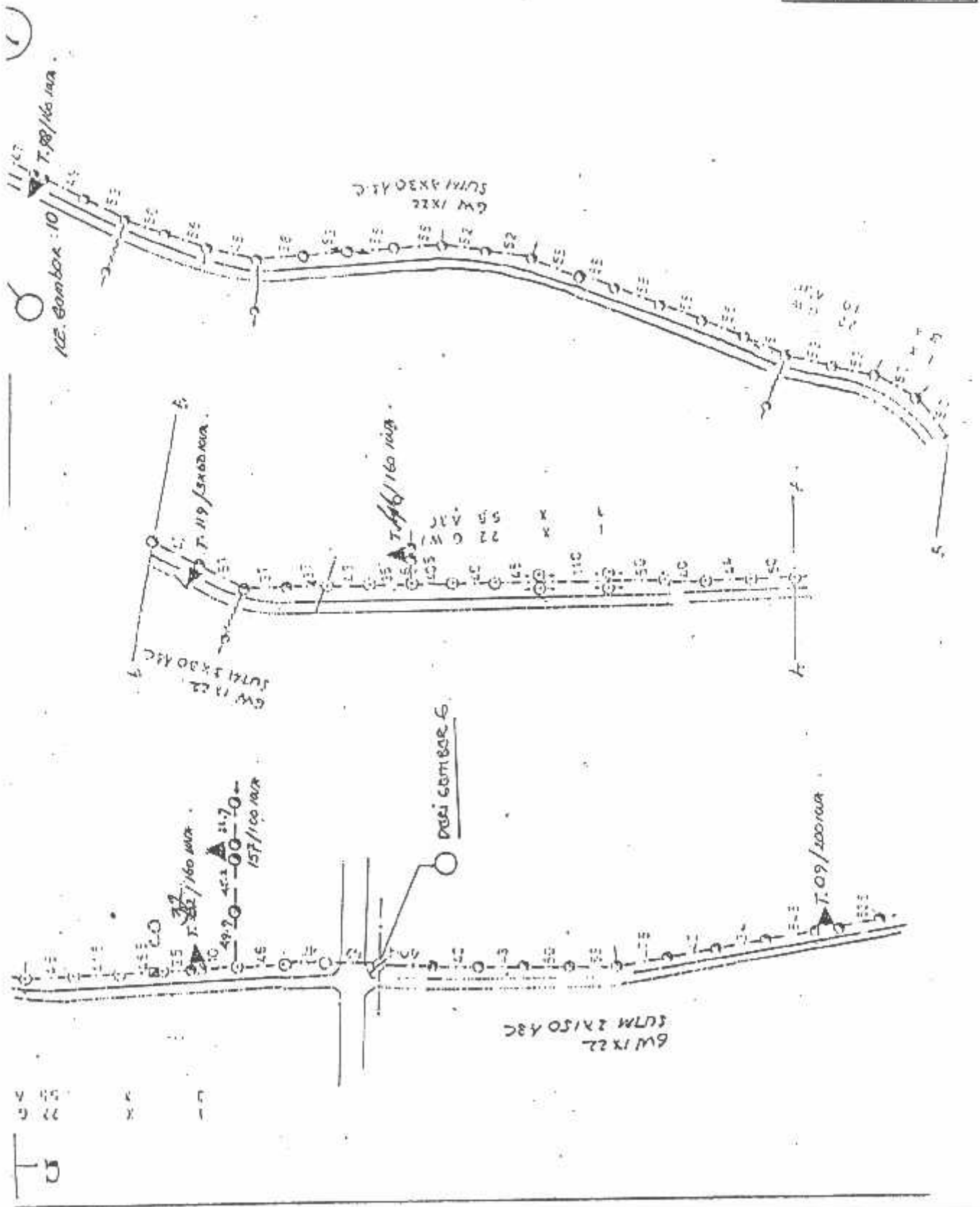
pelaksana pengawas diketahui
WIDDODO SUJOKO MANAGER

No	KETERANGAN	KODE	JUMLAH	UNIT
1	Kapas - respon pobox	A	1	Kns
2	Preseksi Tangk TM	B	2	Bh
3	Preseksi Tangk	C	1	Bh
4	Preseksi Conduktor	D	1	Kns
5	Preseksi Conduktor	E	1	Kns
6	Tangg miring/brogan	F	1	Bh
7	Pracharusa Pand	G	1	Bh
8	Postak Tang/GIT	H	1	Bh
9	Preseksi Groat Wite	I	1	Kns
10	Preseksi COLESAVSS	J	1	Bh
11	Preseksi Penanfan LBS	K	1	Bh
12	Preseksi Lemip Conector	L	1	Bh
13	Preseksi Lohoa	M	1	Bh
14	Preseksi Anator	N	1	Bh
15	Preseksi COLESAVSS	O	1	Bh
16	Preseksi Conduktor	P	1	Bh

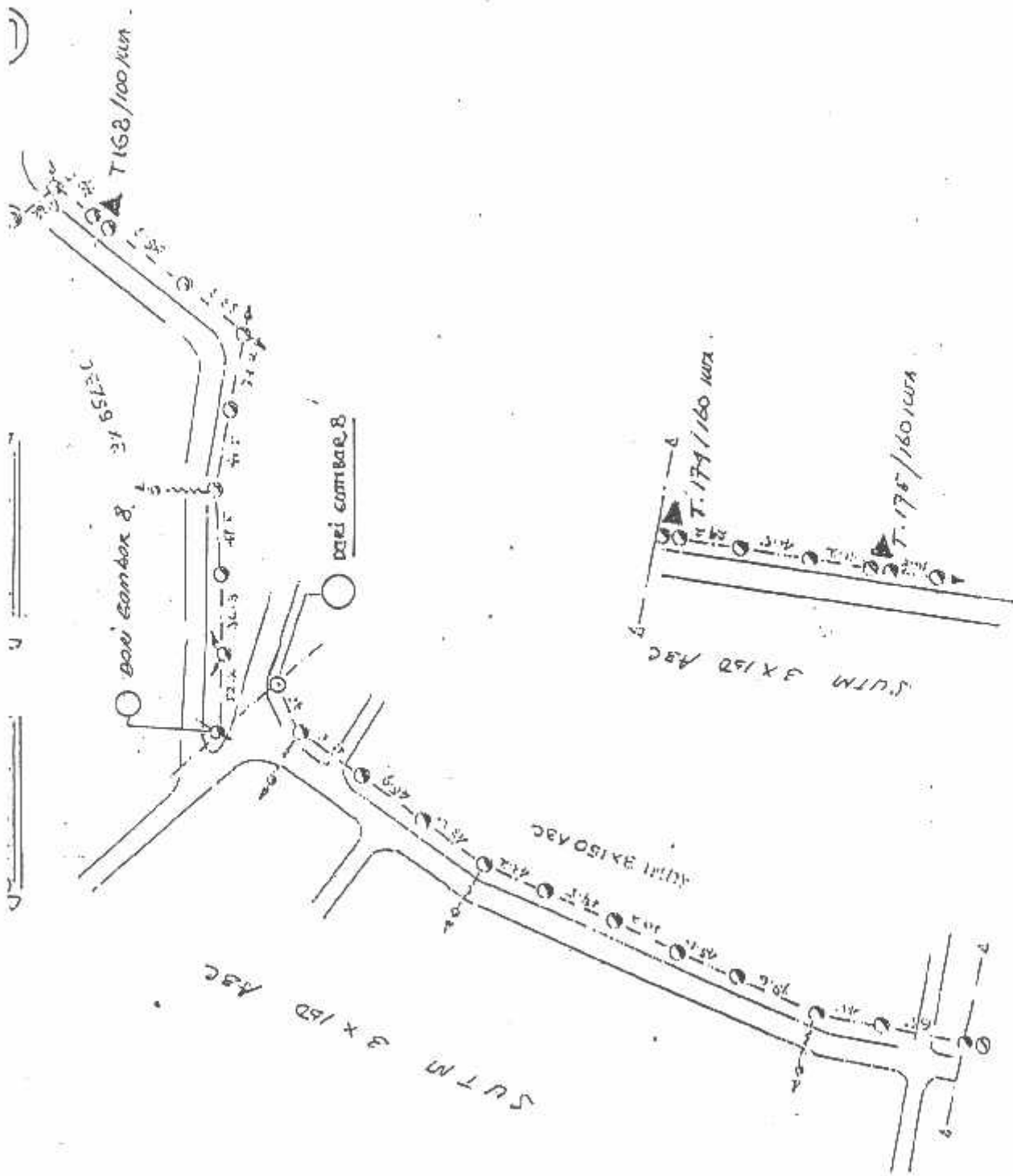
PT. PLN (PERSERO)
DISTRIBUSI JATIM - AP&J MALANG
UNIT PELAYANAN & JARINGAN BATU

PELAYANAN : PUJON

pelaksana pengawas diketahui
WIDODO SUJOKO MANAGER



No	KETERANGAN	KODE	JUMLAH	UNIT
1	Pasang SUTM 3x300k Cu	Ku	6223	
2	TM1	Bh	31	
3	TM2	Bh	11	
4	TM4	Bh		
5	TM3	Bh		
6	TM8	Bh		
7	TM10	Bh		
8	TM11	Bh		
9	TM12	Bh		
10	TM14	Bh		
11	TM15 ARE / SOMA	Bh		
12	TM TYPE POLE	Bh		
13	CONTRA MAS	Bh		
14	SEBOR	Bh		
15	DRUKER SCROOR	Bh		
16	Jumlah Tangk besi TM	Bh		
17	Jumlah Tangk besi TM	Bh		
18	Jumlah Gerde	Bh		
19	Jumlah besi Trela	Kns		
20	Jumlah Kapasitor	Bh		
21	Jumlah PCT	Bh		
22	Jumlah CO	Bh		
23	Jumlah A.V.S	Bh		
24	Jumlah LBS	Bh		



No	Uraian Pekerjaan	Kode	Kuantitas	Uraian
1	Rampung-rampung pokok	A	Kms	
2	Pemetaan Tiang TM	B	Bh	
3	Pemontoran Tiang	C	Bh	
4	Proyeksi Conduktor	D	Kms	
5	Proyeksi Conduktor	E	Kms	
6	Tiang miring/tegak	F	Bh	
7	Pasak-Semam Pasak	G	Bh	
8	Pondasi Tiang/GIT	H	Bh	
9	Pasak-pasak Ground Wire	I	Kms	
10	Pemontoran Pemastian LBS	J	Bh	
11	Pemontoran Pemastian LBS	K	Bh	
12	Pemontoran Lintang Conductor	L	Bh	
13	Proyeksi Lintang	M	Bh	
14	Proyeksi Arcer	N	Bh	
15	Proyeksi Conductor	O	Bh	
16				

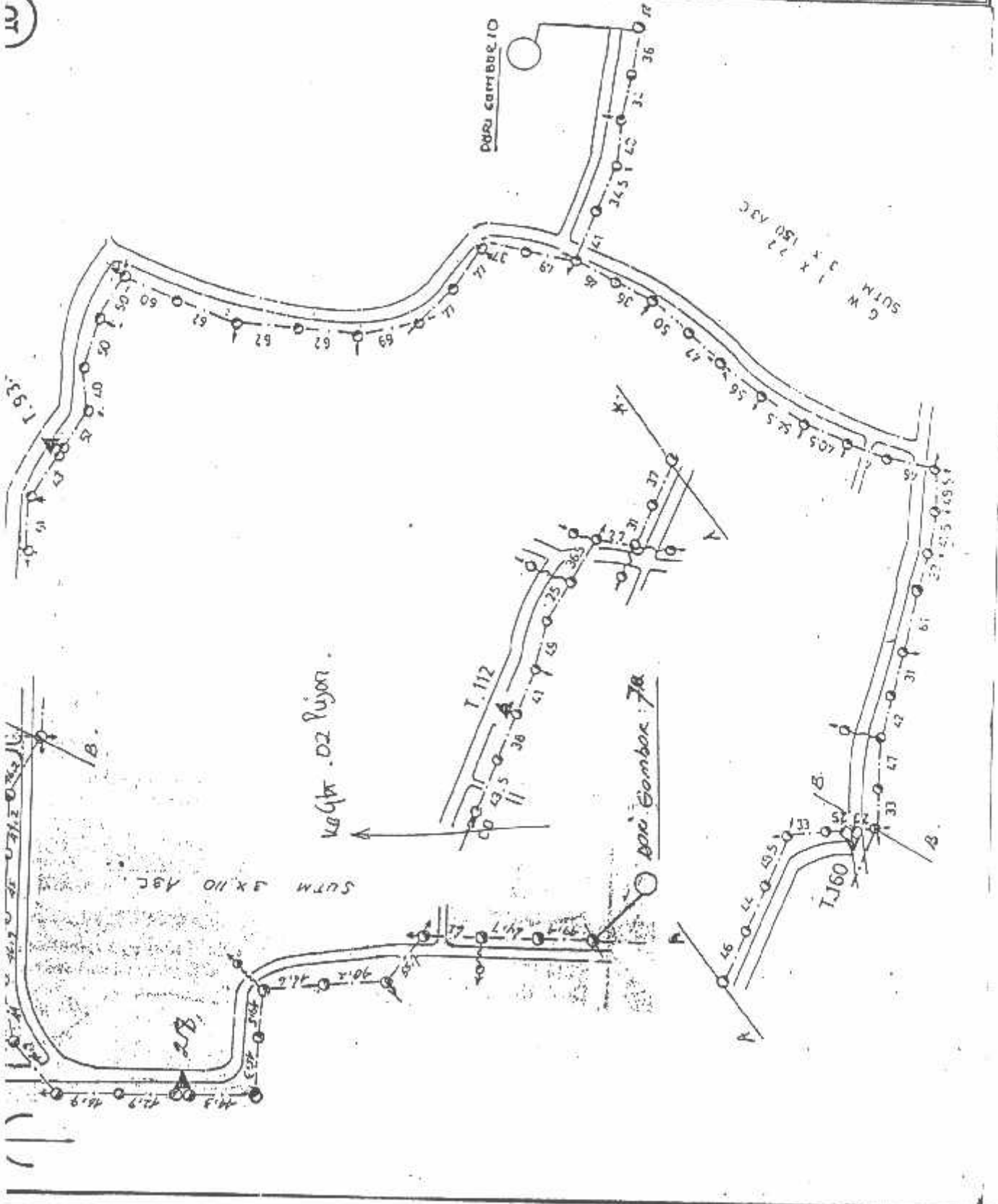
PT. PLN (PERSERO)
DISTRIBUSI JATIM - AP&J MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG : PUJON

pelaksana pengawas diketahui
WIDODO SUJOKO MANAGER

692

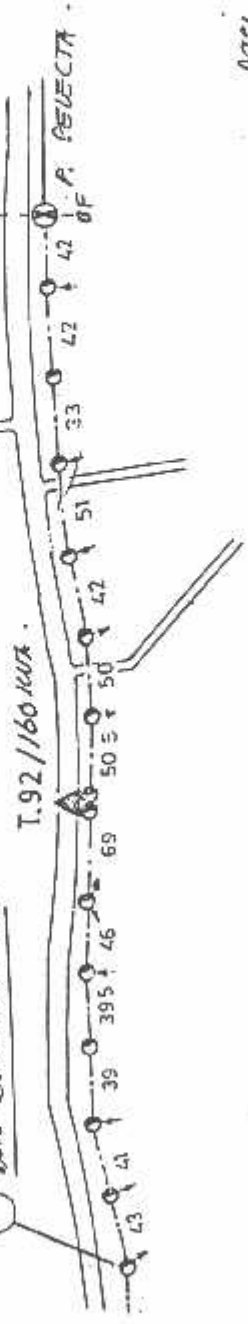
10



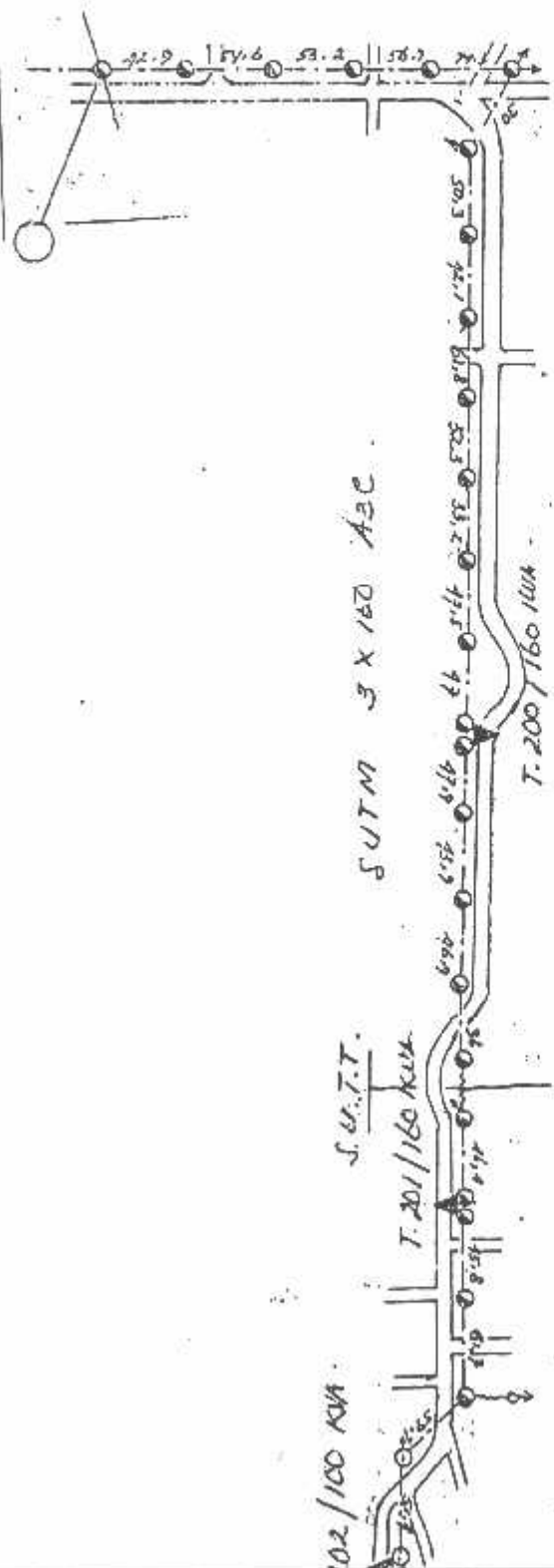
11

G. V. 1 X 22' 83
SUTM KONSI 20 KV 3 X 150 A3C

boni Gambar : 10.



boni Gambar : 8.



No.	KADANG PERTULANG	SATUAN	JUMLAH	
1	Majang SUTM A3C/50	Kms	1718,1	
2	TM 1	Bh	24	
3	TM 2	Bh	12	
4	TM 4	Bh	1	
5	TM 5	Bh	1	
6	TM 6	Bh	1	
7	TM 10	Bh	1	
8	TM 11	Bh	1	
9	TM 12	Bh	12	
10	TM 14	Bh	42	
11	PREYKIN CUT 150	Bh	42	
12	TM TYPE POLE	Bh	1	
13	CONTRA MAS	Bh	12	
14	SCHOOR	Bh	5	
15	DRUCK SCHOOR	Bh	42	
16	Jumlah Tiang boni TM	Bh	4	
17	Jumlah Tiang boni TM	Bh	300	
18	Jumlah Gerdil	Bh	-	
19	Jumlah Iva Tield	Kva	-	
20	Jumlah Kapasitor	Bh	-	
21	Jumlah PCT	Bh	12	
22	Jumlah C.O	Bh	-	
23	Jumlah A.V.S	Bh	1	
24	Jumlah L.B.S	Bh	-	
No.	KADANG PERBAIKAN	CODE	SATUAN	J. V. V.
1	Rapat - rapai poles	A	Kms	-
2	Pengapian Tiang TM	B	Bh	-
3	Pasoveran Tiang	C	Bh	-
4	Pengapian Obedektor	D	Kms	-
5	Pengapian Conduktor	E	Kms	-
6	Tiang miring/kepote	F	Bh	-
7	Pasoveran Pund	G	Bh	-
8	Pondasi Tiang/GTT	H	Bh	-
9	Pasoveran Grouad Wire	I	Kms	-
10	Pasoveran COLEBSAVVS	J	Bh	-
11	Pasoveran pengapian L.B.S	K	Bh	-
12	Pasoveran Lympo Ombuch	L	Bh	-
13	Pasoveran Isobur	M	Bh	-
14	Pasoveran Arrester	N	Bh	-
15	Pasoveran COLEBSAVVS	O	Bh	-
16	Pasoveran Cross Arm	P	Bh	-

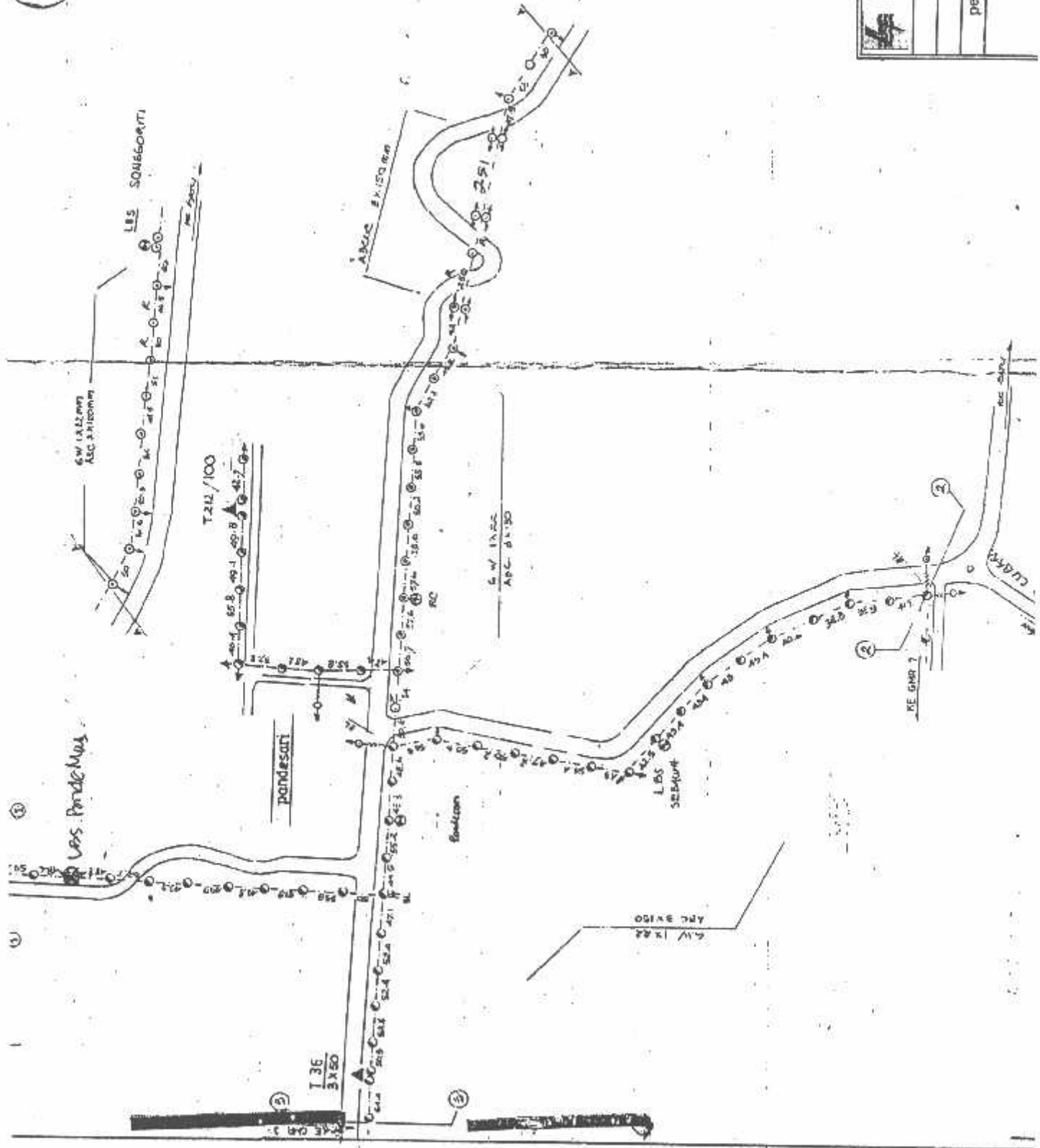
PT. PLN (PERSERO)
DISTRIBUSI JATIM - AP&J MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG : PUJON

pelaksana pengawas diketahui

WIDODO SUJOKO MANAGER

1



305

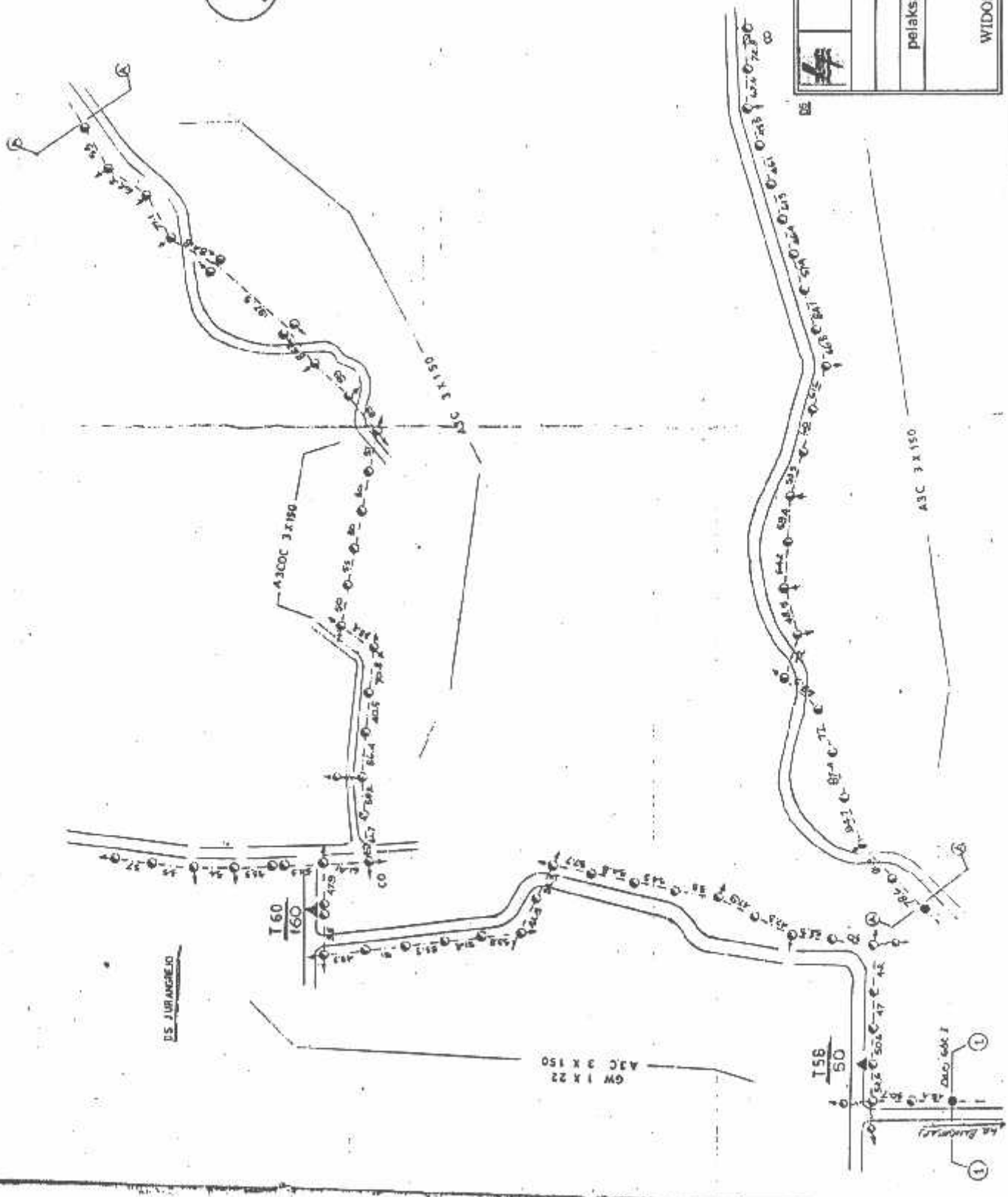
No	Uraian Pekerjaan	Satuan	Jumlah	Uraian
1	Pasang 100VA/110V	Km	1.1	
2	T.M.1	Bk	11	
3	T.M.2	Bk	11	
4	T.M.3	Bk	11	
5	T.M.4	Bk	11	
6	T.M.5	Bk	11	
7	T.M.6	Bk	11	
8	T.M.7	Bk	11	
9	T.M.8	Bk	11	
10	T.M.9	Bk	11	
11	T.M.10	Bk	11	
12	T.M.11	Bk	11	
13	T.M.12	Bk	11	
14	T.M.13	Bk	11	
15	T.M.14	Bk	11	
16	T.M.15	Bk	11	
17	T.M.16	Bk	11	
18	T.M.17	Bk	11	
19	T.M.18	Bk	11	
20	T.M.19	Bk	11	
21	T.M.20	Bk	11	
22	T.M.21	Bk	11	
23	T.M.22	Bk	11	
24	T.M.23	Bk	11	
25	T.M.24	Bk	11	
26	T.M.25	Bk	11	
27	T.M.26	Bk	11	
28	T.M.27	Bk	11	
29	T.M.28	Bk	11	
30	T.M.29	Bk	11	
31	T.M.30	Bk	11	
32	T.M.31	Bk	11	
33	T.M.32	Bk	11	
34	T.M.33	Bk	11	
35	T.M.34	Bk	11	
36	T.M.35	Bk	11	
37	T.M.36	Bk	11	
38	T.M.37	Bk	11	
39	T.M.38	Bk	11	
40	T.M.39	Bk	11	
41	T.M.40	Bk	11	
42	T.M.41	Bk	11	
43	T.M.42	Bk	11	
44	T.M.43	Bk	11	
45	T.M.44	Bk	11	
46	T.M.45	Bk	11	
47	T.M.46	Bk	11	
48	T.M.47	Bk	11	
49	T.M.48	Bk	11	
50	T.M.49	Bk	11	
51	T.M.50	Bk	11	
52	T.M.51	Bk	11	
53	T.M.52	Bk	11	
54	T.M.53	Bk	11	
55	T.M.54	Bk	11	
56	T.M.55	Bk	11	
57	T.M.56	Bk	11	
58	T.M.57	Bk	11	
59	T.M.58	Bk	11	
60	T.M.59	Bk	11	
61	T.M.60	Bk	11	
62	T.M.61	Bk	11	
63	T.M.62	Bk	11	
64	T.M.63	Bk	11	
65	T.M.64	Bk	11	
66	T.M.65	Bk	11	
67	T.M.66	Bk	11	
68	T.M.67	Bk	11	
69	T.M.68	Bk	11	
70	T.M.69	Bk	11	
71	T.M.70	Bk	11	
72	T.M.71	Bk	11	
73	T.M.72	Bk	11	
74	T.M.73	Bk	11	
75	T.M.74	Bk	11	
76	T.M.75	Bk	11	
77	T.M.76	Bk	11	
78	T.M.77	Bk	11	
79	T.M.78	Bk	11	
80	T.M.79	Bk	11	
81	T.M.80	Bk	11	
82	T.M.81	Bk	11	
83	T.M.82	Bk	11	
84	T.M.83	Bk	11	
85	T.M.84	Bk	11	
86	T.M.85	Bk	11	
87	T.M.86	Bk	11	
88	T.M.87	Bk	11	
89	T.M.88	Bk	11	
90	T.M.89	Bk	11	
91	T.M.90	Bk	11	
92	T.M.91	Bk	11	
93	T.M.92	Bk	11	
94	T.M.93	Bk	11	
95	T.M.94	Bk	11	
96	T.M.95	Bk	11	
97	T.M.96	Bk	11	
98	T.M.97	Bk	11	
99	T.M.98	Bk	11	
100	T.M.99	Bk	11	
101	T.M.100	Bk	11	

PT. PLN (PERSERO)
DISTRIBUSI JATIM - AP&J MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG : PUJON

pelaksana pengawas diketahui

2



HASIL INSPEKSI JARINGAN		
No. Lokasi	Perbaikan	Andalan
1	Pengujian UTM	Kms 4273.6
2	TM1	34
3	TM2	34
4	TM3	34
5	TM4	34
6	TM5	34
7	TM6	34
8	TM7	34
9	TM8	34
10	TM9	34
11	TM10	34
12	TM TYPE POLE	34
13	CONTRAK M&I	34
14	SKRIBOR	34
15	BRUCE KROKOR	34
16	Jarak Tiang ke TM	34
17	Jarak Tiang ke TM	34
18	Jarak Ganda	34
19	Jarak ke Tiang	34
20	Jarak Kapasitor	34
21	Jarak PC	34
22	Jarak CO	34
23	Jarak AV1	34
24	Jarak LBT	34
25	Waktu Inspeksi	34
26	Waktu Perbaikan	34
27	Waktu Pengujian	34
28	Waktu Pengukuran	34
29	Waktu Pengukuran	34
30	Waktu Pengukuran	34
31	Waktu Pengukuran	34
32	Waktu Pengukuran	34
33	Waktu Pengukuran	34
34	Waktu Pengukuran	34
35	Waktu Pengukuran	34
36	Waktu Pengukuran	34
37	Waktu Pengukuran	34
38	Waktu Pengukuran	34
39	Waktu Pengukuran	34
40	Waktu Pengukuran	34
41	Waktu Pengukuran	34
42	Waktu Pengukuran	34
43	Waktu Pengukuran	34
44	Waktu Pengukuran	34
45	Waktu Pengukuran	34
46	Waktu Pengukuran	34
47	Waktu Pengukuran	34
48	Waktu Pengukuran	34
49	Waktu Pengukuran	34
50	Waktu Pengukuran	34

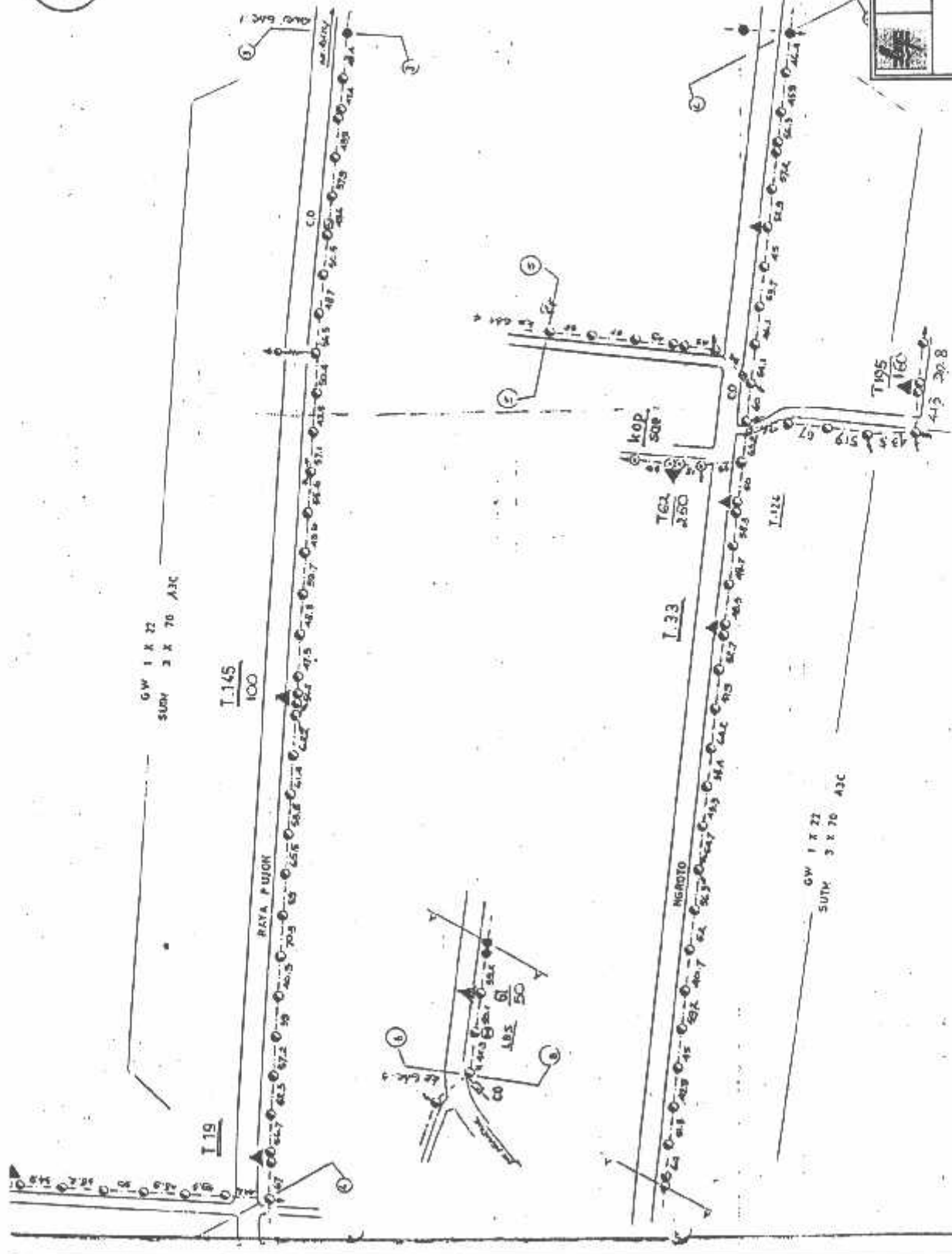
PT. PLN (PERSERO)
DISTRIBUSI JATIM - APJ MALANG
UNIT PELAYANAN & JARINGAN BATU

penyulung : PUJON

pelaksana : pengawas diketahui
WIDODO SUJOKO MANAGER

3

4095



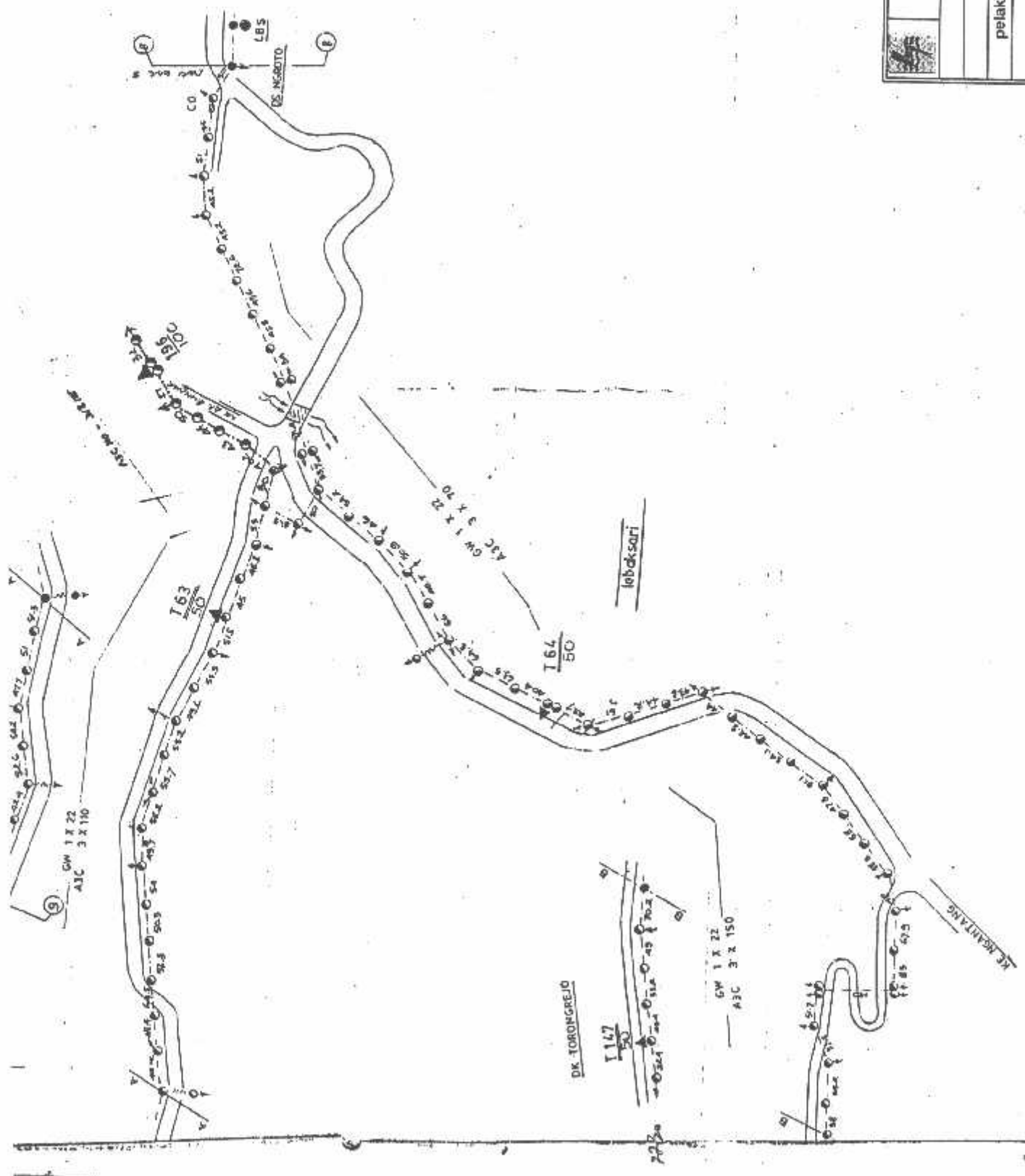
REKAM JEKAL JARINGAN		
No	LOKASI	STATUS
1	Perumahan RTM/A3C 70	RM
2	TM 1	RM
3	TM 2	RM
4	TM 3	RM
5	TM 4	RM
6	TM 5	RM
7	TM 6	RM
8	TM 7	RM
9	TM 8	RM
10	TM 9	RM
11	TM 10	RM
12	TM 11	RM
13	TM 12	RM
14	TM 13	RM
15	TM 14	RM
16	TM 15	RM
17	TM 16	RM
18	TM 17	RM
19	TM 18	RM
20	TM 19	RM
21	TM 20	RM
22	TM 21	RM
23	TM 22	RM
24	TM 23	RM
25	TM 24	RM
26	TM 25	RM
27	TM 26	RM
28	TM 27	RM
29	TM 28	RM
30	TM 29	RM
31	TM 30	RM
32	TM 31	RM
33	TM 32	RM
34	TM 33	RM
35	TM 34	RM
36	TM 35	RM
37	TM 36	RM
38	TM 37	RM
39	TM 38	RM
40	TM 39	RM
41	TM 40	RM
42	TM 41	RM
43	TM 42	RM
44	TM 43	RM
45	TM 44	RM
46	TM 45	RM
47	TM 46	RM
48	TM 47	RM
49	TM 48	RM
50	TM 49	RM
51	TM 50	RM
52	TM 51	RM
53	TM 52	RM
54	TM 53	RM
55	TM 54	RM
56	TM 55	RM
57	TM 56	RM
58	TM 57	RM
59	TM 58	RM
60	TM 59	RM
61	TM 60	RM
62	TM 61	RM
63	TM 62	RM
64	TM 63	RM
65	TM 64	RM
66	TM 65	RM
67	TM 66	RM
68	TM 67	RM
69	TM 68	RM
70	TM 69	RM
71	TM 70	RM
72	TM 71	RM
73	TM 72	RM
74	TM 73	RM
75	TM 74	RM
76	TM 75	RM
77	TM 76	RM
78	TM 77	RM
79	TM 78	RM
80	TM 79	RM
81	TM 80	RM
82	TM 81	RM
83	TM 82	RM
84	TM 83	RM
85	TM 84	RM
86	TM 85	RM
87	TM 86	RM
88	TM 87	RM
89	TM 88	RM
90	TM 89	RM
91	TM 90	RM
92	TM 91	RM
93	TM 92	RM
94	TM 93	RM
95	TM 94	RM
96	TM 95	RM
97	TM 96	RM
98	TM 97	RM
99	TM 98	RM
100	TM 99	RM
101	TM 100	RM

PT. PLN (PERSERO)
DISTRIBUSI JATIM - APAL MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG : PUJON

pelaksana	pengawas	diketahui
WIDODO	SUJOKO	MANAGER

5

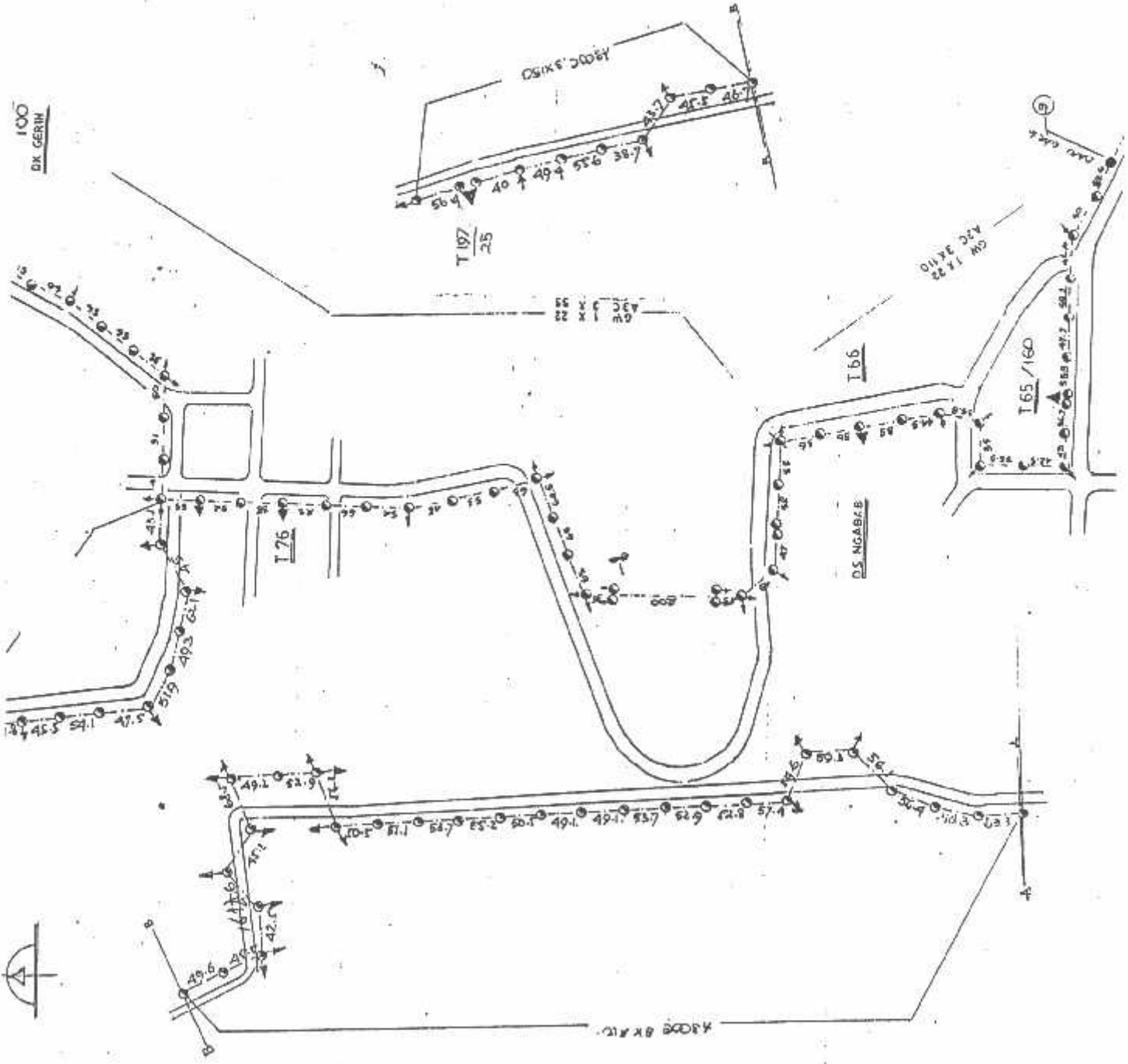


PT. PLN (PERSERO)
DISTRIBUSI JATIM - AP&J MALANG
UNIT PELAYANAN & JARINGAN BATU

PENYULANG : PUJON

pelaksana	pengawas	diketahui
-----------	----------	-----------

6



HASIL INSPEKSI JARINGAN		Luas	jumlah
No.	nama pemakai	km	rumah
1	Pelung WTM 432.55	177.3	
2	TM 1		
3	TM 1		
4	TM 1		
5	TM 2		
6	TM 4		
7	TM 18		
8	TM 18		
9	TM 18		
10	TM 18		
11	TM 18		
12	TM 18		
13	TM 18		
14	TM 18		
15	TM 18		
16	TM 18		
17	TM 18		
18	TM 18		
19	TM 18		
20	TM 18		
21	TM 18		
22	TM 18		
23	TM 18		
24	TM 18		
25	TM 18		
26	TM 18		
27	TM 18		
28	TM 18		
29	TM 18		
30	TM 18		
31	TM 18		
32	TM 18		
33	TM 18		
34	TM 18		
35	TM 18		
36	TM 18		
37	TM 18		
38	TM 18		
39	TM 18		
40	TM 18		
41	TM 18		
42	TM 18		
43	TM 18		
44	TM 18		
45	TM 18		
46	TM 18		
47	TM 18		
48	TM 18		
49	TM 18		
50	TM 18		
51	TM 18		
52	TM 18		
53	TM 18		
54	TM 18		
55	TM 18		
56	TM 18		
57	TM 18		
58	TM 18		
59	TM 18		
60	TM 18		
61	TM 18		
62	TM 18		
63	TM 18		
64	TM 18		
65	TM 18		
66	TM 18		
67	TM 18		
68	TM 18		
69	TM 18		
70	TM 18		
71	TM 18		
72	TM 18		
73	TM 18		
74	TM 18		
75	TM 18		
76	TM 18		
77	TM 18		
78	TM 18		
79	TM 18		
80	TM 18		
81	TM 18		
82	TM 18		
83	TM 18		
84	TM 18		
85	TM 18		
86	TM 18		
87	TM 18		
88	TM 18		
89	TM 18		
90	TM 18		
91	TM 18		
92	TM 18		
93	TM 18		
94	TM 18		
95	TM 18		
96	TM 18		
97	TM 18		
98	TM 18		
99	TM 18		
100	TM 18		

PT. PLN (PERSERO)
 DISTRIBUTI JATIM - APJ MALANG
 UNIT PELAYANAN & JARINGAN BATU

Penyulang : PUJON

aksana pengawas diketahui

9

HASIL INSPEKSI JARINGAN		LANTARAN		SUDUKAN	
No	LOKASI BERTUMBUH	Uraian	Uraian	Uraian	Uraian
1	Pepohonan				
2	TM 1				
3	TM 2				
4	TM 3				
5	TM 4				
6	TM 5				
7	TM 6				
8	TM 7				
9	TM 8				
10	TM 9				
11	TM 10				
12	TM 11				
13	TM 12				
14	TM 13				
15	TM 14				
16	TM 15				
17	TM 16				
18	TM 17				
19	TM 18				
20	TM 19				
21	TM 20				
22	TM 21				
23	TM 22				
24	TM 23				
25	TM 24				
26	TM 25				
27	TM 26				
28	TM 27				
29	TM 28				
30	TM 29				
31	TM 30				
32	TM 31				
33	TM 32				
34	TM 33				
35	TM 34				
36	TM 35				
37	TM 36				
38	TM 37				
39	TM 38				
40	TM 39				
41	TM 40				
42	TM 41				
43	TM 42				
44	TM 43				
45	TM 44				
46	TM 45				
47	TM 46				
48	TM 47				
49	TM 48				
50	TM 49				
51	TM 50				
52	TM 51				
53	TM 52				
54	TM 53				
55	TM 54				
56	TM 55				
57	TM 56				
58	TM 57				
59	TM 58				
60	TM 59				
61	TM 60				
62	TM 61				
63	TM 62				
64	TM 63				
65	TM 64				
66	TM 65				
67	TM 66				
68	TM 67				
69	TM 68				
70	TM 69				
71	TM 70				
72	TM 71				
73	TM 72				
74	TM 73				
75	TM 74				
76	TM 75				
77	TM 76				
78	TM 77				
79	TM 78				
80	TM 79				
81	TM 80				
82	TM 81				
83	TM 82				
84	TM 83				
85	TM 84				
86	TM 85				
87	TM 86				
88	TM 87				
89	TM 88				
90	TM 89				
91	TM 90				
92	TM 91				
93	TM 92				
94	TM 93				
95	TM 94				
96	TM 95				
97	TM 96				
98	TM 97				
99	TM 98				
100	TM 99				
101	TM 100				
102	TM 101				
103	TM 102				
104	TM 103				
105	TM 104				
106	TM 105				
107	TM 106				
108	TM 107				
109	TM 108				
110	TM 109				
111	TM 110				
112	TM 111				
113	TM 112				
114	TM 113				
115	TM 114				
116	TM 115				
117	TM 116				
118	TM 117				
119	TM 118				
120	TM 119				
121	TM 120				
122	TM 121				
123	TM 122				
124	TM 123				
125	TM 124				
126	TM 125				
127	TM 126				
128	TM 127				
129	TM 128				
130	TM 129				
131	TM 130				
132	TM 131				
133	TM 132				
134	TM 133				
135	TM 134				
136	TM 135				
137	TM 136				
138	TM 137				
139	TM 138				
140	TM 139				
141	TM 140				
142	TM 141				
143	TM 142				
144	TM 143				
145	TM 144				
146	TM 145				
147	TM 146				
148	TM 147				
149	TM 148				
150	TM 149				
151	TM 150				
152	TM 151				
153	TM 152				
154	TM 153				
155	TM 154				
156	TM 155				
157	TM 156				
158	TM 157				
159	TM 158				
160	TM 159				
161	TM 160				
162	TM 161				
163	TM 162				
164	TM 163				
165	TM 164				
166	TM 165				
167	TM 166				
168	TM 167				
169	TM 168				
170	TM 169				
171	TM 170				
172	TM 171				
173	TM 172				
174	TM 173				
175	TM 174				
176	TM 175				
177	TM 176				
178	TM 177				
179	TM 178				
180	TM 179				
181	TM 180				
182	TM 181				
183	TM 182				
184	TM 183				
185	TM 184				
186	TM 185				
187	TM 186				
188	TM 187				
189	TM 188				
190	TM 189				
191	TM 190				
192	TM 191				
193	TM 192				
194	TM 193				
195	TM 194				
196	TM 195				
197	TM 196				
198	TM 197				
199	TM 198				
200	TM 199				
201	TM 200				
202	TM 201				
203	TM 202				
204	TM 203				
205	TM 204				
206	TM 205				
207	TM 206				
208	TM 207				
209	TM 208				
210	TM 209				
211	TM 210				
212	TM 211				
213	TM 212				
214	TM 213				
215	TM 214				
216	TM 215				
217	TM 216				
218	TM 217				
219	TM 218				
220	TM 219				
221	TM 220				
222	TM 221				
223	TM 222				
224	TM 223				
225	TM 224				
226	TM 225				
227	TM 226				
228	TM 227				
229	TM 228				
230	TM 229				
231	TM 230				
232	TM 231				
233	TM 232				
234	TM 233				
235	TM 234				
236	TM 235				
237	TM 236				
238	TM 237				
239	TM 238				
240	TM 239				
241	TM 240				
242	TM 241				
243	TM 242				
244	TM 243				
245	TM 244				
246	TM 245				
247	TM 246				
248	TM 247				
249	TM 248				
250	TM 249				
251	TM 250				
252	TM 251				
253	TM 252				
254	TM 253				
255	TM 254				
256	TM 255				
257	TM 256				
258	TM 257				
259	TM 258				
260	TM 259				
261	TM 260				
262	TM 261				
263	TM 262				
264	TM 263				
265	TM 264				
266	TM 265				
267	TM 266				
268	TM 267				
269	TM 268				
270	TM 269				
271	TM 270				
272	TM 271				
273	TM 272				
274	TM 273				
275	TM 274				
276	TM 275				
277	TM 276				
278	TM 277				
279	TM 278				
280	TM 279				
281	TM 280				
282	TM 281				
283	TM 282				
284	TM 283				
285	TM 284				
286	TM 285				
287	TM 286				
288	TM 287				
289	TM 288				
290	TM 289				
291	TM 290				
292	TM 291				
293	TM 292				
294	TM 293				
295	TM 294				
296	TM 295				
297	TM 296				
298	TM 297				
299	TM 298				
300	TM 299				
301	TM 300				
302	TM 301				
303	TM 302				
304	TM 303				
305	TM 304				
306	TM 305				
307	TM 306				
308	TM 307				
309	TM 308				
310	TM 309				
311	TM 310				
312	TM 311				
313	TM 312				
314	TM 313				
315	TM 314				
316	TM 315				
317	TM 316				
318	TM 317				
319					

Listing Program “Optimalisasi Pemilihan Konduktor Pada Saluran Distribusi Primer 20 kV Penyulang Pujon Menggunakan Metode Evolutionary Programming”

```
program CableEP;

uses
  Forms,
  uAbout in 'uAbout.pas' {frmAbout},
  uComplex in 'uComplex.pas',
  uMatrix in 'uMatrix.pas',
  uMenu in 'uMenu.pas' {frmMenu},
  uUtils in 'uUtils.pas',
  uInputLF in 'Loadflow\uInputLF.pas' {frmInput},
  uLoadflow in 'Loadflow\uLoadflow.pas',
  uNewtonPolar in 'Loadflow\NewtonPolar\uNewtonPolar.pas',
  uFitness in 'Genetic\uFitness.pas',
  uGenetic in 'Genetic\uGenetic.pas',
  uObjFunc in 'Genetic\uObjFunc.pas',
  uRandom in 'Genetic\uRandom.pas',
  uEvoPro in 'Genetic\EvoPro\uEvoPro.pas',
  uParamEP in 'Genetic\EvoPro\uParamEP.pas' {frmParamEP},
  uCable in 'uCable.pas',
  uGenerator in 'Generator\uGenerator.pas';

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TfrmMenu, frmMenu);
  Application.CreateForm(TfrmAbout, frmAbout);
  Application.CreateForm(TfrmInput, frmInput);
  Application.CreateForm(TfrmParamEP, frmParamEP);
  Application.Run;
end.
unit uAbout;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;
type
  TfrmAbout = class(TForm)
  private
```

```

begin
  if result.fitness>FParent[i-1].fitness then
    begin
      result:=GetIndividu(FParent[i-1]);
    end;
  end;
end;

procedure TEvoPro1.InitParent;
var i,j:integer;
begin
  SetLength(FParent,PopSize);
  SetLength(FChild,PopSize);
  for i:=1 to PopSize do
    begin
      SetLength(FParent[i-1].chrom,LengthX);
      SetLength(FChild[i-1].chrom,LengthX);
    end;
  SetLength(FBestIndi.chrom,LengthX);
  for i:=1 to PopSize do
    begin
      if i=1 then
        begin
          for j:=1 to LengthX do
            begin
              FParent[i-1].chrom[j-1]:=1;
            end;
          end
        else
          begin
            for j:=1 to LengthX do
              begin
                FParent[i-1].chrom[j-1]:=FRandom.NextDouble;
              end;
            end;
          FParent[i-1].fitness:=CalcFitness(FParent[i-1].chrom);
        end;
    end;

procedure TEvoPro1.Statistik;
var i:integer;
    sumFitness:double;
begin

```

```

FMin1:=FParent[0].fitness;
FMax1:=FParent[0].fitness;
sumFitness:=FParent[0].fitness;
for i:=2 to PopSize do
begin
if FMin1>FParent[i-1].fitness then
begin
FMin1:=FParent[i-1].fitness;
end;
if FMax1<FParent[i-1].fitness then
begin
FMax1:=FParent[i-1].fitness;
end;
sumFitness:=sumFitness+FParent[i-1].fitness;
end;
FAvg1:=sumFitness/PopSize;
end;

```

```

procedure TEvoPro1.Generasi;
var i,j:integer;
    tmpIndi:TIndiDouble1;
    Smin,Sfun:double;
begin
tmpIndi:=FindIndiMin;
Smin:=tmpIndi.fitness;
for i:=1 to PopSize do
begin
Sfun:=FParent[i-1].fitness;
for j:=1 to LengthX do
begin
FChild[i-1].chrom[j-1]:=MutasiEvoPro(Sfun,Smin,Betha,
FParent[i-1].chrom[j-1],FRandom);
end;
FChild[i-1].fitness:=CalcFitness(FChild[i-1].chrom);
end;
end;

```

```

procedure TEvoPro1.Kompetisi;
var i,j,point:integer;
    tmp:TPopDouble1;
    tmpIndi:TIndiDouble1;
    sortIndi:iArr1;
begin

```

```

SetLength(tmp,PopSize*2);
for i:=1 to PopSize do
begin
  tmp[i-1]:=GetIndividu(FParent[i-1]);
  tmp[PopSize+i-1]:=GetIndividu(FChild[i-1]);
end;
SetLength(sortIndi,PopSize*2);
for i:=1 to PopSize*2 do
begin
  sortIndi[i-1]:=0;
  for j:=1 to PopSize*2 do
  begin
    repeat
      point:=FRandom.NextInt(1,PopSize*2);
    until point<>i;
    if tmp[i-1].fitness<tmp[point-1].fitness then
    begin
      sortIndi[i-1]:=sortIndi[i-1]+1;
    end;
  end;
end;
for i:=1 to PopSize*2-1 do
begin
  for j:=i to PopSize*2 do
  begin
    if sortIndi[i-1]<sortIndi[j-1] then
    begin
      Swap(sortIndi[i-1],sortIndi[j-1]);
      tmpIndi:=GetIndividu(tmp[i-1]);
      tmp[i-1]:=GetIndividu(tmp[j-1]);
      tmp[j-1]:=GetIndividu(tmpIndi);
    end;
  end;
end;
for i:=1 to PopSize do
begin
  FParent[i-1]:=GetIndividu(tmp[i-1]);
end;
end;

procedure TEvoPro1.doHitung;
var gen:integer;
    TempIndi:TIndiDouble1;

```

```

    SetLength(FChild[i-1].chrom,FParam,LengthX);
end;
SetLength(FBestIndi.chrom,FParam,LengthX);
for i:=1 to PopSize do
begin
    if i=1 then
    begin
        for j:=1 to FParam do
        begin
            for k:=1 to LengthX do
            begin
                FParent[i-1].chrom[j-1,k-1]:=1;
            end;
        end;
    end
    else
    begin
        for j:=1 to FParam do
        begin
            for k:=1 to LengthX do
            begin
                FParent[i-1].chrom[j-1,k-1]:=FRandom.NextDouble;
            end;
        end;
    end;
    FParent[i-1].fitness:=CalcFitness(FParent[i-1].chrom);
end;
end;

```

```

procedure TEvoPro2.Statistik;
var i:integer;
    sumFitness:double;
begin
    FMin1:=FParent[0].fitness;
    FMax1:=FParent[0].fitness;
    sumFitness:=FParent[0].fitness;
    for i:=2 to PopSize do
    begin
        if FMin1>FParent[i-1].fitness then
        begin
            FMin1:=FParent[i-1].fitness;
        end;
        if FMax1<FParent[i-1].fitness then

```

```

begin
  FMax1:=FParent[i-1].fitness;
end;
sumFitness:=sumFitness+FParent[i-1].fitness;
end;
FAvg1:=sumFitness/PopSize;
end;

procedure TEvoPro2.Generasi;
var i,j,k:integer;
    tmpIndi:TIndiDouble2;
    Smin,Sfun:double;
begin
  tmpIndi:=FindIndiMin;
  Smin:=tmpIndi.fitness;
  for i:=1 to PopSize do
  begin
    Sfun:=FParent[i-1].fitness;
    for j:=1 to FParam do
    begin
      for k:=1 to LengthX do
      begin
        FChild[i-1].chrom[j-1,k-1]:=MutasiEvoPro(Sfun,Smin,Betha,
          FParent[i-1].chrom[j-1,k-1],FRandom);
      end;
    end;
    FChild[i-1].fitness:=CalcFitness(FChild[i-1].chrom);
  end;
end;

procedure TEvoPro2.Kompetisi;
var i,j,point:integer;
    tmp:TPopDouble2;
    tmpIndi:TIndiDouble2;
    sortIndi:iArr1;
begin
  SetLength(tmp,PopSize*2);
  for i:=1 to PopSize do
  begin
    tmp[i-1]:=GetIndividu(FParent[i-1]);
    tmp[PopSize-1+i]:=GetIndividu(FChild[i-1]);
  end;
  SetLength(sortIndi,PopSize*2);

```

```

for i:=1 to PopSize*2 do
begin
  sortIndi[i-1]:=0;
  for j:=1 to PopSize*2 do
  begin
    repeat
      point:=FRandom.NextInt(1,PopSize*2);
    until point < i;
    if tmp[i-1].fitness < tmp[point-1].fitness then
    begin
      sortIndi[i-1]:=sortIndi[point-1]+1;
    end;
  end;
end;
for i:=1 to PopSize*2-1 do
begin
  for j:=i to PopSize*2 do
  begin
    if sortIndi[i-1] < sortIndi[j-1] then
    begin
      Swap(sortIndi[i-1],sortIndi[j-1]);
      tmpIndi:=GetIndividu(tmp[i-1]);
      tmp[i-1]:=GetIndividu(tmp[j-1]);
      tmp[j-1]:=GetIndividu(tmpIndi);
    end;
  end;
end;
for i:=1 to PopSize do
begin
  FParent[i-1]:=GetIndividu(tmp[i-1]);
end;
end;

```

```

procedure TEvoPro2.doHitung;
var gen:integer;
    TempIndi:TIndiDouble2;
begin
  InitParent;
  Statistik;
  FBestIndi:=FindIndiMin;
  gen:=1;
  repeat
    Generasi;
  until

```

```

Ca:=sum.Divide(Za);
Cr[i,j]:=Ca.Real;
Ci[i,j]:=Ca.Imag;
Ca.Free;
Za.Free;
end
else
begin
Cr[i,j]:=0.0;
Ci[i,j]:=0.0;
end;
end;
sum.Free;
SetLength(rG,rNbus,rNbus);
SetLength(rB,rNbus,rNbus);
for i:=0 to rNbus-1 do
begin
for j:=0 to rNbus-1 do
begin
if j=i then
begin
rG[i,j]:=0.0;
rB[i,j]:=0.0;
for k:=0 to rNbus-1 do
begin
rG[i,j]:=rG[i,j]+Cr[i,k];
rB[i,j]:=rB[i,j]+Ci[i,k]+rLc[i,k];
end;
end
else
begin
rG[i,j]:=-Cr[i,j];
rB[i,j]:=-Ci[i,j];
end;
end;
end;
end;
for i:=0 to rNbus-1 do
begin
for j:=0 to rNbus-1 do
begin
if rTr[i,j]<>0 then
begin

```

```

Kompetisi;
Statistik;
TempIndi:=FindIndiMin;
if FBestIndi.fitness>TempIndi.fitness then
begin
  FBestIndi:=GetIndividu(TempIndi);
end;
FMin[gen-1]:=FMin1;
FAvg[gen-1]:=FAvg1;
FMax[gen-1]:=FMax1;
inc(gen);
until (gen>MaxGen);
end;

function TEvoPro2.GetBestChrom:dArr2;
var i,j:integer;
begin
  doHitung;
  SetLength(result,FParam,LengthX);
  for i:=1 to FParam do
  begin
    for j:=1 to LengthX do
    begin
      result[i-1,j-1]:=FBestIndi.chrom[i-1,j-1];
    end;
  end;
end;

end.

unit uFitness;

interface

uses uUtils,uObjFunc,uRandom;

function CalcFitness(var aChrom:bArr1):double;overload;
function CalcFitness(var aChrom:bArr2):double;overload;
function CalcFitness(var aChrom:dArr1):double;overload;
function CalcFitness(var aChrom:dArr2):double;overload;
function CalcFitness(var aChrom:TChromTCSC1):double;overload;
function CalcFitness(var aChrom:TChromUpfc1):double;overload;
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;

```

```

Kompetisi;
Statistik;
TempIndi:=FindIndiMin;
if FBestIndi.fitness>TempIndi.fitness then
begin
  FBestIndi:=GetIndividu(TempIndi);
end;
FMin[gen-1]:=FMin1;
FAvg[gen-1]:=FAvg1;
FMax[gen-1]:=FMax1;
inc(gen);
until (gen>MaxGen);
end;

```

```

function TEvoPro2.GetBestChrom:dArr2;
var i,j:integer;
begin
  doHitung;
  SetLength(result,FParam,LengthX);
  for i:=1 to FParam do
  begin
    for j:=1 to LengthX do
    begin
      result[i-1,j-1]:=FBestIndi.chrom[i-1,j-1];
    end;
  end;
end;

end.

```

```

unit uFitness;

```

```

interface

```

```

uses uUtils,uObjFunc,uRandom;

```

```

function CalcFitness(var aChrom:bArr1):double;overload;
function CalcFitness(var aChrom:bArr2):double;overload;
function CalcFitness(var aChrom:dArr1):double;overload;
function CalcFitness(var aChrom:dArr2):double;overload;
function CalcFitness(var aChrom:TChromTCSC1):double;overload;
function CalcFitness(var aChrom:TChromUpfc1):double;overload;
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;

```

```

    var rRand:TRandomu):double;overload;
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;
    const ip:integer;
    var rRand:TRandomu):double;overload;
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;
    const ip,jp:integer;
    var rRand:TRandomu):double;overload;

```

Implementation

```

function CalcFitness(var aChrom:bArr1):double;
var aX:double;
begin
    aX:=DecodeBinToFloat1(aChrom);
    //insert code obj function here
    result:=aX;
end;

```

```

function CalcFitness(var aChrom:bArr2):double;
var aX:dArr1;
begin
    SetLength(aX,high(aChrom));
    aX:=DecodeBinToFloat2(aChrom);
    //insert code obj function here
    result:=aX[1];
end;

```

```

function CalcFitness(var aChrom:dArr1):double;
begin
    //insert code obj function here
    result:=gObjFunc.doHitung(aChrom);
end;

```

```

function CalcFitness(var aChrom:dArr2):double;
begin
    //insert code obj function here
    result:=0;//gObjFunc.doHitung(aChrom);
end;

```

```

function CalcFitness(var aChrom:TChromTCSC1):double;
begin
    //insert code obj function here
    result:=0;//dObjFunc1.doHitung(aChrom);
end;

```

```
end;
```

```
function CalcFitness(var aChrom:TChromUpfc1):double;
```

```
begin
```

```
  result:=0;//dObjFunc2.doHitung(aChrom);
```

```
end;
```

```
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;  
  var rRand:TRandomu):double;
```

```
begin
```

```
  //insert code obj function here
```

```
  result:=gObjFunc.doMutasi(rSfun,rSmin,rBeta,rValue,rRand);
```

```
end;
```

```
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;  
  const ip:integer;  
  var rRand:TRandomu):double;
```

```
begin
```

```
  //insert code obj function here
```

```
  result:=0;//gObjFunc.doMutasi(rSfun,rSmin,rBeta,rValue,rRand);
```

```
end;
```

```
function MutasiEvoPro(const rSfun,rSmin,rBeta,rValue:double;  
  const ip,jp:integer;  
  var rRand:TRandomu):double;
```

```
begin
```

```
  //insert code obj function here
```

```
  result:=0;
```

```
end;
```

```
end.
```

```
unit uLoadflow;
```

```
interface
```

```
uses uUtils,uComplex,SysUtils;
```

```
type
```

```
  TParam=(psPu,psOhm);
```

```
  TGeneral=record
```

```
    MaxIterasi,Iterasi:integer;
```

```
Vbase,VKonst,Pbase,PKonst,Zbase,Ibase,Toleransi:double;  
Param:TParam;  
sumGen,sumLoad,sumLoss:TComplex;  
end;
```

```
ar gGeneral:TGeneral;  
gLengthBranch:dArr1;  
gBus,gBranch,gGen:dArr2;
```

```
procedure DecodeData(const rBus,rBranch:dArr2;  
var rGeneral:TGeneral;  
var rNbus,rNsal:integer;  
var rE,rF,rPg,rQg,rPL,rQL,rCap:dArr1;  
var rTyp:iArr1;  
var rR,rX,rLc,rTr,rTu,rSu:dArr2);
```

```
function FindSumGen(const rNbus:integer;const rTyp:iArr1):integer;
```

```
procedure Admitansi(const rNbus:integer;  
const rR,rX,rLc,rTr,rTu,rSu:dArr2;  
const rCap:dArr1;  
var rG,rB:dArr2);
```

```
procedure RecToPolar(const rRecRe,rRecIm:dArr1;  
var rPolRe,rPolIm:dArr1);
```

```
procedure PolarToRec(const rPolRe,rPolIm:dArr1;  
var rRecRe,rRecIm:dArr1);
```

```
procedure AliranDaya(const rNbus:integer;const rE,rF:dArr1;  
const rG,rB,rLc:dArr2;  
var rAlirP,rAlirQ:dArr2);
```

```
procedure DayaGen(const rNbus:integer;  
const rE,rF,rPL,rQL:dArr1;  
const rG,rB:dArr2;const rTyp:iArr1;  
var rQg:dArr1);
```

```
procedure DayaSlack(const rNbus:integer;  
const rAlirP,rAlirQ:dArr2;  
const rTyp:iArr1;  
const rPL,rQL:dArr1;  
var rPg,rQg:dArr1);
```

```
procedure ArusBranch(const rNbus:integer;  
const rE,rF:dArr1;  
const rLc,rG,rB:dArr2;  
var rArusRe,rArusIm:dArr2);
```

```
procedure Summary(const rNbus:integer;  
const rPg,rQg,rPL,rQL:dArr1;
```



```
    var rSumGen,rSumLoad,rSumLoss:TComplex);  
function MaxArray(const rData:dArr1):double;
```

```
procedure UpdateBusGen(const rNbus:integer;  
    const rE,rF,rPg,rQg:dArr1;  
    const rGeneral:TGeneral;  
    var rBus:dArr2);  
procedure UpdateBranch(const rNsal:integer;  
    const rAlirRe,rAlirIm,rArusRe,rArusIm:dArr2;  
    const rGeneral:TGeneral;  
    var rBranch:dArr2);
```

Implementation

```
procedure DecodeData(const rBus,rBranch:dArr2;  
    var rGeneral:TGeneral;  
    var rNbus,rNsal:integer;  
    var rE,rF,rPg,rQg,rPL,rQL,rCap:dArr1;  
    var rTyp:iArr1;  
    var rR,rX,rLc,rTr,rTu,rSu:dArr2);  
var i,j,dari,ke:integer;  
begin  
    rGeneral.Zbase:=sqr(rGeneral.Vbase*rGeneral.VKonst)/  
        (rGeneral.Pbase*rGeneral.PKonst);  
    rGeneral.Ibase:=rGeneral.Vbase*rGeneral.VKonst/rGeneral.Zbase;  
    rNbus:=high(rBus)+1;  
    rNsal:=high(rBranch)+1;  
    SetLength(rE,rNbus);  
    SetLength(rF,rNbus);  
    SetLength(rPg,rNbus);  
    SetLength(rQg,rNbus);  
    SetLength(rPL,rNbus);  
    SetLength(rQL,rNbus);  
    SetLength(rCap,rNbus);  
    SetLength(rTyp,rNbus);  
    SetLength(rR,rNbus,rNbus);  
    SetLength(rX,rNbus,rNbus);  
    SetLength(rLc,rNbus,rNbus);  
    SetLength(rTr,rNbus,rNbus);  
    SetLength(rTu,rNbus,rNbus);  
    SetLength(rSu,rNbus,rNbus);  
    for i:=0 to rNbus-1 do  
    begin
```

```

rE[i]:=rBus[i,0];
rF[i]:=rBus[i,1];
rPg[i]:=rBus[i,2]/rGeneral.Pbase;
rQg[i]:=rBus[i,3]/rGeneral.Pbase;
rPL[i]:=rBus[i,4]/rGeneral.Pbase;
rQL[i]:=rBus[i,5]/rGeneral.Pbase;
if rGeneral.Param=psPu then
begin
  rCap[i]:=rBus[i,6];
end
else
begin
  rCap[i]:=rBus[i,6]/rGeneral.Zbase;
end;
rTyp[i]:=round(rBus[i,7]);
for j:=0 to rNbus-1 do
begin
  rR[i,j]:=0.0;
  rX[i,j]:=0.0;
  rLc[i,j]:=0.0;
  rTr[i,j]:=0.0;
  rTu[i,j]:=0.0;
  rSu[i,j]:=0.0;
end;
end;
for i:=0 to rNsal-1 do
begin
  dari:=round(rBranch[i,0])-1;
  ke:=round(rBranch[i,1])-1;
  if rGeneral.Param=psPu then
  begin
    rR[dari,ke]:=rBranch[i,2];
    rX[dari,ke]:=rBranch[i,3];
    rLc[dari,ke]:=rBranch[i,4];
    rR[ke,dari]:=rBranch[i,2];
    rX[ke,dari]:=rBranch[i,3];
    rLc[ke,dari]:=rBranch[i,4];
  end
  else if rGeneral.Param=psOhm then
  begin
    rR[dari,ke]:=rBranch[i,2]/rGeneral.Zbase;
    rX[dari,ke]:=rBranch[i,3]/rGeneral.Zbase;
    rLc[dari,ke]:=rBranch[i,4]/rGeneral.Zbase;
  end
end;

```

```

rR[ke,dari]:=rBranch[i,2]/rGeneral.Zbase;
rX[ke,dari]:=rBranch[i,3]/rGeneral.Zbase;
rLc[ke,dari]:=rBranch[i,4]/rGeneral.Zbase;
end;
rTr[dari,ke]:=rBranch[i,5];
rTu[dari,ke]:=rBranch[i,6];
rSu[dari,ke]:=rBranch[i,7];
end;
end;

```

```

function FindSumGen(const rNbus:integer;
  const rTyp:iArr1):integer;
var i:integer;
begin
  result:=0;
  for i:=0 to rNbus-1 do
  begin
    if rTyp[i]=2 then
    begin
      inc(result);
    end;
  end;
end;
end;

```

```

procedure Admitansi(const rNbus:integer;
  const rR,rX,rLc,rTr,rTu,rSu:dArr2;
  const rCap:dArr1;
  var rG,rB:dArr2);
var i,j,k:integer;
  a,b:double;
  Cr,Ci:dArr2;
  sum,Ca,Za:TComplex;
begin
  SetLength(Cr,rNbus,rNbus);
  SetLength(Ci,rNbus,rNbus);
  sum:=TComplex.Create(1.0,0.0);
  for i:=0 to rNbus-1 do
  begin
    for j:=0 to rNbus-1 do
    begin
      if rX[i,j]<>0 then
      begin
        Za:=TComplex.Create(rR[i,j],rX[i,j]);

```

```

rG[i,i]:=rG[i,i]-Cr[i,j];
rB[i,i]:=rB[i,i]-Ci[i,j]-rLc[i,j];
rG[i,i]:=rG[i,i]+Cr[i,j]/sqr(rTr[i,j]);
rB[i,i]:=rB[i,i]+Ci[i,j]/sqr(rTr[i,j]);
rG[j,j]:=rG[j,j]-Cr[i,j];
rB[j,j]:=rB[j,j]-Ci[i,j]-rLc[i,j];
rG[j,j]:=rG[j,j]+Cr[i,j];
rB[j,j]:=rB[j,j]+Ci[i,j];
rG[i,j]:=-1*Cr[i,j]/rTr[i,j];
rB[i,j]:=-1*Ci[i,j]/rTr[i,j];
rG[j,i]:=rG[i,j];
rB[j,i]:=rB[i,j];
{CLc:=TComplex.Create(0.0,aLc[i,j]);
result[i,i]:=result[i,i]-Cx[i,j]-CLc;
result[i,i]:=result[i,i]+Cx[i,j]/sqr(at);
result[j,j]:=result[j,j]-Cx[i,j]-CLc;
result[j,j]:=result[j,j]+Cx[i,j];
result[i,j]:=-Cx[i,j]/at;
result[j,i]:=result[i,j];
CLc.Free;}
end;
end;
end;
for i:=0 to rNbus-1 do
begin
for j:=0 to rNbus-1 do
begin
if rTu[i,j]<>0 then
begin
a:=rTu[i,j]*cos(rSu[i,j]);
b:=rTu[i,j]*sin(rSu[i,j]);
rG[i,i]:=rG[i,i]-Cr[i,j];
rB[i,i]:=rB[i,i]-Ci[i,j]-rLc[i,j];
rG[i,i]:=rG[i,i]+Cr[i,j]/(sqr(a)+sqr(b));
rB[i,i]:=rB[i,i]+Ci[i,j]/(sqr(a)+sqr(b))+rLc[i,j];
Za:=TComplex.Create(Cr[i,j],Ci[i,j]);
Za:=Za.Negative;
Ca:=TComplex.Create(a,b);
sum:=Za.Divide(Ca);
rG[j,i]:=sum.Real;
rB[j,i]:=sum.Imag;
sum.Free;
Ca:=Ca.Conj;

```

```

sum:=Za.Divide(Ca);
rG[i,j]:=sum.Real;
rB[i,j]:=sum.Imag;
sum.Free;
Ca.Free;
Za.Free;
{sUpfc:=TComplex.Create(aUpfc[i,j].tap*cos(aUpfc[i,j].sudut),
    aUpfc[i,j].tap*sin(aUpfc[i,j].sudut));
CLc:=TComplex.Create(0.0,aLc[i,j]);
result[i,i]:=result[i,i]-Cx[i,j]-CLc;
result[i,i]:=result[i,i]+Cx[i,j]/sqr(sUpfc.Abs)+CLc;
result[i,j]:=-Cx[i,j]/conj(sUpfc);
result[j,i]:=-Cx[i,j]/sUpfc;
CLc.Free;
sUpfc.Free;}
end;
end;
end;
for i:=0 to rNbus-1 do
begin
if rCap[i] > 0 then
begin
rB[i,i]:=rB[i,i]+rCap[i];
end;
end;
end;
end;

```

```

procedure RecToPolar(const rRecRe,rRecIm:dArr1;
    var rPolRe,rPolIm:dArr1);
var i:integer;
begin
if high(rRecRe) > high(rRecIm) then
begin
raise Exception.Create('Dimensi kedua Vector tidak sama!');
end;
SetLength(rPolRe,high(rRecRe)+1);
SetLength(rPolIm,high(rRecRe)+1);
for i:=0 to high(rRecRe) do
begin
rPolRe[i]:=sqrt(sqr(rRecRe[i])+sqr(rRecIm[i]));
rPolIm[i]:=arctan(rRecIm[i]/rRecRe[i]);
end;
end;
end;

```

```

procedure PolarToRec(const rPolRe,rPolIm:dArr1;
    var rRecRe,rRecIm:dArr1);
var i:integer;
begin
    if high(rRecRe) <> high(rRecIm) then
        begin
            raise Exception.Create('Dimensi kedua Vector tidak sama!');
        end;
    SetLength(rRecRe,high(rPolRe)+1);
    SetLength(rRecIm,high(rPolRe)+1);
    for i:=0 to high(rPolRe) do
        begin
            rRecRe[i]:=rPolRe[i]*cos(rPolIm[i]);
            rRecIm[i]:=rPolRe[i]*sin(rPolIm[i]);
        end;
    end;

procedure AliranDaya(const rNbus:integer;const rE,rF:dArr1;
    const rG,rB,rLc:dArr2;
    var rAlirP,rAlirQ:dArr2);
var i,j:integer;
    Vi,Vj,Ya,Lca,Sa,tmp1,tmp2:TComplex;
begin
    SetLength(rAlirP,rNbus,rNbus);
    SetLength(rAlirQ,rNbus,rNbus);
    for i:=0 to rNbus-1 do
        begin
            Vi:=TComplex.Create(rE[i],rF[i]);
            for j:=0 to rNbus-1 do
                begin
                    rAlirP[i,j]:=0.0;
                    rAlirQ[i,j]:=0.0;
                    if j <> i then
                        begin
                            if rB[i,j] <> 0 then
                                begin
                                    Vj:=TComplex.Create(rE[j],rF[j]);
                                    Ya:=TComplex.Create(rG[i,j],rB[i,j]);
                                    Lca:=TComplex.Create(0.0,rLc[i,j]);
                                    tmp1:=Vi.Conj.Multiply(Vi.Subtract(Vj)).Multiply(Ya.Negative);
                                    tmp2:=Vi.Conj.Multiply(Vi).Multiply(Lca);
                                    Sa:=tmp1.Add(tmp2);
                                    rAlirP[i,j]:=Sa.Real;
                                end;
                            end;
                        end;
                end;
        end;
end;

```

```

rAlirQ[i,j]:=-Sa.Imag;
//result[i,j]:=conj(aV[i]*(aV[i]-aV[j]))*(-aY[i,j])+
//      conj(aV[i]*aV[i]*dLc;
//result[i,j]:=conj(result[i,j]);
Sa.Free;
tmp2.Free;
tmp1.Free;
Lca.Free;
Ya.Free;
Vj.Free;
end;
end;
end;
Vi.Free;
end;
end;

procedure DayaGen(const rNbus:integer;
  const rE,rF,rPL,rQL:dArr1;
  const rG,rB:dArr2;const rTyp:iArr1;
  var rQg:dArr1);
var i,j:integer;
  sum:double;
begin
  for i:=0 to rNbus-1 do
  begin
    sum:=0.0;
    if rTyp[i]=2 then
    begin
      for j:=0 to rNbus-1 do
      begin
        //sum:=sum+(Fi*(Ej*Gij+Fj*-Bij)-Ei*(Fj*Gij-Ej*-Bij));
        sum:=sum+(rF[i]*(rE[j]*rG[i,j]+rF[j]*-rB[i,j])-
          rE[i]*(rF[j]*rG[i,j]-rE[j]*-rB[i,j]));
      end;
      rQg[i]:=sum+rQL[i];
    end;
  end;
end;

procedure DayaSlack(const rNbus:integer;
  const rAlirP,rAlirQ:dArr2;
  const rTyp:iArr1;

```

```

    const rPL,rQL:dArr1;
    var rPg,rQg:dArr1);
var i,j:integer;
    sumP,sumQ:double;
begin
for i:=0 to rNbus-1 do
begin
if rTyp[i]=1 then
begin
sumP:=0.0;
sumQ:=0.0;
for j:=0 to rNbus-1 do
begin
if rAlirQ[i,j]<>0 then
begin
sumP:=sumP+rAlirP[i,j];
sumQ:=sumQ+rAlirQ[i,j];
end;
end;
rPg[i]:=sumP+rPL[i];
rQg[i]:=sumQ+rQL[i];
end;
end;
end;
end;

```

```

procedure ArusBranch(const rNbus:integer;
    const rE,rF:dArr1;
    const rLc,rG,rB:dArr2;
    var rArusRe,rArusIm:dArr2);
var i,j:integer;
    Vi,Vj,Ya,xLc,Arus,tmp1,tmp2:TComplex;
begin
SetLength(rArusRe,rNbus,rNbus);
SetLength(rArusIm,rNbus,rNbus);
for i:=0 to rNbus-1 do
begin
Vi:=TComplex.Create(rE[i],rF[i]);
for j:=0 to rNbus-1 do
begin
if rB[i,j]<>0 then
begin
Vj:=TComplex.Create(rE[j],rF[j]);
Ya:=TComplex.Create(rG[i,j],rB[i,j]);

```



```

xLc:=TComplex.Create(0,rLc[i,j]);
tmp1:=Ya.Negative.Multiply(Vi.Subtract(Vj));
tmp2:=Vi.Multiply(xLc);
Arus:=tmp1.Add(tmp2);
rArusRe[i,j]:=Arus.Real;
rArusIm[i,j]:=-Arus.Imag;
Arus.Free;
tmp2.Free;
tmp1.Free;
xLc.Free;
Ya.Free;
Vj.Free;
//result[i,j]:=(aV[i]-aV[j])*(-aY[i,j])+aV[i]*xLc;
//result[i,j]:=Conj(result[i,j]);
end
else
begin
rArusRe[i,j]:=0.0;
rArusIm[i,j]:=0.0;
end;
end;
Vi.Free;
end;
end;

```

```

procedure Summary(const rNbus:integer;
const rPg,rQg,rPL,rQL:dArr1;
var rSumGen,rSumLoad,rSumLoss:TComplex);
var i:integer;
sumPg,sumPL,sumQg,sumQL:double;
begin
sumPg:=0.0;
sumQg:=0.0;
sumPL:=0.0;
sumQL:=0.0;
for i:=0 to rNbus-1 do
begin
sumPg:=sumPg+rPg[i];
sumQg:=sumQg+rQg[i];
sumPL:=sumPL+rPL[i];
sumQL:=sumQL+rQL[i];
end;
rSumGen:=TComplex.Create(sumPg,sumQg);

```

```
rSumLoad:=TComplex.Create(sumPL,sumQL);  
rSumLoss:=rSumGen.Subtract(rSumLoad);  
end;
```

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

```
procedure UpdateBusGen(const rNbus:integer;  
const rE,rF,rPg,rQg:dArr1;  
const rGeneral:TGeneral;  
var rBus:dArr2);  
var i:integer;  
begin  
for i:=0 to rNbus-1 do  
begin  
rBus[i,0]:=rE[i];  
rBus[i,1]:=rF[i];  
rBus[i,2]:=rPg[i]*rGeneral.Pbase;  
rBus[i,3]:=rQg[i]*rGeneral.Pbase;  
end;  
end;
```

```
procedure UpdateBranch(const rNsal:integer;  
const rAlirRe,rAlirIm,rArusRe,rArusIm:dArr2;  
const rGeneral:TGeneral;  
var rBranch:dArr2);  
var i,dari,ke:integer;  
begin  
for i:=0 to rNsal-1 do  
begin  
dari:=round(rBranch[i,0])-1;  
ke:=round(rBranch[i,1])-1;  
rBranch[i,9] :=rAlirRe[dari,ke]*rGeneral.Pbase;
```

```

rBranch[i,10]:=rAlirIm[dari,ke]*rGeneral.Pbase;
rBranch[i,11]:=rAlirRe[ke,dari]*rGeneral.Pbase;
rBranch[i,12]:=rAlirIm[ke,dari]*rGeneral.Pbase;
rBranch[i,13]:=rArusRe[dari,ke]*rGeneral.Ibase;
rBranch[i,14]:=rArusIm[dari,ke]*rGeneral.Ibase;
rBranch[i,15]:=rArusRe[ke,dari]*rGeneral.Ibase;
rBranch[i,16]:=rArusIm[ke,dari]*rGeneral.Ibase;
end;
end;

end.

unit uObjFunc;

interface

uses uUtils,uLoadflow,uNewtonPolar, SysUtils,uCable,uRandom;

type
TObjFunc=class
private
  FNCable,FNbus,FNBranch:integer;
  FLamdaV,FLamdaS,FKp,FKe,FLsf,FAlpha:double;
  FBatasV:TBatas;
  FCable:TCableArr;
  FLBranch:dArr1;
  function getBatasV:TBatas;
  procedure setBatasV(const rBatasV:TBatas);
  procedure UbahParamBranch(const rNoCable,rNoBranch:integer;
    var rR,rX:double);
  function FindTypCable(const rChrom:dArr1):iArr1;
  function DecodeChrom(const rTypCable:iArr1;
    const rBranch:dArr2):dArr2;
  function FindConstV(const rBus:dArr2):integer;
  function FindConstS(const rTypCable:iArr1,const
    rBranch:dArr2):double;
  function CalcTotalCost(const rTypCable:iArr1;
    const rBranch:dArr2):double;
  function DeviasiV(const rBus:dArr2):double;
public
  constructor Create;overload;
  constructor Create(const rCable:TCableArr;

```

```

    const rLBranch:dArr1;
    const rBatasV:TBatas;
    const rLamdaV,rLamdaS,rKp,rKe,rLsf,rAlpha:double);overload;
function doHitung(const rChrom:dArr1):double;
procedure doHitungAkhir(const rChrom:dArr1;
    var rLBranch:dArr2;
    var rTypCable:iArr1;
    var rWjk,rCost:double);
function doHitungSebelum:double;
function doMutasi(const rSfun,rSmin,rBeta,rValue:double;
    var rRand:TRandomu):double;
property NCable:integer read FNCable write FNCable;
property BatasV:TBatas read getBatasV write setBatasV;
property LamdaV:double read FLamdaV write FLamdaV;
property LamdaS:double read FLamdaS write FLamdaS;
end;

```

```
var gObjFunc:TObjFunc;
```

```
implementation
```

```
{ TObjFunc }
```

```
//constructor
```

```
constructor TObjFunc.Create;
```

```
begin
```

```
    inherited Create;
```

```
    FBatasV.min:=0.95;
```

```
    FBatasV.max:=1.05;
```

```
    FNCable:=1;
```

```
    FLamdaV:=1000;
```

```
    FLamdaS:=1000;
```

```
end;
```

```
constructor TObjFunc.Create(const rCable:TCableArr;
```

```
    const rLBranch:dArr1;
```

```
    const rBatasV:TBatas;
```

```
    const rLamdaV,rLamdaS,rKp,rKe,rLsf,rAlpha:double);
```

```
var i:integer;
```

```
begin
```

```
    inherited Create;
```

```
    FNCable:=high(rCable)+1;
```

```
    SetLength(FCable,FNCable);
```

```

for i:=0 to FNCable-1 do
begin
  FCable[i].Nama:=rCable[i].Nama;
  FCable[i].R:=rCable[i].R;
  FCable[i].X:=rCable[i].X;
  FCable[i].Harga:=rCable[i].Harga;
  FCable[i].Cap:=rCable[i].Cap;
end;
FNbus:=high(gBus)+1;
FNBranch:=high(rLBranch)+1;
SetLength(FLBranch,FNBranch);
for i:=0 to FNBranch-1 do
begin
  FLBranch[i]:=rLBranch[i];
end;
FBatasV.min:=rBatasV.min;
FBatasV.max:=rBatasV.max;
FLamdaV:=rLamdaV;
FLamdaS:=rLamdaS;
FKp:=rKp;
FKe:=rKe;
FLsf:=rLsf;
FAlpha:=rAlpha;
end;

//data accessing

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

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

//data processing
function TObjFunc.FindTypCable(const rChrom:dArr1):iArr1;
var i,Ncek:integer;
begin

```

```

    const rPL,rQL:dArr1;
    var rPg,rQg:dArr1);
ar i,j:integer;
  sumP,sumQ:double;
egin
for i:=0 to rNbus-1 do
begin
  if rTyp[i]=1 then
  begin
    sumP:=0.0;
    sumQ:=0.0;
    for j:=0 to rNbus-1 do
    begin
      if rAlirQ[i,j]<>0 then
      begin
        sumP:=sumP+rAlirP[i,j];
        sumQ:=sumQ+rAlirQ[i,j];
      end;
    end;
    rPg[i]:=sumP+rPL[i];
    rQg[i]:=sumQ+rQL[i];
  end;
end;
nd;

```

```

rocedure ArusBranch(const rNbus:integer;
  const rE,rF:dArr1;
  const rLc,rG,rB:dArr2;
  var rArusRe,rArusIm:dArr2);
ar i,j:integer;
  Vi,Vj,Ya,xLc,Arus,tmp1,tmp2:TComplex;
egin
SetLength(rArusRe,rNbus,rNbus);
SetLength(rArusIm,rNbus,rNbus);
for i:=0 to rNbus-1 do
begin
  Vi:=TComplex.Create(rE[i],rF[i]);
  for j:=0 to rNbus-1 do
  begin
    if rB[i,j]<>0 then
    begin
      Vj:=TComplex.Create(rE[j],rF[j]);
      Ya:=TComplex.Create(rG[i,j],rB[i,j]);
    end;
  end;
end;

```

```
    dTyp,Typ:integer;
begin
  Typ:=round(GetBatasToReal(rValue,0,FNCable-1));
  tho:=rBeta*(FNCable-1)*rSfun/rSmin;
  tho:=sqr(tho);
  dTyp:=round(rRand.NextGaussian(0,tho));
  Typ:=Typ+dTyp;
  if Typ>FNCable-1 then Typ:=FNCable-1;
  if Typ<0 then Typ:=0;
  result:=GetRealToBatas(Typ,0,FNCable-1);
end;

end.
```



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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : AGUS HARIYANTO
N.I.M. : 00.12.028
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Energi Listrik
Judul Skripsi : OPTIMALISASI PEMILIHAN KONDUKTOR
PADA SALURAN DISTRIBUSI PRIMER 20 kV
PENYULANG PUJON MENGGUNAKAN
METODE *EVOLUTIONARY PROGRAMMING*

Dipertabakan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)

Hari : Jum'at
Tanggal : 01 April 2005
Dengan Nilai : 73.25 (B) *g*



Panitia Ujian Skripsi

(Ir. Mochtar Asroni, MSME)
Ketua

(Ir. F. Yudi Limpraptono, MT.)
Sekretaris

Anggota Penguji

(Ir. Yusuf Ismail Nakhoda, MT.)
Penguji Pertama

(Ir. I Made Wartana, MT.)
Penguji Kedua