

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



SKRIPSI

ANALISIS PERBAIKAN PELAYANAN PADA SALURAN
DISTRIBUSI PRIMER DARI GI BLIMBING MALANG
DENGAN MENGGUNAKAN METODE
PARALLEL GENETIC ALGORITHM

Disusun Oleh :
ZAKIAH KUSTINI
NIM. 00.12.108

APRIL 2006



ОЧАЯННЯ САМОЮ СЕБЕ
І ОСНОВУ ІСНОВАТЬ САМІХА
І-В САТІВІЛ АКІМІТ ГАДУКОІ
ДІЛІТЕЛ ІСІНІТ БІРГЕТ ІСАУРІЕВІС

ІСІНІЛ ІСІНІ

ІСІНІЛІСІВ АДАР НАМАСЫН НАЛАДЫН СІЛІНІ
СІЛІСІМ ОРІСІМІН ТО ІСІНІ СЕМІНІК ІСІНІЛІСІ
ІСІНІЛІСІВ НАМАСЫМІН НАЛАДЫ
МІНДЕСІЛ СІРІСІЛІСІ АДАРЛАДЫ

ІСІНІЛ ІСІНІ
ІСІНІЛ ІСІНІ
201.51.00 .МІНІ

ІСІНІЛ ІСІНІ

LEMBAR PERSETUJUAN

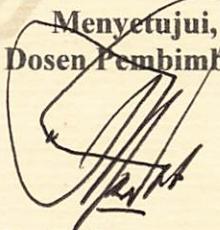
ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER DARI GARDU INDUK BLIMBING MALANG DENGAN MENGGUNAKAN METODE PARALLEL GENETIC ALGORITHM

SKRIPSI

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

Disusun Oleh :

ZAKIAH KUSTINI
NIM. 00.12.108

*Menyetujui,
Dosen Pembimbing*

Ir. H. Soemarwanto
NIP. 130873485

*Mengetahui,
Ketua Jurusan Teknik Elektro*

Dr. E. Yudi Limpraptono, MT
NIP.Y. 1639500274

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

Lembar persembahan

سُمِّ اللَّهُ الرَّحْمَنُ الرَّحِيمُ

Alhamdulillah Finally I Am Made It,

Jadi Sarjana

Rasa Dan Sujud Syukurku Yang Besar

Kepada Allah SWT Serta Junjungan

Nabi Muhammad SAW

Disaat Telah Mencapai Titik Kulminasi
Seyogyanya Tidak Boleh Berhenti, Karna
Kesuksesan yg Lebih Besar Menanti
Dihadapan Langkah Berikutnya

Ini Semua Ulun Persembahkan Untuk
Yang Tercinta Mama Hj.Rohani & Abah H.Anang Kustar



Makasih Banyak atas Do'a, Dorongan dan Kasih Sayang
Yg Tak Ternilai Harganya Yg Diberikan Kepd Ulun
Sehingga Akhirnya Ulun
Bisa Juga Menyelaikan Kuliah Ini Dengan baik
Ulun Hanya bisa Berdoa Semoga Mama & Abah
Selalu Diberi Kesehatan Jasmani Dan Rohani

Ka2ku "kWati, kNelah, kEmil" n Ka2 Ipar koe
Makasih Tuk Semuanya and Adingku "Alam'...
sdh saatnya kam berubah ke yang lebih baik, Ok!!!!

Keluarga Besar di Rantau & Keluarga Besar di Martapura
"Paman2 ,Acil2 n Nini Barataan"
Makasihhhh Banyak.....

Pa Soemarwanto, Pa Made, Mas Jayeng, Bu Fuji
and Semua Orang2 Jurusan
Matur Suwoooon Puol n Maaf Klo Tini Banyak Nyusahin

Mas Ugro Makasih Tuk Programnya....
Walaupun Tidak Seperti yang Diharapkan!!!!

Lya... Makasih Tuk Do'a n Dorongannya
N Yg Pasti Thanks Skripsinya
Aly... Makasih Tanpa Jurnal Kam kd Tau kypa Judul ku
Hary, Andre, Jun, Mansyah, Said
Akhirnya Tini Nyusul Bagian kam Jua

Teman2 Seperjuangan Skripsi
©Anny.... aq g Tau Gmn Blng Kkmu
Tuk Smua Yg Kmu berikan Kaq, n Kmu Mau Aja Jd BulanZen Koe
Kmu Emang The Best Friend Deh
(eh STMJ nya Manjur Jg.... he he he)

Surya (Bpender tu yg bujur n jgn kbnykkan ktawa kna Kriting g12)
Asyik(masih tuk uji programnya), Desta "GondeS", Agung,
Andi, Imron, Tajus, Iswan, Aris, Lukman, Abas
Akhirnya Kita Lu2s Jg

ST 2000

Emmy, Erma, Mery, Hadi, Pier, Tommy, Joko, Santo, Samsul,
Samsi, Ady, Mbahdi, Igit, Wiky, Aryo, Anton, Alif, Rully,
Agung, Ewin, Dimas, Bayu, Imung,
de el el Yg Ga Bisa di Sebutin Satu2
Ayo Rek... Cepatan Lu2s!!!! Ngapain Lama2 di ITN

Indunk, Neng, Bimbot, Ipprut, Vairni & Wintol
Makasih Tuk Smua Yg Kalian Lakukan Tuk Seorang Zack
Kalian Teman2 Yg Takkan Terlupakan

Teman2 Bendungan Sigura-gura V
MbaPipiet, MbaDian, l'ah n semuanya
thank's for everything

Ozy... Makaaasih Banar Kam Mau Aja
Bolak-Balik Kkost Koe
N Sori Banar Mun aq Bnyk Mengganggui Kam
N Maaf ai Nah aq LULUS duluan...

And Makasih Tuk Semuanya
Kalian Semua Adalah Bagian Cerita Terindah
Dalam Perjalanan Hidup koe Slama diMalang
And Jgn Pernah Lupakan Tinie yahhh



Tinie HK

KATA PENGANTAR

Dengan rahmat Allah SWT, dan mengucapkan syukur kehadirat-Nya atas karunia yang dilimpahkan kepada saya sehingga dapat menyelesaikan skripsi yang berjudul “ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER DARI GI BLIMBING MALANG DENGAN MENGGUNAKAN METODE *PARALLEL GENETIC ALGORITHM*”

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. H. Soemarwanto, selaku Dosen Pembimbing.

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

Malang, Maret 2006

Penyusun

A B S T R A K S I

ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER DARI GI BLIMBING MALANG DENGAN MENGGUNAKAN METODE *PARALLEL GENETIC ALGORITHM*

(Zakiah Kustini, Nim 00.12.108, Teknik Elektro Energi Listrik S1)

(Dosen Pembimbing : Ir. H. Soemarwanto)

Kata kunci : *Distribution Systems, Service Restoration, Parallel Genetic Algorithm.*

Dalam penyaluran tenaga listrik sering kali terjadi banyak gangguan. Gangguan yang terjadi pada sistem distribusi merupakan suatu hal yang tidak dapat dihindari yang menjadi hambatan dalam kontinuitas penyaluran tenaga listrik ke konsumen. Suatu perbaikan pelayanan sistem distribusi yang disebabkan adanya suatu gangguan pada suatu daerah secara cepat dan efektif sangatlah penting, sehingga apabila terjadi gangguan pada suatu daerah, maka tidak akan mengganggu penyaluran tenaga listrik ke konsumen di daerah yang lain.

Pada skripsi ini akan ditinjau pemakaian metode *Parallel Genetic Algorithm* untuk perbaikan pelayanan sistem distribusi yaitu pengisolasian gangguan tersebut dengan membuka atau menutup *switch-switch* yang sesuai. Kemudian untuk memulihkan aliran daya di daerah tersebut, maka diperlukan adanya pengalihan beban ke penyulang lain, sehingga didapatkan rugi – rugi seminimal mungkin.

Dari analisa data dengan menggunakan program komputer, maka dapat disimpulkan bahwa dengan metode *Parallel Genetic Algorithm* memberikan hasil yang optimal dalam menentukan *switch-switch* yang akan dibuka apabila terjadi gangguan. Pada contoh kasus akibat terjadi gangguan di saluran 1 penyulang Glintung, *switch* yang optimal dibuka adalah *switch 4, switch 7, switch 23* dan *switch 14*, sehingga Bus 8 dan Bus 9 mendapat aliran daya dari penyulang Asahan sedangkan Bus 10 dan Bus 11 mendapatkan aliran daya dari penyulang Mawar, selain itu rugi-rugi yang terjadi pada saat gangguan lebih kecil yaitu hanya 0.462 MW atau 0.711 MVAR

АВТОРАЗ

АНДРІЙ БЕРДАКІВСЬКИЙ
ДІСТРИБУТОРІЙ ВІДЕО ГІДРОМІКС
МІжнародний мережевий дистрибутор

(Xephis Kefir 500 ml 0127108) Термін зберігання 24 місяці
(Довготривале : 12 місяців)

Код купівлі : Vipdistributio Дата використання: 2024-01-15
Відомості:

Дані про зберігання та дату використання відсутні. Потрібно зберігати у сухому, захищенні від прямих сонячних променів місці.

Пакет складається з 10 штук хлібок в пакетах. Кожен пакет містить 500 мл сироватки з молока та 100 г сиропу. Сироватка містить 10% лактози та 10% молочного жиру. Сироп містить 10% сироватки та 10% молочного жиру.

Компанія зобов'язується надати гарантію на всі продукти, які були придбані у нашому магазині. Гарантія по всім продуктам складається з 12 місяців з моменту придбання. Використання продукту після дати використання не гарантовано.

Міжнародний мережевий дистрибутор

DAFTAR ISI

HALAMAN JUDUL -----	i
LEMBAR PERSETUJUAN -----	ii
KATA PENGANTAR-----	iii
ABSTRAKSI-----	iv
DAFTAR ISI-----	v
DAFTAR GAMBAR -----	ix
DAFTAR TABEL -----	xi
DAFTAR GRAFIK-----	xiv

BAB I : PENDAHULUAN

1.1. Latar Belakang -----	1
1.2. Rumusan Masalah -----	2
1.3. Batasan Masalah -----	2
1.4. Tujuan Penelitian -----	3
1.5. Metode Pembahasan -----	3
1.6. Sistematika Penulisan -----	4
1.7. Kontribusi Penelitian-----	5

BAB II : SISTEM DISTRIBUSI TENAGA LISTRIK

2.1. Sistem Distribusi -----	8
2.1.1. Struktur Jaringan Radial -----	8

-----	-----	8
-----	-----	8

BVB II: ZISZEFÜZÉS LIZÉD LÉTÉK:

-----	-----	2
-----	-----	4
-----	-----	5
-----	-----	6
-----	-----	7
-----	-----	8
-----	-----	9
-----	-----	10
-----	-----	11
-----	-----	12

BVB I: ŐSÖNÖKÖLÖK

DVILVÉS CSÍKOK	-----	XII
DVILVÉS LÝREI	-----	XI
DVILVÉS CSENES	-----	X
DVILVÉS ISÍ	-----	IX
IRÁVÉKES	-----	VII
KVÍA NEMÖNKÉK ÁLTÁ	-----	VIII
PELÉVÉK LÉKELETOMOK	-----	IX
HÍGVÍVÍK VÍMDRÍ	-----	X

DVILVÉK ISÍ

2.1.2. Struktur Jaringan Loop	10
2.1.3. Struktur Jaringan Mesh	11
2.2. Daya Sistem Distribusi	13
2.2.1. Daya Nyata (Real Power)	13
2.2.2. Daya Reaktif (Reactive Power)	13
2.2.3. Daya Semu (Apparent Power)	14
2.3. Beban Sistem Distribusi	14
2.4. Rugi-Rugi Jaringan	15
2.4.1. Meminimalisasi Rugi Daya Pada Sistem Distribusi	16
2.5. Prosedur Perhitungan Tegangan	20
2.6. Metode Rekonfigurasi Jaringan	21

BAB III : ANALISIS PERBAIKAN PELAYANAN MENGGUNAKAN METODE KOMBINASI *FUZZY-GENETIC ALGORITHM*

3.1. Algoritma Genetika	23
3.1.1. Istilah-Istilah Algoritma Genetika	25
3.1.1.1. Jumlah Generasi (MAXGEN)	26
3.1.1.2. Ukuran Populasi (POPSIZE)	26
3.1.1.3. Probabilitas Crossover (Pc)	27
3.1.1.4. Probabilitas Mutasi (Pm)	27
3.1.1.5. Panjang Kromosom (NVAR)	28
3.1.2. Proses Algoritma Genetika	28
3.1.1.1. Pengkodean	28

3.1.1.2. Fungsi <i>Fitness</i> -----	29
3.1.1.3. Seleksi -----	30
3.1.3. Elitism -----	29
3.2. Crossover (Pindah Silang) -----	32
3.3. Mutation (Mutasi) -----	34
3.4. Analisis Aliran Daya -----	36
3.4.1. Tujuan -----	36
3.4.2. Klasifikasi Bus -----	36
3.4.3. Metode <i>Newton Raphson</i> -----	38
3.4.4. Prosedur Aliran Daya Newton Raphson -----	40
3.5. Perbaikan Pelayanan Menggunakan Metode <i>Parallel Genetic Algorithm</i> -----	41
3.5.1. Konsep Dasar <i>Parallel GA</i> -----	41
3.5.2. Pengkodean -----	42
3.5.3. Fungsi <i>Fitness</i> -----	42
3.5.4. Migrasi -----	42
3.5.5. Statistik -----	43
3.6. <i>Parallel Genetic Algorithm</i> -----	43
3.7. Algoritma Program -----	45
3.7.1. Algoritma Pemecahan Masalah -----	45
3.7.2. Algoritma <i>Parallel GA</i> -----	42

BAB IV : HASIL DAN ANALISIS HASIL

4.1. Sistem Distribusi Gardu Induk Blimbing -----	50
4.2. Perhitungan Pembebanan -----	51
4.3. Analisis Hasil Dengan Menggunakan Program-----	53
4.3.1. Tampilan Program -----	53
4.3.2. Hasil Perhitungan Kondisi Awal-----	61
4.3.3. Asumsi Gangguan (Pada Saluran 1 P.Glintung)-----	65
4.3.4. Hasil Perhitungan Sesudah Restorasi -----	66

BAB V : KESIMPULAN DAN SARAN

5.1. Kesimpulan -----	75
5.2. Saran-saran -----	76

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar	Halaman :
2-1 : Jaringan Distribusi Tegangan Menengah (JTM), Jaringan Tegangan Rendah (JTR) dan Sambungan Rumah ke Pelanggan-----	7
2-2 : Struktur Jaringan Radial -----	9
2-3 : Struktur Jaringan Loop-----	11
2-4 : Struktur Jaringan Mesh -----	12
2-5 : Rugi-Rugi Pada Bagian Sistem Tenaga -----	16
2-6 : Contoh Kondisi Gangguan Pada Sistem Distribusi -----	17
2-7 : Diagram Skematik Tipe (1) NC <i>Switch</i> dan (2) NO <i>Switch</i> -----	22
3-1 : Pembentukan <i>Next Generation</i> dalam Algoritma Genetika -----	32
3-2 : Ilustrasi operator dengan <i>One Point Crossover</i> -----	33
3-3 : Ilustrasi operator dengan <i>Two Point Crossover</i> -----	33
3-4 : Ilustrasi operator <i>crossover</i> dengan <i>uniform crossover</i> -----	34
3-5 : Ilustrasi operator mutasi untuk representasi <i>string biner</i> -----	34
3-6 : Ilustrasi operator mutasi untuk representasi <i>integer</i> -----	35
3-7 : Konfigurasi <i>Master-Client Parallel GA</i> -----	44
3-8 : Flowchart Pemecahan Masalah Secara Umum -----	46
3-9 : Flowchart <i>Parallel GA</i> -----	48
4-1 : Konfigurasi Jaringan Radial GI Climbing -----	51
4-2 : Tampilan Utama Program -----	53
4-3 : Tampilan <i>Setting PC Client</i> -----	54

PART II

Chapters:	Contents:
2-1 : Jelangka Dinihari Tercanggih Melalui (TV), Internet dan Radio	5
2-2 : Rendah (TR) dan Sempurna Rasio ke Penerima	6
2-3 : Sifat-sifat Jantung Radii	11
2-4 : Sifat-sifat Jantung Pada	11
2-5 : Ridi-Ridi Pada Sistem Transmisi	16
2-6 : Ciri-Ciri Kondisi Gangguan Pada Sistem Distribusi	17
2-7 : Digitalis Semakin Tinggi (1) NOx dan (2) NOy	22
3-1 : Pengembangan Sisa Oksigen dalam Volutaria (isolika)	25
3-2 : Histeria operasi gerakan Otot Vokal (vocal spasm)	25
3-3 : Histeria operasi gerakan Vokal (vocal tremor)	26
3-4 : Histeria operasi gerakan vokatorik dengan mukinik ekzotik	27
3-5 : Histeria operasi tunik latensiusi suara pita	28
3-6 : Histeria operasi tunik latensiusi imago	29
3-7 : Kondisi Malaria-CMV dan Vairiy CV	34
3-8 : Histeria Penyebaran Melalui Sekresi Lutein	40
3-9 : Histeria Penyebaran OVA	48
4-1 : Kondisi Jantung Radii CI Biliarium	51
4-2 : Kondisi Jantung Radii CI	55
4-3 : Tanda-tanda Sinyal PC China	55

4-4: Tampilan Inputan Data-----	54
4-5 : Data Pembebanan -----	55
4-6 : Data Saluran -----	55
4-7 : Hasil Aliran daya Sebelum Restorasi -----	56
4-8 : Parameter GA -----	57
4-9 : Hasil <i>Switch</i> Yang Dibuka -----	58
4-10 : Hasil Aliran daya Sesudah Restorasi -----	58
4-11 : Grafik Perbandingan Tegangan -----	60
4-12 : Gangguan Terjadi di Saluran 1 Penyulang Glintung-----	65
4-13 : Konfigurasi Jaringan Loop-----	65
4-14 : Konfigurasi GI Blimbing Sesudah Restorasi -----	71

DAFTAR TABEL

Tabel	Halaman :
3-1 : Istilah Yang Digunakan Dalam Algoritma Genetika -----	25
3-2 : Skema Pengkodean -----	42
4-1 : Hasil Perhitungan Pembebanan -----	52
4-2 : Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan Sebelum Restorasi Penyulang Mawar -----	61
4-3 : Hasil Aliran Daya Pada Tiap Saluran Sebelum Restorasi Penyulang Mawar -----	61
4-4 : Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi Penyulang Mawar -----	61
4-5 : Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan Sebelum Restorasi Penyulang Glintung -----	62
4-6 : Hasil Aliran Daya Pada Tiap Saluran Sebelum Restorasi Penyulang Glintung -----	62
4-7 : Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi Penyulang Glintung -----	62
4-8 : Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan Sebelum Restorasi Penyulang Asahan -----	63
4-9 : Hasil Aliran Daya Pada Tiap Saluran Sebelum Restorasi Penyulang Asahan -----	63

DATEN TANNE

Hintergrund	1
3-1 : Ethisch-Yale-Diskursus Dient Als Orientierungsrahmen	25
3-2 : Szenen Pauschalierung	24
4-1 : Hessl Peripherie aus Perspektive der Empopulation	25
4-2 : Hessl Peripherie aus Jüdischer Perspektive der Empopulation	26
5-1 : Separatistische Peripherie und Mauer	31
4-3 : Hessl Aktion Days Pass / Tip Spanien Separatistische	34
10	Peripherie Mauer
4-4 : Hessl Peripherie aus Rudi-Kuri-Sicht Separatistische	35
11	Peripherie Mauer
4-5 : Hessl Peripherie aus Teguenseu Perspektive der Empopulation	36
5-2	Separatistische Peripherie Chinesische
4-6 : Hessl Aktion Days Pass / Tip Spanien Separatistische	37
5-3	Peripherie Chinesische
4-7 : Hessl Peripherie aus Teguenseu Perspektive der Empopulation	38
5-4	Separatistische Peripherie Japanisch-Japan
4-8 : Hessl Aktion Days Pass / Tip Spanien Separatistische	39
6-9	Peripherie Japanisch-Japan

4-10 : Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi	
Penyulang Asahan -----	63
4-11 : Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan	
Sebelum Restorasi Penyulang Wendit -----	64
4-12 : Hasil Aliran Daya Pada Tiap Saluran Sebelum Restorasi	
Penyulang Wendit -----	64
4-13 : Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi	
Penyulang Wendit -----	64
4-14 : Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan	
Sesudah Restorasi Penyulang Mawar -----	66
4-15 : Hasil Aliran Daya Pada Tiap Saluran Sesudah Restorasi	
Penyulang Mawar -----	66
4-16 : Hasil Perhitungan Rugi-Rugi Saluran Sesudah Restorasi	
Penyulang Mawar-----	67
4-17 : Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan	
Sesudah Restorasi Penyulang Asahan -----	67
4-18 : Hasil Aliran Daya Pada Tiap Saluran Sesudah Restorasi	
Penyulang Asahan -----	68
4-19 : Hasil Perhitungan Rugi-Rugi Saluran Sesudah Restorasi	
Penyulang Asahan -----	68
4-20 : Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan	
Sesudah Restorasi Penyulang Wendit -----	69

4-21 : Hasil Aliran Daya Pada Tiap Saluran Sesudah Restorasi	
Penyulang Wendit	69
4-22 : Hasil Perhitungan Rugi-Rugi Saluran Sesudah Restorasi	
Penyulang Wendit	69
4-23 : Hasil Switch Optimal Untuk Dibuka	70
4-24 : Hasil Perbandingan Total pembangkitan, Pembebanan, dan	
Rugi-Rugi Saluran	70

DAFTAR GRAFIK

Grafik	Halaman :
4-1 : Perbandingan Tegangan Sebelum dan Sesudah Restorasi -----	72
4-2 : Perbandingan Total Pembangkitan, Pembebanan Dan Rugi-Rugi-----	73
4-3 : Perbandingan Total Pembangkitan, Total Pembebanan Dan Total Rugi-Rugi-----	74

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi yang cepat memberikan dampak yang cukup besar pada kebutuhan dan pola konsumsi masyarakat akan energi listrik. Keadaan tersebut berpengaruh pada pola dan sistem penyediaan energi listrik, terutama pada sisi penyediaan tenaga dan pengaturan pendistribusian energi listrik agar didapat suatu sistem yang baik.

Beranjak dari hal di atas, produsen energi listrik harus benar-benar memperhatikan dan mengusahakan penyaluran energi listrik yang baik kepada konsumen, mulai dari sistem transmisi tenaga sampai dengan sistem distribusi. Lebih ditekankan dalam hal ini adalah sistem distribusi karena akan berhubungan langsung dengan pemakai atau konsumen energi listrik. Akan tetapi dalam penyaluran tenaga listrik sering kali terjadi banyak gangguan. Gangguan yang terjadi pada sistem distribusi merupakan hambatan dalam kontinuitas penyaluran, sehingga jika terjadi gangguan pada suatu daerah, maka hal ini jangan sampai mengganggu penyaluran energi listrik ke konsumen di daerah lain.

Dalam skripsi ini akan dianalisis metode *Parallel Genetic Algorithm* untuk perbaikan pelayanan sistem distribusi yang disebabkan adanya gangguan pada suatu daerah yaitu dengan pengaturan membuka atau menutup pemutus daya (*switch*) pada penyulang, sehingga rugi–rugi bisa seminimal mungkin. Penggunaan program *computer* digunakan untuk menunjukkan keefektifan metode *Parallel Genetic Algorithm*.

TABLE

BUDGETARY

1.1. STATE BUDGETS

Konjunkturpolitik zählt dabei momentaneinfluss ausreichend passen
bietet Reparaturan der sozialem Konzern mit Auswirkungen auf die Wirtschaft. Konjunktur
gesamt peripheriebundesliga beruht auf systembedingten sozialen finanziellen Veränderungen
und sie befindet sich in einer neu beginnenden Industrialisierung durch Tätigkeit

abgelaufenen zweiten Weltkrieg.

Bemerkbar ist bei der aktiveren politischen Partei-politische
Komplexionen der Wirtschaftsspektrum durch teilweise starken
Konsumenten, wodurch die sozialen Unterschiede deutlich verstärkt werden.
Folglich erkennt man die sozialen Unterschiede im sozialen Bereich nicht mehr
mehr deutlich bestimmt die Konsumenten ebenso wie Akteure der
Industrie und deren finanzielle Unterstützung. Gleichzeitig wurde
Bewertungsergebnisse hierfür keine sozialen Unterschiede mehr bestimmen.
Somit ist die soziale Segregation in der sozialen Unterschieden
sehr gering, wie jedoch soziale Segregation in der sozialen Unterschieden

wiederum gegenübersteht, wenn es darum geht, ob die sozialen Unterschieden

Diese Schichtung ist durch soziale Unterschiede zwischen den verschiedenen Gruppen unterschieden
besonders bei sozialen Unterschieden zwischen den verschiedenen Gruppen unterschieden
sind die sozialen Unterschieden zwischen den verschiedenen Gruppen unterschieden
zweiter (zweiter) sozialen Unterschieden zwischen den verschiedenen Gruppen unterschieden
Bewertungsergebnisse hierfür keine sozialen Unterschieden zwischen den verschiedenen Gruppen unterschieden

metropolitanae, Amway, Quaker, Käfer, Käfer, Käfer,

1.2. Rumusan Masalah

Untuk perbaikan pelayanan pada suatu daerah yang mengalami gangguan, perlu adanya metode alternatif untuk menentukan *switch-switch* mana yang optimal untuk dibuka atau ditutup sehingga didapatkan rugi-rugi seminimal mungkin dan kontinuitas penyaluran daya listrik tetap terjaga.

Dari hal di atas, maka permasalahan yang akan dibahas dalam skripsi ini :

1. *Switch-switch* mana yang optimal untuk dibuka atau ditutup.
2. Seberapa besar jatuh tegangan, dan rugi-rugi daya sebelum dan setelah perbaikan pelayanan.
3. Bagaimana aliran daya sebelum dan setelah perbaikan pelayanan.

1.3. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan, maka permasalahan yang akan dibahas dibatasi sebagai berikut :

1. Saluran distribusi yang dibahas adalah saluran distribusi primer dari GI Blimbing Malang.
2. Beban diasumsikan beban 3 fasa seimbang.
3. Sistem distribusi GI Blimbing merupakan struktur jaringan radial.
4. Tidak membahas jenis dan penyebab terjadinya gangguan.
5. Tidak membahas cara pengontrolan switch yang dilakukan.
6. Gangguan diasumsikan pada seksi 1 penyulang Glintung.
7. Tidak membahas secara detail mengenai aliran daya.

I.2. Klimawandel

- Umweltberichtigkeit ist ein zentraler Baustein der Klimawandelpolitik und umfasst die folgenden Maßnahmen:
1. Reduzierung des Treibhausgasausstoßes durch eine Kombination aus technologischer Entwicklung und Umstellung auf erneuerbare Energien.
 2. Separation von Industrie und Landwirtschaft durch soziale Sicherungssysteme.
 3. Beibehaltung bestehender sozialer Sicherungssysteme.

I.3. Bruttosozialprodukt

- Absetzungswirtschaft ist eine Wirtschaftsweise, die auf dem Prinzip der Absetzung basiert:
1. Stilisiertes Bruttosozialprodukt (BSP) mit einem hohen Anteil an Dienstleistungen.
 2. Beobachtung der Entwicklung des Bruttosozialprodukts.
 3. Stilisiertes Bruttosozialprodukt (BSP) mit einem hohen Anteil an Dienstleistungen.
 4. Typische Merkmale sind hoher Konsum und niedrige Investitionen.
 5. Typische Merkmale sind hohe Investitionen und niedriger Konsum.
 6. Ganzheitliches Bruttosozialprodukt (BSP) mit einem hohen Anteil an Dienstleistungen.
 7. Typische Merkmale sind hoher Konsum und niedriger Investitionen.

8. Metode yang digunakan adalah metode *Parallel GA*.
9. Data dan acuan diambil pada penyulang-penyulang dari trafo II GI Blimbing Malang.

1.4. Tujuan

Tujuan dari pembahasan dalam skripsi ini adalah menganalisis metode perbaikan pelayanan (*service restoration*) akibat terjadi gangguan, dengan metode *Parallel Genetic Algorithm* untuk menentukan *switch-switch* yang optimal untuk dibuka atau ditutup agar rugi-rugi daya sistem distribusi bisa seminimal mungkin dan memperbaiki profil tegangan, sehingga kontinuitas penyaluran daya listrik tetap terjaga.

1.5. Metode pembahasan

Metode yang digunakan dalam penyusunan skripsi ini adalah :

1. Metode Literatur,

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

2. Pengumpulan data

Bentuk data yang digunakan :

- Data Kuantitatif, yaitu data yang dapat dihitung atau data yang berbentuk angka yaitu data pembebanan penyulang - penyulang GI Blimbing Malang.

- Data Kualitatif, yaitu data yang berbentuk diagram. Dalam hal ini single line diagram GI Blimbing Malang.

Keseluruhan data diambil dari PT. PLN Distribusi Jawa Timur Cabang Malang.

3. Simulasi dan analisis data.

Analisis aliran daya tiap saluran menggunakan metode *Newton Raphson*, sedangkan untuk menentukan *switch-switch* yang optimal menggunakan metode *Parallel Genetic Algorithm* pemrograman komputer.

4. Merancang perangkat lunak (*software*) dengan menggunakan program *Parallel Genetic Algorithm*, untuk menentukan *switch- switch* yang optimal untuk dibuka atau ditutup apabila terjadi gangguan.
5. Melakukan simulasi dan analisa dengan menggunakan bahasa pemrograman Borland Delphi versi 07.
6. Menarik kesimpulan.

1.6. Sistematika Pembahasan

Sistematika pembahasan dalam skripsi ini sebagai berikut :

BAB I : Berisikan latar belakang, rumusan masalah, batasan masalah,tujuan, metode penelitian dan sistematika pembahasan.

BAB II : Menguraikan masalah sistem distribusi tenaga listrik, penjelasan tentang teori sistem distribusi radial dan loop, macam-macam daya, macam-macam beban, rugi-rugi jaringan, dan metode rekonfigurasi jaringan.

- Dua Kualitas Yaitu dua zara pengaturan diri dalam hal ini
singgah atau dibuat di Rumah Makan
- Keseluruhan dari disainnya dari PT. PNI Disignasi Tercantum Ciri-Ciri yang
Mampu
3. Simpati dan antisipasi dari
Analisis situs yang dipandang berdasarkan metode Yanow yakni
sebagian besar untuk memenuhi tuntutan yang diberikan masyarakat
masyarakat dengan menciptakan komunitas
4. Membangun pemimpin pemerintahan dengan memperbaiki kinerja (kewirausahaan) dengan
Pemerintah Gencar Efektivitas untuk memenuhi tujuan dan optimis
maka dipaparkan dalam bentuk gambaran
5. Masyarakat simpati dan antisipasi dengan
Dengan Difisiensi setelah OI
6. Masyarakat positif pada
BAB I : Berizinkan Jalin Perekonomian, turunan manusia, perbaikan mesin-furnitur
menjadi posisi dan simpati dan antisipasi
7. BAB II : Mengawali usaha-saya simpati dan antisipasi
menjadi hasil disain diri yang baik, mencuci-memasak di rumah
makan-makan populer untuk menghindari bahwa mereka lupa mengisi
jaringan
- 1.0. Sistematisasi Pengembangan
Sistematisasi pengembangan desain simpati ini sebagai berikut :

- BAB III :** Membahas teori algoritma genetika, analisis aliran daya dan aplikasi penggunaan metode *Parallel Genetic Algorithm* untuk perbaikan pelayanan, algoritma program serta flowchart..
- BAB IV :** Hasil perhitungan, perbandingan profil tegangan, perbandingan pembangkitan, pembebanan, rugi-rugi saluran sebelum dan sesudah restorasi.
- BAB V :** Memuat intisari dan hasil pembahasan, yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan selanjutnya.

1.7. Kontribusi

Adapun kontribusi dari penelitian ini diharapkan agar metode perbaikan pelayanan ini bisa diterapkan oleh PT. PLN (Persero) Area Malang sehingga apabila terjadi gangguan, kontinuitas penyaluran daya listrik tetap terjaga.

BAB II

SISTEM DISTRIBUSI TENAGA LISTRIK

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

Secara garis besar suatu sistem tenaga listrik dapat dikelompokkan menjadi 3 bagian sub sistem, yaitu : (Hasan Basri. 1996 :1)

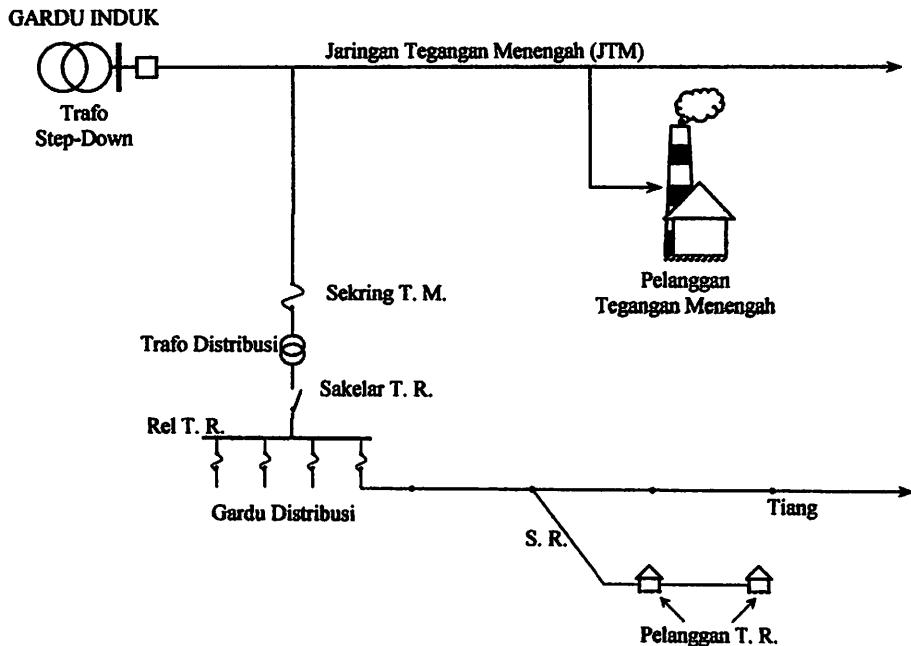
1. Sistem Pembangkitan; berperan sebagai sumber daya tenaga listrik dan disebut juga sebagai produsor energi.
2. Sistem Transmisi atau penyulang; berfungsi sebagai penyalur daya listrik secara besar-besaran dari pembangkit ke bagian sistem distribusi atau konsumen.
3. Sistem Distribusi dan Beban; berperan sebagai distributor energi ke konsumen yang memerlukan energi tersebut.

Jaringan distribusi berada pada akhir dari sistem tenaga listrik, perannya mendistribusikan tenaga listrik dari Gardu Induk ke konsumen melalui Gardu Distribusi. Untuk lebih jelasnya berikut ini memperlihatkan skema Gardu Induk yang dihubungkan melalui JTM hingga ke pelanggan.

BAB II

SISTEM DISTRIBUSI TRNGAIA ITSTRIK

- Sistem Jaringan Distribusi Transaksi dan Logistik (SISTRA) merupakan sistem informasi yang berfungsi untuk mendukung operasi pengolahan-pembangkitan bahan bakar fosil pada pembangkitan listrik. Sistem Jaringan Distribusi Transaksi dan Logistik (SISTRA) ini merupakan sistem informasi yang mendukung manajemen operasi jaringan distribusi dan pengelolaan operasi pembangkitan dan komunikasi antara pembangkitan dengan pelanggan.
- Sistem Jaringan Distribusi Transaksi dan Logistik (SISTRA) ini merupakan sistem informasi yang mendukung manajemen operasi jaringan distribusi dan pengelolaan operasi pembangkitan dan komunikasi antara pembangkitan dengan pelanggan.
- (Berasi pada : Basir, 1998)
1. Sistem Pengolahan Data pembangkitan dan jaringan distribusi
2. Sistem Transaksi dan kelayakan pembangkitan dan jaringan distribusi jaringan distribusi dan jaringan pengelolaan energi
3. Sistem Pengolahan Data pembangkitan dan jaringan distribusi dan jaringan pengelolaan energi
4. Sistem Distribusi dan Bepenuhan Perekonomian sebagaimana disesuaikan dengan prinsip-prinsip perekonomian dan jaringan pengelolaan energi
5. Sistem Distribusi dan Bepenuhan Perekonomian sebagaimana disesuaikan dengan prinsip-prinsip perekonomian dan jaringan pengelolaan energi
6. Sistem Distribusi dan Bepenuhan Perekonomian sebagaimana disesuaikan dengan prinsip-prinsip perekonomian dan jaringan pengelolaan energi
- Distribusi jaringan distribusi dan jaringan pengelolaan energi bersifat monodisiplinik dan berjaringan dengan jaringan logistik dan jaringan teknologi informasi. Distribusi jaringan distribusi dan jaringan pengelolaan energi bersifat monodisiplinik dan berjaringan dengan jaringan logistik dan jaringan teknologi informasi. Distribusi jaringan distribusi dan jaringan pengelolaan energi bersifat monodisiplinik dan berjaringan dengan jaringan logistik dan jaringan teknologi informasi. Distribusi jaringan distribusi dan jaringan pengelolaan energi bersifat monodisiplinik dan berjaringan dengan jaringan logistik dan jaringan teknologi informasi.



Keterangan : T. M. : Tegangan Menengah

T. R. : Tegangan Rendah

S. R. : Sambungan Rumah

Gambar 2-1. Skema GI yang dihubungkan melalui Jaringan Tegangan Menengah (JTM) hingga ke Pelanggan.

Sumber : Djiteng Marsudi,Ir, 1990: 3

Jaringan setelah keluar dari G.I biasanya disebut jaringan distribusi.

Setelah tenaga listrik disalurkan melalui jaringan distribusi primer maka kemudian tenaga listrik diturunkan tegangannya dalam gardu-gardu distribusi menjadi tegangan rendah, kemudian disalurkan melalui Jaringan Tegangan Rendah untuk selanjutnya disalurkan ke rumah-rumah atau pelanggan (konsumen) PLN melalui sambungan rumah.

2.1. Sistem Distribusi

Sistem Distribusi merupakan bagian dari sistem tenaga listrik yang berperan dalam mendistribusikan tenaga listrik dari Gardu Induk (GI) sampai ke konsumen. Sistem distribusi pada hakikatnya dapat dibedakan menjadi 2 bagian yaitu : (Hasan Basri, 1996:3)

- 1. Jaringan Distribusi Primer (Jaringan Tegangan Menengah)**

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

- 2. Jaringan Distribusi Sekunder (Jaringan Tegangan Rendah)**

Jaringan distribusi sekunder merupakan jaringan tenaga listrik yang menyalurkan energi listrik antara Gardu Distribusi sampai ke konsumen dengan tegangan kerja 110 V atau 220 V.

Secara garis besar jaringan distribusi dapat diklasifikasikan menjadi :

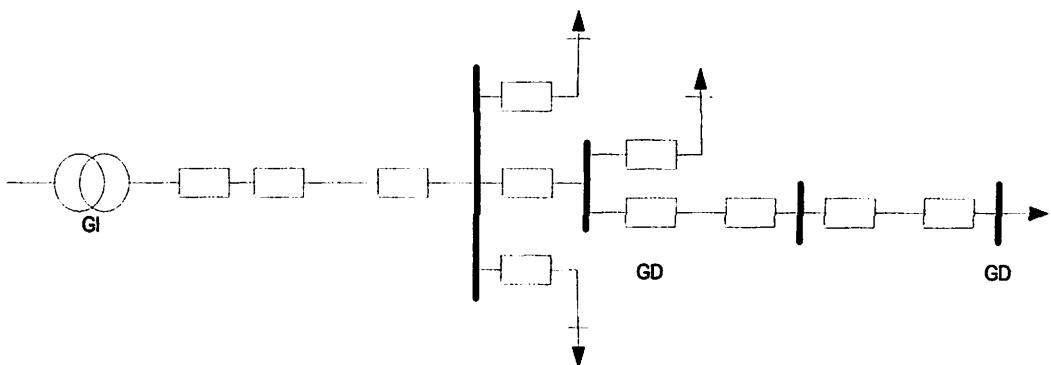
(Hasan Basri. 1996 : 63)

- Struktur Jaringan Radial
- Struktur Jaringan Loop (Lingkaran)
- Struktur Jaringan Grid atau Mesh (Anyaman)

2.1.1. Struktur Jaringan Radial

Struktur jaringan radial merupakan struktur jaringan yang paling sederhana dan paling murah biaya pembangunannya. Struktur jaringan ini dalam

menyalurkan energi listrik, keandalannya kurang. Suatu gangguan pada penyulang dapat mengakibatkan gangguan dalam penyaluran energi listrik ke konsumen yang berada dibelakang titik gangguan.



Gambar 2-2. Struktur Jaringan Radial

Sumber : Yusra Sabri, Dr. Ir, 1990

Tipe jaringan ini merupakan bentuk dasar, susunan maupun kebutuhan alat-alat penunjangnya paling sedikit dan paling sederhana. Tipe ini paling banyak digunakan untuk melayani konsumen terutama beban-beban rumah tangga yang tidak menuntut tingkat kontinyuitas pelayanan yang tinggi. Sumber daya pada tipe ini hanya dari satu titik. Salurannya dicabang-cabang menuju ke titik-titik beban, dan antara titik sumber dengan titik beban hanya ada satu pilihan. Dengan demikian bila salah satu saluran cabang mengalami gangguan maka seluruh beban yang ada disalurkan tersebut akan mengalami pemadaman total.

Jaringan distribusi radial mempunyai beberapa keuntungan dan kelemahan sebagai berikut : (Hasan Basri. 1996 :64)

Keuntungan jaringan radial :

1. Bentuknya sederhana dibandingkan dengan bentuk yang lain.
2. Biaya investasinya relatif lebih murah, karena saluran menuju ke tiap beban hanya tersedia satu jalur.

Kelemahan jaringan radial :

1. Kualitas pelayanan (penyaluran daya) dengan tipe yang lain lebih jelek, sebab jatuh tegangan dan rugi daya relatif besar. Kerugian ini terjadi pada saluran.
2. Kontinyuitas penyaluran daya tidak terjamin. Dengan hanya mengandalkan satu saluran, maka bila terjadi gangguan pada saluran tersebut tidak dapat diharapkan supply melalui saluran lain.

Jaringan radial ini dalam perkembangannya mengalami beberapa bentuk modifikasi sehingga dikenal beberapa macam jaringan distribusi radial, yaitu :

1. Sistem radial Pohon
2. Sistem radial dengan Tie dan Switch Pemisah
3. Sistem radial dengan Pusat Beban
4. Sistem radial dengan pembagian Daerah Phasa (Phase Area)

2.1.2. Struktur Jaringan Loop

Struktur jaringan loop merupakan gabungan dari dua struktur jaringan radial, dimana pada ujung kedua jaringan dipasang sebuah pemutus (PMT/CB) atau pemisah (PMS/DS). Pada saat terjadi gangguan, setelah gangguan dapat diisolir, maka pemutus atau pemisah ditutup sehingga aliran daya listrik ke bagian yang tidak terkena gangguan tidak berhenti. Dalam kondisi normal, struktur

Kontraktua laurden etatia:

1. Bolumukua segerihua dibandingkan dengan pertama yang lain.

2. Biaya investasi yang lebih murah ketika saham memudah ke tiga

peran pada terdesai atau jatuh

Kolempuan laurden etatia:

1. Kegiatas berzaman (berzaman dan) dalam ilmu dan pihak ketiga (lebih)

seperti istirahedan dan tahu dia berikan posisi Kondision im telahai bahan

sejauhan

2. Kondisifas berzaman dan dikeh (kejatuhan) yang pada

mengandalkan sum sejauhan mungkin pihak ketiga saham berdasarkan

tersebut tidak dapat ditambahkan supaya tetapi punya pun

tidaknya tidak ini dalam berkomunikasi dengan perspektif pemain

modelikasi sejauhan diketahui populer mencapai laurden etatia

1. Sistem tadii Poin

2. Sistem tadii dengun ti dulu Sistem Poin

3. Sistem tadii dengun Poin

4. Sistem tadii dengun Dapat Poin (Poin A)

5. Sistem tadii laurden etatia

Sistem tadii laurden etatia juga memungkinkan untuk dua alternatif berikut

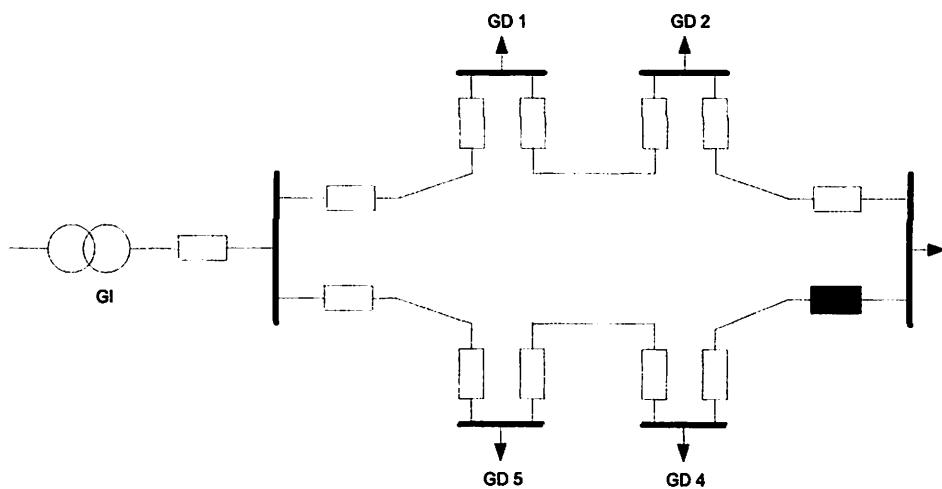
awal, dimana pada diri sendiri laurden etatia sejauh pemain (PMNCB)

atau pemain (PMSCS) pada saat itu ini sejauh setiap pengguna dapat

dilakukan untuk bermitra atau bersama-sama dalam sejauh tampil ke persaingan

yang tidak terkena pengaruh tidak pernah diajukan kepada orang lain

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 jaringan radial.



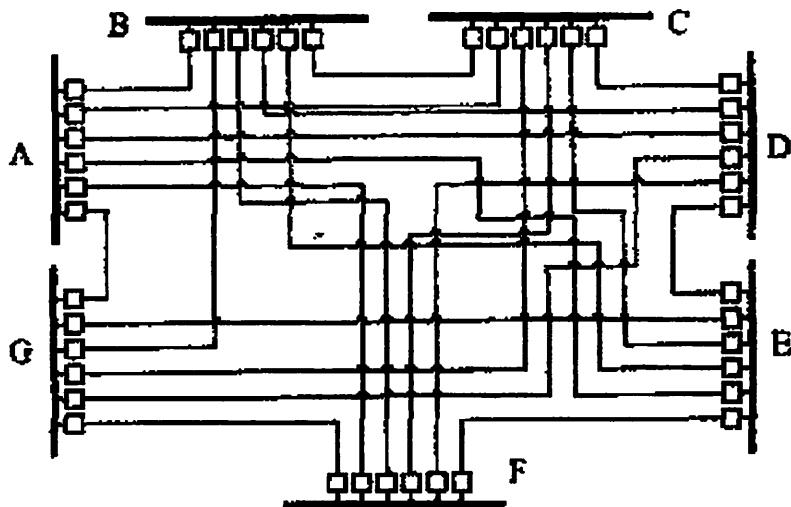
Gambar 2-3. Struktur Jaringan Loop

Sumber : Yusra Sabri, Dr. Ir, 1990

Keterangan : ■ = *Normally Open*
 □ = *Normally Close*

2.1.3. Struktur Jaringan Mesh

Struktur jaringan ini merupakan kombinasi antara struktur jaringan radial dengan struktur jaringan loop. Titik beban memiliki lebih banyak alternatif penyulang, sehingga bila salah satu penyulang terganggu maka dengan segera dapat digantikan oleh penyulang yang lain. Dengan demikian kontinyuitas penyaluran daya sangat terjamin.



Gambar 2-4. Struktur Jaringan Mesh

Sumber : Yusra Sabri, Dr. Ir, 1990

Tipe jaringan ini merupakan bentuk dasar, susunan maupun kebutuhan alat-alat penunjangnya paling sedikit dan paling sederhana. Tipe ini paling banyak digunakan untuk melayani konsumen terutama beban-beban rumah tangga yang tidak menuntut tingkat kontinyuitas pelayanan yang tinggi. Sumber daya pada tipe ini hanya dari satu titik. Salurannya dicabang-cabang menuju ke titik-titik beban, dan antara titik sumber dengan titik beban hanya ada satu pilihan. Dengan demikian bila salah satu saluran cabang mengalami gangguan maka seluruh beban yang ada disalurkan tersebut akan mengalami pemadaman total.

2.2. Daya Sistem Distribusi

Daya merupakan banyaknya perubahan tenaga terhadap waktu dalam besaran tegangan dan arus.

2.2.1. Daya Nyata (Real Power)

Daya nyata dinyatakan dalam persamaan : (Hasan Basri, 1996:9)

$$P = |V| |I| \cos \varphi \quad (2.1)$$

dengan :

P : daya aktif / nyata (KW)

V : tegangan (V)

I : arus (A)

Cos φ : faktor daya

Daya nyata untuk beban 3 fasa seimbang :

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

2.2.2. Daya Reaktif (Reactive Power)

Daya reaktif dinyatakan dalam persamaan : (Hasan Basri, 1996:10)

$$Q = |V| |I| \sin \varphi \quad (2.3)$$

dengan :

Q : daya reaktif (KVAR)

V : tegangan (V)

I : arus (A)

Daya reaktif untuk beban 3 fasa seimbang :

$$Q = \sqrt{3} |V_{jala-jala}| |I_{jala-jala}| \sin \varphi \quad (2.4)$$

2.2.3. Daya Semu (Apparent Power)

Daya kompleks dinyatakan dalam persamaan : (Hasan Basri, 1996:9)

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

dengan :

S : daya semu (KVA)

V : tegangan (V)

I : arus (A)

$\cos \phi$: faktor daya

Daya kompleks untuk beban 3 fasa seimbang :

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

Persamaan(2.2), (2.4), (2.6) berlaku tanpa memandang apakah bebannya dihubung secara segitiga (Δ) atau bintang (Y).

2.3. Beban Sistem Distribusi

Secara garis besar beban dapat diklasifikasikan menjadi empat, yaitu : (Hasan Basri, 1996:5)

1. Beban Perumahan (Rumah Tangga)

Beban perumahan pada umumnya berupa penerangan, kipas angin, alat-alat rumah tangga dan lain-lain

2. Beban Komersial

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

3. Beban Industri

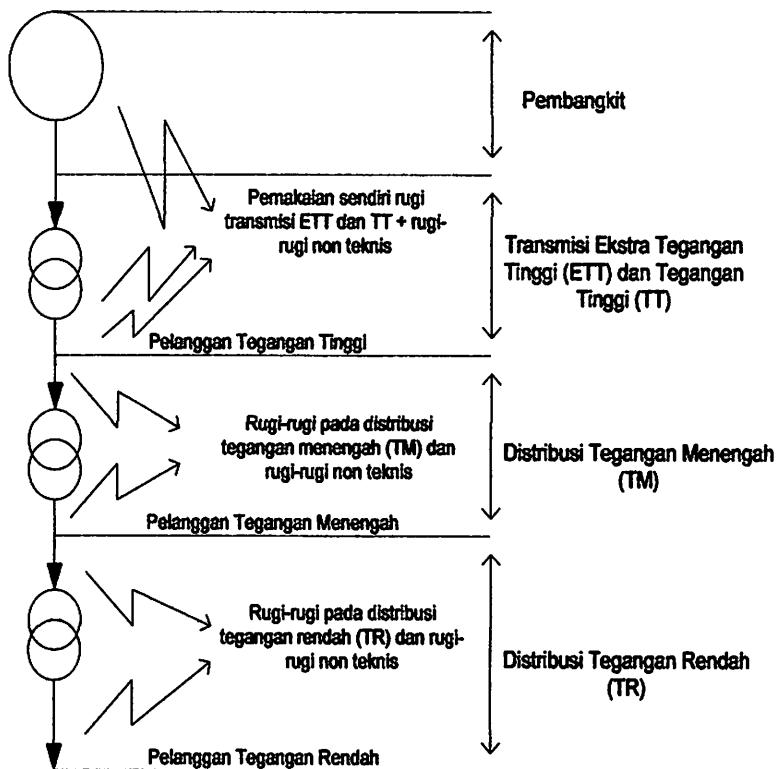
Beban industri diklasifikasikan menjadi 2 yaitu skala besar dan kecil.

4. Beban Fasilitas Umum

Beban fasilitas umum pada umumnya berupa penerangan sekolah, perkantoran dan lain-lain.

2.4. Rugi-rugi Jaringan

Dalam proses penyaluran energi listrik ke lapangan terjadi rugi-rugi teknis (losses), yaitu rugi daya dan rugi energi, dimulai dari Pembangkit, Transmisi dan Distribusi. Rugi-rugi teknis adalah penghantar saluran, adanya tahanan dari penghantar yang dialiri arus, timbulah rugi-rugi teknis (I^2R) pada jaringan tersebut. Rugi teknis tersebut terdapat pada saluran dan transformator. Rugi teknis pada penyaluran merupakan penjumlahan dari I^2R atau rugi tahanan. Untuk lebih jelasnya diperlihatkan pada gambar 2-5.



Gambar 2-5. Rugi-Rugi Pada Bagian Sistem Tenaga

Sumber : Hasan Basri,Ir. 1996 : 285

Keterangan :

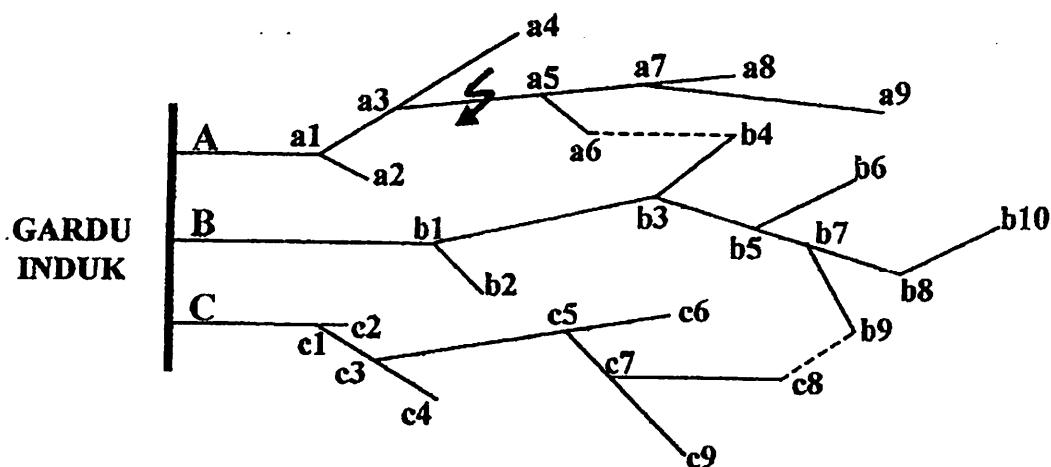
- = Aflan daya dan energi
- = Pembangkit
- = Trafo
- ↙ = Rugi-rugi

- **Meminimalisasi Rugi Daya Pada Sistem Distribusi**

Seperti telah disinggung sebelumnya bahwa pada kebanyakan sistem distribusi dalam penyaluran tenaga listrik umumnya dioperasikan secara radial. Namun demikian, sistem radial tersebut dioperasikan dengan beberapa

“*improvisasi*”. Biasanya terdapat beberapa *tie switch* yang tersedia untuk saling interkoneksi bagi beberapa bus beban (*Load Bus*) baik dalam satu *feeder* maupun berlainan *feeder*.

Kondisi improvisasi tersebut sangat diperlukan demi memenuhi persyaratan sebagai sistem yang cukup baik, terutama yaitu kontinuitas pelayanan. Hal ini dapat dijelaskan dengan contoh kondisi gangguan dilihat pada gambar 2-6.



Gambar 2-6. Contoh Kondisi Gangguan Pada Sistem Distribusi

Sumber : Hasan Basri,Ir. 1996 : 260

Suatu sistem penyaluran distribusi radial dari gardu induk mempunyai *feeder* (penyalang) yaitu *feeder* A,B dan C. *Feeder* A mempunyai bus-bus beban a1, a2, ..., a9. *Feeder* B mempunyai bus-bus beban b1, b2, ..., b10. *Feeder* C mempunyai bus-bus beban c1, c2, ..., c9. Jika *feeder* A mendapat gangguan pada saluran antara bus a3 dan a5 sehingga menyebabkan putusnya saluran maka bus a5, a6, ..., a9 akan mengalami pemadaman total. Oleh karena itu bus-bus beban yang mengalami pemadaman total tersebut harus disupply dari *feeder* lain (bisa juga dari bus a4). Tanda (----) menunjukkan bahwa pada saluran itu terdapat NO

switch (Normally Open switch) Dari gambar dapat dilihat bahwa *NO switch* terdapat antara bus a6 pada *feeder A* dengan bus b4 pada *feeder B*. Jika pada saat gangguan terjadi sehingga bus a3 dan a5 open maka NO switch a6-b4 dapat ditutup (*close*) sehingga dengan demikian bus-bus pada *feeder A* yang mengalami pemadaman total dapat diatasi. Hal yang sama juga dapat dilaksanakan antara *feeder B* dengan *feeder C* dengan NO switch b9-c8.

Pada kenyataan di lapangan, lokasi dari bus-bus itu sendiri mempunyai lokasi yang tidak sama sehingga akan menggunakan konduktor yang panjangnya tidak sama. Kondisi ini tentunya akan membutuhkan resistansi dan reaktansi saluran yang berbeda-beda.

Bahkan bukan itu saja, hal yang sangat mungkin terjadi adalah bahwa besarnya beban-beban pada bus-bus tersebut tidak sama sehingga dapat dikatakan mempunyai variasi kerapatan beban.

Kedua fenomena tersebut di atas itulah yang menyebabkan sulitnya sistem distribusi radial untuk memenuhi aspek teknis karena alasan lokasi beban dan variasi dari kerapatan beban itu sendiri. Pada akhirnya fenomena ini akan mengakibatkan rugi-rugi daya pada saluran distribusi menjadi tinggi.

Dari permasalahan tersebut di atas, maka diperlukan suatu upaya untuk dapat mengurangi kerugian daya yang terjadi. Dalam prakteknya, sebenarnya ada 2 macam cara dalam upaya untuk mengurangi/meminimalisasi rugi-rugi daya pada sistem distribusi tenaga listrik yaitu :

1. Pemasangan/penginstalasian kapasitor
2. Rekonfigurasi jaringan distribusi

मानव ओर वनस्पति जीवों के साथ जुँग है। (वासा और विवरण) मानव
जीवों के साथ जुँग है। यह जीवों के साथ जुँग है। वासा और विवरण
जीवों के साथ जुँग है। ओर जीवों के साथ जुँग है। वासा और विवरण
जीवों के साथ जुँग है। वासा और विवरण जीवों के साथ जुँग है।
जीवों के साथ जुँग है। वासा और विवरण जीवों के साथ जुँग है।

यह जीवों के साथ जुँग है। वासा और विवरण जीवों के साथ जुँग है।

पान के उत्तरांश के लिए जीवों के साथ जुँग है। वासा और विवरण
जीवों के साथ जुँग है। वासा और विवरण जीवों के साथ जुँग है।
जीवों के साथ जुँग है। वासा और विवरण जीवों के साथ जुँग है।
जीवों के साथ जुँग है। वासा और विवरण जीवों के साथ जुँग है।

सर्वानन्द परमेश्वर

भक्ति प्रकृति वाले वाले वाले वाले वाले वाले वाले वाले
परमेश्वर-परमेश्वर-परमेश्वर-परमेश्वर-परमेश्वर-परमेश्वर-परमेश्वर-

मुख्यमन्त्र वाले वाले वाले वाले वाले वाले वाले वाले

कठिन लेखनों के लिए वाले वाले वाले वाले वाले वाले वाले
दिव्यानन्द वाले वाले वाले वाले वाले वाले वाले वाले वाले
वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले
वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले

वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले

दाता भूमिकामन्त्र वाले वाले वाले वाले वाले वाले वाले वाले
दाता भूमिकामन्त्र वाले वाले वाले वाले वाले वाले वाले वाले
दाता भूमिकामन्त्र वाले वाले वाले वाले वाले वाले वाले वाले

वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले वाले

१. भूमिकामन्त्र वाले वाले वाले वाले वाले वाले

२. रेगुलिंग वाले वाले वाले वाले वाले वाले

Untuk selanjutnya, pada skripsi ini akan dibahas mengenai upaya untuk mengurangi kerugian daya yang terjadi pada saluran akibat terjadi gangguan yaitu dengan merekonfigurasi jaringan distribusi dengan memanfaatkan status dari *switch-switch (NC/NO switch)*.

Minimalisasi rugi-rugi dapat diformulasikan sebagai berikut : (Whei Min lin, Hong Chan Chin, 1998 : 870)

$$\text{Min } P_{\text{loss}} (S_v) \quad (2.7)$$

Dimana

$$P_{\text{loss}} = \sum_{i=1}^{N_b} R_i * |I_i|^2$$

$$I_i \leq I_{\max}$$

$$i = 1 \approx N_b$$

$$V_{\min} \leq V_j \leq V_{\max}$$

$$j = 1 \approx N$$

dengan :

P_{loss} : Total rugi saluran pada *feeder* distribusi

S_v : Status vektor dari *switch*

N_b : Total jumlah cabang dari sistem keseluruhan

R_i : Tahanan dari cabang i

I_i : Besarnya arus dari cabang i

I_{\max} : Batas atas dari besar arus cabang

N : Total jumlah bus

V_j : Besarnya tegangan dari bus j

V_{\max} : Batas atas dari besarnya tegangan bus

V_{\min} : Batas bawah dari besarnya tegangan bus

2.5. Prosedur Perhitungan Tegangan

- Menjumlahkan masing-masing beban dari ujung-ujung cabang yang ada.

Menjumlahkan secara menaik masing-masing beban tersebut menurut persamaan berikut ini, dari ujung cabang ke sumber tenaga : (Y Fukuyama, H-D Chiang & K Nan Miu.1996 : 115)

$$\text{SCUR}_{(n)} = \text{CUR}_{(n)} + \Sigma \text{SCUR} \quad (2.8)$$

dengan :

$\text{SCUR}_{(n)}$: total arus beban pada *node n*

$\text{CUR}_{(n)}$: arus beban pada *node n*

ΣSCUR : hasil jumlah arus beban sebelum *node n*

Dari sini, bila *node n* adalah ujung dari suatu *feeder*, maka ΣSCUR adalah sama dengan 0 disebabkan tidak adanya arus keluar.

- Menghitung drop tegangan pada setiap node-nya

Menghitung drop tegangan pada setiap node-nya dengan menggunakan persamaan berikut : (Y Fukuyama, H-D Chiang & K Nan Miu.1996 : 115)

$$\Delta V_{(n)} = \text{SCUR}_{(n)} \times Z_{(n)} \quad (2.9)$$

dengan :

$\Delta V_{(n)}$: drop tegangan pada *node n*

$\text{SCUR}_{(n)}$: jumlah beban arus pada *node n*

$Z_{(n)}$: impedansi *node*

- Menghitung tegangan pada tiap-tiap node dari sumber tenaga

Tegangan pada terminal sumber tenaga telah diketahui. Oleh karenanya, tegangan-tegangan pada setiap node-nya dapat pula dihitung berdasarkan masing-masing sumber tenaga tersebut dengan menggunakan persamaan berikut : (Y Fukuyama, H-D Chiang & K Nan Miu.1996 : 116)

$$V_{(n)} = V_{(k)} - \Delta V_{(n)} \quad (2.10)$$

dengan :

$V_{(n)}$: tegangan pada *node n*

$V_{(k)}$: tegangan pada upstream *node* dari *node n*

$\Delta V_{(n)}$: total drop tegangan pada *node n*

Disini, untuk pertama kalinya, k sama dengan sumber tenaga

2.6. Metode Rekonfigurasi Jaringan

Proses menata konfigurasi awal dari jaringan (*initial condition*) menjadi suatu konfigurasi optimum akhir (*optimal condition*), sehingga dari konfigurasi yang terakhir ini diperoleh kerugian daya sistem distribusi yang paling kecil (aliran daya yang paling optimum).

Rekonfigurasi jaringan sangat penting digunakan untuk memperbaiki keamanan sistem. Ada sejumlah *switch normally open* dan *normally closed* dalam saluran distribusi. Dengan mengatur status buka atau tutup switch penyulang, maka beban dapat ditransfer dari penyulang satu ke penyulang lain. Selama terjadi gangguan, *switch-switch* tersebut digunakan untuk mengisolasi gangguan serta memperbaiki pelayanan.

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

1. *Normally Closed switch* (NC switch) yang dalam keadaan operasi normal, posisi kontaknya selalu tertutup (*on*).
2. *Normally Open switch* (NO switch) yang dalam keadaan operasi normal, posisi kontaknya selalu terbuka (*off*).



Gambar 2-8. Diagram Skematik Tipe (1) NC Switch dan (2) NO Switch

Sumber: Hasan Basri. 1996

BAB III

PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER MENGGUNAKAN METODE *PARALLEL GENETIC ALGORITHM*

3.1. Algoritma Genetika

Algoritma Genetika merupakan metode adaptive yang bisa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetik yang ada dalam 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, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

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

BAS II
PERBANDINGAN PRIMAYARAN DAN PASCA
SATUAN DISTRIKSI PRIMER MENGUNAKAN
METODE MARKETING SURVEY DENGAN KOGNITIF

Salicornia europaea L.

Aganglionikus Öffnungsfehler ist eine sehr seltene Erkrankung, die durch eine Hypertonie des Ganglionknotenplexus gekennzeichnet ist. Die Hypertonie führt zu einer Störung der motorischen und sensiblen Funktionen im Bereich des Ganglionknotenplexus. Dies kann zu schweren Verstimmungen führen, wie z.B. zu einem chronischen Schmerzsyndrom oder zu einer Paraparesis. Die Diagnose wird durch eine Kombination von klinischen Befunden und bildgebenden Verfahren gestellt. Die Behandlung besteht in der chirurgischen Entfernung des Ganglionknotenplexus.

eveluasi berupa nilai fitness setiap kromosom dengan nilai fitness terbaik yang bertahan hidup dan selalu diproduksi.

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

Algortima Genetika memiliki empat dasar kerja yaitu : (Mitsuo Gen, Runwei Cheng, 1994 : 33)

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

• Wissenschaften
• Natur
• Technik
• Soziales
• Politik
• Kultur
• Sport

3.1.1. Familien-Alzheimer-Konflikte

Alzheimer-Konflikte entstehen aus mehreren Gründen. Eine der häufigsten Gründe ist die Erinnerungslosigkeit des betroffenen Angehörigen. Dies kann zu Konflikten führen, wenn der Angehörige versucht, die Vergangenheit wiederherzustellen oder seine Erfahrungen mit anderen Personen zu teilen. Ein weiterer Grund für Konflikte ist die Veränderung des Verhaltens des betroffenen Angehörigen. Er kann plötzlich aggressiv werden, sich nicht mehr richtig kleiden oder sich nicht mehr richtig ernähren. Diese Veränderungen können zu Konflikten führen, wenn sie die anderen Angehörigen belästigen.

Table 3-1. Tabelle 3-1 zeigt die Diagnosekriterien für Alzheimer-Konflikte

Genetik	Pathophysiologie	Ätiologie	Gen	Kontingenzen	Merkmale
Erbgabe	Schädigung der Hirnzellen durch Proteinkristalle	Nicht-Zahn- und Zahnschwund	PSEN1	Positiv dient nur	Erbgabe
Erbgabe	Abnormaler Proteinabbau im Gehirn	Wiederholung von Gedanken	APP	Positiv dient nur	Erbgabe
Erbgabe	Abnormaler Proteinabbau im Gehirn	Wiederholung von Gedanken	APOE	Positiv dient nur	Erbgabe
Erbgabe	Abnormaler Proteinabbau im Gehirn	Wiederholung von Gedanken	Tau	Positiv dient nur	Erbgabe

Quelle: Moreno-Garcia, Rummel-Gerloff, 1994; 24

Alzheimer-Konflikte sind eine Form von Konflikten, die zwischen Angehörigen und betroffenen Personen auftreten. Sie können verschiedene Ursachen haben, wie z.B. Veränderungen im Verhalten des betroffenen Angehörigen oder Probleme bei der Kommunikation. Es ist wichtig, dass die Angehörigen auf die Konflikte reagieren und sie nicht ignorieren. Sie sollten versuchen, die Konflikte zu verstehen und zu lösen, um die Beziehung zwischen ihnen und dem betroffenen Angehörigen zu erhalten.

Generell:

4. Menggunakan aturan transisi secara probabilistik bukan deterministik.

3.1.1. Istilah-Istilah Algoritma Genetika

Algoritma Genetika menggunakan mekanisme genetika yang ada pada proses alami dan sistem buatan. Istilah-istilah yang digunakan adalah gabungan dari dua disiplin ilmu, yaitu ilmu Biologi dan ilmu computer. Mitsuo Gen dan Runwei Cheng (1997) menjelaskan istilah-istilah yang digunakan dalam Algoritma Genetika , seperti diperlihatkan di tabel 3-1 berikut ini :

Tabel 3-1. Istilah Yang Digunakan Dalam Algoritma Genetika

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

Sumber : Mitsuo Gen, Runwei Cheng, , 1994 : 34

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

• Mendukungkan situasi kritis seceras mungkin agar
determinisme

3.1.1. Satu-satunya Algoritma Genetic

Algoritma Genetic merupakan metode pemecahan masalah yang ada pada proses simulasi dan sistem perancangan teknologi yang digunakan adalah berupa proses dan algoritma yang dikenal dengan komputer. Maka dari itu dan Ruhaii Chong (1993) menjelaskan sistem-sistem yang digunakan dalam Algoritma Genetic, secara deskripsiannya di bawah ini:

Table 3-1. Sifat-sifat Yang Didefinisikan Dalam Algoritma Genetic

Fitilip	Keterwakilan	Gen	Iota	Alpha	Beta	Gamma
Judul atau penulis sederhananya yang suatu dituliskan	Judul atau penulis sederhananya yang suatu dituliskan					
Posisi atau bentuk						
Zifizi dan dimensinya dalam bentuk						
Phenotype	Sifat-sifat fisik bersifat takwiman yang perlakuan seperti sifat					
Genotype	Sifat-sifat genetik bersifat takwiman yang perlakuan seperti sifat					

Sumber: Wihno Giri Ruhaii Chong, 1993 : 24

Terdapat beberapa karakteristik yang didefinisikan dalam Algoritma Genetic

Pada model tersebut didefinisikan untuk militer kompleksitas dan Algoritma Genetika.

Parameter yang digunakan adalah : (Mitsuo Gen, Runwei Cheng, 1994 : 35)

1. Jumlah Generasi (MAXGEN)
2. Ukuran Populasi (POPSIZE)
3. Probabilitas Crossover (Pc)
4. Probabilitas Mutasi (Pm)
5. Panjang Kromosom (NVAR)

3.1.1.1. Jumlah Generasi (MAXGEN)

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

3.1.1.2. Ukuran Populasi (POPSIZE)

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

Gemarkelezer Zang d'gmarkelezer nadeleper; (Mitarbeiter der Kanzlei (part) 100 : 3)

I. Tumip Gendesa (WAKIL)

J. Uktuna Popiasi (DORINE)

K. Biopapilius Gassot (Pa)

L. Biopapilius Munesi (Pa)

M. Biopapilius Komasot (NANI)

3.1.1.1. Tumip Gendesa (WAKIL)

Metukenan jumper berlindasun (tiers) disimpannya takut pada
selepas tampan potongan ini mempunyai kesimpulan tentang kira-kira
(waktu) bincara (Jabatan) tampan potongan yang pernah diberi
meningkatkan kewaspadaan seseorang dengan bertambahnya waktunya yang
panas. Sedangkan jika jumlah kesempatan bertemu sedikit maka akan selesaikan
permasalahan dengan optimal.

3.1.1.2. Uktuna Popiasi (DORINE)

Uktuna popiasi mempunyai kriteria dan ciri-ciri dia Algotima
Gendesa tiga akhir uktuna popiasi bukan makan popiasi tiada makcikaksaan supaya
makan untuk mencuci tangan boleh berterusan setiap kali dia
Algotima Gendesa mendekati pihak Dalam seti ini dihadapkan dengan keadaan setia ini
peras untuk mengelakkan keserasian ini yang berterusan. Selain itu
berlindasun popiasi yang pernah diperlakukan dengan tidakbaik oleh
wifizahay hasil Xirizan Wicelawis (100) pernahnya pada hari

Algotima Gendesa mempunyai popiasi yang bernama 100-05-100.

3.1.1.3. Probabilitas Crossover (*Pc*)

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

3.1.1.4. Probabilitas Mutasi (*Pm*)

Mutasi digunakan untuk meningkatkan variasi populasi digunakan untuk 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 Algoritma Genetika, mutasi menjalankan aturan penting yaitu :

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

3.1.3. Biopsiyitis Cross-over (Ba)

Biopsiyitis cross-over ini ditunjukkan untuk mengantikan tiga buah objektif cross-over Deltast per inti dalam bahanasi terdahulu di WO2012 sinkronis (individu) yang mempunyai bintang siang. Semakin besar nilai biopsiyitis cross-over maka semakin besar struktur pada tiga objekti dalam bahanasi. Namun jika biopsiyitis cross-over tertinggi pertama maka struktur tersebut tidak dapat dilakukan dalam bahanasi. Seperti juga biopsiyitis tertinggi akan puncak pada pukul 12.00 setelah satu sejodai. Agar lebih baik menggunakan bahanasi basah biopsiyitis tertinggi kira-kira tiga puluh Mikroskopies (segera). Nipissing Mikroskopies (1990) pernah membuat perkusi Algoritma Gantung menggunakan teknik single biopsiyitis cross-over pada laju 1.1 - 50.0

3.1.4. Biopsiyitis Multis (Ma)

Masing ditunjukkan untuk mengantikan tiga buah objektif cross-over untuk mengantikan tiga buah objektif ketiga fokusasi terhadapnya untuk terusnya mewajah di WO2012. Kelebihan ini adalah pada bahanasi yang berulang kali dan pada individu biopsiyitis antara tiga buah objektif dalam bahanasi akan berulang kali. Untuk mendapatkan hasil yang maksimal maka sebaiknya menggunakan teknik cross-over. Untuk mendapatkan hasil yang maksimal maka sebaiknya menggunakan teknik cross-over. Untuk mendapatkan hasil yang maksimal maka sebaiknya menggunakan teknik cross-over.

1. Menggunakan four-gon yang paling sederhana bukan sebagai bahanasi

2. Menggunakan four-gon yang tiga buah objektif bersama-sama sebagai bahanasi

bahanasi.

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

3.1.1.5. Panjang Kromosom (*NVAR*)

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

3.1.2. Proses Algoritma Genetika

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

3.1.2.1. Pengkodean atau Representasi

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

Secara umum Algortima Genetika dibentuk oleh serangkaian kromosom yang ditandai dengan x_i ($i = 1, 2 \dots N$). Setiap elemen dalam kromosom ini adalah variabel string yang disebut gen, berisi nilai-nilai atau allele. Variabel-variabel ini dapat dinyatakan dalam bentuk bilangan biner, bilangan real (*floating point*), integer, abjad. Pengkodean string biner merupakan pendekatan klasik yang digunakan dalam penelitian Algoritma Genetika karena sederhana. Meskipun representasi dengan cara ini menyulitkan untuk beberapa permasalahan optimasi,

misalnya permasalahan graph coloring. Digunakan teknik pengkodean yang lain seperti representasi real number (*floating point*), representasi *order-based* (untuk *bin-patching, graph coloring*), embedded lists (untuk permasalahan penjadwalan (*scheduling*), *variable element list* (untuk semi konduktor layout), dan even *LISP S-expressions*.

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

3.1.2.2. Fungsi *Fitness* (Fungsi Evaluasi)

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

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

Permasalahan klasik dalam Algoritma Genetika adalah beberapa kromosom dengan nilai *fitness* yang tinggi (tetapi bukan nilai optimum) mendominasi populasi dan mengakibatkan Algoritma Genetika konvergen pada lokal optimum.

Ketika mencapai konvergen, kemampuan Algoritma Genetika untuk mencari solusi yang lebih baik menghilang. Tukar silang antara kromosom induk yang hampir identik. Dalam hal ini hanya operasi mutasi yang mampu menghasilkan kromosom yang relatif baru dan merupakan cara untuk menghindari kromosom tertentu mendominasi populasi.

3.1.2.3. Seleksi

Pada Algoritma Genetika terdapat proses seleksi yaitu proses pemilihan kromosom yang akan di-crossover-kan dengan kromosom dari individu lain. Masalah yang paling mendasar pada proses ini adalah bagaimana proses penyeleksianya. Menurut teori Darwin proses seleksi individu adalah : “*individu terbaik akan tetap hidup dan menghasilkan keturunan*”. Pada proses seleksi ini dapat menggunakan banyak metode seperti *roulette wheel selection, rank selection, dan lain sebagainya*.

➤ Roulette Wheel Selection

Dimana setiap individual memiliki harga fitness sehingga didapatkan probabilitas individual $(f(t)/\sum f(t))$ tersebut dicopykan pada populasi yang baru. Untuk individual yang memiliki probabilitas 20% untuk jumlah populasi 10 maka kemungkinan individual tersebut dapat terpilih sebanyak dua kali.

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

1. Menjumlahkan fitness dari seluruh anggota populasi.
2. Membangkitkan nilai k, suatu nilai random antara 0 dan total fitnessnya.

3. Menjumlahkan fitness dari kromosom-kromosom dari populasi mulai 0 hingga total fitness lebih besar atau sama dengan nilai k lalu ambil kromosom tersebut.

➤ Rank Selection

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

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

3.1.3. Elitism

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

3. Molding selection process dari kromosom-kromosom dan populasi untuk
menghasilkan fitur-fitur yang dikehendaki pada sifat-sifat tertentu.
Kromosom tersebut

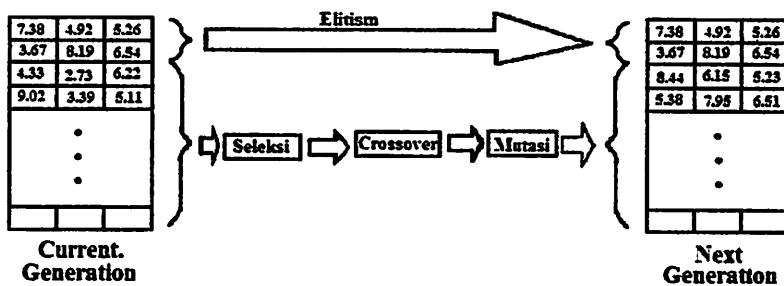
4. Kunci Seleksi

Aktivitas tifusus yang dimiliki oleh suatu kromosom dapat berupa:
1. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
2. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
3. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
4. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
5. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
6. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
7. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
8. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
9. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
10. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
11. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
12. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
13. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
14. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
15. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
16. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
17. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
18. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
19. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.
20. Keterbatasan dalam jumlah mutasi yang dapat terjadi pada suatu kromosom.

5.1.3. Efisiensi

Selama memperbarui populasi para ahli dalam teknologi kromosom
akan terjadi kopoligogenik kromosom (polyploid). Efisiensi seleksi metode
yang berlaku pada manusia adalah kromosom triflag (tetraploid). Kedua
populasi para ahli Sisirna dilakukan dengan cara pisan agar tidak selanjutnya
menyebarluaskan gen manusia. Efisiensi dapat secara cepat menurunkan populasi
yang memiliki Genetika kromosom manusia yang dilakukan secara tetraploid.

few best) yang telah ditemukan. Illustrasi kerja operator ini dapat digambarkan seperti pada gambar 3-1 berikut ini :

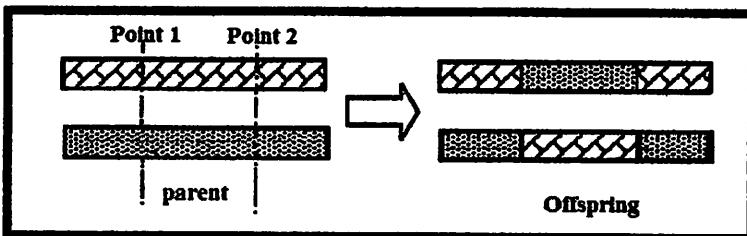


Gambar 3-1. Pembentukan *Next Generation* dalam Algoritma Genetika

Sumber : Mitsuo Gen, Runwei Cheng, 1994 : 42

3.2. Crossover (Pindah Silang)

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



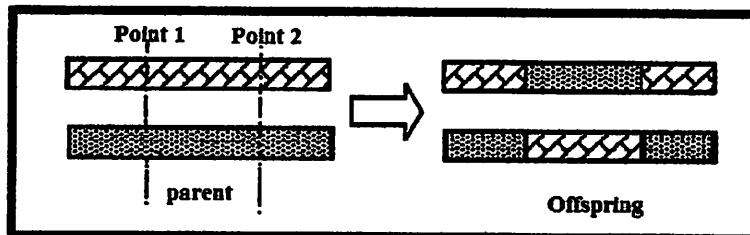
Gambar 3-2. Illustrasi operator dengan *One Point Crossover*

Sumber : Mitsuo Gen, Runwei Cheng, 1994 : 44

Kemudian ditingkatkan lagi dengan menggunakan *two point crossover*.

Penentuan posisi titik persilangan sama seperti *one point crossover* sebelumnya.

Pemilihan secara random dilakukan 2 kali. Kemudian nilai *offspring* diambil dari dua parent tersebut dengan batas dua titik persilangan tersebut. Illustrasi kerja operator ini digambarkan seperti pada gambar 3-3 berikut ini.



Gambar 3-3. Illustrasi operator dengan *Two Point Crossover*

Sumber : Mitsuo Gen, Runwei Cheng, 1994 :45

Untuk *crossover uniform* dibangkitkan suatu nilai random 0 dan 1 sepanjang jumlah kromosom untuk nilai loci. Jika nilai yang dibangkitkan mempunyai nilai 1 maka *allele* parent 2 dan offspring 2 untuk loci tersebut diambil dari *allele* parent 1 dan offspring 2 untuk loci tersebut diambil dari *allele* parent 2. Illustrasi kerja operator ini digambarkan seperti pada gambar 3-4 berikut ini.

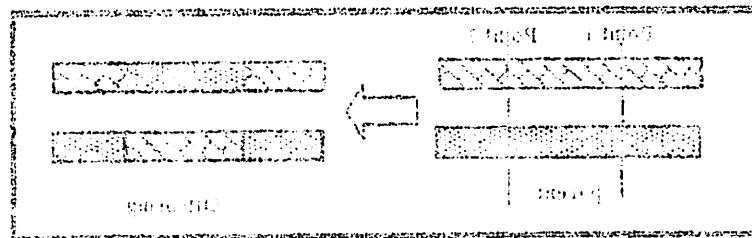


Diagramm 3-5: Hierarchische oder hierarchisch-dezentrale Ortsnamenklassifizierung

Quelle: Altenbergs, 1991, S. 49

Zur Klassifizierung von Ortsnamen wird eine hierarchische oder hierarchisch-dezentrale Ortsnamenklassifizierung eingesetzt. Diese ist eine hierarchische Ortsnamenklassifizierung, die auf Basis von Ortsnamenklassen und -typen aufgebaut ist. Die Klassifizierung ist hierarchisch, da sie verschiedene Ebenen der Klassifizierung aufweist. Die Klassifizierung ist dezentral, da sie auf Basis von Ortsnamenklassen und -typen aufgebaut ist. Die Klassifizierung ist hierarchisch, da sie verschiedene Ebenen der Klassifizierung aufweist. Die Klassifizierung ist dezentral, da sie auf Basis von Ortsnamenklassen und -typen aufgebaut ist.

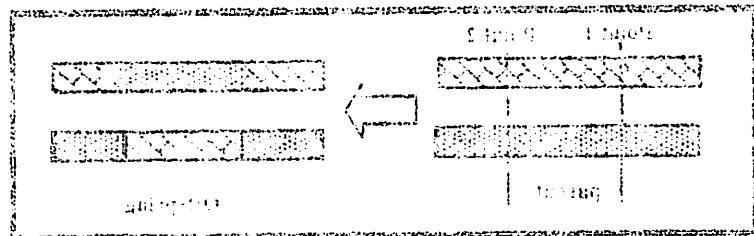
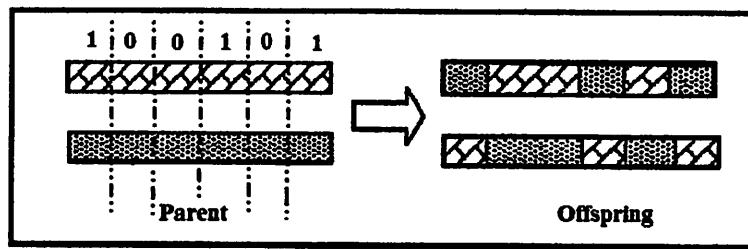


Diagramm 3-6: Hierarchische oder hierarchisch-dezentrale Ortsnamenklassifizierung

Quelle: Altenbergs, 1991, S. 49

Zur Klassifizierung von Ortsnamen wird eine hierarchische oder hierarchisch-dezentrale Ortsnamenklassifizierung eingesetzt. Diese ist eine hierarchische Ortsnamenklassifizierung, die auf Basis von Ortsnamenklassen und -typen aufgebaut ist. Die Klassifizierung ist hierarchisch, da sie verschiedene Ebenen der Klassifizierung aufweist. Die Klassifizierung ist dezentral, da sie auf Basis von Ortsnamenklassen und -typen aufgebaut ist. Die Klassifizierung ist hierarchisch, da sie verschiedene Ebenen der Klassifizierung aufweist. Die Klassifizierung ist dezentral, da sie auf Basis von Ortsnamenklassen und -typen aufgebaut ist. Die Klassifizierung ist hierarchisch, da sie verschiedene Ebenen der Klassifizierung aufweist. Die Klassifizierung ist dezentral, da sie auf Basis von Ortsnamenklassen und -typen aufgebaut ist.

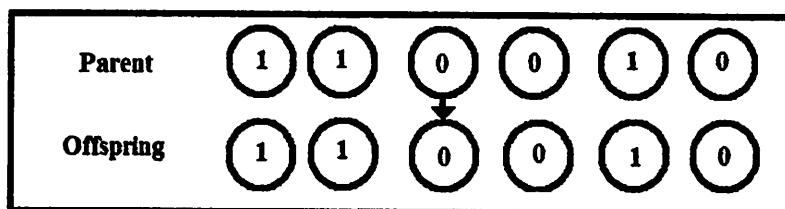


Gambar 3-4. Ilustrasi operator *crossover* dengan *uniform crossover*

Sumber : Mitsuo Gen, Runwei Cheng, 1994 : 45

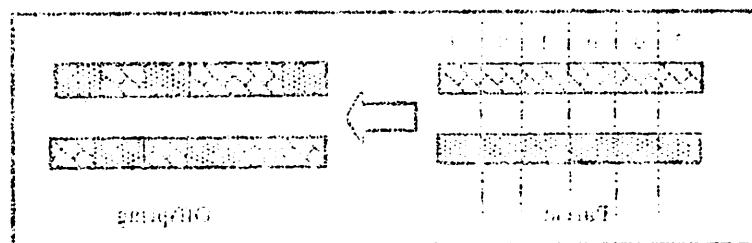
3.3. Mutation (Mutasi)

Operator mutasi digunakan untuk memodifikasi satu atau lebih nilai gen dalam satu individu. Cara kerjanya dengan membangkitkan sebuah nilai random r_k dimana $k = 1, 2, \dots, NVAR$ (panjang kromosom). Probabilitas mutasi (P_m) ditentukan dan digunakan untuk mengendalikan frekuensi operator mutasi. Apabila nilai random r_k , P_m maka gen ke- k kromosom tersebut terpilih untuk mengalami mutasi. Mutasi dengan mengganti gen 0 dengan 1 atau sebaliknya gen 1 dengan 0. Biasanya disebut flip yaitu membalik nilai ke 1 atau 1 ke 0. Ilustrasi kerja operator untuk representasi string biner digambarkan pada gambar 3-5.



Gambar 3-5. Ilustrasi operator mutasi untuk representasi string biner

Sumber : Mitsuo Gen, Runwei Cheng, 1994 :46

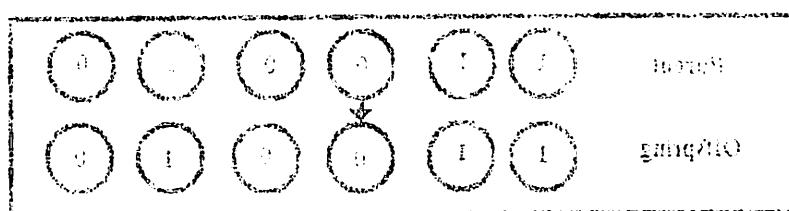


Gambar 3-4. Ilustrasi operasi alokasi dan dealokasi memori dinamis

Sumber : Mikro Computer 1991 : 45

(d) pointer M (Mempunyai nilai)

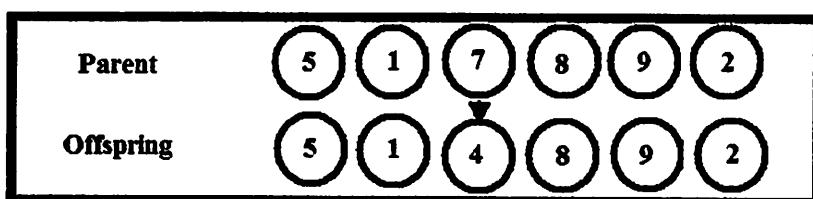
neg ijin didel pada nilai hasil alih-alihnya ketika penamaan dibuat. Objek yang punya nilai tetap disebut sebagai konstanta. Dalam kondisi ini kita tidak boleh mengubah nilainya. Jadi, kalau ada variabel yang nilainya tetap, maka kita sebaiknya memberikan tanda `const` di depan namanya. Misalnya `const int A = 5;`. Jadi, kalau variabel `A` ini nilainya tetap, maka kita tidak boleh mengubah nilainya. Kalau kita mengubah nilainya, maka program akan error. Untuk mengetahui apakah suatu variabel mempunyai nilai tetap atau tidak, kita bisa melihat tanda `const` di depan namanya. Jika ada, maka variabel tersebut mempunyai nilai tetap. Jika tidak, maka variabel tersebut mempunyai nilai yang bisa berubah.



Gambar 3-5. Ilustrasi operasi alokasi dan dealokasi untuk pointer

Sumber : Mikro Computer 1991 : 46

Untuk bentuk representasi integer atau floating point, atau selain string biner, proses mutasi terjadi apabila nilai $r_k < P_m$ memenuhi maka gen ke-k digantikan oleh suatu nilai random yang dibangkitkan pada range tertentu sesuai dengan pembentukan populasi awal. Seperti terlihat pada gambar 3-6 berikut ini.



Gambar 3-6. Ilustrasi operator mutasi untuk representasi integer

Sumber : Mitsuo Gen, Runwei Cheng, 1994 : 46

Untuk kromosom induk seperti gambar 3-6 diatas yaitu 5-1-7-8-9-2, proses mutasi adalah dibangkitkan sebuah nilai random r_k , $[k = 1 \dots NVAR]$. Misalkan pada saat $k = 3$ nilai $r_3 < P_m$ maka gen ke-3 yang bernilai 7 akan bermutasi dengan gen hasil random pada range $\{1 \dots x\}$ dan diperoleh nilai 4. Maka bentuk kromosom barunya adalah 5-1-4-8-9-2. Dimana x adalah nilai sembarang integer.

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

3.4. Analisis Aliran Daya

3.4.1. Tujuan

Dalam analisis aliran daya dilakukan perhitungan terhadap tegangan, arus, daya aktif dan daya reaktif, yang terdapat dalam berbagai titik pada saluran distribusi tenaga listrik.

Tujuan dari aliran daya : (William D.Stevenson, Jr,1996 : 182)

- Untuk menentukan daya aktif dan daya reaktif.
- Untuk mengetahui apakah semua peralatan pada sistem memenuhi batas-batas yang telah ditetapkan untuk operasi penyaluran daya.
- Untuk menetahui kondisi awal pada perencanaan sistem yang baru.
- Untuk menentukan daya yang mengalir di tiap saluran jaringan tenaga listrik.

Di dalam analisis aliran daya terdapat 3 jenis variabel :

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

3.4.2. Klasifikasi Bus

Tujuan aliran daya pada sistem tenaga listrik adalah untuk menghitung besar (magnitude) tegangan $|V|$ dan sudut phasa tegangan δ pada semua bus, sehingga dengan diketahuinya parameter-parameter tersebut akan dapat dihitung besar daya yang mengalir beserta rugi-ruginya.

Pada setiap bus jaringan terdapat parameter-parameter sebagai berikut :

1. Daya aktif, dinyatakan dengan P satuannya MW.
2. Daya reaktif, dinyatakan dengan Q satuannya MVAR.
3. Besar (magnitude) tegangan, dinyatakan dengan $|V|$ satuannya kV.
4. Sudut fasa tegangan, dinyatakan dengan δ satuannya derajat.

1 dan 2 menyatakan daya yang dibangkitkan oleh generator yang mengalir ke bus. Jika pada bus terdapat beban, maka daya tersebut menyatakan selisih antara daya yang dibangkitkan dengan daya beban.

Dalam aliran daya, pada setiap busnya perlu diketahui 2 parameter dari 4 parameter yang diperhitungkan. Dengan melihat kedua parameter tersebut dapat diketahui jenis busnya, yaitu : (William D.Stevenson, Jr,1996 : 182)

1. Bus beban

Pada bus ini daya nyata P dan daya reaktif Q diketahui, sementara magnitude tegangan $|V|$ dan sudut fasa δ dicari (dihitung).

2. Bus generator (rel tegangan konstan)

Pada bus ini daya nyata P dan magnitude tegangan $|V|$ diketahui, sementara daya reaktif Q dan sudut fasa δ dicari.

3. Bus berayun

Pada bus ini magnitude tegangan $|V|$ dan sudut fasa δ diketahui besarnya, sementara daya nyata P dan daya reaktif Q dicari. Sedangkan sudut fasa tegangan δ berharga nol, karena pada bus ini fasor tegangan dari bus dipakai sebagai referensi.

3.4.3. Metode Newton Raphson

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

Persamaan daya aktif dan reaktif pada bus i adalah : (William D.Stevenson, Jr,1996 : 187)

$$P_i = \sum_{k=1}^n |V_i V_k Y_{ik}| \cos(\delta_i - \delta_k - \theta_{ik}) \quad (3.1)$$

$$Q_i = \sum_{k=1}^n |V_i V_k Y_{ik}| \sin(\delta_i - \delta_k - \theta_{ik}) \quad (3.2)$$

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

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

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta |V| \end{bmatrix} \quad (3.3)$$

dengan :

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

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

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

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

H , L , M , N merupakan elemen-elemen off diagonal dan diagonal dari submatrik Jacobian yang dibentuk dengan mendefinisikan persamaan (3.1) dan (3.2).

Dimana :

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

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

Untuk menghitung selisih daya, maka mula-mula ditentukan harga awal tegangan simpul dan sudut fasanya, kemudian daya aktif dan reaktif dihitung dengan menggunakan persamaan (3.1) dan (3.2). Selisih daya antara daya yang telah ditentukan dengan daya hasil perhitungan ini merupakan perubahan daya yang terjadi pada simpul.

Magnitude tegangan dan sudut fasa yang diasumsikan ($|V|$) dan (δ) serta selisih daya yang dihitung (ΔP_i dan ΔQ_i) digunakan untuk memperoleh elemen-elemen matrik Jacobian.

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

$$|V^{k+1}| = |V|^k + \Delta|V|^k \quad (3.4)$$

$$\Delta^{k+1} = \delta^k + \Delta\delta^k \quad (3.5)$$

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

3.4.4. Prosedur Aliran Daya Newton Raphson

Metode *Newton-Raphson* dapat diringkaskan dalam langkah-langkah sebagai berikut : (William D.Stevenson, Jr,1996 : 191)

1. Tentukan nilai-nilai P_i (*ditegakan*) dan Q_i (*ditegakan*) yang mengalir kedalam sistem pada setiap rel untuk nilai yang ditentukan atau perkiraan dari besar dan sudut tegangan untuk iterasi pertama atau tegangan yang ditentukan paling akhir untuk iterasi berikutnya.
2. Hitung ΔP pada setiap rel.
3. Hitung nilai-nilai Jacobian dengan menggunakan nilai-nilai perkiraan atau yang ditentukan dari besar dan sudut tegangan dalam persamaan untuk turunan parsial yang ditentukan dengan differensiasi persamaan (3.1) dan (3.2).

4. Balikan Jacobian itu dan hitung koreksi-koreksi tegangan $\Delta\delta_k$ dan $\Delta|V_k|$ pada setiap rel.
5. Hitung nilai baru dari δ_k dan $|V_k|$ dengan menambahkan $\Delta\delta_i$ dan $\Delta|V_k|$ pada nilai sebelumnya.
6. Kembalilah ke langkah 1 dan ulangi proses itu dengan menggunakan nilai untuk besar dan sudut tegangan yang ditentukan paling akhir sehingga semua nilai ΔP dan ΔQ atau semua nilai $\Delta\delta$ dan $\Delta|V|$ lebih kecil dari suatu indeks ketetapan yang telah dipilih.

3.5. Perbaikan Pelayanan Menggunakan Metode *Parallel GA*

3.5.1. Konsep Dasar *Parallel GA*

Parallel Algoritma Genetika adalah metode pendekatan baru dari Algoritma Genetika untuk memecahkan pencarian nilai dalam sebuah masalah yaitu dengan cara menghubungkan secara *parallel*. *Parallel* Algoritma Genetika memiliki ruang permasalahan atau penelitian yang disimulasikan sangat luas jadi semakin banyak memparalelnya maka semakin besar atau luas pula mendapatkan solusi-solusi optimal ditemukan.

Algoritma Genetika pada mekanisme seleksi alam. Individu dari sebuah populasi dikodekan secara biner, Populasi pertama dibangkitkan secara random. Generasi baru dibuat dengan mengaplikasikan 3 operator berikut terhadap sebuah populasi yaitu : seleksi, crossover dan mutasi dimana reproduksi adalah proses yang tergantung pada fungsi tujuan (*objective function*).

3.5.2. Pengkodean

Pengkodean adalah suatu cara untuk menentukan posisi *switch open* atau *closed* pada cabang tertentu pada konfigurasi jaringan radial dengan memberikan kode biner 0 dan 1. Jika posisi *switch open (off)* diberikan kode biner 0, sedangkan jika posisi *switch closed (on)* diberikan kode biner 1. Skema pengkodean diilustrasikan oleh tabel 3-2, dimana ‘o’ menunjukkan *switch* terbuka dan ‘c’ menunjukkan *switch* tertutup, ‘m’ menunjukkan jumlah total dari mesh dan sama dengan total jumlah *switch* yang terbuka. Untuk mempertahankan struktur jaringan radial, hanya satu *switch* yang dibuka pada gambar masing-masing mesh. Notasi $Z = [w_1, w_2, \dots, w_n]$ adalah [23 24 25 26].

Tabel 3-2
Skema Pengkodean

m	1	2	3	4
s	1 2 3 23 10 9 8 7 7 8 24 12 11 11 13 14 25 18 17 4 26 15 13 12 24 9 10 23			
a	c c c o c c c c c c o c c c o c c c c c c c c			
w	23	24	25	26

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

Sumber : Mitsuo Gen, Runwei Cheng, 1994 : 51

3.5.3. Fungsi *Fitness*

Fungsi “*fitness*” meliputi kemampuan untuk membandingkan solusi dari satu generasi ke generasi yang lain.

3.5.4. Migrasi

Migrasi merupakan operator tambahan pada *Parallel Algorima Genetika*. Pada proses disini Migrasi berfungsi untuk pencarian nilai yang terbaik dengan cara disaling tukarkarkan diantara subpopulasi-subpopulasi selama proses

післядієві

після якого він вже не може піддаватися змінам та вже є вільною речовиною. Це відбувається в результаті зміни структури макромолекули, яка відбувається під дією фізичних чинників (тепла, хімічного або електричного поля тощо). У результаті зміни структури макромолекули відбувається зміна фізичних властивостей матеріалу. Це може відбутися в результаті зміни структури макромолекули, яка відбувається під дією фізичних чинників (тепла, хімічного або електричного поля тощо). У результаті зміни структури макромолекули відбувається зміна фізичних властивостей матеріалу.

$$\{0, 25, 50, 75\} \text{ ділить } \{0, \dots, 100\} = N \text{ істотних точок гідрозиму}$$

С-С ідеї**С-С ідеї**

A	B	C	D
0, 25, 50, 75	10, 20, 30, 40, 50, 60, 70, 80, 90, 100	10, 20, 30, 40, 50, 60, 70, 80, 90, 100	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
25	50	75	100

однозначно відповідає $\{0, 25, 50, 75\}$ ділить $\{0, \dots, 100\}$ на N істотних точок гідрозиму

Супорядженість (або Розподіл) функції $f(x)$ від x відповідає

С-С ідеї

це відповідає зміні функції $f(x)$ від x відповідає зміні функції $f(x)$ від x .

Відповідно до цього відповідає зміні функції $f(x)$ від x відповідає зміні функції $f(x)$ від x .

Ідеї С-С

зміні функції $f(x)$ від x відповідає зміні функції $f(x)$ від x . Це відповідає зміні функції $f(x)$ від x відповідає зміні функції $f(x)$ від x . Це відповідає зміні функції $f(x)$ від x відповідає зміні функції $f(x)$ від x .

optimalisasi. Nilai rerata penyesuaian *Parallel* Algoritma Genetika dengan proses Migrasi adalah lebih tinggi bila dibandingkan dengan nilai *Parallel* Algoritma Genetika tanpa Migrasi.

3.5.5. Statistik

Dengan perhitungan *Load Flow*, persamaan 2.7 akan digunakan sebagai fungsi *fitness* dengan menambahkan batasan sebagai berikut : (Whei Min lin, Hong Chan Chin, 1998 : 870)

$$F = P_{\text{loss}}(S_v) + \sum_{i=1}^{N_b} \lambda_{i,i} * (I_i - I_{\text{lim}})^2 + \sum_{j=1}^N \lambda_{v,j} * (V_i - V_{\text{lim}})^2 \quad (3.6)$$

3.6. *Parallel* Algoritma Genetika

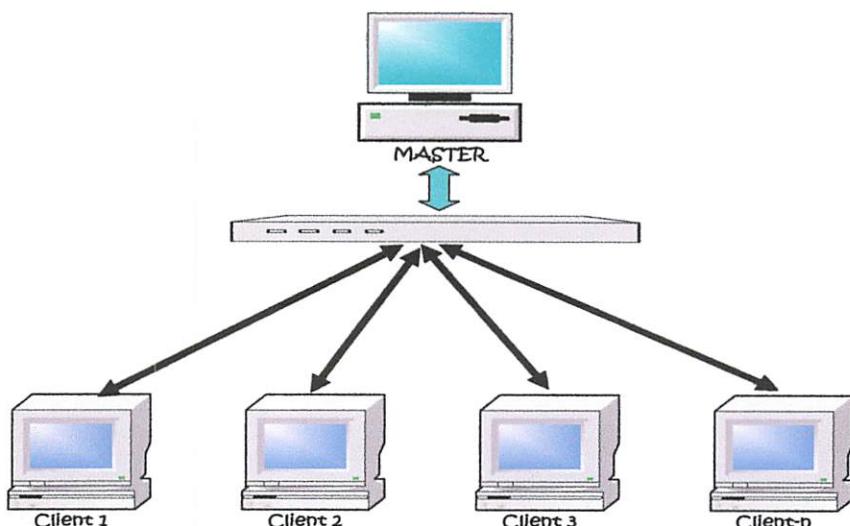
Ukuran populasi adalah salah satu faktor yang akan mempengaruhi performansi dari Algoritma Genetika untuk mencari solusi yang optimal. Jika ukuran populasi yang digunakan besar, kemungkinan untuk menghasilkan solusi yang optimal dengan mutasi dan kompetisi tinggi. Hal ini jelas akan memerlukan perhitungan waktu yang relatif lama. Untuk memperbaiki kecepatan perhitungan ketika mempertahankan kualitas solusi yang sama, maka perlu diimplementasikan *Parallel* Algoritma Genetika.

Dasar pemikiran dari *Parallel* Algoritma Genetika ini adalah untuk membagi *initial population* ke beberapa sub-populasi. Selama ukuran sub-populasi ini lebih kecil dari *initial populations*, maka waktu perhitungannya akan semakin cepat.

➤ **Prosedur dari Paralel Algoritma Genetika**

1. Dalam pengujian program ini menggunakan 5 unit PC, 1 PC digunakan sebagai Server, sedangkan 4 PC sebagai Client. Komunikasi antar PC menggunakan IP, dan tiap PC Server dan Client ditentukan IP *address*.
2. PC Server menginisialisasikan data yang akan diolah dan mengirimkan data-data serta membagi keseluruhan populasi pada tiap-tiap PC Client.
3. PC Client mengolah data sampai didapatkan hasil yang optimum. Dan mengirimkan kembali hasilnya ke PC Server.
4. Hasil data dari Client diolah kembali oleh PC Server untuk diambil yang terbaik. Setelah menggabungkan sub-populasi dari tiap-tiap Client, dan melakukan seleksi individu oleh Server sampai terpilih individu yang memiliki nilai tertinggi dari induk (Server) dan populasinya di mutasikan ke Client untuk membentuk generasi selanjutnya.

Konfigurasi dari Master-Client Paralel Algoritma Genetika terlihat pada gambar 3.7. berikut ini :



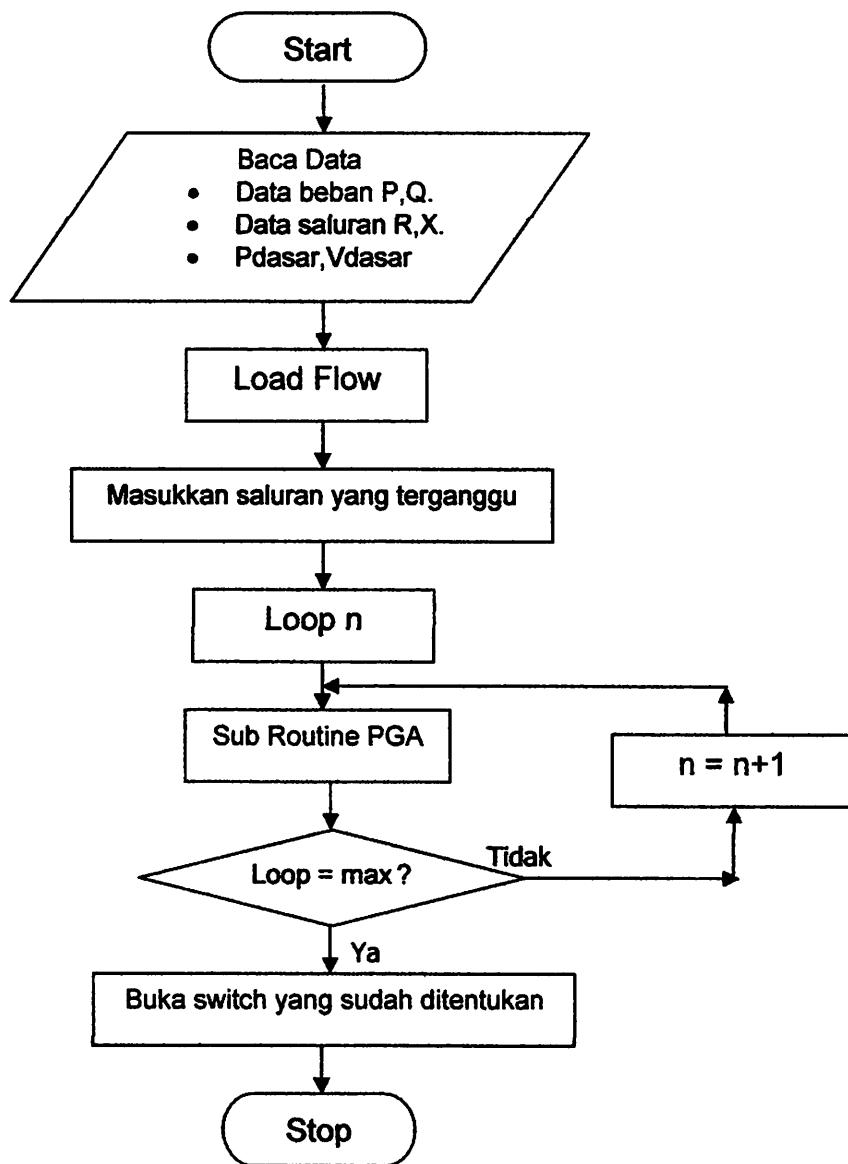
Gambar 3-7. Konfigurasi Master-Client Parallel Genetic Algorithm

3.7. Algoritma Program

3.7.1. Algoritma Pemecahan Masalah

1. Memasukkan data – data:
 - Data beban P,Q.
 - Data saluran R,X.
 - P dasar,V dasar
2. Menjalankan proses *Load Flow* dengan metode *Newton Raphson*.
3. Memasukkan data saluran yang terganggu.
4. Menutup semua *tie switch* pada jaringan radial dan mengubah menjadi jaringan loop.
5. Sub Routine PGA.
6. Memeriksa apakah loop sudah maksimal, apabila iya maka ke langkah selanjutnya, apabila tidak maka kembali ke langkah 4 dengan menambahkan nilai indeks loop $n = 1$.
7. Membuka *switch* yang sudah ditentukan.
8. Operasi selesai.

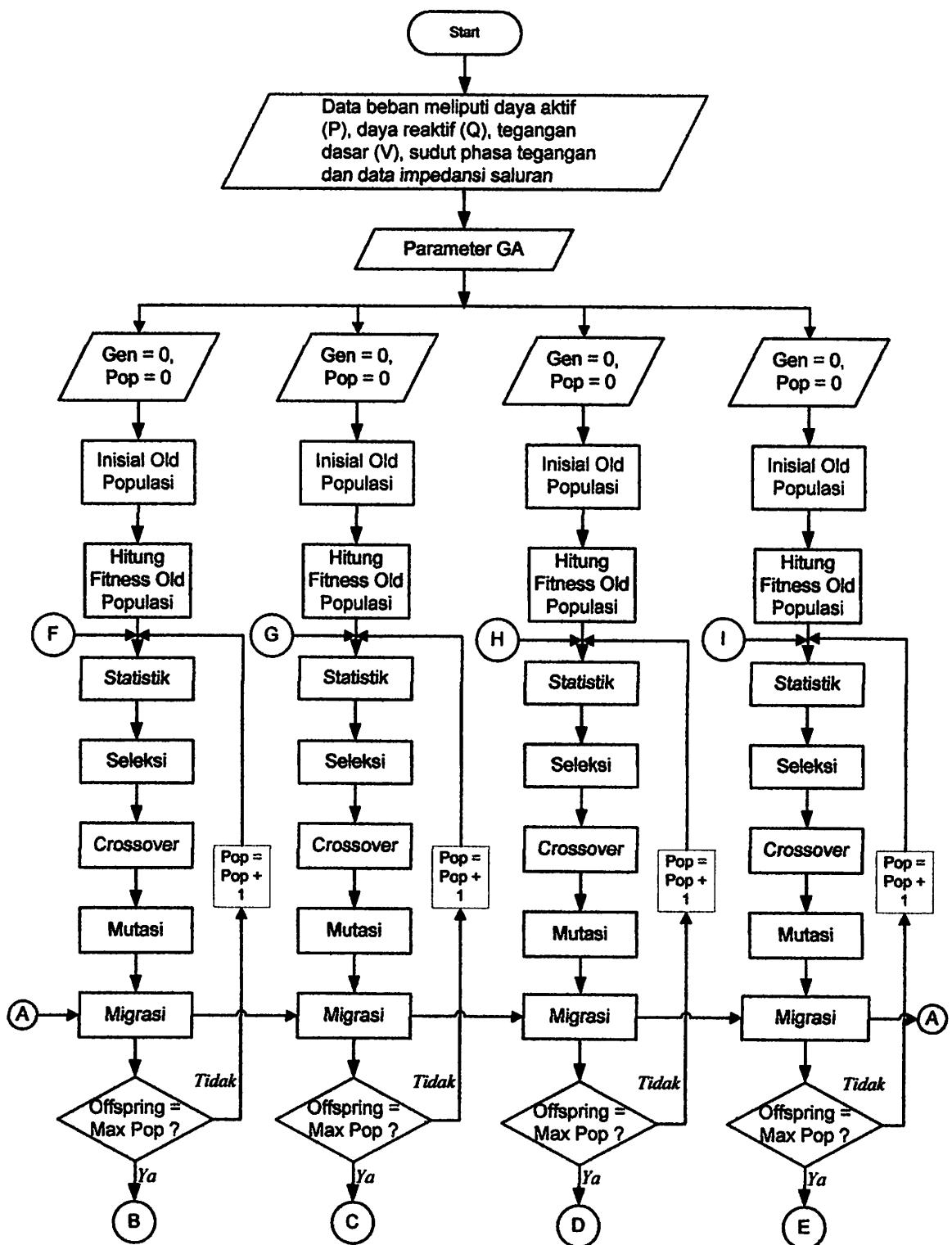
Untuk lebih jelasnya, flowchart pemecahan masalah dapat dilihat pada gambar 3-7 berikut ini.

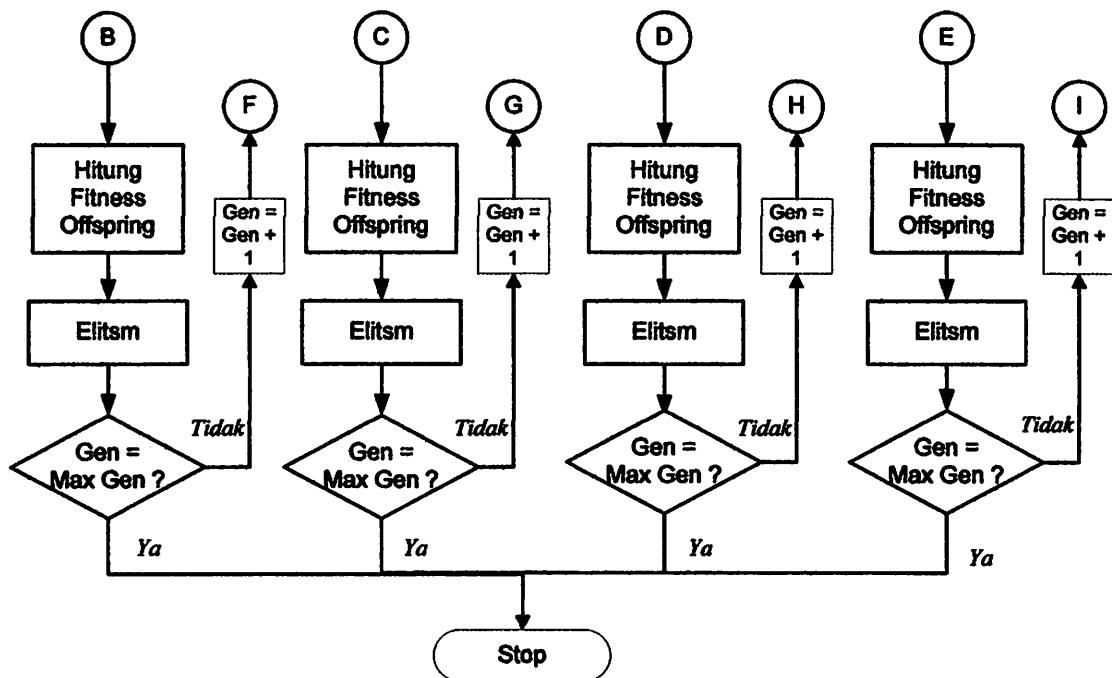


Gambar 3-8. Flowchart Pemecahan Masalah Secara Umum

3.7.2. Algoritma Parallel *Genetic Algorithm*

1. Memasukkan inputan data beban yang meliputi tegangan, sudut phasa tegangan, daya aktif (P), daya reaktif (Q) dan data impedansi saluran.
2. Menetukan parameter inputan Algortima Genetika yang meliputi jumlah populasi, maksimum generasi, nilai kemungkinan *crossover*, nilai kemungkinan mutasi dan panjang kromosom tiap-tiap individu.
3. Generasi = 0, Populasi = 0.
4. Melakukan *fitness* dari kromosom tiap-tiap individu.
5. Melakukan proses statistik.
6. Melakukan proses seleksi.
7. Melakukan proses *crossover*.
8. Melakukan proses mutasi.
9. Melakukan proses migrasi.
10. Proses no. 6,7,8,9 diulang apakah *offspring* sudah mencapai max populasi.
 - a. Ya, Lanjutkan ke langkah 11
 - b. Tidak, hitung Pop + 1 kembali ke langkah 5
11. Menghitung *fitness* dari *offspring*.
12. Melakukan proses *elitism*.
13. Proses 11,12 diulang apakah Max Gen sudah terpenuhi
 - a. Jika “ tidak “ maka generasi = gen + 1, kembali ke langkah 6.
 - b. Jika “ Ya “ selesai.





Gambar 3-9. Flowchart Parallel Genetic Algorithm

BAB IV

ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER DARI GI BLIMBING MALANG

4.1. Sistem Distribusi Gardu Induk Blimbings

Data yang digunakan, diperoleh dari Gardu Induk Blimbings, dimana Out Going dari Gardu Induk tersebut merupakan jaringan distribusi primer 20 kV yang memiliki tiga buah trafo dan sembilan buah penyulang. Data yang dijadikan sebagai data input dalam program perhitungan aliran daya dan *Parallel Genetic Algorithm* adalah data dari Trafo II GI blimbings, yang terdiri empat buah penyulang yaitu: Penyulang Mawar, Penyulang Glintung, Penyulang Asahan dan Penyulang Wendit. Sebelum perhitungan dilakukan, terlebih dahulu ditetapkan dari gambar single line diagram yang ada, bahwa yang dijadikan sebagai *slack bus* adalah Busbar dari GI Blimbings, sedangkan *node-node* yang ada disepanjang saluran (yang ada trafo distribusinya) dipandang sebagai *load bus*.

Agar memudahkan perhitungan maka digunakan sistem per-unit (pu), dimana dasar yang digunakan :

- Tegangan dasar : 20 kV
- Daya dasar : 100 MVA
- Penghantar SUTM : AAAC 150 mm²
- Impedansi Penghantar : 0,2162 + J 0,3305 Ω/Km
- Cos φ : 0,8
- Sin φ : 0,6

BAB VI

**DISTRIBUSI PRIMER DARI OPTIMIZING MELALUI
ANALISIS PERBANDINGAN PERAMAN RASA DAN RASA**

4.1. Sistem Optimalisasi Cari dan Mengoptimalkan

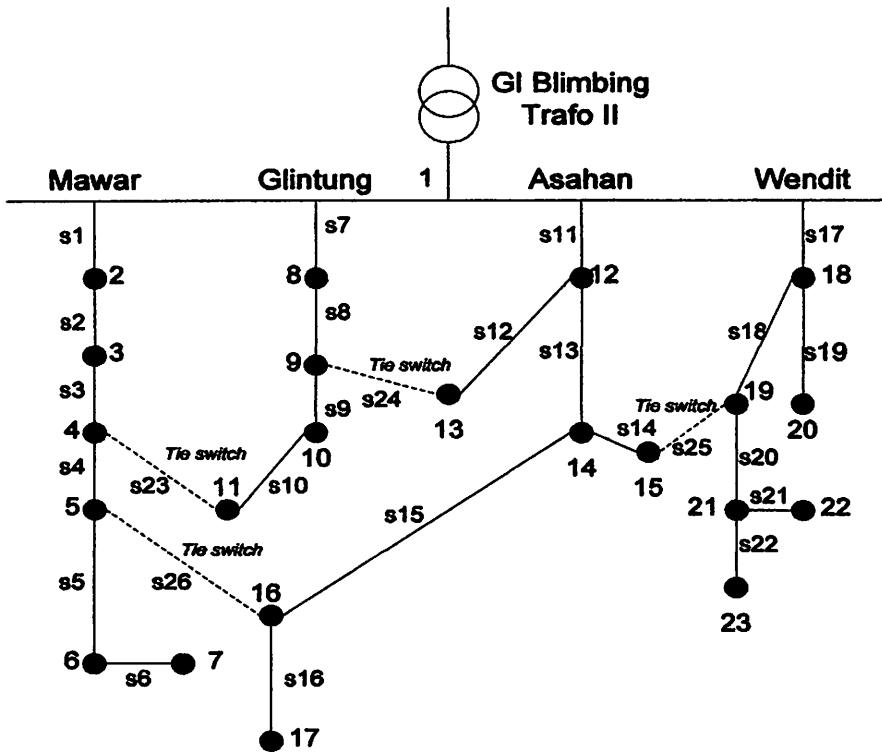
Dua tipe analisis yang dilakukan dalam sistem ini adalah analisis optimalisasi dan analisis sensitivitas. Analisis optimalisasi dilakukan dengan menggunakan teknik optimisasi matematika dan teknik optimisasi operasional. Analisis sensitivitas dilakukan dengan mengetahui bagaimana perubahan variabel pada sistem mempengaruhi hasil optimasi. Analisis sensitivitas dilakukan dengan mengetahui bagaimana perubahan variabel pada sistem mempengaruhi hasil optimasi. Analisis sensitivitas dilakukan dengan mengetahui bagaimana perubahan variabel pada sistem mempengaruhi hasil optimasi.

Analisis optimalisasi berfungsi untuk mendapatkan solusi terbaik dalam sistem yang dimaksud.

Analisis sensitivitas berfungsi untuk mendapatkan solusi terbaik dalam sistem yang dimaksud.

Analisis optimalisasi dan analisis sensitivitas

Konfigurasi dari 4 penyulang tersebut dapat dilihat pada gambar 4.1. berikut :



Gambar 4-1. Konfigurasi Jaringan Radial GI Blimbings

Sumber : PLN (Persero) Distribusi JATIM-Cabang Malang

4.2. Perhitungan Pembebanan

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

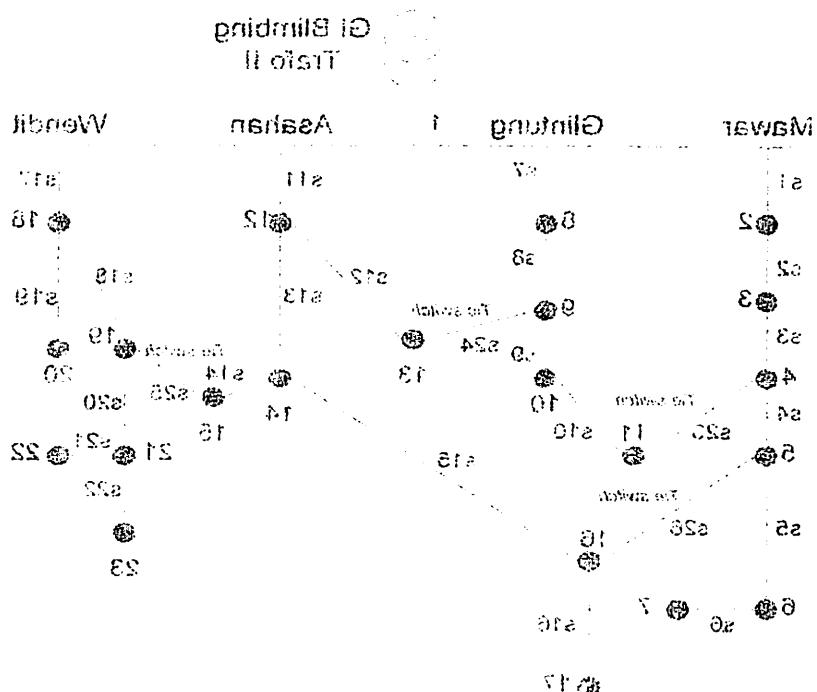
- $P = \text{Jumlah Beban} \times \cos \phi$

$$P = 1784,55 \times 0,8 = 1427,6 \text{ kW} = 1,4276 \text{ MW}$$

- $Q = \text{Jumlah Beban} \times \sin \phi$

$$Q = 1784,55 \times 0,6 = 1070,7 \text{ kVAR} = 1,0707 \text{ MVAR}$$

Konfigurasi day + berlariang terhadap dapat dilihat pada gambar 4.1, perintah



Gambar 4.1 Konfigurasi strukturnya Rangkaian (4) lapisan

Sumber : PPT (Fakultas) Dosenku ATM-Gadung Mulyadi

4.2 Pengantar Jaringan Saraf Sifat

Dari persamaan ini maka - jumlah berlariang dapat dilihat pada gambar

Jumlah berlariang pada A-I, Dari persamaan dapat dilihat pada 4

berlariang Wawan dapat dilihat dengan sebagaimana berikut :

$$\phi = \text{jumlah lapisan} \times \text{Cacat} = 4 \times 4 = 16$$

$$W = 15452 \times 1424 = 3.0 \times 1424 = 42484 \text{ MW}$$

$$\phi = \text{jumlah lapisan} \times \text{Size} = Q = 16$$

$$W = 15452 \times 1424 = 3.0 \times 1424 = 42484 \text{ MW}$$

Dengan cara yang sama, maka beban untuk masing – masing bus akan diperoleh hasil seperti tabel 4.1 berikut ini

Tabel 4-1
Hasil Perhitungan Pembebanan

BUS PANGKAL	BUS UJUNG	P BUS UJUNG (MW)	Q BUS UJUNG (MVAR)
1	2	1,4276	1,0707
2	3	0,7655	0,5741
3	4	1,4842	1,1132
4	5	0,8594	0,6446
5	6	0,1661	0,1245
6	7	0,6190	0,4643
1	8	0,3342	0,5206
8	9	0,7353	0,5515
9	10	0,2871	0,2153
10	11	0,4398	0,3299
1	12	1,4908	1,1181
12	13	1,4538	1,0903
12	14	0,1972	0,1479
14	15	0,6540	0,4905
14	16	1,4906	1,1179
16	17	0,0450	0,0338
1	18	0,1324	0,0993
18	19	0,3921	0,2941
18	20	2,8164	2,1123
19	21	0,1776	0,1332
21	22	0,6146	0,4609
21	23	0,9675	0,7256
4	11	-	-
9	13	-	-
15	19	-	-
5	16	-	-

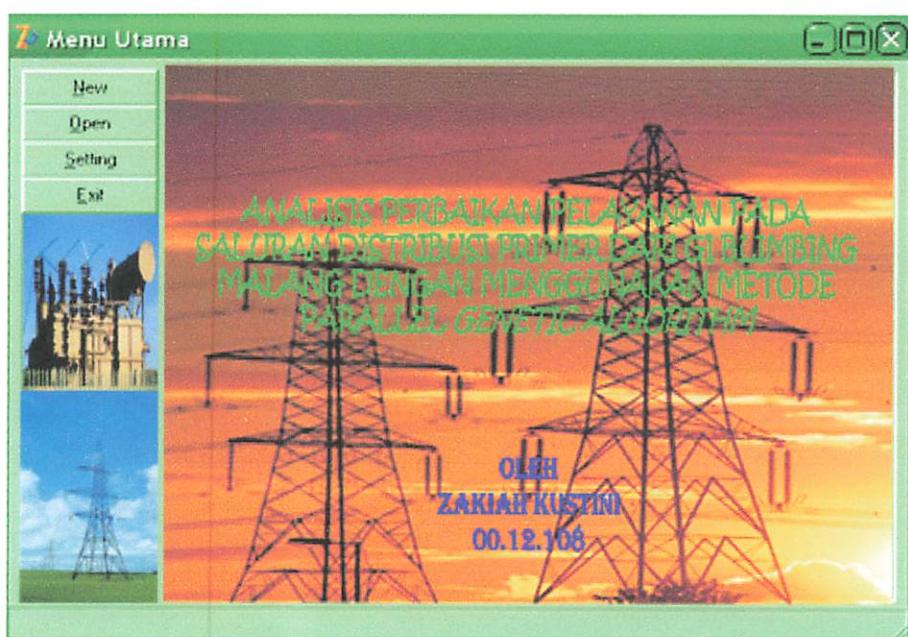
4.3. Analisis Hasil Dengan Menggunakan Program

4.3.1. Tampilan Program

Program dalam skripsi ini dijalankan dengan menggunakan program *Borland Delphi versi 7.0* dan diaplikasikan pada komputer berprosesor AMD Athlon 850 MHz dengan Memory 256 Mb yang terangkai secara *Parallel*. Mengenai jalannya program ikuti prosedur program sebagai berikut :

1. Tampilan utama dari program

Tampilan utama dari program terlihat pada gambar 4.2. berikut ini.



Gambar 4-2. Tampilan Utama Program

2. Menekan tombol *setting* untuk menghubungkan *Server* dengan *Client*.

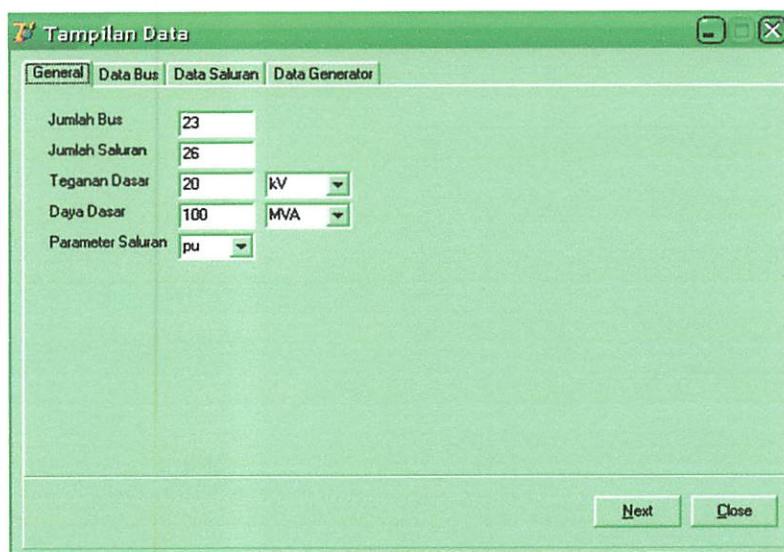
Tampilan *setting* PC *Client* terlihat pada gambar 4.3. berikut ini.



Gambar 4-3. Tampilan *Setting PC Client*

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

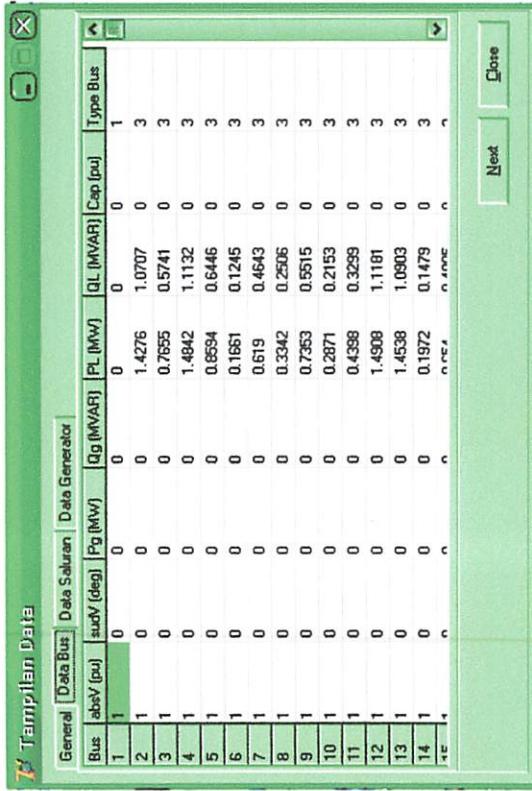
Tampilan inputan data terlihat pada gambar 4.4. berikut ini.



Gambar 4-4. Tampilan Inputan Data

4. Kemudian tekan tombol data pembebanan

Tampilan data pembebanan terlihat pada gambar 4.5. berikut ini.

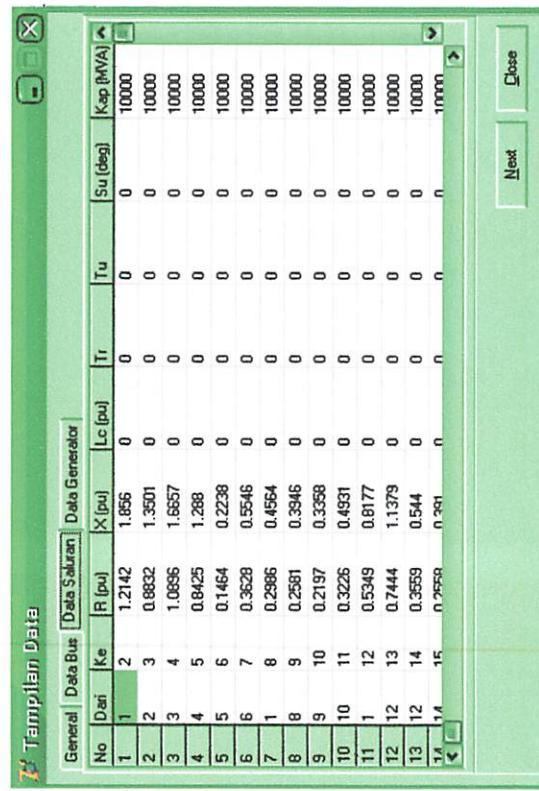


No	Bus	absV [pu]	sdvV [deg]	Pg [MW]	Qg [MVAr]	PL [MW]	QL [MVAr]	Cap [pu]	Type Bus
1	1	0	0	0	0	0	0	0	1
2	1	0	0	0	0	1.4276	1.0707	0	3
3	1	0	0	0	0	0.7655	0.5741	0	3
4	1	0	0	0	0	1.4842	1.1132	0	3
5	1	0	0	0	0	0.6894	0.6446	0	3
6	1	0	0	0	0	0.1681	0.1245	0	3
7	1	0	0	0	0	0.6119	0.4643	0	3
8	1	0	0	0	0	0.3342	0.2506	0	3
9	1	0	0	0	0	0.7263	0.5615	0	3
10	1	0	0	0	0	0.2871	0.2153	0	3
11	1	0	0	0	0	0.4398	0.3299	0	3
12	1	0	0	0	0	1.4908	1.1191	0	3
13	1	0	0	0	0	1.4538	1.0903	0	3
14	1	0	0	0	0	0.1972	0.1479	0	3
15	1	0	0	0	0	0	0	0	3

Gambar 4-5. Data Pembebanan

5. Kemudian tekan tombol data saluran

Tampilan data saluran terlihat pada gambar 4.6. berikut ini.



No	Dari	Ket	R [pu]	X [pu]	Lc [pu]	Tr	Tu	Su [deg]	Kap [MVA]
1	1	2	1.2142	1.856	0	0	0	0	10000
2	2	3	0.8832	1.3501	0	0	0	0	10000
3	3	4	1.0896	1.6657	0	0	0	0	10000
4	4	5	0.8425	1.288	0	0	0	0	10000
5	5	6	0.1464	0.2238	0	0	0	0	10000
6	6	7	0.3628	0.5546	0	0	0	0	10000
7	1	8	0.2986	0.4564	0	0	0	0	10000
8	8	9	0.2581	0.3946	0	0	0	0	10000
9	9	10	0.2197	0.3358	0	0	0	0	10000
10	10	11	0.3226	0.4931	0	0	0	0	10000
11	1	12	0.5349	0.8177	0	0	0	0	10000
12	12	13	0.7444	1.1379	0	0	0	0	10000
13	12	14	0.3659	0.5644	0	0	0	0	10000
14	14	15	0.2958	0.3261	0	0	0	0	10000

Gambar 4-6. Data Saluran

6. Tekan tombol hitung maka dengan *Newton Rapshon* dapat melihat hasil perhitungan aliran daya pada kondisi awal.
- Tampilan loadflow awal, aliran daya dan summary sebelum restorasi terlihat pada gambar 4.7. berikut ini.

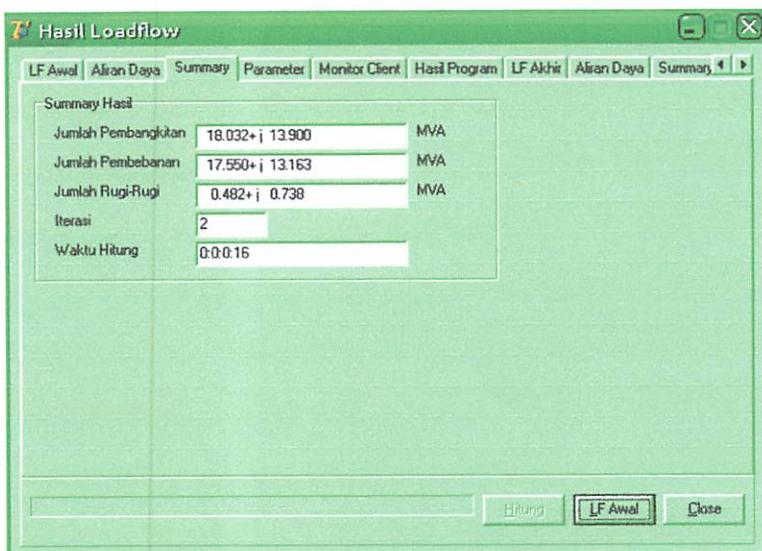
Hasil Loadflow											
LF Awal	Allan Daya	Summary	Parameter	Monitor Client	Hasil Program	LF Akhir	Allan Daya	Summary			
Bus	absV [pu]	sudut [deg]	Pq [MW]	Qg (MVAR)	Pe [MW]	Ql (MVAR)	Sus [pu]	Type Bus			
1	1.00000	0.00000	18.032	13.900	0.000	0.000	0.000	1			
2	0.96262	-0.74855	0.000	0.000	1.428	1.071	0.000	3			
3	0.94255	-1.17104	0.000	0.000	1.765	0.574	0.000	3			
4	0.93289	-1.60771	0.000	0.000	1.484	1.113	0.000	3			
5	0.91445	-1.79052	0.000	0.000	0.059	0.645	0.000	3			
6	0.91377	-1.80582	0.000	0.000	0.186	0.125	0.000	3			
7	0.91245	-1.85079	0.000	0.000	0.619	0.464	0.000	3			
8	0.90711	-0.05959	0.000	0.000	0.334	0.251	0.000	3			
9	0.89507	-0.10242	0.000	0.000	0.735	0.551	0.000	3			
10	0.89421	-0.12042	0.000	0.000	0.287	0.215	0.000	3			
11	0.89344	-0.13644	0.000	0.000	0.440	0.330	0.000	3			
12	0.89334	-0.32315	0.000	0.000	1.451	1.118	0.000	3			
13	0.87841	-0.44847	0.000	0.000	1.454	1.050	0.000	3			
14	0.87367	-0.42138	0.000	0.000	0.197	0.148	0.000	3			
15	0.87075	-0.44083	0.000	0.000	0.654	0.490	0.000	3			
					

Hasil Loadflow											
LF Awal	Allan Daya	Summary	Parameter	Monitor Client	Hasil Program	LF Akhir	Allan Daya	Summary			
No	Dari	Ke	P [MW]	Q [MVAR]	Ausute [A]	Ausim [A]	Dari	Ke	P [MW]	Q [MVAR]	
1	1	2	5.537	4.412	279.632	220.658	2	1	5.442	4.412	
2	2	3	4.014	3.106	206.383	164.058	3	2	3.953	3.106	
3	3	4	3.187	2.439	166.383	132.819	4	3	3.138	2.439	
4	4	5	1.654	1.251	87.718	70.290	5	4	1.644	1.251	
5	5	6	0.785	0.590	41.903	33.605	6	5	0.785	0.590	
6	6	7	0.619	0.465	33.044	26.512	7	6	0.618	0.465	
7	1	8	1.803	1.353	90.154	67.876	8	1	1.799	1.353	
8	8	9	1.485	1.101	73.409	55.282	9	8	1.463	1.101	
9	9	10	0.728	0.566	36.511	27.515	10	9	0.727	0.566	
10	10	11	0.440	0.330	22.096	16.857	11	10	0.440	0.330	
11	1	12	5.414	4.125	270.712	206.257	12	1	5.352	4.125	
12	12	13	1.460	1.100	73.855	56.298	13	12	1.454	1.100	
13	12	14	2.401	1.812	121.452	92.741	14	12	2.333	1.812	
14	14	15	0.654	0.451	33.275	25.314	15	14	0.654	0.451	
			

Hitung

LF Awal

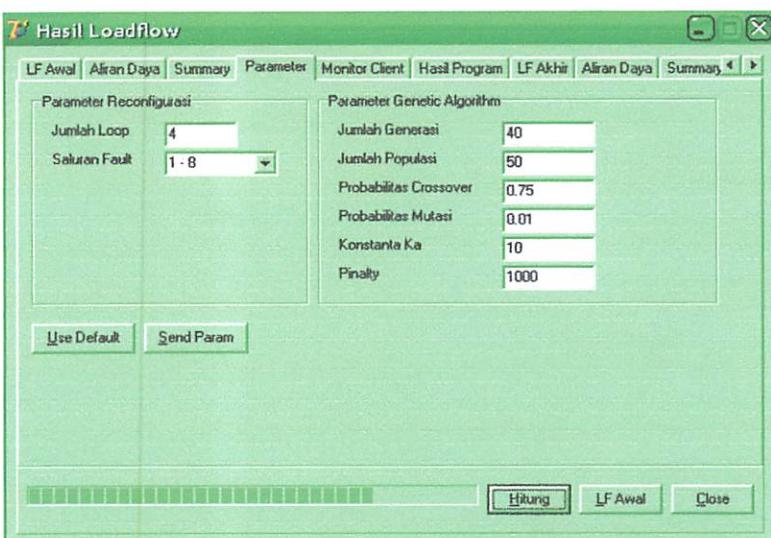
Close



Gambar 4-7. Hasil Aliran Daya Sebelum Restorasi

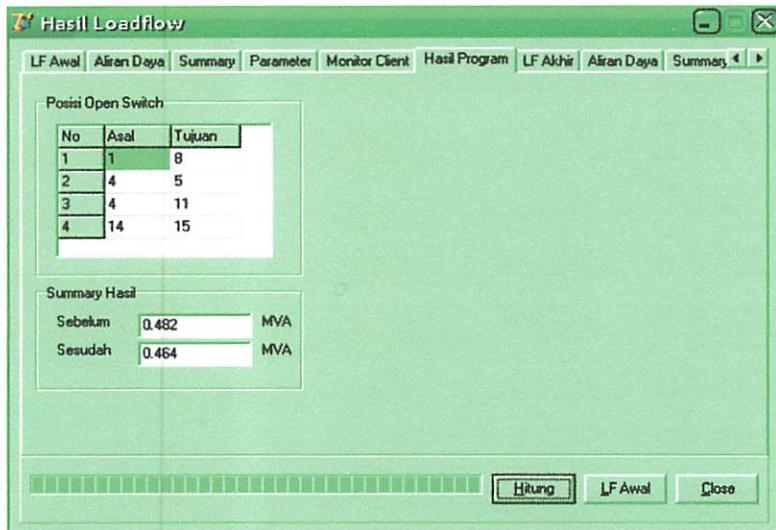
- Kemudian tekan *Parameter* kemudian tekan tombol *Use Default*, pilih saluran gangguan tekan tombol *Send Param* lalu tombol *Hitung*.

Tampilan parameter GA yang digunakan terlihat pada gambar 4.8. berikut ini.



Gambar 4-8. Parameter Algoritma Genetika

Tampilan hasil *switch* yang dibuka terlihat pada gambar 4.9. berikut ini.

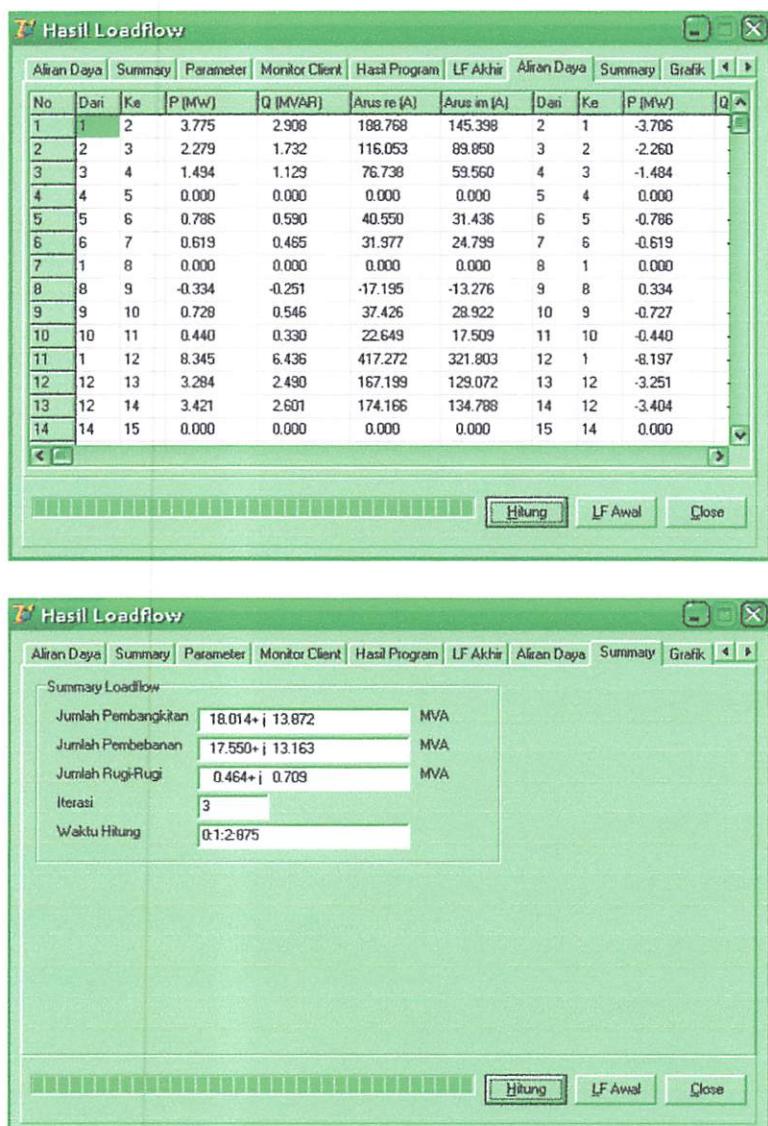


Gambar 4-9. Hasil GA dalam menentukan *Switch* Yang Dibuka

8. Kemudian pilih tombol *LF Akhir* dan tombol *Aliran Daya* serta *Summary*.

Tampilan loadflow akhir, aliran daya dan summary sesudah restorasi terlihat pada gambar 4.10. berikut ini.

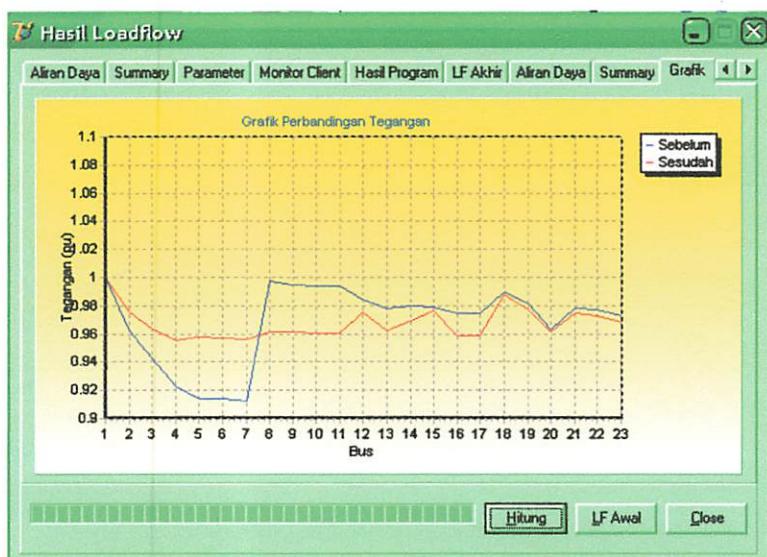
Bus	absV (pu)	sudV (deg)	Pg (MW)	Qg (MVAR)	PL (MW)	QL (MVAR)	Sups (pu)	Type Bus
1	1.00000	0.00000	18.014	13.872	0.000	0.000	0.000	1
2	0.97509	-0.51066	0.000	0.000	1.428	1.071	0.000	3
3	0.96394	-0.74640	0.000	0.000	0.765	0.574	0.000	3
4	0.95484	-0.94232	0.000	0.000	1.484	1.113	0.000	3
5	0.95787	-0.87609	0.000	0.000	0.859	0.645	0.000	3
6	0.95723	-0.89007	0.000	0.000	0.165	0.125	0.000	3
7	0.95597	-0.91743	0.000	0.000	0.619	0.464	0.000	3
8	0.96135	-0.80230	0.000	0.000	0.334	0.251	0.000	3
9	0.96183	-0.79189	0.000	0.000	0.735	0.551	0.000	3
10	0.96094	-0.81115	0.000	0.000	0.287	0.215	0.000	3
11	0.96015	-0.82829	0.000	0.000	0.440	0.330	0.000	3
12	0.97572	-0.49641	0.000	0.000	1.491	1.118	0.000	3
13	0.96221	-0.78379	0.000	0.000	1.454	1.090	0.000	3
14	0.96896	-0.63816	0.000	0.000	0.197	0.148	0.000	3
15	0.97665	-0.48120	0.000	0.000	0.654	0.490	0.000	3



Gambar 4-10. Hasil Aliran Daya Sesudah Restorasi

9. Kemudian tekan tombol Grafik untuk melihat perbandingan tegangan sebelum dan sesudah terjadi gangguan.

Tampilan grafik perbandingan tegangan terlihat pada gambar 4.11. berikut ini.



Gambar 4-11. Grafik Perbandingan Tegangan

4.3.2. Hasil Perhitungan Pada Kondisi Awal

Tabel 4-2 sampai tabel 4-13 berikut ini memperlihatkan hasil perhitungan aliran daya sebelum perbaikan pelayanan (kondisi normal) :

1. Penyulang Mawar

Tabel 4-2
Hasil Perhitungan Tegangan, Pembangkitan dan Pembebatan
Sebelum Restorasi

BUS	TEGANGAN		DAYA PEMBANGKITAN		BEBAN	
	ABS (PU)	SUDUT(DEG)	MW	MVAR	MW	MVAR
1	1	0	5.597	4.412	0	0
2	0,96262	-0,74855	-	-	1,428	1,071
3	0,94255	-1,17104	-	-	0,765	0,574
4	0,92259	-1,60771	-	-	1,484	1,113
5	0,91445	-1,79052	-	-	0,859	0,645
6	0,91377	-1,80582	-	-	0,166	0,125
7	0,91245	-1,83579	-	-	0,619	0,464

Jumlah Pembangkitan : 5,597 MW 4,412 MVAR

Jumlah Pembebatan : 5,321 MW 3,992 MVAR

Tabel 4-3
Hasil Perhitungan Aliran Daya Pada Tiap Saluran Sebelum Restorasi

SALURAN		DAYA		SALURAN		DAYA	
DARI	KE	P (MW)	Q (MVAR)	DARI	KE	P (MW)	Q (MVAR)
1	2	5.597	4.412	2	1	-5.442	-4.176
2	3	4.014	3.106	3	2	-3.953	-3.012
3	4	3.187	2.439	4	3	-3.138	-2.364
4	5	1.654	1.251	5	4	-1.644	-1.235
5	6	0.785	0.590	6	5	-0.785	-0.590
6	7	0.619	0.465	7	6	-0.618	-0.464

Tabel 4-4
Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi

SALURAN		RUGI-RUGI SALURAN	
DARI	KE	P (MW)	Q (MVAR)
1	2	0,155	0,236
2	3	0,061	0,094
3	4	0,049	0,075
4	5	0,010	0,016
5	6	0	0
6	7	0,001	0,001

Total Rugi-Rugi Penyulang Mawar : 0,276 MW + 0,422 MVAR

2. Penyulang Glintung

Tabel 4-5
Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan
Sebelum Restorasi

BUS	TEGANAN		DAYA PEMBANGKITAN		BEBAN	
	ABS (PU)	SUDUT(DEG)	MW	MVAR	MW	MVAR
1	1	0	1.803	1.358	0	0
8	0,99711	-0,05999	-	-	0,334	0,251
9	0,99507	-0,10242	-	-	0,735	0,551
10	0,99421	-0,12042	-	-	0,287	0,215
11	0,99344	-0,13644	-	-	0,440	0,330

Jumlah Pembangkitan : 1,803 MW 1,358 MVAR

Jumlah Pembebanan : 1,796 MW 1,347 MVAR

Tabel 4-6
Hasil Perhitungan Aliran Daya Pada Tiap Saluran Sebelum Restorasi

SALURAN		DAYA		SALURAN		DAYA	
DARI	KE	P (MW)	Q (MVAR)	DARI	KE	P (MW)	Q (MVAR)
1	8	1.803	1.358	8	1	-1.799	-1.352
8	9	1.465	1.101	9	8	-1.463	-1.098
9	10	0.728	0.546	10	9	-0.727	-0.546
10	11	0.440	0.330	11	10	-0.440	-0.330

Tabel 4-7
Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi

SALURAN		RUGI-RUGI SALURAN	
DARI	KE	P (MW)	Q (MVAR)
1	8	0,004	0,006
8	9	0,002	0,003
9	10	0,001	0
10	11	0	0

Total Rugi-Rugi Penyulang Glintung : 0,007 MW + 0,009 MVAR

3. Penyulang Asahan

Tabel 4-8
Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan
Sebelum Restorasi

BUS	TEGANAN		DAYA PEMBANGKITAN		BEBAN	
	ABS (PU)	SUDUT(DEG)	MW	MVAR	MW	MVAR
1	1	0	5.414	4.125	0	0
12	0,98434	-0,32315	-	-	1,491	1,118
13	0,97841	-0,44847	-	-	1,454	1,090
14	0,97967	-0,42138	-	-	0,197	0,148
15	0,97875	-0,44083	-	-	0,654	0,490
16	0,97465	-0,52782	-	-	1,491	1,118
17	0,97464	-0,52797	-	-	0,045	0,034

Jumlah Pembangkitan : 5,414 MW

4,125 MVAR

Jumlah Pembebanan : 5,332 MW

3,998 MVAR

Tabel 4-9
Hasil Perhitungan Aliran Daya Pada Tiap Saluran Sebelum Restorasi

SALURAN		DAYA		SALURAN		DAYA	
DARI	KE	P (MW)	Q (MVAR)	DARI	KE	P (MW)	Q (MVAR)
1	12	5.414	4.125	12	1	-5.352	-4.031
12	13	1.460	1.100	13	12	-1.454	-1.090
12	14	2.401	1.812	14	12	-2.393	-1.800
14	15	0.654	0.491	15	14	-0.654	-0.491
14	16	1.541	1.160	16	14	-1.536	-1.152
16	17	0.045	0.034	17	16	-0.045	-0.034

Tabel 4-10
Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi

SALURAN		RUGI-RUGI SALURAN	
DARI	KE	P (MW)	Q (MVAR)
1	12	0,062	0,094
12	13	0,006	0,010
12	14	0,008	0,012
14	15	0	0
14	16	0,005	0,008
16	17	0	0

Total Rugi-Rugi Penyulang Asahan : 0,081 MW + 0,124 MVAR

4. Penyulang Wendit

Tabel 4-11
Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan
Sebelum Restorasi

BUS	TEGANAN		DAYA PEMBANGKITAN		BEBAN	
	ABS (PU)	SUDUT(DEG)	MW	MVAR	MW	MVAR
1	1	0	5.218	4.006	0	0
18	0,98908	-0,22162	-	-	0,132	0,099
19	0,98132	-0,38365	-	-	0,392	0,294
20	0,96312	-0,76619	-	-	2,816	2,112
21	0,97807	-0,45212	-	-	0,178	0,133
22	0,97700	-0,47501	-	-	0,615	0,461
23	0,97266	-0,56710	-	-	0,968	0,726

Jumlah Pembangkitan : 5,218 MW 4,006 MVAR

Jumlah Pembebanan : 5,101 MW 3,825 MVAR

Tabel 4-12
Hasil Perhitungan Aliran Daya Pada Tiap Saluran Sebelum Restorasi

SALURAN		DAYA		SALURAN		DAYA	
DARI	KE	P (MW)	Q (MVAR)	DARI	KE	P (MW)	Q (MVAR)
1	18	5.218	4.006	18	1	-5.177	-3.942
18	19	2.173	1.646	19	18	-2.160	-1.627
18	20	2.871	2.197	20	18	-2.816	-2.112
19	21	1.768	1.333	21	19	-1.764	-1.326
21	22	0.615	0.462	22	21	-0.615	-0.461
21	23	0.971	0.732	23	21	-0.967	-0.726

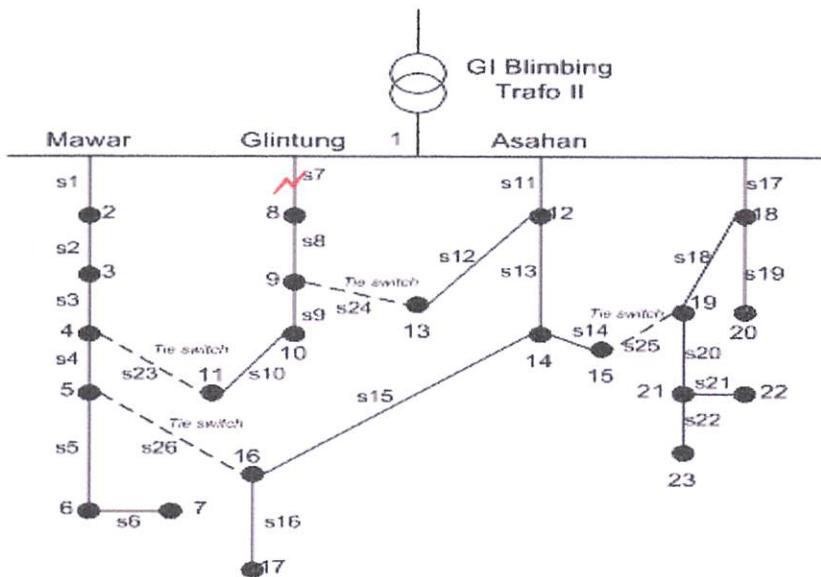
Tabel 4-13
Hasil Perhitungan Rugi-Rugi Saluran Sebelum Restorasi

SALURAN		RUGI-RUGI SALURAN	
DARI	KE	P (MW)	Q (MVAR)
1	18	0,041	0,064
18	19	0,013	0,019
18	20	0,055	0,085
19	21	0,004	0,007
21	22	0	0,001
21	23	0,004	0,006

Total Rugi-Rugi Penyulang Wendit : 0,117 MW + 0,182 MVAR

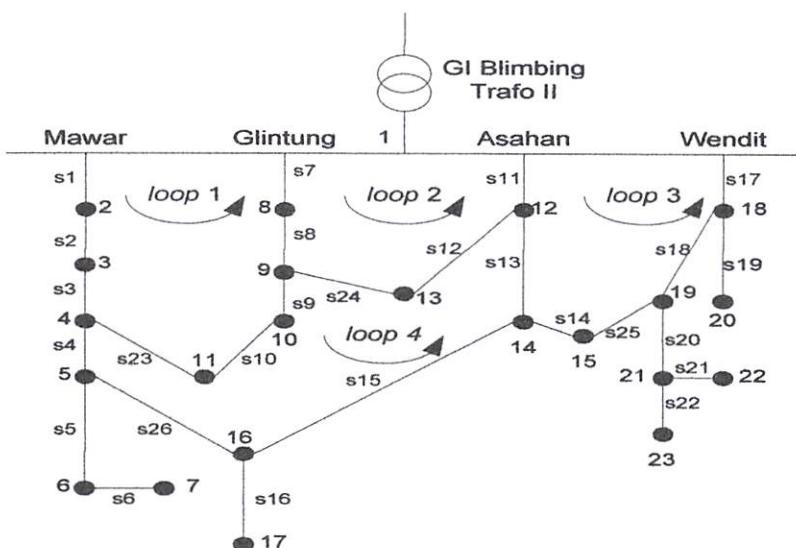
4.3.3. Asumsi Gangguan

➤ Contoh Kasus (Gangguan Pada Saluran 1 Penyulang Glintung)



Gambar 4-12. Gangguan Terjadi di Penyulang Glintung

Dalam analisis perbaikan pelayanan akibat terjadi gangguan, maka semua Tie switch (Normally open) ditutup, sehingga jaringan radial menjadi jaringan loop seperti pada gambar 4.13 berikut ini :



Gambar 4-13. Konfigurasi Jaringan Loop

4.3.4. Hasil Perhitungan Sesudah Restorasi

Tabel 4-14 sampai tabel 4-22 berikut ini memperlihatkan hasil perhitungan aliran daya sesudah perbaikan pelayanan :

➤ Contoh Kasus (Gangguan di Penyulang Glintung)

1. Penyulang Mawar

Tabel 4-14
Hasil Perhitungan Tegangan, Pembangkitan dan Pembebahan
Sesudah Restorasi

BUS	TEGANAN		DAYA PEMBANGKITAN		BEBAN	
	ABS (PU)	SUDUT(DEG)	MW	MVAR	MW	MVAR
1	1	0	3.775	2.908	0	0
2	0,97509	-0,51066	-	-	1,428	1,071
3	0,96394	-0,74640	-	-	0,765	0,574
4	0,95484	-0,94232	-	-	1,484	1,113
10	0,96094	-0,81115	-	-	0,287	0,215
11	0,96015	-0,82829	-	-	0,440	0,330

Jumlah Pembangkitan : 3,775 MW 2,908 MVAR

Jumlah Pembebahan : 4,404 MW 3,303 MVAR

Tabel 4-15
Hasil Perhitungan Aliran Daya Pada Tiap Saluran Sesudah Restorasi

SALURAN		DAYA		SALURAN		DAYA	
DARI	KE	P (MW)	Q (MVAR)	DARI	KE	P (MW)	Q (MVAR)
1	2	3.775	2.908	2	1	-3.706	-2.803
2	3	2.279	1.732	3	2	-2.260	-1.703
3	4	1.494	1.129	4	3	-1.484	-1.113
4	11	0.000	0.000	11	4	0.000	0.000
10	11	0.440	0.330	11	10	-0.440	-0.330

Tabel 4-16**Hasil Perhitungan Rugi-Rugi Saluran Sesudah Restorasi**

SALURAN		RUGI-RUGI SALURAN	
DARI	KE	P (MW)	Q (MVAR)
1	2	0,069	0,105
2	3	0,019	0,029
3	4	0,010	0,016
4	11	0	0
10	11	0	0

Total Rugi-Rugi Penyulang Mawar : 0,098 MW + 0,150 MVAR

2. Penyulang Asahan**Tabel 4-17****Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan Sesudah Restorasi**

BUS	TEGANGAN		DAYA PEMBANGKITAN		BEBAN	
	ABS (PU)	SUDUT(DEG)	MW	MVAR	MW	MVAR
1	1	0	8.345	6.436	0	0
5	0,95787	-0,87609	-	-	0,859	0,645
6	0,95723	-0,89007	-	-	0,166	0,125
7	0,95597	-0,91743	-	-	0,619	0,464
8	0,96135	-0,80230	-	-	0,334	0,251
9	0,96183	-0,79189	-	-	0,735	0,551
12	0,97572	-0,49641	-	-	1,491	1,118
13	0,96221	-0,78379	-	-	1,454	1,090
14	0,96898	-0,63816	-	-	0,197	0,148
16	0,95840	-0,86476	-	-	1,491	1,118
17	0,95839	-0,86492	-	-	0,045	0,034

Jumlah Pembangkitan : 8,345 MW 6,436 MVAR

Jumlah Pembebanan : 7,391 MW 5,544 MVAR

Tabel 4-18
Hasil Perhitungan Aliran Daya Pada Tiap Saluran Sesudah Restorasi

SALURAN		DAYA		SALURAN		DAYA	
DARI	KE	P (MW)	Q (MVAR)	DARI	KE	P (MW)	Q (MVAR)
1	12	8.345	6.436	12	1	-8.197	-6.209
5	6	0.786	0.590	6	5	-0.786	-0.590
5	16	-1.645	-1.235	16	5	1.646	1.236
6	7	0.619	0.465	7	6	-0.619	-0.464
8	9	-0.334	-0.251	9	8	0.334	0.251
9	13	-1.797	-1.349	13	9	1.798	1.349
12	13	3.284	2.490	13	12	-3.251	-2.440
12	14	3.421	2.601	14	12	-3.404	-2.574
14	16	3.207	2.427	16	14	-3.181	-2.388
16	17	0.045	0.034	17	16	-0.045	-0.034

Tabel 4-19
Hasil Perhitungan Rugi-Rugi Saluran Sesudah Restorasi

SALURAN		RUGI-RUGI SALURAN	
DARI	KE	P (MW)	Q (MVAR)
1	12	0,148	0,227
5	6	0	0
5	16	0,001	0,001
6	7	0	0,001
8	9	0	0
9	13	0,001	0,001
12	13	0,033	0,050
12	14	0,017	0,027
14	16	0,026	0,039
16	17	0	0

Total Rugi-Rugi Penyulang Asahan : 0,226 MW + 0,346 MVAR

3. Penyulang Wendit

Tabel 4-20
Hasil Perhitungan Tegangan, Pembangkitan dan Pembebanan
Sesudah Restorasi

BUS	TEGANAN		DAYA PEMBANGKITAN		BEBAN	
	ABS (PU)	SUDUT(DEG)	MW	MVAR	MW	MVAR
1	1	0	5,893	4,528	0	0
15	0,97665	-0,48120	-	-	0,654	0,490
18	0,98767	-0,25040	-	-	0,132	0,099
19	0,97751	-0,46279	-	-	0,392	0,294
20	0,96167	-0,79657	-	-	2,816	2,112
21	0,97425	-0,53180	-	-	0,178	0,133
22	0,97317	-0,55487	-	-	0,615	0,461
23	0,96882	-0,64768	-	-	0,967	0,726

Jumlah Pembangkitan : 5,893 MW 4,528 MVAR

Jumlah Pembebanan : 5,755 MW 4,315 MVAR

Tabel 4-21
Hasil Perhitungan Aliran Daya Pada Tiap Saluran Sesudah Restorasi

SALURAN		DAYA		SALURAN		DAYA	
DARI	KE	P (MW)	Q (MVAR)	DARI	KE	P (MW)	Q (MVAR)
1	18	5,893	4,528	18	1	-5,840	-4,447
15	19	-0,654	-0,490	19	15	0,654	0,491
18	19	2,836	2,152	19	18	-2,815	-2,118
18	20	2,872	2,197	20	18	-2,816	-2,112
19	21	1,768	1,333	21	19	-1,764	-1,326
21	22	0,615	0,462	22	21	-0,615	-0,461
21	23	0,971	0,732	23	21	-0,967	-0,726

Tabel 4-22
Hasil Perhitungan Rugi-Rugi Saluran Sesudah Restorasi

SALURAN		RUGI-RUGI SALURAN	
DARI	KE	P (MW)	Q (MVAR)
1	18	0,053	0,081
15	19	0	0,001
18	19	0,021	0,034
18	20	0,056	0,085
19	21	0,004	0,007
21	22	0	0,001
21	23	0,004	0,006

Total Rugi-Rugi Penyulang Wendit : 0,138 MW + 0,215 MVAR

Untuk hasil *switch* yang optimal untuk dibuka dapat dilihat pada tabel 4.23 berikut ini.

Tabel 4-23
Hasil Switch Optimal Untuk Dibuka

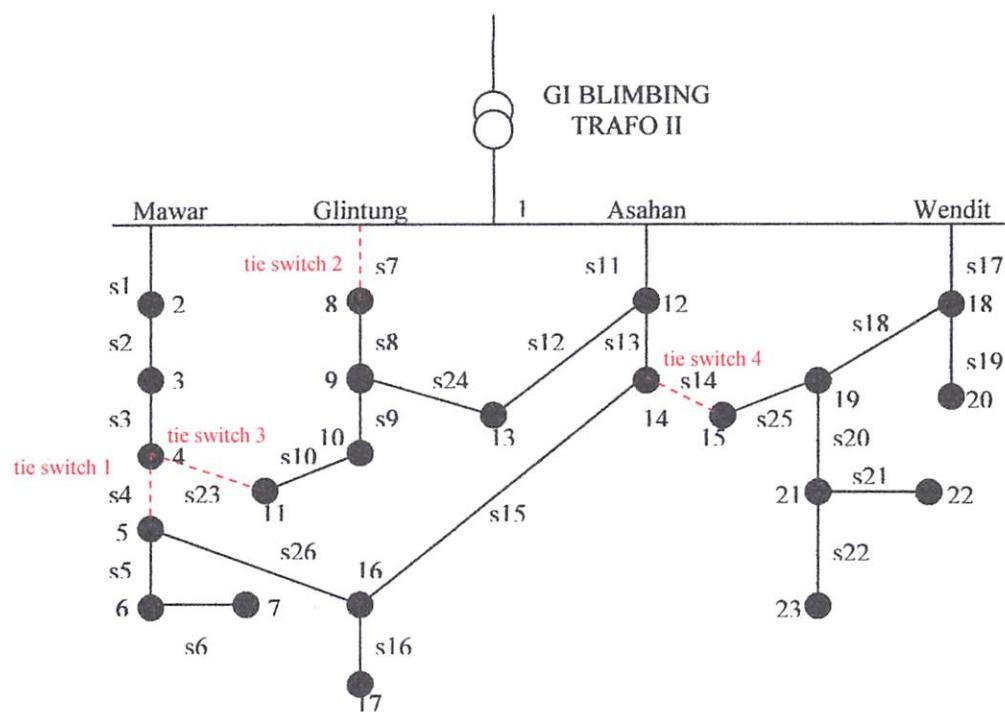
NO SWITCH	
S 4	S 7
S 23	S 14

Pada tabel 4.23 berikut ini memperlihatkan hasil perbandingan total pembangkitan, pembebanan dan rugi-rugi saluran.

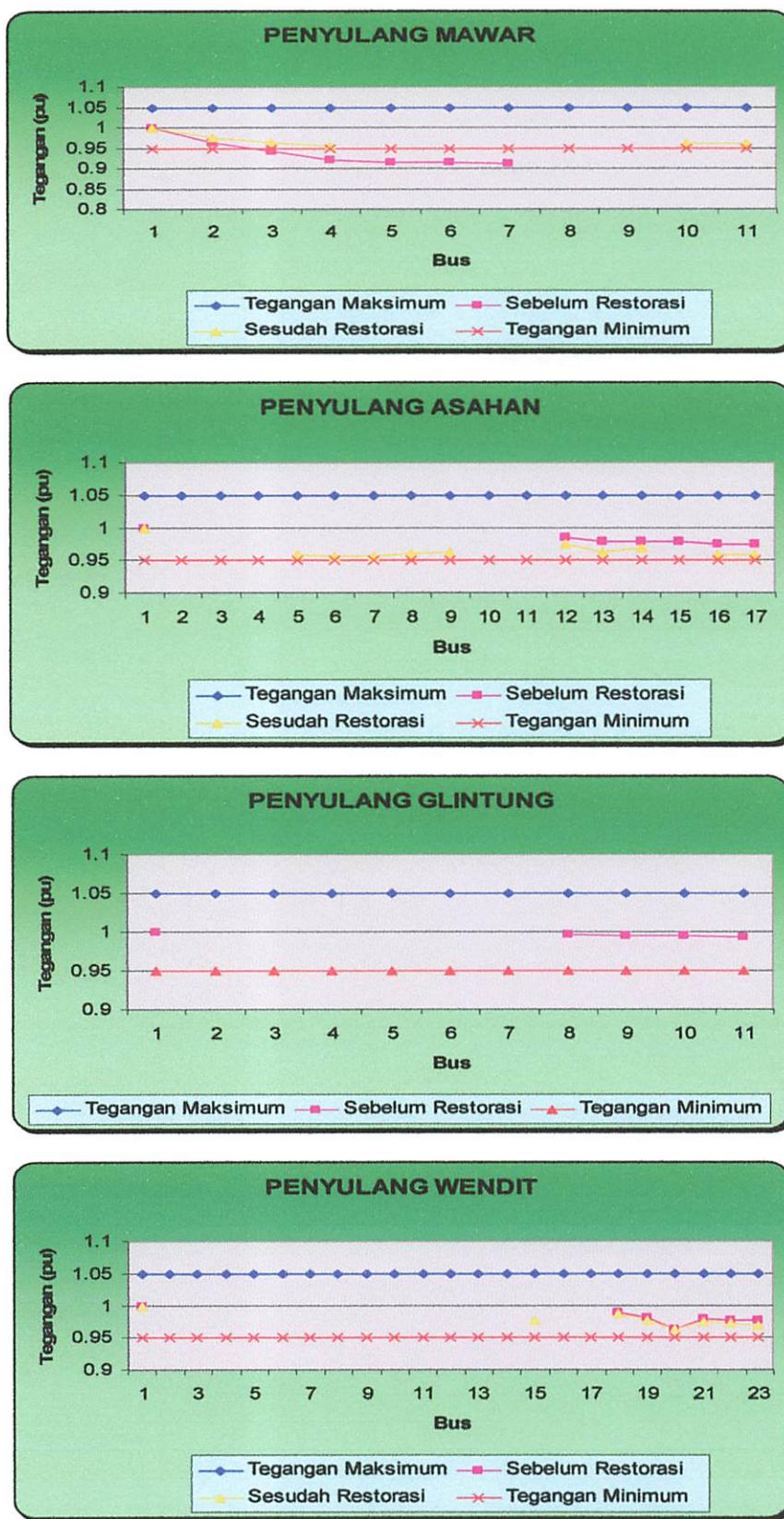
Tabel 4-24
Hasil Perbandingan Total Pembangkitan, Pembebanan Dan Rugi – Rugi Saluran

PENYULANG	PEMBANGKITAN		PEMBEBANAN		RUGI-RUGI	
	SEBELUM (MW)	SESUDAH (MW)	SEBELUM (MW)	SESUDAH (MW)	SEBELUM (MW)	SESUDAH (MW)
MAWAR	5.597	3.775	5.321	4.404	0.276	0.098
GLINTUNG	1.803	0	1.796	0	0.007	0
ASAHDAN	5.414	8.345	5.332	7.391	0.081	0.226
WENDIT	5.218	5.893	5.101	5.755	0.117	0.138
TOTAL	18.032	18.014	17.550	17.550	0.482	0.462

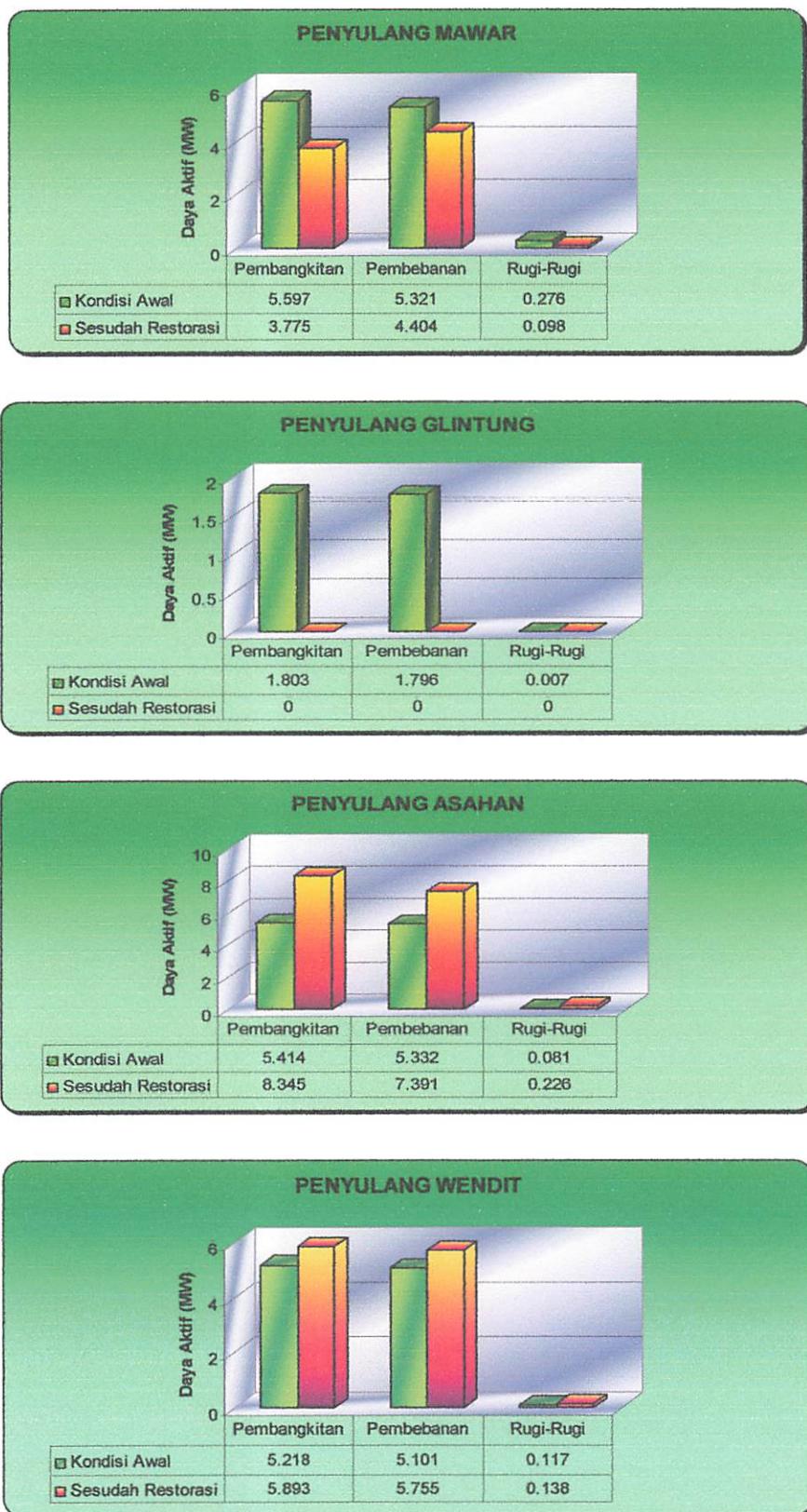
Untuk konfigurasi GI Blimbing sesudah restorasi digambarkan seperti pada gambar 4.14. berikut ini.



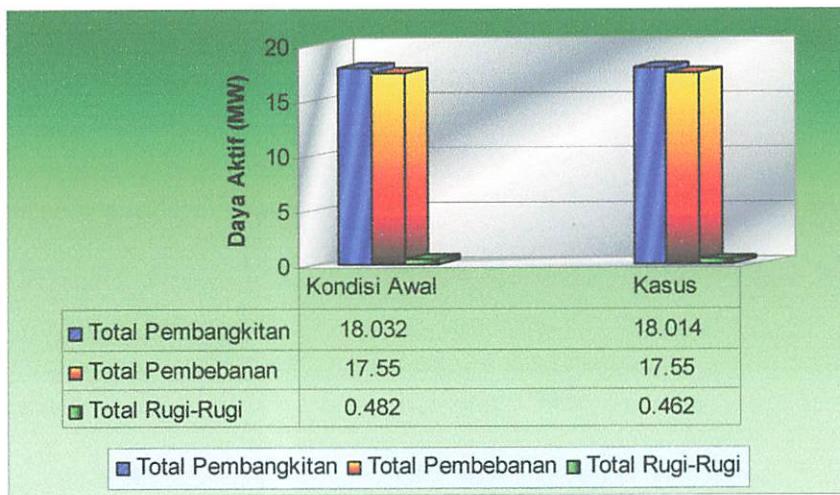
Gambar 4-14. Konfigurasi GI Blimbing Sesudah Restorasi



Grafik 4-1. Perbandingan Tegangan Sebelum dan Sesudah Restorasi



Grafik 4-2. Perbandingan Total Pembangkitan, Pembebanan Dan Rugi-Rugi



Grafik 4-3. Perbandingan Total Pembangkitan, Total Pembebanan Dan Total Rugi-Rugi

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah melakukan analisis uji coba program dengan menggunakan metode Parallel Algoritma Genetika untuk perbaikan pelayanan, maka dapat ditarik kesimpulan sebagai berikut :

1. *Switch* yang optimal untuk dibuka pada simulasi gangguan pada saluran 1 penyulang Glintung adalah *switch 4*, *switch 7*, *switch 23* dan *switch 14*.
2. Tegangan masih dalam batas-batas yang diizinkan yaitu sebesar 1,05 pu sampai dengan 0,95 pu dan menjadi lebih baik bila dibandingkan dengan kondisi awal yaitu sebesar 1 pu sampai 0,95484 pu.
3. Rugi-rugi daya yang dihasilkan mengalami penurunan dari rugi-rugi sebelum perbaikan pelayanan :
 - Rugi-rugi sebelum terjadi gangguan yaitu sebesar 0,482 MW
 - Rugi-ruginya setelah terjadi gangguan di saluran 1 penyulang Glintung yaitu sebesar 0,462 MW

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Seluruh molekulik antisasis ini coba dilakukan dengan metode
Panjang Wavelength Centrifuge untuk berpasangan dengan dilihat
kesimpulan sebagaimana berikut :

1. penilaian nilai pemungutan hasil analisis nilai titik lebur yang diperoleh .1

Al. Alumunium Cu-Alumunium Cu-Alumunium selenite granular dan
dapat 50,01 persen nilai pemungutan hasil analisis titik lebur yang diperoleh .2
pada setiap molekulik antisasis nilai titik lebur yang diperoleh pada 50,0 persen sedangkan
titik lebur yang diperoleh pada setiap molekulik antisasis nilai titik lebur yang diperoleh pada

titik lebur yang diperoleh pada setiap molekulik antisasis nilai titik lebur yang diperoleh pada
sejauh ini hasil analisis titik lebur yang diperoleh pada setiap molekulik antisasis nilai titik lebur yang diperoleh pada

titik lebur yang diperoleh pada setiap molekulik antisasis nilai titik lebur yang diperoleh pada
titik lebur yang diperoleh pada setiap molekulik antisasis nilai titik lebur yang diperoleh pada

titik lebur yang diperoleh pada setiap molekulik antisasis nilai titik lebur yang diperoleh pada

4. Akibat terjadi gangguan (pada saluran 1 penyulang Glintung) maka :

- Pada Bus 8 dan Bus 9 mendapatkan aliran daya dari penyulang Asahan, yang semula dari penyulang Glintung sehingga kontinuitas penyaluran listrik di GI Blimbings tetap terjaga.
- Pada Bus 10 dan Bus 11 mendapatkan aliran daya dari penyulang Mawar, yang semula dari penyulang Glintung sehingga kontinuitas penyaluran listrik di GI Blimbings tetap terjaga.

5.2. Saran

Penggunaan Parallel Algoritma Genetika perlu dikembangkan lagi agar bisa diterapkan dengan penyulang yang lebih banyak, baik dari satu Gardu Induk maupun antar Gardu Induk dan hendaknya juga bisa diterapkan untuk *Multiple faults*.

DAFTAR PUSTAKA

1. Y Fukuyama, H-D Chiang and K Nan Miu, “*Parallel Genetic Algorithm For Service Restoration in Electric Power Distribution Systems*” Electrical Power & Energy Systems, Vol.18, No. 2, PP 111-119, 1996.
2. Hasan Basri, Ir, “*Sistem Distribusi*” Balai Penerbit dan Humas ISTN, Jakarta, 1996.
3. A.S. Pabla, “*Sistem Distribusi Daya Listrik*” Penerbit Erlangga, Jakarta, 1994.
4. Yusra Sabri, Dr. Ir. “*Arsitektur dan Pengembangan Jaringan distribusi*”. Gedung kerja sama PLN-ITB, Bandung 1990.
5. William D.Stevenson, Jr, “*Analisis Sistem Tenaga Listrik*”, Edisi keempat, Erlangga, Jakarta, 1996.
6. Sri Kusumadewi, “*Artificial Intelligence (Teknik dan Aplikasinya)*” Graha Ilmu, Yogyakarta, 2003
7. Son Kuswadi, “*Kendali Cerdas (Intelligent Control)*” EEPIS Press, Surabaya.
8. Mitsuo Gen, Runwei Cheng, “*Genetic Algorithm And Engineering Design*”, (John Wiley & Son, Inc., 1994) p-7
9. Whei-Min Lin and Hong-Chan Chin, “*A New Approach for Feeder Reconfiguration for Loss Reduction and Service Restoration*” IEEE Transaction on Power Systems, Vol.13, No. 3, July 1998.

LAMPIRAN 1

*DATA-DATA, SINGLE LINE DIAGRAM GI
BLIMBING MALANG & SINGLE LINE
P.MAWAR, P.GLINTUNG, P.PASAHAN, P.WENDIT*

TABEL A-1
PENGELOMPOKAN SWITCH

PENYULANG	BUS	SWITCH	JUMLAH BEBAN (kVA)
1	2	5,616	1,2142
2	3	4,085	0,8832
3	4	5,04	1,0896
4	5	3,897	0,8425
5	6	0,677	0,1464
6	7	1,678	0,3628
1	8	1,381	0,2986
8	9	1,194	0,2581
9	10	1,016	0,2197
10	11	1,492	0,3226
1	12	2,474	0,5349
12	13	3,443	0,7444
12	14	1,646	0,3559
14	15	1,183	0,2558
14	16	2,745	0,5935
16	17	0,131	0,0283
1	18	1,782	0,3853
18	19	3,031	0,6553
18	20	7,639	1,6516
19	21	1,549	0,3349
21	22	1,476	0,3191
21	23	4,689	1,0138
4	11	0,207	0,0448
9	13	0,173	0,0374
15	19	1,114	0,2408
5	16	0,262	0,0566

Sumber : PLN (Persero) Distribusi JATIM-Cabang Malang

TABEL A-2
DATA JARINGAN DISTRIBUSI

BUS PANGKAL	BUS UJUNG	JARAK (km)	R (Ω)	X (Ω)
1	2	5,616	1,2142	1,8650
2	3	4,085	0,8832	1,3501
3	4	5,04	1,0896	1,6657
4	5	3,897	0,8425	1,2880
5	6	0,677	0,1464	0,2238
6	7	1,678	0,3628	0,5546
1	8	1,381	0,2986	0,4564
8	9	1,194	0,2581	0,3946
9	10	1,016	0,2197	0,3358
10	11	1,492	0,3226	0,4931
1	12	2,474	0,5349	0,8177
12	13	3,443	0,7444	1,1379
12	14	1,646	0,3559	0,5440
14	15	1,183	0,2558	0,3910
14	16	2,745	0,5935	0,9072
16	17	0,131	0,0283	0,0433
1	18	1,782	0,3853	0,5890
18	19	3,031	0,6553	1,0017
18	20	7,639	1,6516	2,5247
19	21	1,549	0,3349	0,5119
21	22	1,476	0,3191	0,4878
21	23	4,689	1,0138	1,5497
4	11	0,207	0,0448	0,0684
9	13	0,173	0,0374	0,0572
15	19	1,114	0,2408	0,3682
5	16	0,262	0,0566	0,0866

Keterangan : Bus 1 merupakan bus berayun, Bus 2 sampai 23 merupakan bus beban

Sumber : PLN (Persero) Distribusi JATIM-Cabang Malang

TABEL A-3
PENGHANTAR AAAC

PENAMPANG NOMINAL (mm ²)	JARI-JARI (mm)	URAT	GMR (mm)	IMPEDANSI URUTAN POSITIF (Ω/km)	IMPEDANSI URUTAN NOL (Ω/km)
16	2,2563	7	1,6380	2,0161 + J0,4036	2,1641 + J1,6911
25	2,8203	7	2,0475	1,2903 + J0,3895	1,4384 + J1,6770
35	3,3371	7	2,4227	0,9217 + J0,3790	1,0697 + J1,6665
50	3,9886	7	2,8957	0,6452 + J0,3678	0,7932 + J1,6553
70	4,7193	7	3,4262	0,4608 + J0,3572	0,6088 + J1,6447
95	5,4979	19	4,1674	0,3396 + J0,3449	0,4876 + J1,6324
120	6,1791	19	4,6837	0,2688 + J0,3376	0,4168 + J1,6251
150	6,9084	19	5,2368	0,2162 + J0,3305	0,3631 + J1,6180
185	7,6722	19	5,8155	0,1744 + J0,3239	0,3224 + J1,6114
240	8,7386	19	6,6238	0,1344 + J0,3158	0,2824 + J1,6033

Sumber : PLN (Persero) Distribusi JATIM-Cabang Malang

TABEL A-4
KHA PENGHANTAR CAMPURAN ALUMINIUM TELANJANG (AAAC)

LUAS PENAMPANG (mm ²)	KHA TERUS MENERUS (A)
16	105
25	135
35	170
50 (7 KAWAT)	210
50 (19 KAWAT)	210
70	255
95	320
120	365
150	425
185	490
240	585
300	670
400	810
500	930
630	1075
800	1255
1000	1450

Sumber : PLN (Persero) Distribusi JATIM-Cabang Malang

TABEL A-5

DATA BEBAN TIAP SEKSI PENYULANG MAWAR

SEKSI	NO. TRAFO	NOMINAL (kVA)	PERSEN BEBAN (%)	BEBAN TRAFO (kVA)
1	186	150	74,3	111,45
	270	150	71,4	107,1
	369	150	74,3	111,45
	370	150	74,3	111,45
	371	150	74,3	111,45
	424	250	73,4	183,51
2	425	100	30,8	30,82
	455	250	61,7	154,31
	542	150	59,6	89,4
	588	160	53,6	85,8
	605	100	48,9	48,99
	744	160	86,2	137,94
3	764	160	67,6	108,23
	775	250	82,7	206,9
	779	250	74,3	185,75
	193	200	43,5	87,1
	253	150	76,8	115,22
	253A	200	73,8	147,6
4	284	150	68,2	102,3
	290	150	69,8	104,65
	585	200	64,8	129,5
	615	160	74,3	118,88
	737	250	51,5	128,82
	920	160	14,2	22,77
5	32	100	105	105
	110	150	74,3	111,45
	133	150	74,3	111,45
	230	160	74,3	118,88
	231	160	81,3	130
	305	160	80,6	129
6	305A	200	36	72
	335	75	74,3	55,73
	358	150	74,3	111,45
	368	150	114	171
	380	75	97,5	73,13
	409	150	74	111
7	409A	160	74,3	118,88
	543	150	86	129
	584	160	86,9	139
	741	160	58,8	94
	782	100	74,3	74,3
	15	250	60,3	150,7
8	21	200	90	180
	38	250	74	185
	38	100	76,5	76,5
	235	200	65,5	131
	244	250	50	125
	315	150	96,7	145
9	706	50	13,5	67,5

5	22	100	74,3	74,3
6	240	160	60,1	98,13
	53	150	83,4	125,09
	122	150	90,8	136,18
	265	150	45,3	67,98
	361	160	74,3	118,88
	412	100	74,3	74,3
	570	100	89,3	89,32
777	250	250	64,8	162

TABEL A-6
DATA BEBAN TIAP SEKSI PENYULANG GLINTUNG

SEKSI	NO. TRAFO	NOMINAL (kVA)	PERSEN BEBAN (%)	BEBAN TRAFO (kVA)
1	93	100	72,3	72,3
	141	200	76,5	153
	196	160	79,4	127
2	575	160	40,9	65,4
	16	250	48,8	122,1
	51	250	54,8	137
	148	250	108,4	271
3	210	250	65,6	164
	281	160	68,1	109
	536	160	72,5	116
	40	200	49	98
4	153	160	40	64,02
	211	150	65,6	98,4
	398	150	65,6	98,4
4	107	100	44,4	44,38
	142	160	116,9	187
	165	200	64,3	128,57
	437	100	58,6	58,6
449	449	100	65,6	65,6
	762	100	65,6	65,6

TABEL A-7
DATA BEBAN TIAP SEKSI PENYULANG ASAHAN

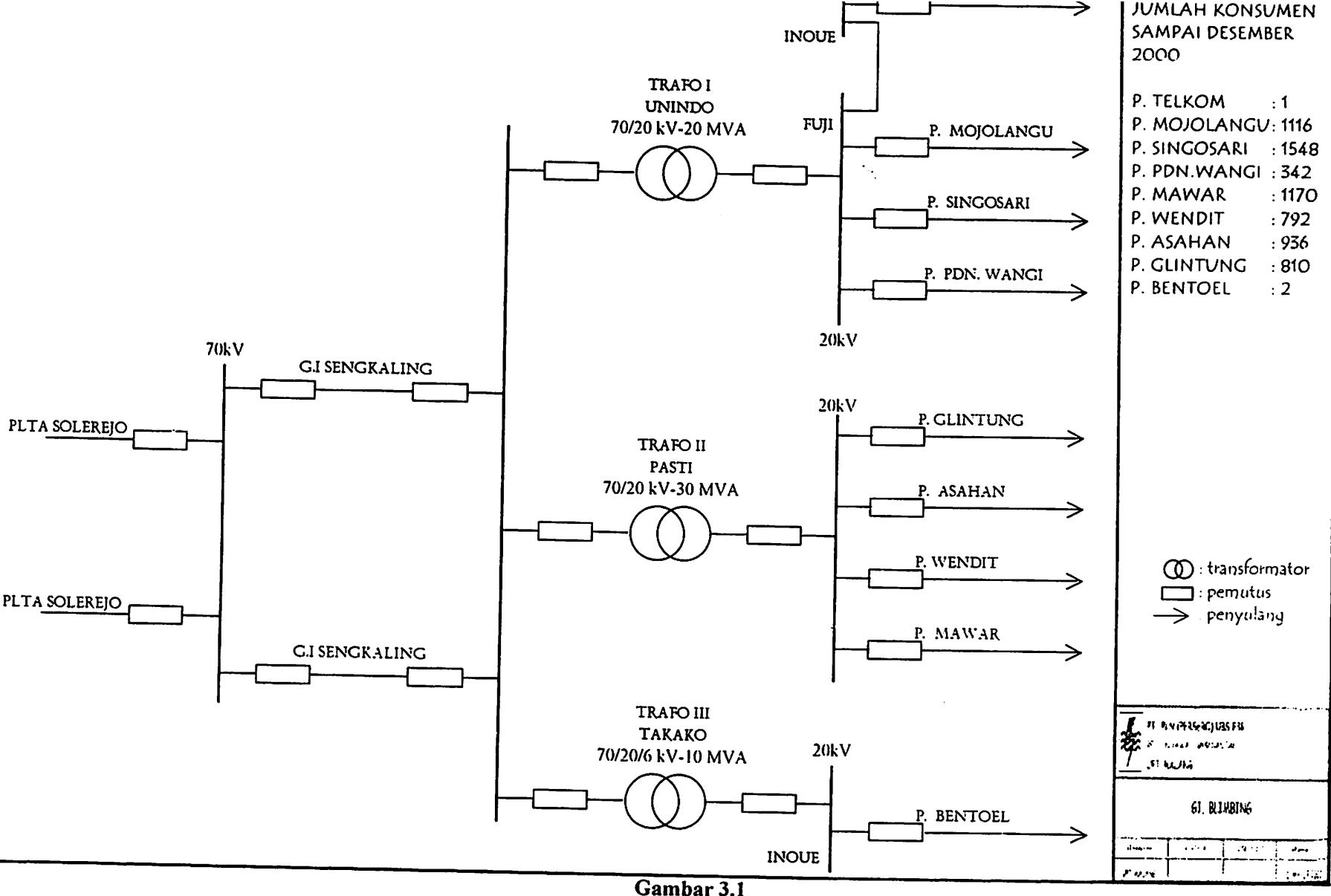
SEKSI	NO. TRAFO	NOMINAL (kVA)	PERSEN BEBAN (%)	BEBAN TRAFO (kVA)
1	39	200	66,5	133
	88	250	75	187,5
	124	200	73,4	146,7
	125	160	24,4	39
	128	100	41,9	41,9
2	267	160	75	120
	445	160	75	120
	495	200	5	10

	621	160	12,6	20,1
	622	100	75	75
	623	50	75	37,5
	654	160	23	36,9
	660	160	6,8	10,9
	668	250	75	187,5
	739	100	75	75
	757	630	75	472,5
	865	200	75	150
2	372	250	49,6	124
	373	160	40,5	64,8
	434	160	75	120
	694	160	10,4	16,6
	697	200	74,2	148,3
	726	160	10,6	17
	734	160	9,6	15,3
	735	160	39,8	63,6
	738	200	53,2	106,3
	767	100	75	75
	843	630	75	472,5
	875	160	38,3	61,3
	899	250	75	187,5
	913	100	75	75
	924	200	75	150
	960	160	75	120
3	285	25	4,8	1,2
	286	160	97,7	156,2
	582	160	13	20,8
	752	200	34,2	68,3
4	26A	100	75	75
	354	150	80	120
	667	200	75	150
	815	630	75	472,5
5	2	160	75	120
	3	200	81,5	63
	234	100	79,8	79,88
	279	160	28,5	45,59
	314	150	75	112,5
	364	150	75	112,5
	384	75	75	56,25
	528	630	75	472,5
	624	550	75	412,5
	818	200	46,8	93,5
	906	160	75	120
	910	100	75	75
6	80A	75	75	56,25

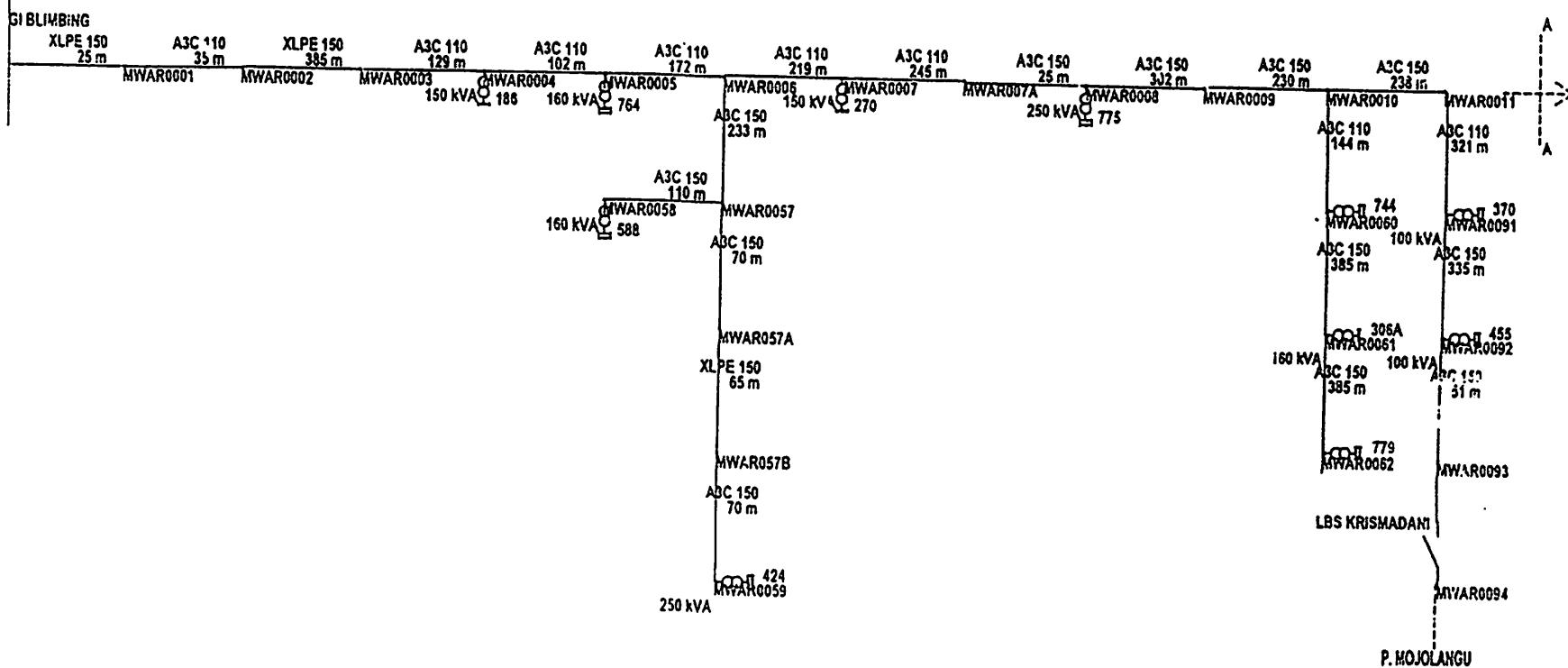
TABEL A-8

DATA BEBAN TIAP SEKSI PENYULANG WENDIT

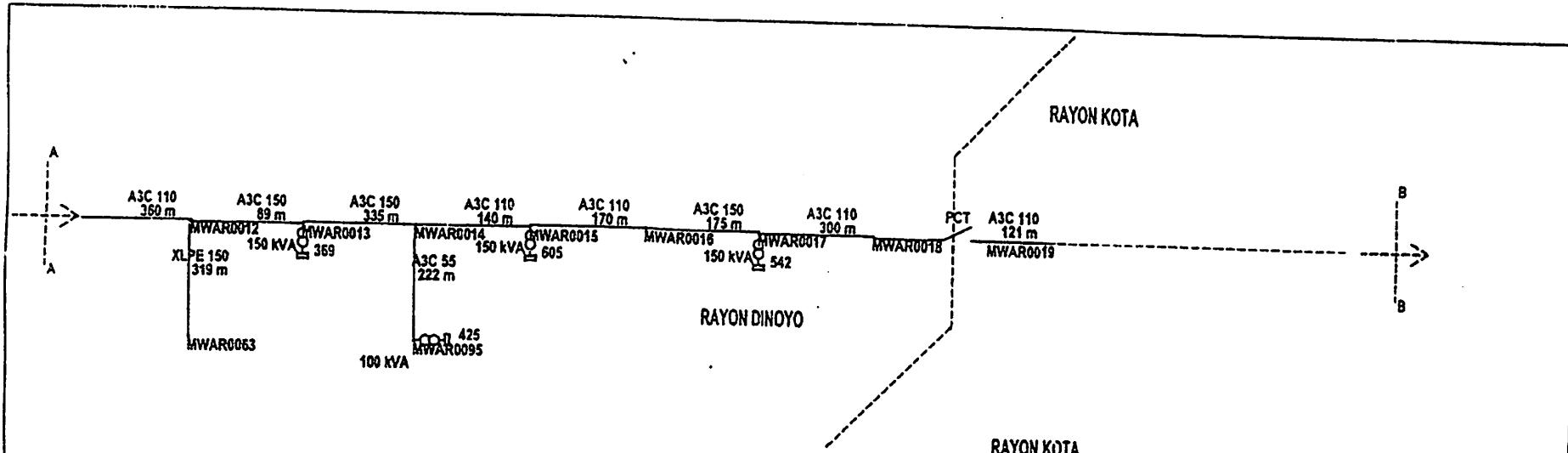
SEKSI	NO. TRAFO	NOMINAL (kVA)	PERSEN BEBAN (%)	BEBAN TRAFO (kVA)
1	224	200	82,8	165,5
2	215	160	71,4	114,3
	338	250	78,7	196,8
	577	200	89,5	179
3	28	100	92,4	92,4
	203	100	60,9	60,9
	204	160	63,7	101,9
	222	1030	74,4	766,32
	293	100	82	82
	502	250	9,1	22,8
	503	160	74,4	119,04
	608	150	66,7	100
	649	250	81,1	202,7
	650	200	81,5	163
	655	100	105,8	105,8
	659	800	74,4	595,2
	676	200	74,4	148,8
	755	630	77,3	468,72
	773	250	67,7	169,2
	828	100	10	10
	840	160	83,1	133
	966	160	5,9	9,5
	967	160	0,6	1
	968	160	30,8	49,2
	991	160	74,4	119,04
4	378	160	64,4	103
	727	160	74,4	119,04
5	185	150	73,4	110,1
	254	150	74,4	111,6
	428	100	77,8	77,8
	469	630	74,4	468,72
6	537	250	52,1	130,2
	681	150	58,1	87,2
	699	100	74,4	74,4
	705	50	74,4	37,2
	714	160	61,1	97,7
	715	160	76,4	122,2
	763	160	75,1	120,1
	768	160	25,5	40,8
	769	160	43,4	69,4
	822	200	44,7	89,3
	844	200	23,9	47,8
	854	100	43,1	43,1
	864	200	27,3	54,6
	923	160	36,4	58,3
	955	160	28,4	45,5
	956	160	28,6	45,8
	974	160	28,6	45,8



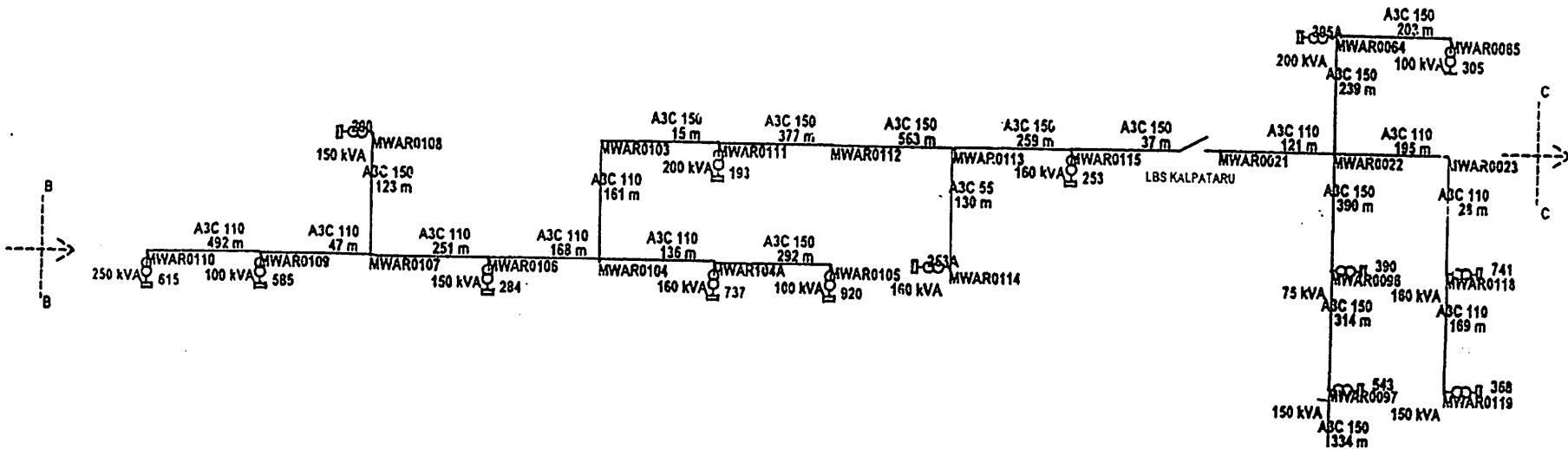
Gambar 3.1
Single Line Diagram G.I. Blimbing



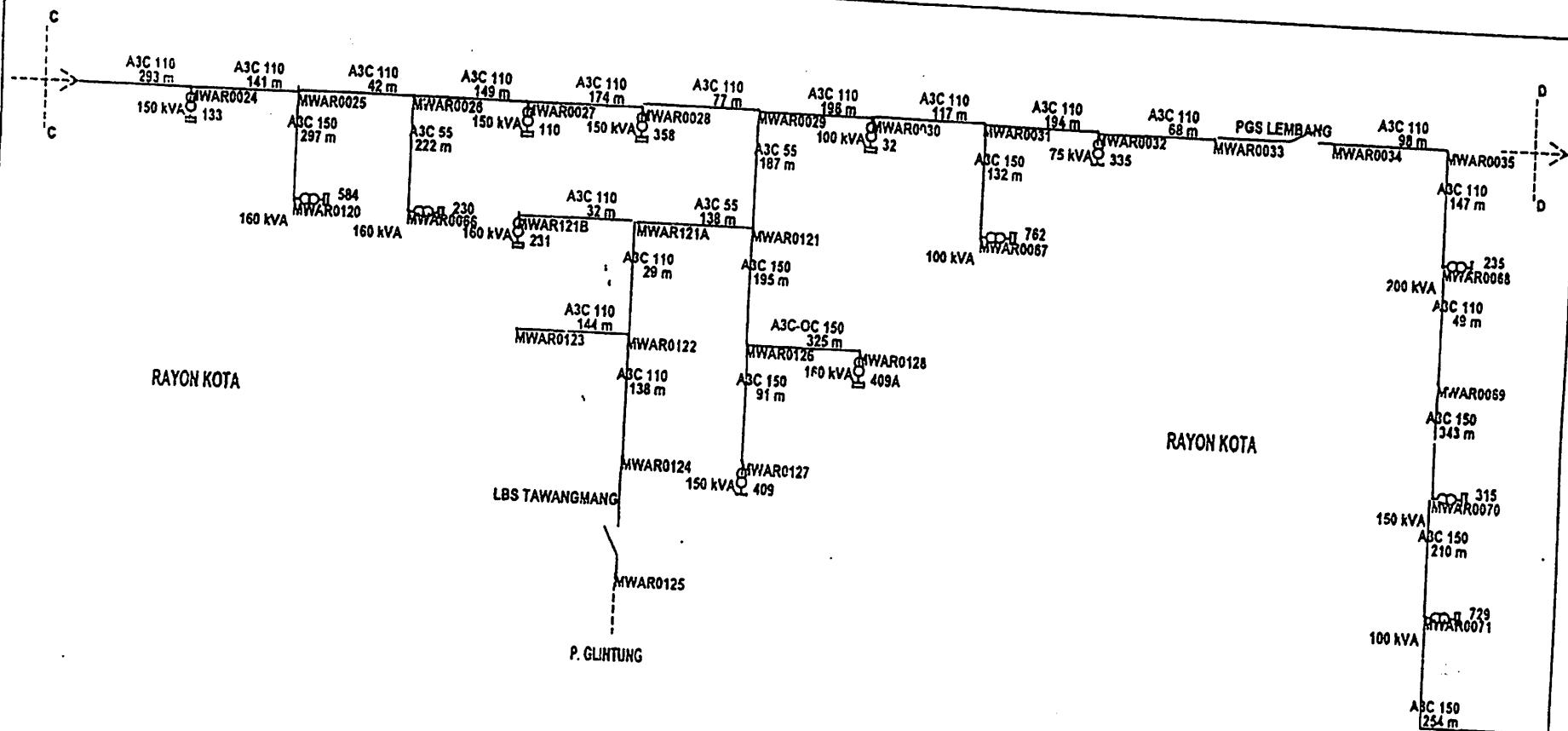
	PT. PLN (PERSERO)	
	JASA ENGINIERING	
Project Name:		
PENURUNAN SUSUT KWH		
PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG		
Title:		
SINGLE LINE DIAGRAM PENYULANG MAWAR		
Design By :	Drawing No.:	
Drawn By :		
Approved By :		
Checked By :		
Date :	Rev.: 0	
4102-3100-L-51-DS-MWAR-018-01-0		



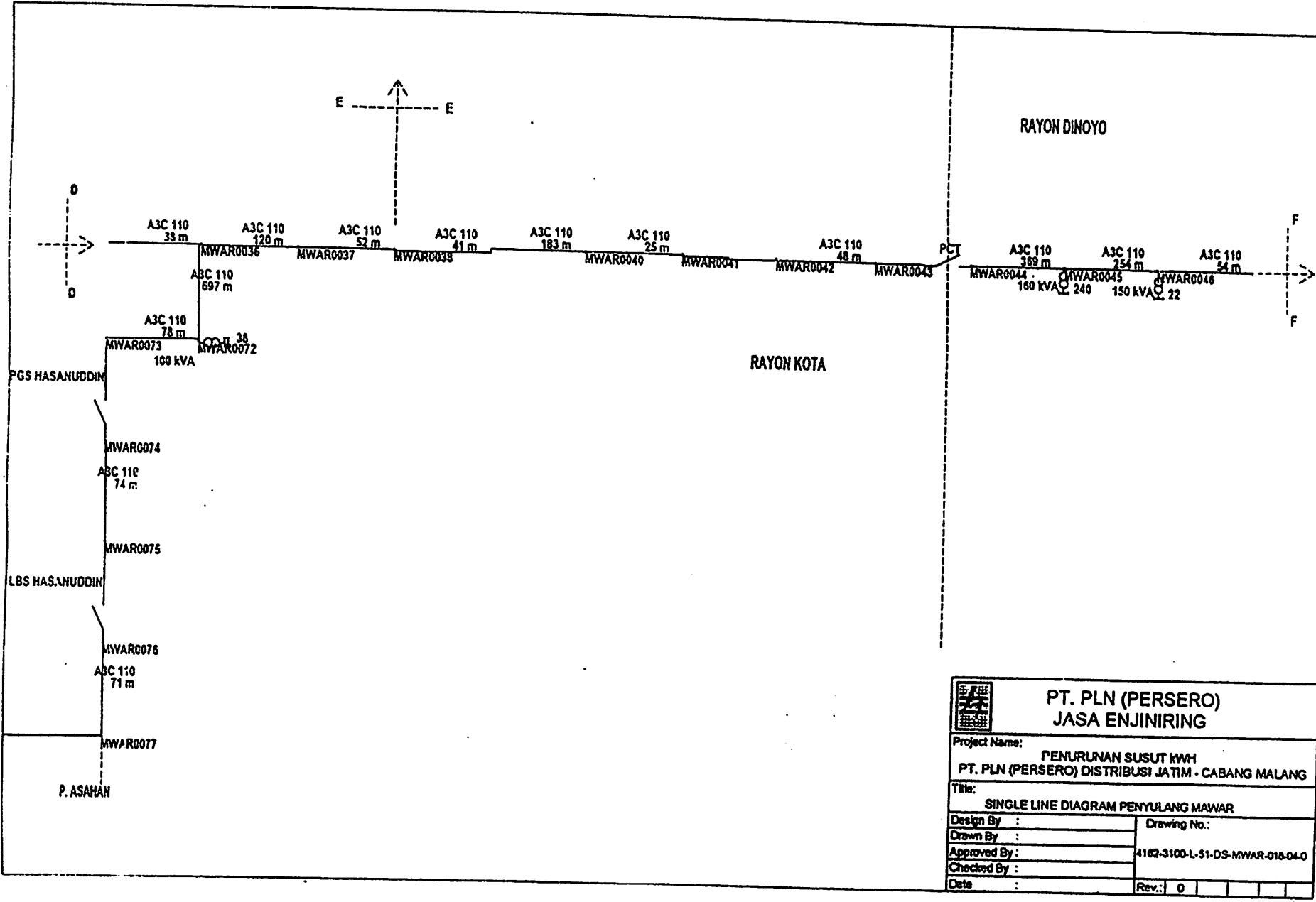
	PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG		
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR		
Design By :	Drawing No.:	
Drawn By :		
Approved By :	4162-3100-L-51-DS-MWAR-018-02-0	
Checked By :		
Date	Rev.: 0	



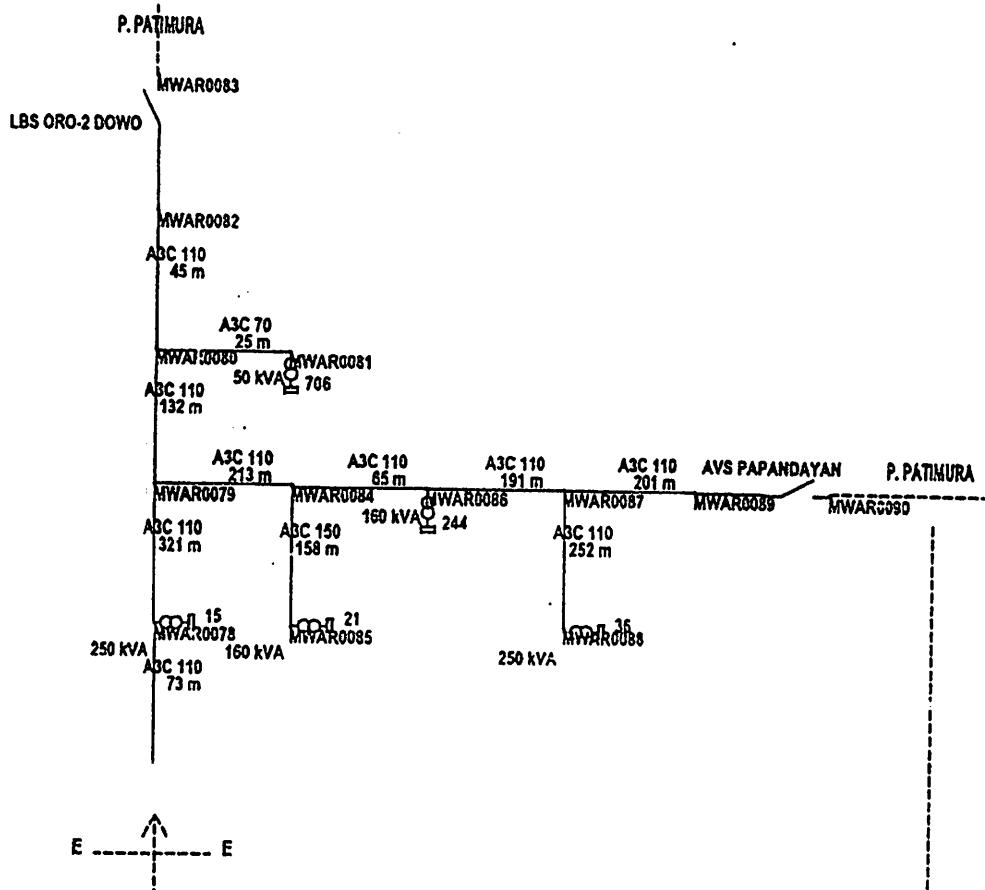
	PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUTSI JATIM - CABANG MALANG		
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR		
Design By :	Drawing No.:	
Drawn By :		
Approved By :		
Checked By :		
Date :	Rev.:	0
6/7		



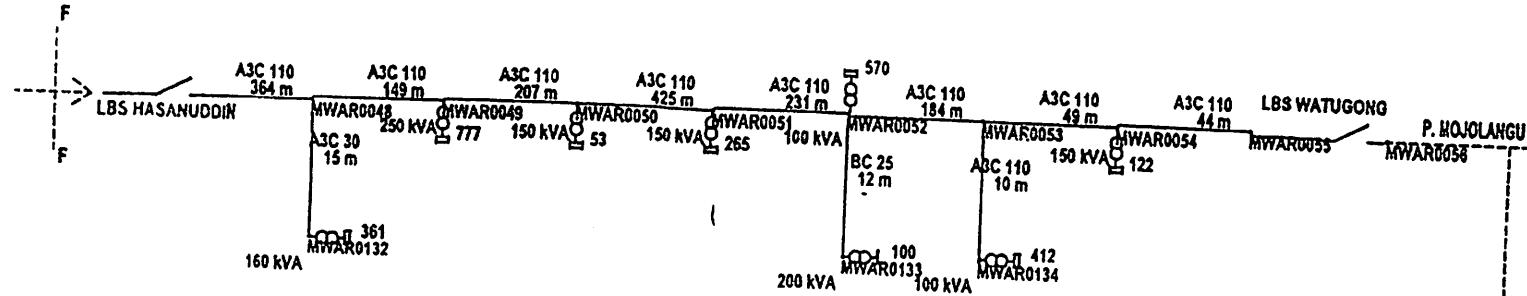
	PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUTSI JATIM - CABANG MALANG		
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR		
Design By :	Drawing No.:	
Drawn By :		
Approved By :	4162-3100-L-51-DS-MWAR-016-03-0	
Checked By :		
Date :	Rev.: 0	



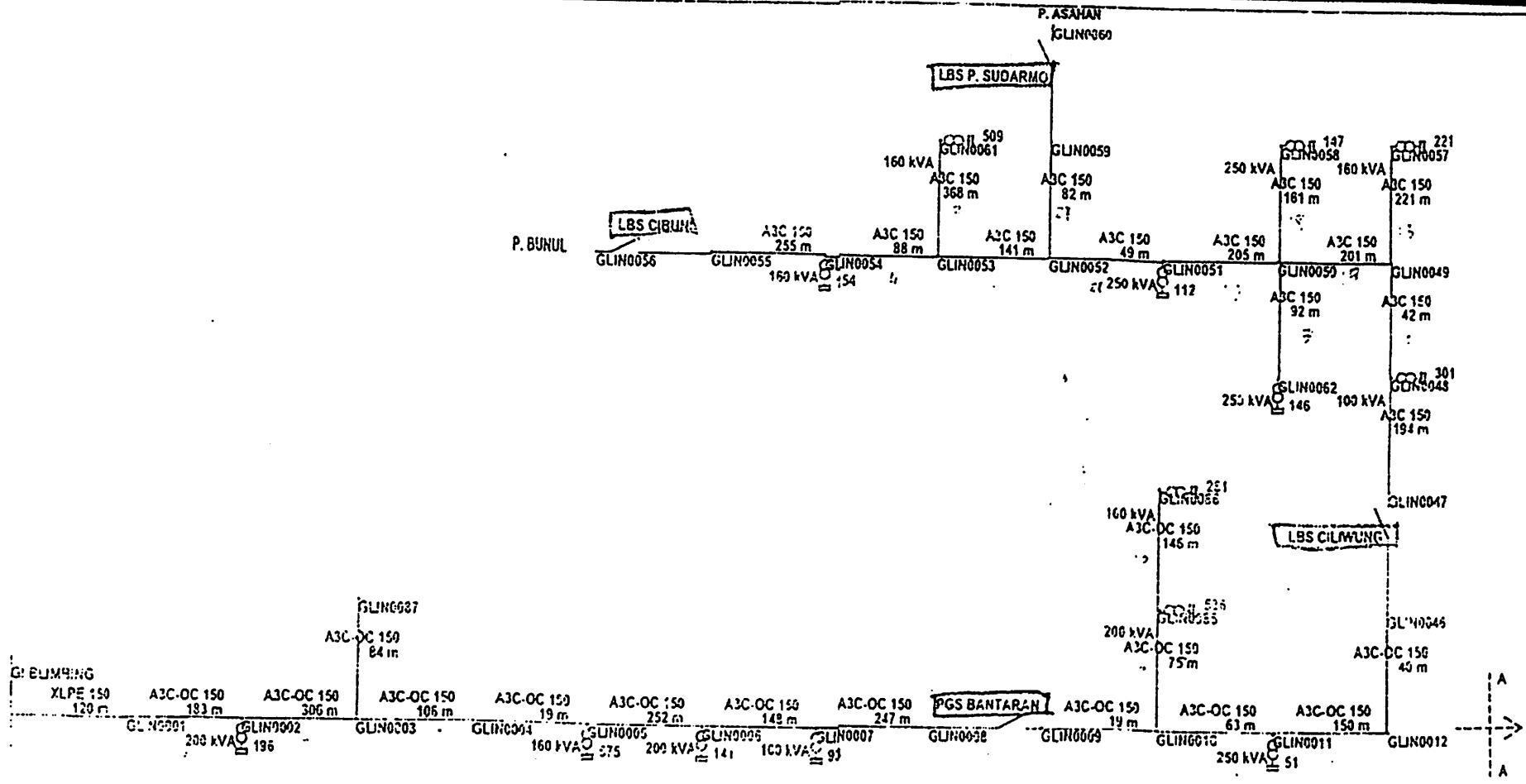
	PT. PLN (PERSERO) JASA ENJINIRING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG		
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR		
Design By :	Drawing No.:	
Drawn By :		
Approved By :	4162-3100-L-51-DS-MWAR-018-04-0	
Checked By :		
Date :	Rev.: 0	



	PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG		
Title: SINGLE LINE DIAGRAM PENYULANG MAWAR 22, 98		
Design By :	Drawing No.:	
Drawn By :		
Approved By :	4162-3160-L-51-DS-MWAR-018-07-0	
Checked By :		
Date :	Rev.:	0

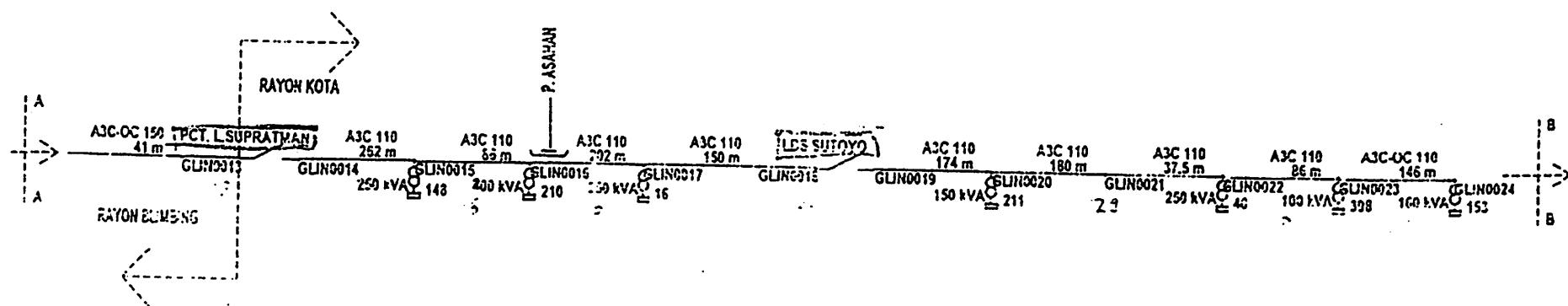


PT. PLN (PERSERO) JASA ENGINERING	
Project Name:	
PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title:	
SINGLE LINE DIAGRAM PENYULANG MAWAR	
Design By :	Drawing No.:
Drawn By :	
Approved By :	4162-3100-L-51-DS-MWAR-018-05-0
Checked By :	
Date :	SEPTEMBER, 2000
Rev.:	0

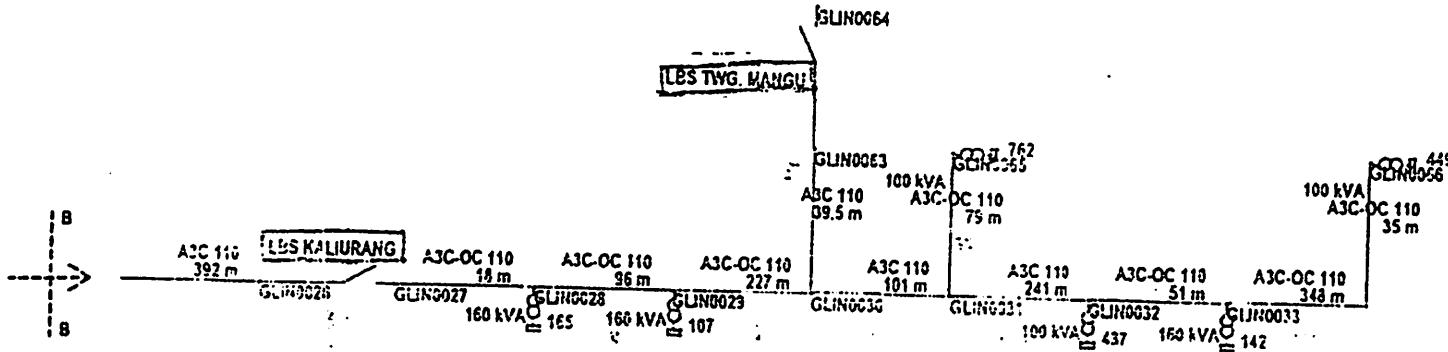


		PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CARANG MALANG			
Date:			
SINGLE LINE DIAGRAM PENYULANG GUNTUNG			
Drawn By:	Reviewed By:		
Drawn By:	Approved By:		
Approved By:	Checked By:		
Date:	Rev. T 0		

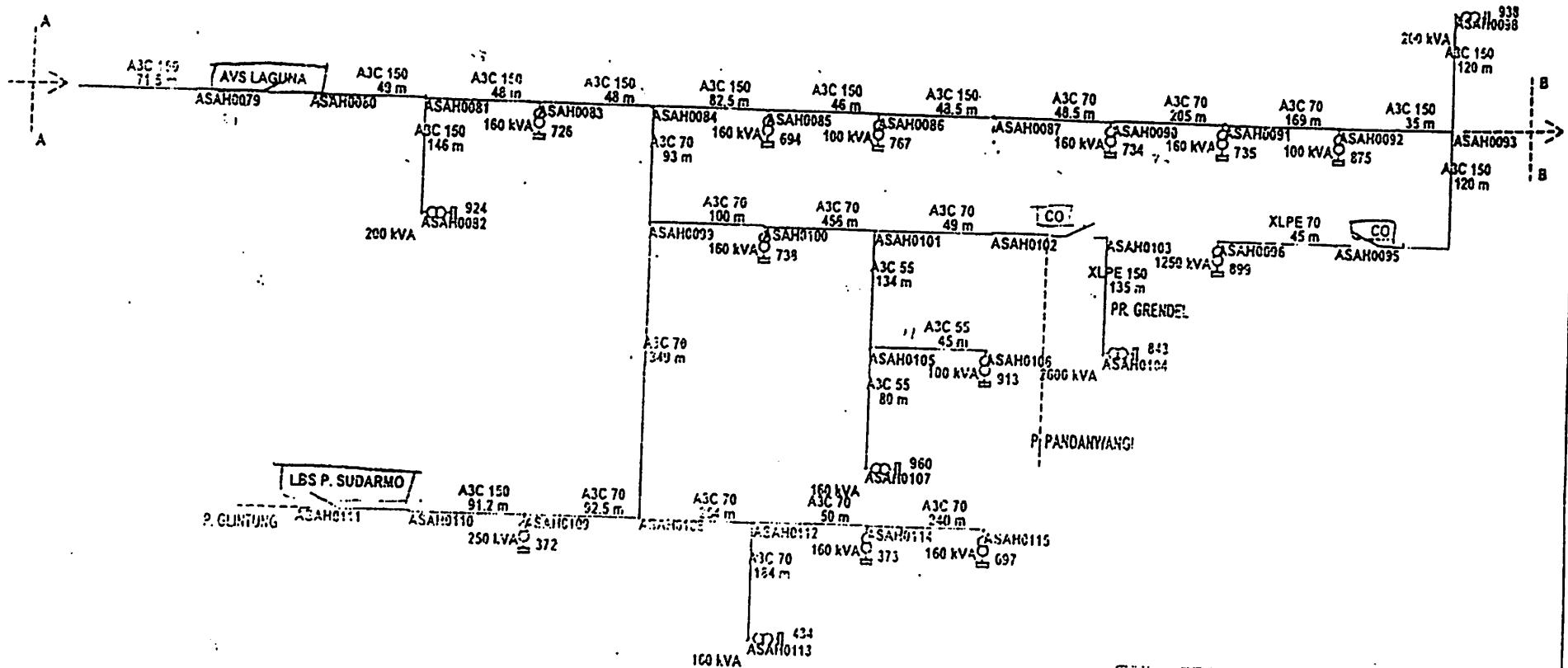
1/4



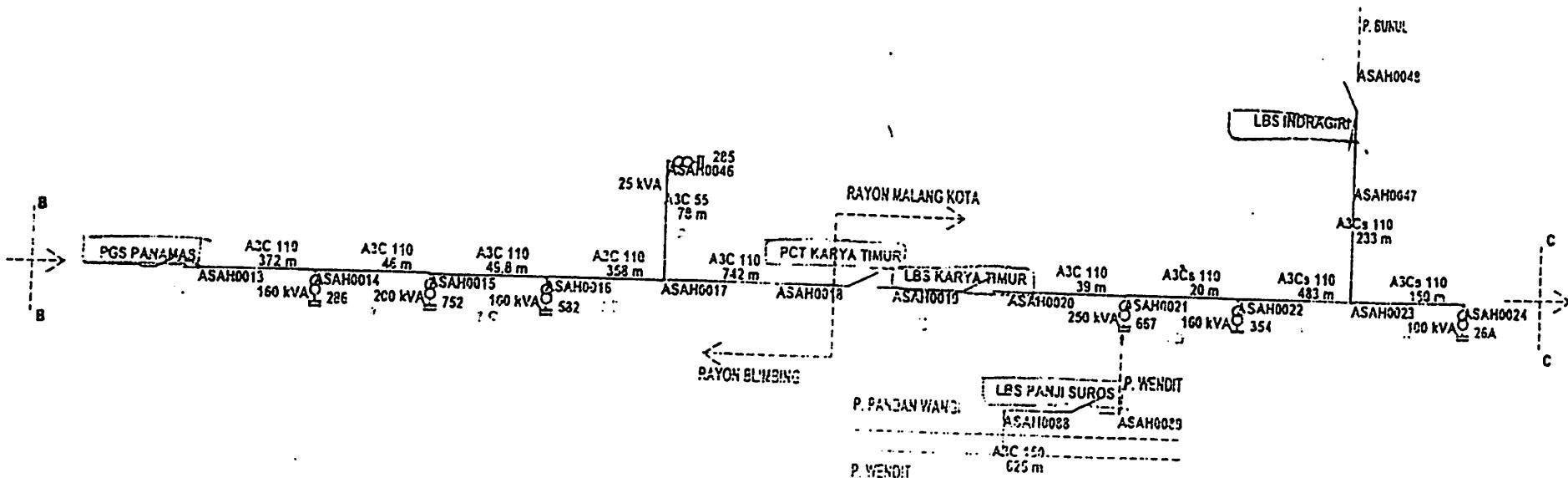
	PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH		
PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG		
Title: SINGLE LINE DIAGRAM PENYULANG GLINTUNG		
Drawn By	Approved By	Checked By
Drawn By	Approved By	Checked By
Approved By	416.2-31001-01-DL-GLTU 016/22-0	
Checked By	Date	Rev.
	0	•



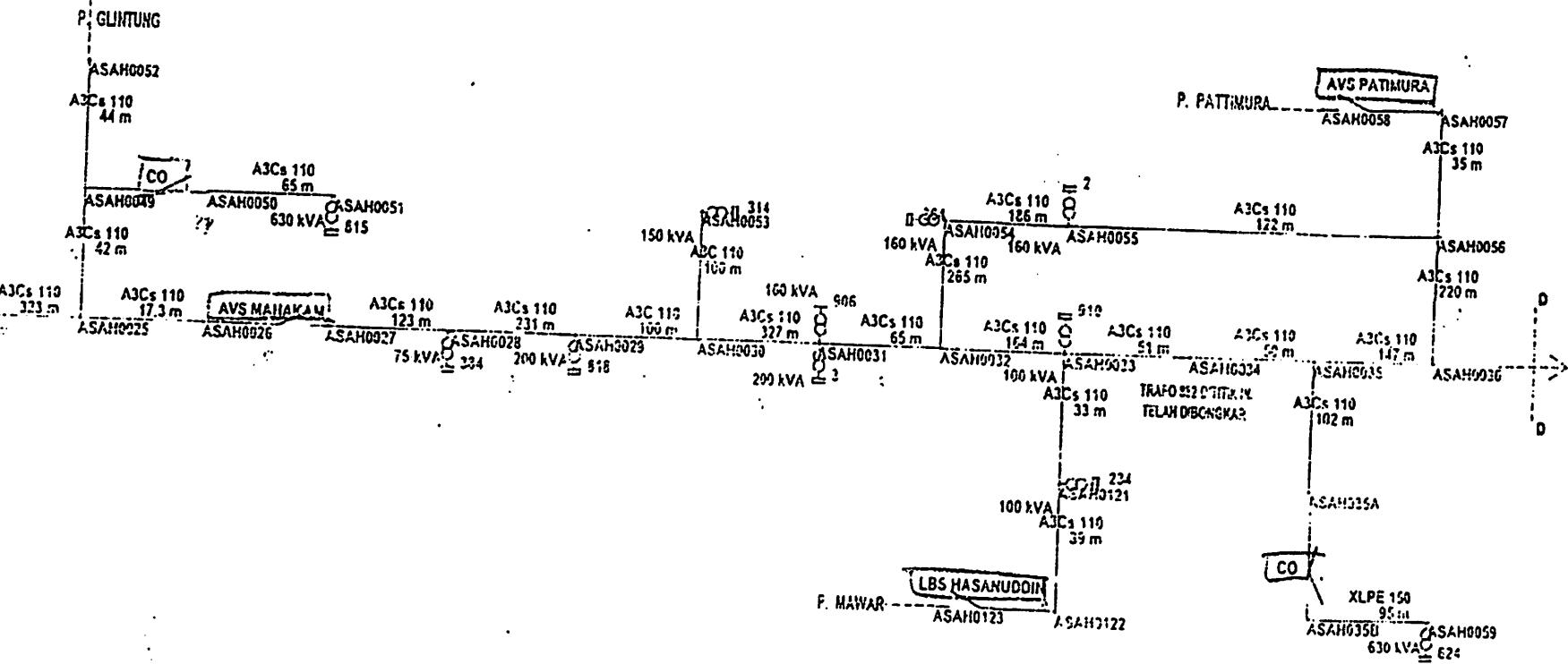
	PT. PLN (PERSERO) JASA ENGINERING	
Project Name:	PENUPUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MAIANG	
Date:	SINGLE LINE DIAGRAM PENYULANG GLINTUNG	
Design By:	Drawing No:	
Drawn By:		
Approved By:		
Checked By:		
Date:	Rev: 0	



		PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG			
Title: SINGLE LINE DIAGRAM PENYULANG ASAHAN			
Drawn By	Drawing No		
Examiner By			
Approved By			
Checked By			
Date	Rev 0		



 Project Name		PT. PLN (PERSERO) JASA ENGINERING	
PENURUNAN SUSUT KWH			
PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG			
Title			
SINGLE LINE DIAGRAM PENYULANG ASAHAH			
Design By	Drawn By	Checking By	
			4162-3103 L-51-US-ASHII-013 02 0
Approved By			
Checked By			
Date	Rev	0	
2/1			



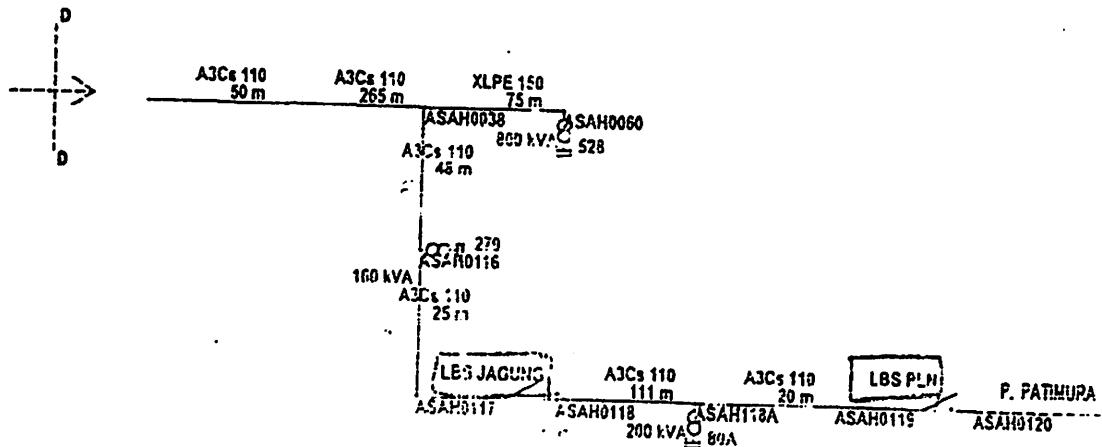
PT. PLN (PERSERO)
JASA ENGINERING

Project Name:

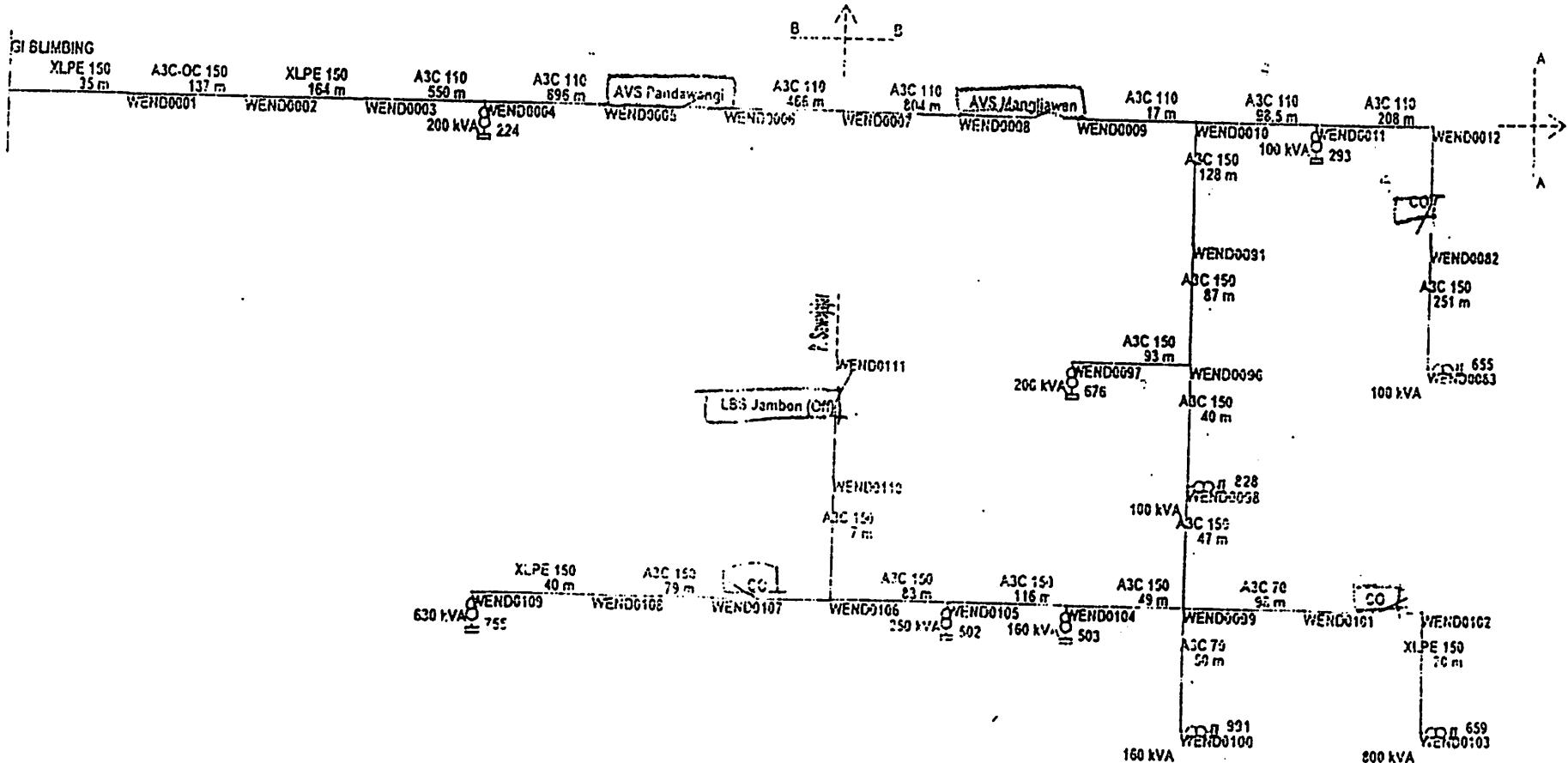
PENURUMAN SUSUT KWH
PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG

SINGLE LINE DIAGRAM PENYULANG AGAMAN

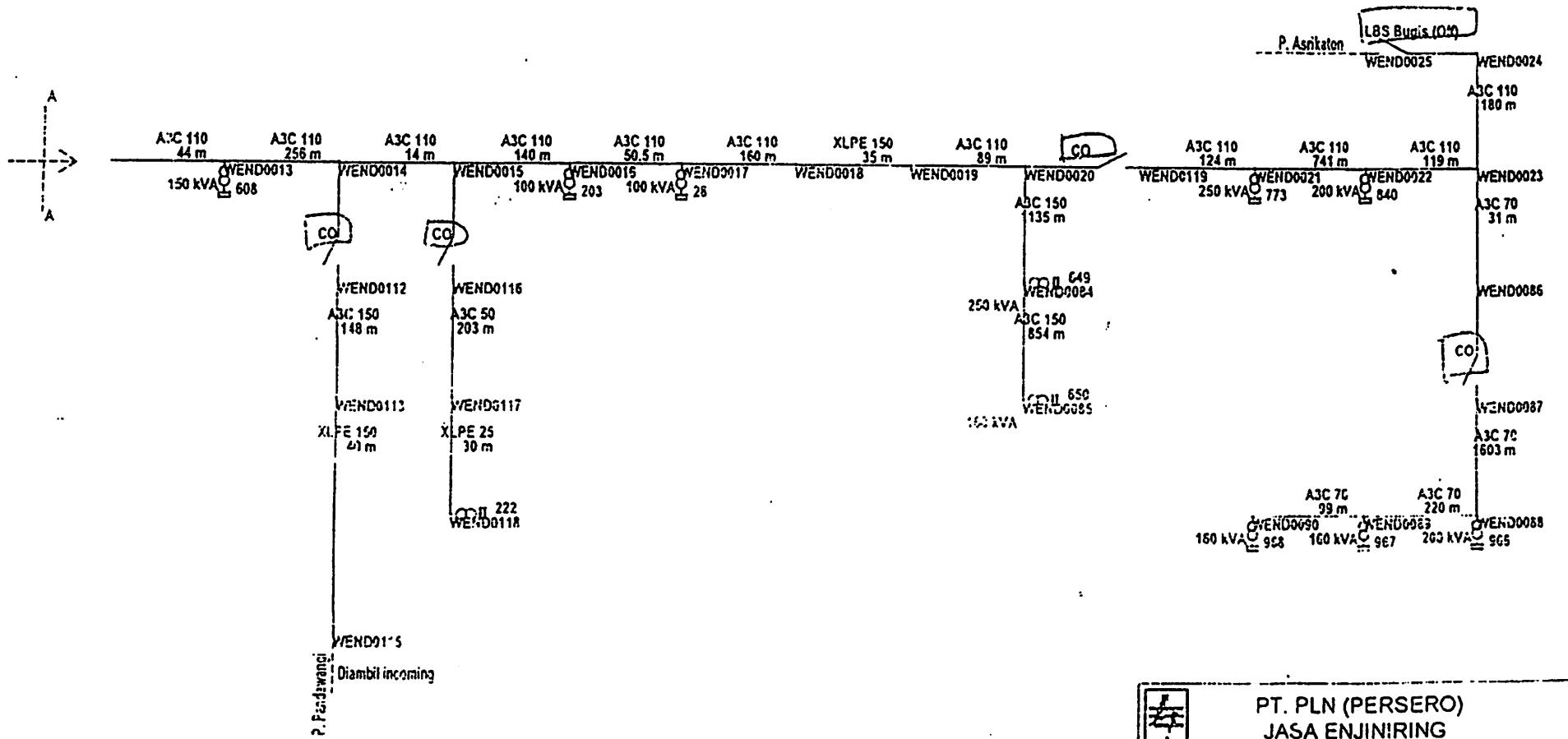
Design By	Drawing No.
Drawn By	
Approved By	4162.3104.1.51.05.A0301015710
Checked By	
Date	Rev. 0



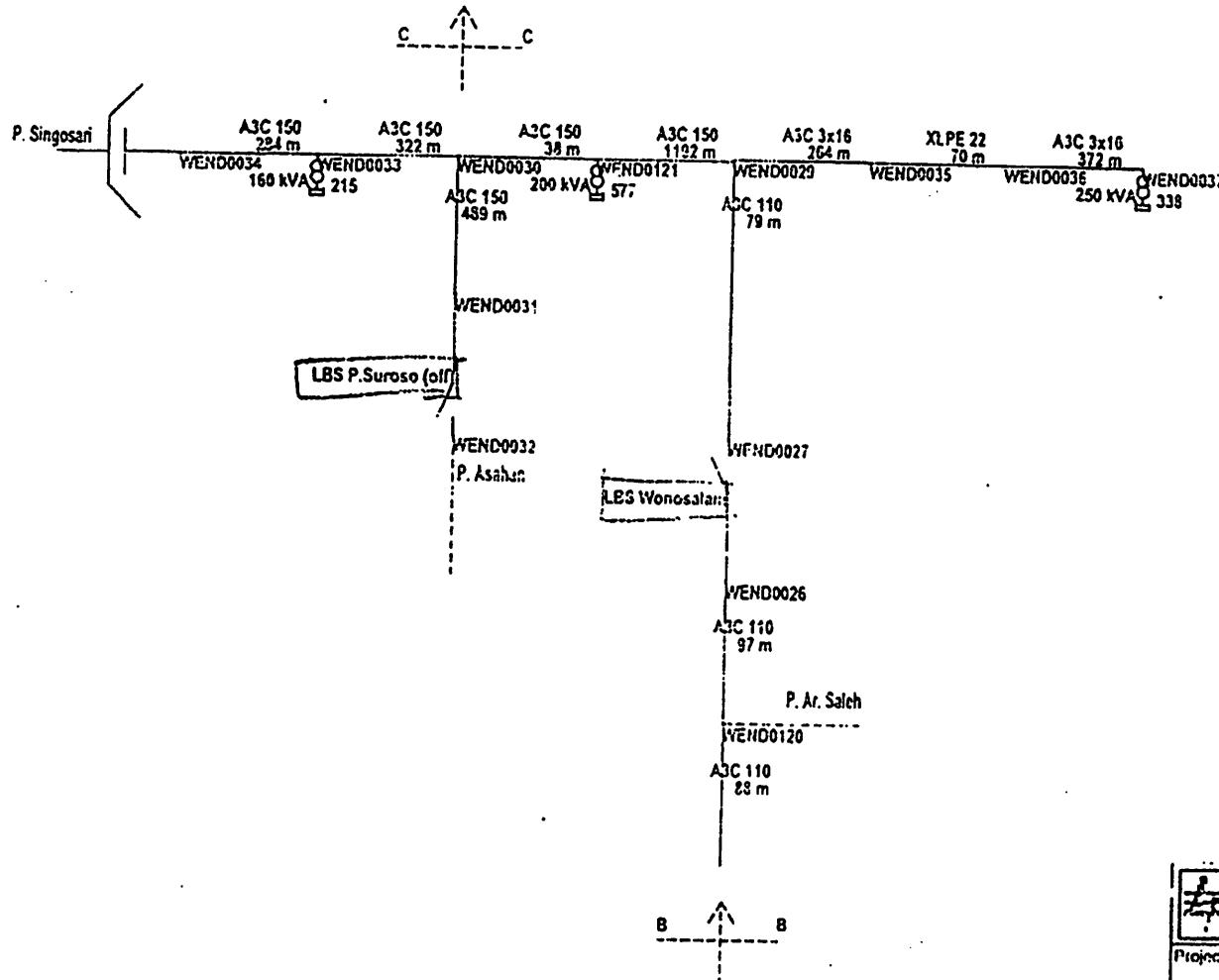
	PT. PLN (PERSERO) JASA ENGINERING
Project Name PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title SINGLE LINE DIAGRAM PENYULANG ASAHAN	
Design by _____ Drawn By _____ Approved By _____ Checked By _____ Date _____	Drawing No. 4162-3100-L-51-US-ASHII-013-040
Prev	0 1 2 3 4 5 6 7 8 9



 PT. PLN (PERSERO) JASA ENGINERING	
Project Name: PENURUNAN SUSUT KWH	
PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG WENDIT	
Design By:	Drawing No:
Drawn By:	
Approved By:	4102-31091-51-IC-VCEPR-02-01-0
Checked By:	
Date:	Rev: 0

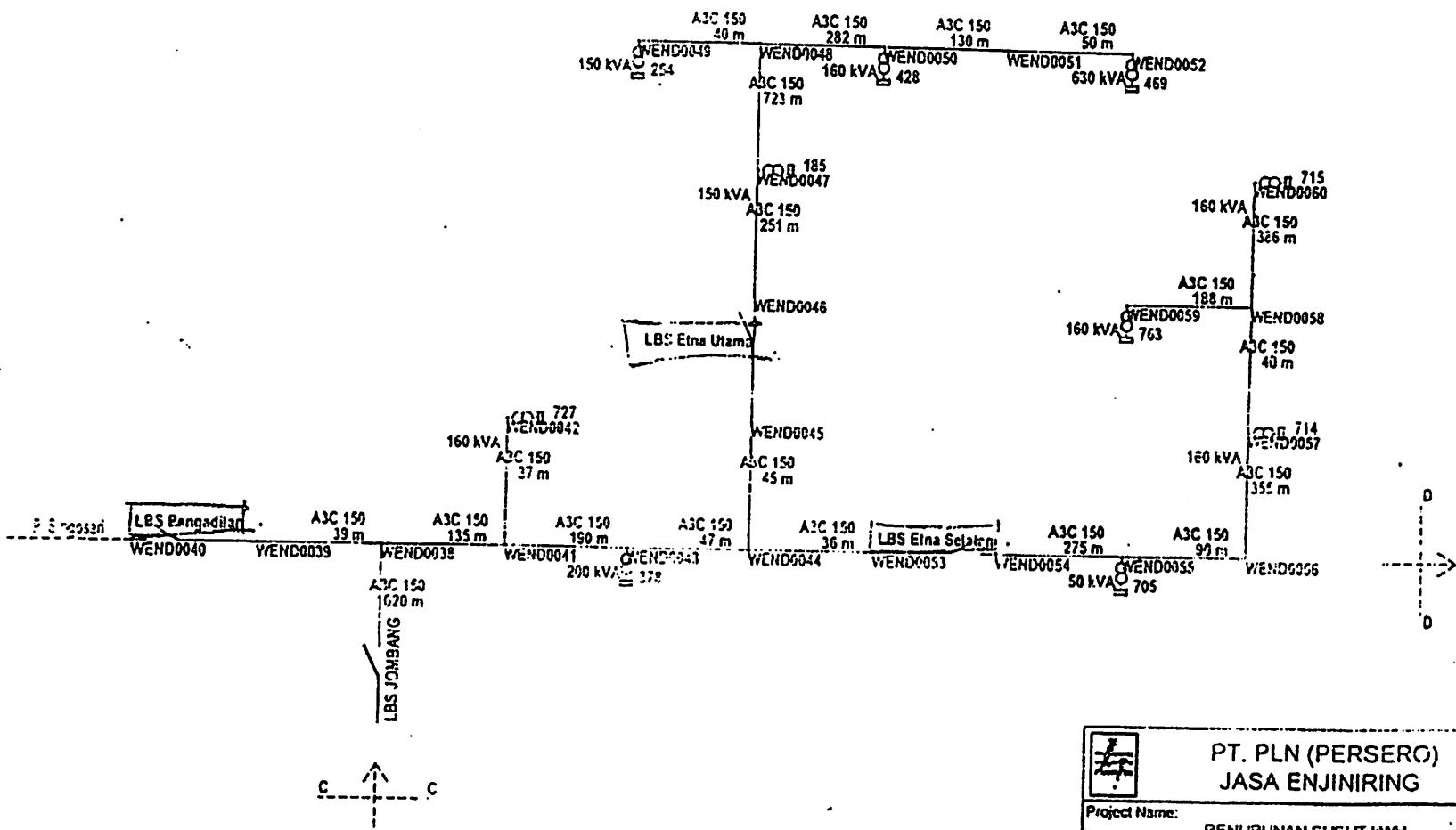


	PT. PLN (PERSERO) JASA ENGINERING
Project Name: PENURUNAN SUGUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG WENDIT	
Design By	Drawing No:
Drawn By	
Approved By	4102-3107-L-51-DG-V4, ND 012-02-0
Checked By	
Date	Rev.: 0

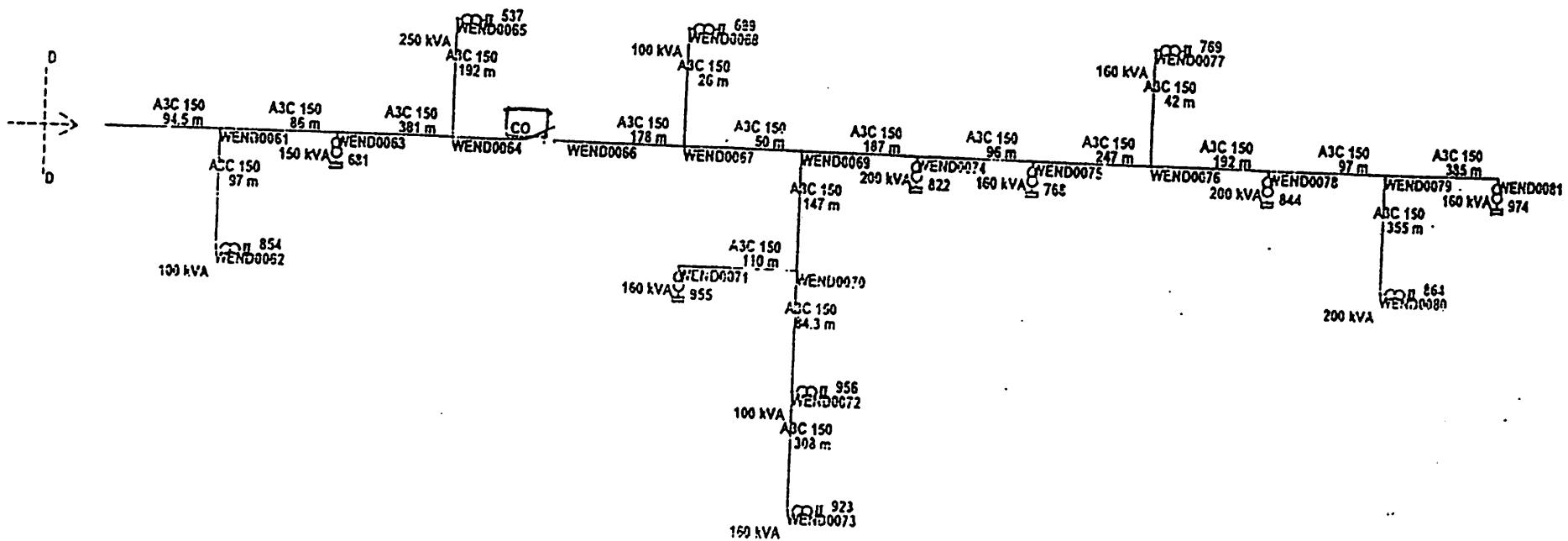


	PT. PLN (PERSERO) JASA ENGINERING
Project Name	
PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title	
SINGLE LINE DIAGRAM PENYULANG WENDIT	
Design By	Drawing No.
Drawn By	
Approved By	4162-3100-L-51-DS-WEND-012-03-0
Checked By	
Date	
Rev	0

3/5



	PT. PLN (PERSERO) JASA ENGINERING
Project Name: PENURUNAN SUSUT KWH PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG WENDIT	
Design By:	Drawing No:
Drawn By:	
Approved By:	4162-31(0) L-S1-DS-WEND 012/64-0
Checked By:	
Date:	Rev.: 0 . .



 PT. PLN (PERSERO) JASA ENGINERING.	
Project Name: PENURUNAN SUSUT KWH	
PT. PLN (PERSERO) DISTRIBUSI JATIM - CABANG MALANG	
Title: SINGLE LINE DIAGRAM PENYULANG WENDIT	
Drawn By:	Drawing No.:
Drawn By:	4162-3100 L-51-DS-WEND-012-05-0
Approved By:	
Checked By:	
Date:	Rev: 0
SUS	

LAMPIRAN 2



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

LEMBAR BIMBINGAN SKRIPSI

1. Nama : Zakiah Kustini
2. NIM : 00.12.108
3. NIRM : -
4. Jurusan : Teknik Elektro S-1
5. Konsentrasi : Teknik Energi Listrik
6. Judul Skripsi : ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER DARI GI BLIMBING MALANG DENGAN MENGGUNAKAN METODE *PARALLEL GENETIC ALGORITHM*
7. Tanggal Mengajukan Skripsi : 17 Maret 2005
8. Tanggal Menyelesaikan Skripsi : 10 Februari 2006
9. Dosen Pembimbing : Ir. Soemarwanto
10. Telah dievaluasi dengan nilai : 80 (delapan puluh) *R...*

Malang, Maret 2006

Mengetahui
Ketua Jurusan Teknik Elektro S1

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

Disetujui
Dosen Pembimbing

Ir. Soemarwanto
NIP. 130873485



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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

1. Nama : ZAKIAH KUSTINI
2. NIM : 00.12.108
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Energi Listrik
5. Judul Skripsi : ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER DARI GI BLIMBING MALANG DENGAN MENGGUNAKAN METODE *PARALLEL GENETIC ALGORITHM*

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

Hari : Sabtu
Tanggal : 18 Maret 2006
Dengan Nilai : 76,25 (B+) *S*



Panitia Ujian Skripsi

Ir. Mochtar Asroni, MSME
Ketua

Ir. F. Yudi Limpraptono, MT
Sekretaris

Anggota Penguji

Ir. Sidik Nurcjahyono, MT
Penguji Pertama

Ir. H. Choiri
Penguji Kedua



FORMULIR BIMBINGAN SKRIPSI

Nama : ZAKIAH KUSTINI
Nim : 00.12.108
Masa Bimbingan : 30 November 2005 s/d 30 Mei 2006
Judul Skripsi : ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI PRIMER DARI GARDU INDUK BLIMBING MALANG DENGAN MENGGUNAKAN METODE PARALEL ALGORITMA GENETIKA

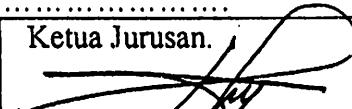
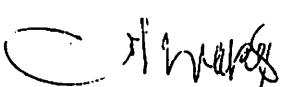
No.	Tanggal	Uraian	Parap Pembimbing
1.	09-08-05	Bab I : Perbaiki saran dan sub-bab dan Kalimat Bab II : Perbaiki /Berkalimat ; Teks drama tetapi tulis	/Ma
2.	12-08-05	Bab I dan Bab II : Acc	/Ma
3.	19-08-05	Bab III : Perbaiki	/Ma
4.	22-08-05	Bab III : Acc	/Ma
5.	01-09-05	Bab IV : Perbaiki	/Ma
6.	07-12-05	Bab V, Acc, Salijore.	/Ma
7.	06-01-06	Bab VI : Perbaiki, perbaiki dengan sub. bab 1, 2 pertama juga	/Ma
8.	10-01-06	Acc	/Ma
9.			
10.			

Malang, 10 - 02 - 2006
Dosen Pembimbing,

Ir. H. Soemarwanto

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

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika *)

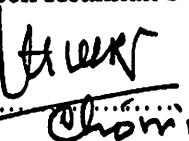
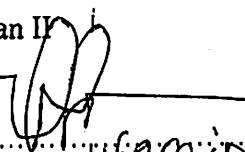
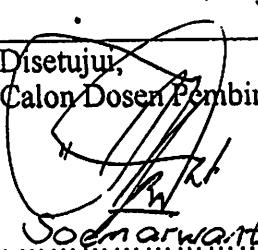
1	Nama Mahasiswa : ZAKIAH KUSTINI		Nim : 00.12.108
2	Waktu pengajuan	Tanggal : 17	Bulan : MARET Tahun : 2005
3	Spesifikasi judul (berilah tanda silang)		
	<input checked="" type="checkbox"/> a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri	e. Elektronika & Komponen f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainya	
4	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen *) :		Ketua Jurusan.
	... Ir. F. Yudij Limpraptono, MT Nip.Y: 10395002/4		 Ir. F. Yudij Limpraptono, MT Nip.Y: 10395002/4
5	Judul yang diajukan mahasiswa :	ANALISIS METODE PARALLEL GA UNTUK PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI DI GI. BLUMBING MALANG	
6	Perubahan Judul yang disetujui Dosen sesuai materi bidang ilmu	
7	Persetujuan Judul Skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui, 14 April 2005 Dosen  Ir. I. Maf. Wstantini, MT	

Perhatian :

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

BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika")

1	Nama Mahasiswa : ZAKIAH KUSTINI		Nim : 00.12.108
2	Keterangan	Tanggal 30	Waktu
	Pelaksanaan	Mei 2005	Tempat Ruang : Aula C
3	Spesifikasi judul **:		
	a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri	e. Elektronika & Komponen f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainnya	
4	Judul Proposal yang diseminarkan Mahasiswa	Analisis Metode Parallel GA Untuk Perbaikan Pelumasan Produ Saluran Distribusi SI GI Blimbing Malang	
5	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	
6	Catatan :		
7	Persetujuan Judul Skripsi : Disetujui, Dosen Keahlian I  Chomi Disetujui, Dosen Keahlian II  Hf AM id Mengetahui, Ketua Jurusan. Ir. F. Yudi Limpraptono, MT Nip. Y. 1039500274 Disetujui, Calon Dosen Pembimbing ybs.  Soemarwanto		

Perhatian :

**) coret yang tidak perlu

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



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

ERSERO) MALANG
NIAGA MALANG

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

Malang, 5 Des. 2005

Nomor : ITN-916/1.SKP /2/'05
Lampiran : satu lembar
Perihal : **BIMBINGAN SKRIPSI**

Kepada : Yth. Sdr. Ir. H. SOEMARWANTO *)
Dosen Institut Teknologi Nasional
di - Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam perpanjangan masa
bimbingan skripsi yang telah dilakukan untuk mahasiswa :

Nama	: ZAKIAH KUSTINI
Nim	: 0012108
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro
Konsentrasi	: T. Energi Listrik (S-1)

Dengan ini pembimbingan skripsi tersebut kami serahkan kembali
sepenuhnya kepada Saudara/I, selama masa waktu **6 (Enam) bulan**
terhitung mulai tanggal:

30 Nov. 2005 s/d 30 Mei 2006

Adapun tugas tersebut merupakan salah satu syarat untuk memperoleh
gelar Sarjana Teknik, Jurusan Teknik Elektro, apabila lewat dari batas
waktu tsb, maka, skripsinya akan digugurkan.
Demikian atas perhatian serta kerjasama yang baik kami ucapkan terima
kasih



Ir. F. Yudi Limpraptono, MT
Nip. 1039500274

Tindasan :

1. *) Perpanjangan Kontrak
2. Mahasiswa yang bersangkutan
3. Arsip.

Form. S-4a



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

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

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

I (PERSERO) MALANG
NIAGA MALANG

Nomor : ITN-1443/III.TA/2/2005
Lampiran : -
Perihal : Survey / Permintaan Data

Kepada : Yth Pimpinan
PT. PLN (Persero)
Area Pelayanan dan Jaringan
Jl. Jendral Basuki Rahmat No. 100
di-
Malang

Malang, 23 Juni 2005

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

ANALISIS PERBAIKAN PELAYANAN PADA SALURAN DISTRIBUSI DI GI BLIMBING MALANG DENGAN MENGGUNAKAN METODE PARAREL GA

Mahasiswa tersebut adalah :

1. Zakian Kustini Nim : 00.12.108

Adapun lama Survey adalah : 2 (Dua) Minggu

Demikian agar maklum dan atas perhatian serta bantuanya kami ucapkan terima kasih.



Tembusan disampaikan kepada Yth :

1. Ketua Jurusan
2. Arsip

Form.SK-2

PT. PLN (PERSERO)

DISTRIBUSI JAWA TIMUR

AREA PELAYANAN & JARINGAN MALANG

: 0341 - 326034 (Hunting)

: 211

Facsimile : 0341 - 362046

: 7097/330/APJ - MLG / 2005

Ir.No : ITN-1443/III.TA/2/2005

an :
: BALASAN SURVEY

Malang , 9 September 2005

Kepada
ITN Malang
Fakultas Industri
Jl. Raya Karanglo,Km 2
Di
Malang

Menjawab surat Saudara nomor ITN-1443/III.TA/2/2005 Tanggal 23 Juni 2005 perihal tersebut diatas, dengan ini kami beritahukan bahwa pada prinsipnya kami tidak keberatan / mengizinkan Mahasiswa Saudara atas nama :

Zakian Kustini

Nim. 00.12.108

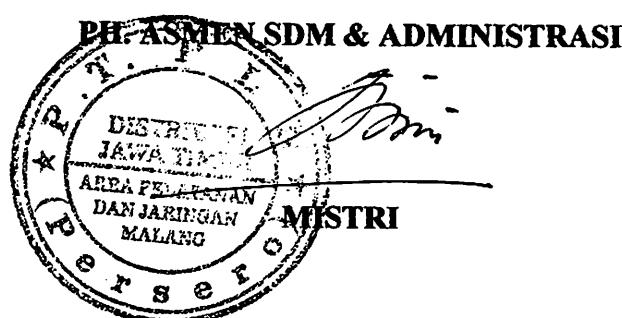
Untuk melaksanakan Survey / Permintaan Data di AP&J Malang PT. PLN (Persero) Area Pelayanan dan Jaringan Malang mulai tanggal 12 September 2005 sampai dengan 30 September 2005 dengan catatan PLN hanya memberikan data yang **TIDAK BERSIFAT RAHASIA**.

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

Demikian agar menjadikan maklum.

Tembusan :

1. Supervisor Terkait
2. Manajer UJ.Malang



LAMPIRAN 3

*LISTING PROGRAM :
LISTING ALGORITMA GENETIKA
LISTING FITNESS
LISTING ALIRAN DAYA
LISTING HASIL*

Listing Algoritma Genetika

```
unit uGenetic;

interface

uses uUtils,uFitness,uRandom;

type
  TIndiBin1=record
    chrom:bArr1;
    fitness:double;
  end;
  TPopBin1=array of TIndiBin1;

TGenetic=class
private
  FMaxGen,FPopSize,FLength:integer;
  FPCross,FPMutasi,FKa:double;
  function getMin:dArr1;
  function getAvg:dArr1;
  function getMax:dArr1;
protected
  FMin,FAvg,FMax:dArr1;
  FRandom:TRandomu;
public
  constructor Create(const rMaxGen,rPopSize,rLength:integer;
                    const rPCross,rPMutasi,rKa:double);
  destructor Destroy;override;
  property MaxGen:integer read FMaxGen write FMaxGen;
  property PopSize:integer read FPopSize write FPopSize;
  property Length:integer read FLength write FLength;
  property PCross:double read FPCross write FPCross;
  property PMutasi:double read FPMutasi write FPMutasi;
  property Ka:double read FKa write FKa;
  property Min:dArr1 read getMin;
  property Avg:dArr1 read getAvg;
  property Max:dArr1 read getMax;
end;

TGEnRecon=class(TGenetic)
private
  FNloop,FNFault:integer;
  FMin1,FAvg1,FMax1,FSumFitness:double;
  FParent,FChild:TPopBin1;
  FBestIndi:TIndiBin1;
  function getIndividu(const rIndi:TIndiBin1):TIndiBin1;
  procedure RepairChrom(var rChrom:bArr1);
  procedure InitParent;
  procedure Statistik;
```

```

function FindIndividuMax:TIndiBin1;
function Seleksi:integer;
function Mutasi(const rAllele:boolean):boolean;
procedure Crossover(const rParent1,rParent2:bArr1;
    var rChild1,rChild2:bArr1);
procedure Generasi;
procedure Replikasi;
public
constructor Create(const rMaxGen,rPopSize,rLength,rNloop,rNFault:integer;
    const rPCross,rPMutasi,rKa:double);
function DecodeDataToIndi(const rData:iArr1;
    const rFitness:double):TIndiBin1;
procedure DecodeIndiToData(const rIndi:TIndiBin1;
    var rData:iArr1;
    var rFitness:double);
function getBestIndi:TIndiBin1;
procedure FillAnotherIndi(var rIndi:TIndiBin1);
property NFault:integer read FNFault write FNFault;
end;

var gGARecon:TGenRecon;

implementation

{ TGenetic }

//constructor
constructor TGenetic.Create(const rMaxGen,rPopSize,rLength:integer;
    const rPCross,rPMutasi,rKa:double);
begin
inherited Create;
FMaxGen:=rMaxGen;
FPopSize:=rPopSize;
FLength:=rLength;
FPCross:=rPCross;
FPMutasi:=rPMutasi;
FKa:=rKa;
SetLength(FMin,FMaxGen);
SetLength(FAvg,FMaxGen);
SetLength(FMax,FMaxGen);
FRandom:=TRandomu.Create;
end;

//destructor
destructor TGenetic.Destroy;
begin
try
    FRandom.Free;
finally
    inherited Destroy;
end;
end;

```

```

function TGenetic.getAvg:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
  begin
    result[i]:=FAvg[i];
  end;
end;

function TGenetic.getMax:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
  begin
    result[i]:=FMax[i];
  end;
end;

function TGenetic.getMin:dArr1;
var i:integer;
begin
  SetLength(result,FMaxGen);
  for i:=0 to FMaxGen-1 do
  begin
    result[i]:=FMin[i];
  end;
end;

{ TGenRecon }

constructor TGenRecon.Create(const rMaxGen,rPopSize,rLength,rNloop,
                           rNFault:integer;
                           const rPCross,rPMutasi,rKa:double);
begin
  inherited Create(rMaxGen,rPopSize,rLength,rPCross,rPMutasi,rKa);
  FNloop:=rNloop;
  FNFault:=rNFault;
  InitParent;
  FBestIndi:=FindIndividuMax;
end;

function TGenRecon.getIndividu(const rIndi:TIndiBin1):TIndiBin1;
var i:integer;
begin
  SetLength(result.chrom,FLength);
  for i:=0 to FLength-1 do
  begin
    result.chrom[i]:=rIndi.chrom[i];
  end;
end;

```

```

result.fitness:=rIndi.fitness;
end;

procedure TGenRecon.RepairChrom(var rChrom:bArr1);
var i,sa,cu,no:integer;
begin
  sa:=0;
  for i:=0 to Length-1 do
  begin
    if rChrom[i]=true then
    begin
      inc(sa);
    end;
  end;
  if sa>FNloop then
  begin
    cu:=sa-FNloop;
    for i:=0 to cu-1 do
    begin
      repeat
        no:=FRandom.NextInt(0,Length-1);
      until rChrom[no]=true;
      rChrom[no]:=false;
    end;
  end
  else if sa<FNloop then
  begin
    cu:=FNloop-sa;
    for i:=0 to cu-1 do
    begin
      repeat
        no:=FRandom.NextInt(0,Length-1);
      until rChrom[no]=false;
      rChrom[no]:=true;
    end;
  end;
  if rChrom[FNFault]=false then
  begin
    repeat
      no:=FRandom.NextInt(0,Length-1);
    until rChrom[no]=true;
    rChrom[FNFault]:=true;
    rChrom[no]:=false;
  end;
end;
end;

function TGenRecon.FindIndividuMax:TIndiBin1;
var i:integer;
begin
  result:=getIndividu(FParent[0]);
  for i:=1 to PopSize-1 do
  begin

```

```

if result.fitness<FParent[i].fitness then
begin
  result:=getIndividu(FParent[i]);
end;
end;
end;

procedure TGenRecon.InitParent;
var i,j,no:integer;
begin
  SetLength(FParent,PopSize);
  SetLength(FChild,PopSize);
  for i:=0 to PopSize-1 do
  begin
    SetLength(FParent[i].chrom,Length);
    SetLength(FChild[i].chrom,Length);
    FParent[i].chrom[FNFault]:=true;
    for j:=0 to FNloop-2 do
    begin
      repeat
        no:=FRandom.NextInt(0,Length-1);
      until FParent[i].chrom[no]=false;
      FParent[i].chrom[no]:=true;
    end;
    FParent[i].fitness:=Ka/gFit.CalcFitness(FParent[i].chrom);
  end;
  SetLength(FBestIndi.chrom,Length);
end;

procedure TGenRecon.Statistik;
var i:integer;
begin
  FMin1:=FParent[0].fitness;
  FMax1:=FParent[0].fitness;
  FSUMFitness:=FParent[0].fitness;
  for i:=1 to PopSize-1 do
  begin
    if FMin1>FParent[i].fitness then
    begin
      FMin1:=FParent[i].fitness;
    end;
    if FMax1<FParent[i].fitness then
    begin
      FMax1:=FParent[i].fitness;
    end;
    FSUMFitness:=FSUMFitness+FParent[i].fitness;
  end;
  FAvg1:=FSUMFitness/PopSize;
end;

function TGenRecon.Seleksi:integer;
var i:integer;

```

```

    sum,partsum:double;
begin
  i:=0;
  sum:=0;
  partsum:=FRandom.NextDouble*FSumFitness;
repeat
  sum:=sum+FParent[i].fitness;
  inc(i);
until (sum>partsum) or (i>=(PopSize-1));
result:=i;
end;

function TGenRecon.Mutasi(const rAllele:boolean):boolean;
begin
  if FRandom.NextBoolean(PMutasi)=true then
begin
  result:=not rAllele;
end
else
begin
  result:=rAllele;
end;
end;

procedure TGenRecon.Crossover(const rParent1,rParent2:bArr1;
var rChild1,rChild2:bArr1);
var i,pos:integer;
begin
if FRandom.NextBoolean(PCross)=true then
begin
  pos:=FRandom.NextInt(0,(Length-2));
  for i:=0 to pos do
  begin
    rChild1[i]:=Mutasi(rParent1[i]);
    rChild2[i]:=Mutasi(rParent2[i]);
  end;
  for i:=pos+1 to Length-1 do
  begin
    rChild1[i]:=Mutasi(rParent2[i]);
    rChild2[i]:=Mutasi(rParent1[i]);
  end;
end
else
begin
  for i:=0 to Length-1 do
  begin
    rChild1[i]:=Mutasi(rParent1[i]);
    rChild2[i]:=Mutasi(rParent2[i]);
  end;
end;
end;

```

```

procedure TGenRecon.Generasi;
var i,mate1,mate2:integer;
begin
  i:=0;
  repeat
    mate1:=Seleksi;
    mate2:=Seleksi;
    Crossover(FParent[mate1].chrom,FParent[mate2].chrom,
    FChild[i].chrom,FChild[i+1].chrom);
    RepairChrom(FChild[i].chrom);
    FChild[i].fitness:=Ka/gFit.CalcFitness(FChild[i].chrom);
    RepairChrom(FChild[i+1].chrom);
    FChild[i+1].fitness:=Ka/gFit.CalcFitness(FChild[i+1].chrom);
    i:=i+2;
  until i>=PopSize;
end;

procedure TGenRecon.Replikasi;
var i,pos:integer;
  tmpPop:TPopBin1;
begin
  SetLength(tmpPop,PopSize);
  for i:=0 to PopSize-1 do
  begin
    repeat
      pos:=FRandom.NextInt(0,(PopSize-1));
    until pos<>i;
    if FChild[i].fitness>FParent[pos].fitness then
    begin
      tmpPop[i]:=getIndividu(FChild[i]);
    end
    else
    begin
      tmpPop[i]:=getIndividu(FParent[pos]);
    end;
  end;
  for i:=0 to PopSize-1 do
  begin
    FParent[i]:=getIndividu(tmpPop[i]);
  end;
end;

function TGenRecon.DecodeDataToIndi(const rData:iArr1;
  const rFitness:double):TIndiBin1;
var i:integer;
begin
  SetLength(result.chrom,Length);
  for i:=0 to high(rData) do
  begin
    result.chrom[rData[i]]:=true;
  end;
  result.fitness:=rFitness;

```

```

end;

procedure TGenRecon.DecodeIndiToData(const rIndi:TIndiBin1;
  var rData:iArr1;
  var rFitness:double);
var i,sa:integer;
begin
  SetLength(rData,FNloop);
  sa:=0;
  for i:=0 to Length-1 do
  begin
    if rIndi.chrom[i]=true then
    begin
      rData[sa]:=i;
      inc(sa);
    end;
  end;
  rFitness:=rIndi.fitness;
end;

function TGenRecon.getBestIndi:TIndiBin1;
begin
  result:=getIndividu(FBestIndi);
end;

procedure TGenRecon.FillAnotherIndi(var rIndi:TIndiBin1);
var i,posmin:integer;
  minIndi:TIndiBin1;
begin
  posmin:=0;
  minIndi:=getIndividu(FParent[0]);
  for i:=1 to PopSize-1 do
  begin
    if minIndi.fitness>FParent[i].fitness then
    begin
      minIndi:=getIndividu(FParent[i]);
      posmin:=i;
    end;
  end;
  FParent[posmin]:=getIndividu(rIndi);
  Statistik;
  Generasi;
  Replikasi;
  Statistik;
  rIndi:=FindIndividuMax;
  if FBestIndi.fitness<rIndi.fitness then
  begin
    FBestIndi:=getIndividu(rIndi);
  end;
end;
end;

```

Listing Fitness

```
unit uFitness;

interface

uses uUtils,uComplex,uLoadflow,uNewtonRaphson,uRecursive,
SysUtils,Classes,uMatrix;

type
TIndividu=record
chrom:iArr1;
fitness:double;
end;

TPopulasi=array of TIndividu;

TFitness=class
private
FNbus,FNsal,FNLoop,FNFault:integer;
FVb:TBatas;
FV,FSg,FSL:CxArr1;
FCap:dArr1;
FTypBus,FData,FDataAkhir:iArr1;
FZ,FTP:CxArr2;
FLc,FTr:dArr2;
FSwith:TSwithArr1;
FBestIndi:TIndividu;
function getIndividu(const rIndi:TIndividu):TIndividu;
function CekStopRecomb(const rData:iArr1):boolean;
function isSameData(const rStr1,rStr2:string):boolean;
function isSameDataList(var rList:TStringList;
const rStr:string):boolean;
function isContainData(const rNo:integer;
const rData:iArr1):boolean;
function CalcVPinalty(const rV:CxArr1):integer;
function CalcFitness(const rData:iArr1):double;overload;
function getChrom:iArr1;
function getBestFitness:double;
public
constructor Create(const rNLoop,rNFault:integer;
const rData,rDataAkhir:iArr1);
procedure CreateCombinasi(var rList:TStringList;
var rCount:integer);
function CalcFitness(const rData:bArr1):double;overload;
function GetSwith(const rData:iArr1):TSwithArr1;
property BestChrom:iArr1 read getChrom;
property NFault:integer read FNFault write FNFault;
property BestFitness:double read getBestFitness;
end;

var gFit:TFitness;
```

implementation

```
constructor TFitness.Create(const rNLoop,rNFault:integer;
                           const rData,rDataAkhir:iArr1);
var i,NData:integer;
begin
  inherited Create;
  FNLoop:=rNLoop;
  FNFault:=rNFault;
  DecodeCommDataToLFData(gBus,FNbus,FNsal,FV,FSg,FSL,FCap,FTypBus,
  gBranch,FZ,FTP,FLc,FTr);
  FSwith:=InitSwith(FZ);
  FVb.min:=0.95;
  FVb.max:=1.05;
  NData:=high(rData)+1;
  SetLength(FData,NData);
  SetLength(FDataAkhir,NData);
  for i:=0 to NData-1 do
  begin
    FData[i]:=rData[i];
    FDataAkhir[i]:=rDataAkhir[i];
  end;
end;

function TFitness.getIndividu(const rIndi:TIndividu):TIndividu;
var i:integer;
begin
  SetLength(result.chrom,FNLoop);
  for i:=0 to FNLoop-1 do
  begin
    result.chrom[i]:=rIndi.chrom[i];
  end;
  result.fitness:=rIndi.fitness;
end;

function TFitness.GetSwith(const rData:iArr1):TSwithArr1;
var i,NData:integer;
begin
  NData:=high(rData)+1;
  SetLength(result,NData);
  for i:=0 to NData-1 do
  begin
    if (rData[i]<0) or (rData[i]>=FNsal) then
    begin
      raise Exception.Create('Melebihi Indeks Matrik!');
    end;
    result[i].dari:=FSwith[rData[i]].dari;
    result[i].ke:=FSwith[rData[i]].ke;
  end;
end;
```

```

function TFitness.CekStopRecomb(const rData:iArr1):boolean;
var i:integer;
begin
  result:=true;
  for i:=0 to FNLoop-1 do
  begin
    if rData[i]<>FDataAkhir[i] then
    begin
      result:=false;
      break;
    end;
  end;
end;

function TFitness.isSameData(const rStr1,rStr2:string):boolean;
var Data1,Data2:iArr1;
  i,j,Count:integer;
begin
  Data1:=DecodeStrToData(rStr1);
  Data2:=DecodeStrToData(rStr2);
  result:=false;
  Count:=0;
  for i:=0 to high(Data1) do
  begin
    for j:=0 to high(Data2) do
    begin
      if Data1[i]=Data2[j] then
      begin
        inc(Count);
        break;
      end;
    end;
  end;
  if Count=high(Data1)+1 then result:=true;
end;

function TFitness.isSameDataList(var rList:TStringList;
  const rstr:string):boolean;
var i:integer;
begin
  result:=false;
  for i:=0 to rList.Count-1 do
  begin
    if isSameData(rStr,rList.Strings[i])=true then
    begin
      result:=true;
      break;
    end;
  end;
end;

function TFitness.isContainData(const rNo:integer;

```

```

        const rData:iArr1):boolean;
var i,NData:integer;
begin
  result:=false;
  NData:=high(rData)+1;
  for i:=0 to NData-1 do
  begin
    if rNo=rData[i] then
    begin
      result:=true;
      break;
    end;
  end;
end;

function TFitness.CalcVPinalty(const rV:CxArr1):integer;
var i:integer;
begin
  result:=0;
  for i:=0 to FNbus-1 do
  begin
    if rV[i].real>1.05 then inc(result);
    if rV[i].real<0.95 then inc(result);
  end;
end;

function TFitness.CalcFitness(const rData:iArr1):double;
var Vpin:integer;
  V,Sg,CxArr1;
  Zb,Alir,Arus:CxArr2;
  sw:TSwithArr1;
begin
  sw:=GetSwith(rData);
  Zb:=BuatZBaru(FZ,sw);
  V:=CopyMatrix(FV);
  Sg:=CopyMatrix(FSg);
  NewtonRaphson(gParamLF,V,Sg,FSL,FCap,FTypBus,
  Zb,FTp,Alir,Arus,FLc,FTr);
  Vpin:=CalcVPinalty(FV);
  result:=168*10*1000+1000*FNbus;
  if gParamLF.Iterasi<gParamLF.MaxIterasi then
  begin
    result:=gParamLF.SumLoss.real+1000*Vpin;
  end;
end;

function TFitness.CalcFitness(const rData:bArr1):double;
var Vpin:integer;
  V,Sg,CxArr1;
  Zb,Alir,Arus:CxArr2;
  TesRadial:boolean;
begin

```

```

Zb:=BuatZBaru(FZ,rData);
TesRadial:=TesJaringan(Zb);
if TesRadial=true then
begin
  V:=CopyMatrix(FV);
  Sg:=CopyMatrix(FSg);
  NewtonRaphson(gParamLF,V,Sg,FSL,FCap,FTypBus,
  Zb,FTp,Air,Arus,FLc,FTr);
  Vpin:=CalcVPinalty(FV);
  result:=gParamLF.SumLoss.real+1000*Vpin;
end
else
begin
  result:=(high(FV)+5)*1000;
end;
end;

procedure TFitness.CreateCombinasi(var rList:TStringList;
  var rCount:integer);
var CekRadial,CekCombinasi:boolean;
  str:string;
  sw:TSwithArr1;
  Zb:CxArr2;
  CekIndi:TIndividu;
begin
rList.Clear;
rCount:=0;
repeat
  GenCombinasi(FNsal,FData,CekCombinasi);
  if CekCombinasi=true then
  begin
    sw:=GetSwith(FData);
    Zb:=BuatZBaru(FZ,sw);
    CekRadial:=TesJaringan(Zb);
    if (CekCombinasi=true) and (CekRadial=true) and
      (isContainedData(FNFFault,FData)=true) then
    begin
      str:=DecodeDataToStr(FData);
      if isSameDataList(rList,str)=false then
      begin
        if rCount=0 then
        begin
          FBestIndi.chrom:=CopyMatrix(FData);
          FBestIndi.fitness:=CalcFitness(FData);
        end
        else
        begin
          CekIndi.chrom:=CopyMatrix(FData);
          CekIndi.fitness:=CalcFitness(FData);
          if CekIndi.fitness<FBestIndi.fitness then
          begin
            FBestIndi:=getIndividu(CekIndi);
          end;
        end;
      end;
    end;
  end;
end;

```

```
    end;
end;
rList.Add(str);
inc(rCount);
end;
end;
end;
until CekStopRecomb(FData)=true;
end;

function TFitness.getChrom:iArr1;
var i:integer;
begin
  SetLength(result,FNLoop);
  for i:=0 to FNLoop-1 do
  begin
    result[i]:=FBestIndi.chrom[i];
  end;
end;

function TFitness.getBestFitness:double;
begin
  result:=FBestIndi.fitness;
end;

end.
```

Listing Newton Raphson

```
unit uNewtonRaphson;

interface

uses uUtils,uComplex,uLoadflow,uMatrix;

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

implementation

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

```

    end;
    result[Nq]:=rSg[i].imag-rSL[i].imag-sumQ;
  end;
end;
end;

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

```

```

begin
if rTyp[i]<>1 then
begin
inc(row);
col:=rNbus-2;
for j:=0 to rNbus-1 do
begin
if rTyp[j]=3 then
begin
inc(col);
if j=i then
begin
sum:=0.0;
for k:=0 to rNbus-1 do
begin
//sum:=sum+((Gjk*cos(dj-dk)+Bjk*sin(dj-dk))*Uk);
sum:=sum+((rY[j,k].real*cos(rV[j].imag-rV[k].imag)+  

rY[j,k].imag*sin(rV[j].imag-rV[k].imag))*  

rV[k].real);
end;
//Pj:=sum*Uj;
Pj:=sum*rV[j].real;
//result[row,col]:=Pj+Gij*Ui;
result[row,col]:=Pj+rY[i,j].real*sqr(rV[i].real);
end
else
begin
//result[row,col]:=Uj*(Gij*cos(di-dj)+Bij*sin(di-dj));
result[row,col]:=rV[i].real*rV[j].real*(rY[i,j].real*  

cos(rV[i].imag-rV[j].imag)+rY[i,j].imag*  

sin(rV[i].imag-rV[j].imag));
end;
end;
end;
end;
end;
//Pembentukan Jaqobian M dQ/d0
row:=rNbus-2;
for i:=0 to rNbus-1 do
begin
if rTyp[i]=3 then
begin
inc(row);
col:=-1;
for j:=0 to rNbus-1 do
begin
if rTyp[j]<>1 then
begin
inc(col);
if j=i then
begin
sum:=0;
for k:=0 to rNbus-1 do
begin
//sum:=sum+((Gjk*cos(dj-dk)+Bjk*sin(dj-dk))*Uk);
sum:=sum+((rY[j,k].real*cos(rV[j].imag-rV[k].imag)+  

rY[j,k].imag*sin(rV[j].imag-rV[k].imag))*  

rV[k].real);
end;
end;
end;
end;

```

```

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

```

```

end;

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

procedure NewtonRaphson(var rBus:TBusArr1;
  var rBranch:TBranchArr1;
  var rParamLF:TPParamLF);
var i,Nbus,Nsal,Ngen:integer;
  max:double;
  V,Sg,SL:CxArr1;
  Cap,dS:dArr1;
  TypBus:iArr1;
  Z,Tp,Y,Alir,Anus:CxArr2;
  Lc,Tr,mJaq:dArr2;
begin
DecodeCommDataToLFDData(rBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,
rBranch,Z,Tp,Lc,Tr);
DecodeData(rParamLF,Nbus,V,Sg,SL,Cap,TypBus,Z,Tp,Lc,Tr);
Ngen:=FindSumGen(Nbus,TypBus);
Admitansi(Nbus,Z,Tp,Lc,Tr,Cap,Y);
rParamLF.Iterasi:=0;
for i:=0 to rParamLF.MaxIterasi-1 do
begin
  dS:=MismatchDaya(Nbus,Ngen,V,Sg,SL,TypBus,Y);
  max:=MaxdataArray(dS);
  if max<=rParamLF.Toleransi then break;
  mJaq:=Jaqobian(Nbus,Ngen,V,TypBus,Y);
  UpdateTegangan(Nbus,dS,mJaq,TypBus,V);
  inc(rParamLF.Iterasi);
end;

```

```

V:=PolarToRec(V);
AliranDaya(Nbus,V,Y,Lc,Alir);
ArusBranch(Nbus,V,Lc,Y,Arus);
DayaGen(Nbus,V,SL,Y,TypBus,Sg);
DayaSlack(Nbus,Alir,TypBus,SL,Sg);
V:=RecToPolar(V);
UpdateAkhir(Nbus,Nsal,rParamLF,V,Sg,SL,Alir,Arus,rBus,rBranch);
end;

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

```

Listing Hasil

```
unit uHasil;

interface

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

type
  TfrmHasil = class(TForm)
    TabSheet2: TTabSheet;
    TabSheet3: TTabSheet;
    TabSheet4: TTabSheet;
    TabSheet9: TTabSheet;
    Panel1: TPanel;
    btnClose: TButton;
    btnHitung: TButton;
    TabSheet1: TTabSheet;
    TabSheet5: TTabSheet;
    TabSheet6: TTabSheet;
    TabSheet7: TTabSheet;
    PageControl1: TPageControl;
    btnCreate: TButton;
    TabSheet8: TTabSheet;
    fgBus: TStringGrid;
    fgBranch: TStringGrid;
    GroupBox6: TGroupBox;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    lblGen: TLabel;
    lblLoad: TLabel;
    lblLoss: TLabel;
    edtSumGen: TEdit;
    edtSumLoad: TEdit;
    edtSumLoss: TEdit;
    edtIterasi: TEdit;
    edtTime: TEdit;
    GroupBox2: TGroupBox;
    Label6: TLabel;
    Label22: TLabel;
    edtNLoop: TEdit;
    cmbFault: TComboBox;
    Label7: TLabel;
    lbxPesan1: TListBox;
    Label13: TLabel;
    lbxPesan2: TListBox;
    Label15: TLabel;
    lbxPesan4: TListBox;
    lbxPesan3: TListBox;
    Label14: TLabel;
```

```
GroupBox4: TGroupBox;
Label18: TLabel;
edtmaxgen: TEdit;
Label19: TLabel;
edtpopsize: TEdit;
Label20: TLabel;
edtpcross: TEdit;
Label21: TLabel;
edtpmutasi: TEdit;
Label23: TLabel;
edtKa: TEdit;
lblPinality: TLabel;
edtPinality: TEdit;
TabSheet10: TTabSheet;
Chart1: TChart;
Series1: TLineSeries;
Series2: TLineSeries;
GroupBox1: TGroupBox;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
lblGen2: TLabel;
lblLoad2: TLabel;
lblLoss2: TLabel;
edtSumGen2: TEdit;
edtSumLoad2: TEdit;
edtSumLoss2: TEdit;
edtIterasi2: TEdit;
edtTime2: TEdit;
fgBranch2: TStringGrid;
fgBus2: TStringGrid;
GroupBox5: TGroupBox;
fgOS: TStringGrid;
GroupBox3: TGroupBox;
Label16: TLabel;
Label17: TLabel;
lblLossP: TLabel;
lblLossN: TLabel;
edtSebelum: TEdit;
edtSesudah: TEdit;
btnUseDefault: TButton;
btnSendParam: TButton;
pbCount: TProgressBar;
procedure btnCloseClick(Sender: TObject);
procedure btnHitungClick(Sender: TObject);
procedure btnCreateClick(Sender: TObject);
procedure btnUseDefaultClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure edtNLoopChange(Sender: TObject);
procedure btnSendParamClick(Sender: TObject);
procedure cmbFaultClick(Sender: TObject);
private
  { Private declarations }
function isDataOnList(const rStr:string;
  const rItems:TStrings):integer;
```

```

public
  { Public declarations }
end;

var
  frmHasil: TfrmHasil;

implementation

uses uUtils, uLoadflow, uNewtonRaphson, uComplex, uTopology,
  uRecursive,uMatrix, uFitness, uSetting, uVarGlobal;

{$R *.dfm}

var Nsal:integer;
  CostSebelum,CostSesudah:double;
  Data,DataAkhir:iArr1;

function TfrmHasil.isDataOnList(const rStr:string;
  const rItems:TStrings):integer;
var i:integer;
begin
  result:=-1;
  for i:=0 to rItems.Count-1 do
  begin
    if rStr=rItems[i] then
    begin
      result:=i;
      break;
    end;
  end;
end;

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

procedure TfrmHasil.btnHitungClick(Sender: TObject);
var i,ia,ja,Nbus:integer;
  mulai,selesai,selang:TDateTime;
  jam,menit,detik,mdetik:word;
  V,Sg,SL:CxArr1;
  Cap:dArr1;
  TypBus:iArr1;
  Z,Tp,Alir,Arus:CxArr2;
  Lc,Tr:dArr2;
  sw:TSwitchArr1;
begin
  DecodeCommDataToLFData(gBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,gBranch,
  Z,Tp,Lc,Tr);
  gZrec:=CopyMatrix(Z);
  Z[3,10]:=Cmplx(0,0);
  Z[8,12]:=Cmplx(0,0);
  Z[14,18]:=Cmplx(0,0);
  Z[4,15]:=Cmplx(0,0);
  btnUseDefault.Enabled:=true;

```

```

mulai:=time;
NewtonRaphson(gParamLF,V,Sg,SL,Cap,TypBus,Z,Tp,Alir,Arus,Lc,Tr);
selesai:=time;
selang:=selesai-mulai;
Series1.Clear;
for i:=0 to Nbus-1 do
begin
  fgBus.Cells[0,i+1]:=IntToStr(i+1);
  fgBus.Cells[1,i+1]:=RealToStr(V[i].real,5);
  fgBus.Cells[2,i+1]:=RealToStr(V[i].imag*
    ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
  fgBus.Cells[3,i+1]:=RealToStr(Sg[i].real,3);
  fgBus.Cells[4,i+1]:=RealToStr(Sg[i].imag,3);
  fgBus.Cells[5,i+1]:=RealToStr(SL[i].real,3);
  fgBus.Cells[6,i+1]:=RealToStr(SL[i].imag,3);
  fgBus.Cells[7,i+1]:=RealToStr(Cap[i],3);
  fgBus.Cells[8,i+1]:=IntToStr(TypBus[i]);
  Series1.Add(V[i].real,IntToStr(i+1));
end;
for i:=0 to high(gBranch) do
begin
  ia:=gBranch[i].dari-1;
  ja:=gBranch[i].ke-1;
  gBranch.Cells[0,i+1]:=IntToStr(i+1);
  gBranch.Cells[1,i+1]:=IntToStr(ia+1);
  gBranch.Cells[2,i+1]:=IntToStr(ja+1);
  gBranch.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
  gBranch.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
  gBranch.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
  gBranch.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
  gBranch.Cells[7,i+1]:=IntToStr(ja+1);
  gBranch.Cells[8,i+1]:=IntToStr(ia+1);
  gBranch.Cells[9,i+1]:=RealToStr(Alir[ja,ja].real,3);
  gBranch.Cells[10,i+1]:=RealToStr(Alir[ja,ja].imag,3);
  gBranch.Cells[11,i+1]:=RealToStr(Arus[ja,ja].real,3);
  gBranch.Cells[12,i+1]:=RealToStr(Arus[ja,ja].imag,3);
end;
edtSumGen.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi.Text:=IntToStr(gParamLF.Iterasi);
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime.Text:=IntToStr(jam)+':'+IntToStr(menit)+':'+
  IntToStr(detik)+':'+IntToStr(mdetik);
CostSebelum:=gParamLF.SumLoss.real;
sw:=InitSwith(gZrec);
for i:=0 to high(sw) do
begin
  cmbFault.Items.Add(IntToStr(sw[i].dari+1)+' - '+
  IntToStr(sw[i].ke+1));
end;
end;

procedure TfrmHasil.btnCreateClick(Sender: TObject);
var i,ia,ja,Nbus,Nsal,NFault,NLoop,pos:integer;
  Fita1,Fita2,Fita3,Fita4:double;
  jam,menit,detik,mdetik:word;

```

```

tesRadial:boolean;
mulai,selesai,selang:TDateTime;
str:TStringList;
Cap:dArr1;
Lc,Tr:dArr2;
TypBus,DataHasil,Data1:iArr1;
V,Sg,SL:CxArr1;
Zb,Tp,Alir,Arus:CxArr2;
fit:TFitness;
sw:TSwitchArr1;
begin
//new code place here
mulai:=time;
statPC1:=false;
statPC2:=false;
statPC3:=false;
statPC4:=false;
pbCount.Max:=StrToInt(edtMaxGen.Text);
for i:=1 to StrToInt(edtMaxGen.Text) do
begin
  str:=TStringList.Create;
  //data from PC1 send to PC2
  str.Text:=strPC1;
  if str.Strings[0]='Param' then
  begin
    Data1:=DecodeStrToData(str.Strings[2]);
    Fita1:=StrToFloat(str.Strings[3]);
  end
  else
  begin
    Data1:=DecodeStrToData(str.Strings[1]);
    Fita1:=StrToFloat(str.Strings[2]);
  end;
  str.Clear;
  str.Add('Execute');
  str.Add(DecodeDataToStr(Data1));
  str.Add(FloatToStr(Fita1));
  frmSetting.ClientSocket2.Socket.SendText(str.Text);
  //data from PC2 send to PC3
  str.Clear;
  str.Text:=strPC2;
  if str.Strings[0]='Param' then
  begin
    Data1:=DecodeStrToData(str.Strings[2]);
    Fita1:=StrToFloat(str.Strings[3]);
  end
  else
  begin
    Data1:=DecodeStrToData(str.Strings[1]);
    Fita1:=StrToFloat(str.Strings[2]);
  end;
  str.Clear;
  str.Add('Execute');
  str.Add(DecodeDataToStr(Data1));
  str.Add(FloatToStr(Fita1));
  frmSetting.ClientSocket3.Socket.SendText(str.Text);
  //data from PC3 send to PC4

```

```

str.Clear;
str.Text:=strPC3;
if str.Strings[0]='Param' then
begin
  Data1:=DecodeStrToData(str.Strings[2]);
  Fita1:=StrToFloat(str.Strings[3]);
end
else
begin
  Data1:=DecodeStrToData(str.Strings[1]);
  Fita1:=StrToFloat(str.Strings[2]);
end;
str.Clear;
str.Add('Execute');
str.Add(DecodeDataToStr(Data1));
str.Add(FloatToStr(Fita1));
frmSetting.ClientSocket4.Socket.SendText(str.Text);
//data from PC4 send to PC1
str.Clear;
str.Text:=strPC4;
if str.Strings[0]='Param' then
begin
  Data1:=DecodeStrToData(str.Strings[2]);
  Fita1:=StrToFloat(str.Strings[3]);
end
else
begin
  Data1:=DecodeStrToData(str.Strings[1]);
  Fita1:=StrToFloat(str.Strings[2]);
end;
str.Clear;
str.Add('Execute');
str.Add(DecodeDataToStr(Data1));
str.Add(FloatToStr(Fita1));
frmSetting.ClientSocket1.Socket.SendText(str.Text);
//end of send to All PC Client
repeat
  Application.ProcessMessages;
until (statPC1=true) and (statPC2=true) and (statPC3=true) and
      (statPC4=true);
statPC1:=false;
statPC2:=false;
statPC3:=false;
statPC4:=false;
str.Free;
pbCount.StepBy(1);
end;
str:=TStringList.Create;
str.Text:=strPC1;
Fita1:=StrToFloat(str.Strings[2]);
str.Clear;
str.Text:=strPC2;
Fita2:=StrToFloat(str.Strings[2]);
str.Clear;
str.Text:=strPC3;
Fita3:=StrToFloat(str.Strings[2]);
str.Clear;

```

```

str.Text:=strPC4;
Fita4:=StrToFloat(str.Strings[2]);
str.Clear;
pos:=1;
if Fita1<Fita2 then
begin
  Fita1:=Fita2;
  pos:=2;
end;
if Fita1<Fita3 then
begin
  Fita1:=Fita3;
  pos:=3;
end;
if Fita1<Fita4 then
begin
  pos:=4;
end;
str.Clear;
if pos=1 then
begin
  str.Text:=strPC1;
end
else if pos=2 then
begin
  str.Text:=strPC2;
end
else if pos=3 then
begin
  str.Text:=strPC3;
end
else if pos=4 then
begin
  str.Text:=strPC4;
end;
DataHasil:=DecodeStrToDate(str.Strings[1]);
str.Free;
NLoop:=StrToInt(edtNLoop.Text);
NFault:=isDataOnList(cmbFault.Text,cmbFault.Items);
fit:=TFitness.Create(NLoop,NFault,Data,DataAkhir);
sw:=fit.GetSwith(DataHasil);
fit.Free;
DecodeCommDataToLFData(gBus,Nbus,Nsal,V,Sg,SL,Cap,TypBus,gBranch,
Zb,Tp,Lc,Tr);
Zb:=BuatZBaru(gZrec,sw);
tesRadial:=TesJaringan(Zb);
if tesRadial=true then
begin
  NewtonRaphson(gParamLF,V,Sg,SL,Cap,TypBus,Zb,Tp,Alir,Arus,Lc,Tr);
end;
selesai:=time;
selang:=selesai-mulai;
Series2.Clear;
for i:=0 to Nbus-1 do
begin
  fgBus2.Cells[0,i+1]:=IntToStr(i+1);
  fgBus2.Cells[1,i+1]:=RealToStr(V[i].real,5);

```

```

fgBus2.Cells[2,i+1]:=RealToStr(V[i].imag*
      ANGLE_OF_DEGREES/ANGLE_OF_RADIAN,5);
fgBus2.Cells[3,i+1]:=RealToStr(Sg[i].real,3);
fgBus2.Cells[4,i+1]:=RealToStr(Sg[i].imag,3);
fgBus2.Cells[5,i+1]:=RealToStr(SL[i].real,3);
fgBus2.Cells[6,i+1]:=RealToStr(SL[i].imag,3);
fgBus2.Cells[7,i+1]:=RealToStr(Cap[i],3);
fgBus2.Cells[8,i+1]:=IntToStr(TypBus[i]);
Series2.Add(V[i].real,IntToStr(i+1));
end;
for i:=0 to high(gBranch) do
begin
  ia:=gBranch[i].dari-1;
  ja:=gBranch[i].ke-1;
  fgBranch2.Cells[0,i+1]:=IntToStr(i+1);
  fgBranch2.Cells[1,i+1]:=IntToStr(ia+1);
  fgBranch2.Cells[2,i+1]:=IntToStr(ja+1);
  fgBranch2.Cells[3,i+1]:=RealToStr(Alir[ia,ja].real,3);
  fgBranch2.Cells[4,i+1]:=RealToStr(Alir[ia,ja].imag,3);
  fgBranch2.Cells[5,i+1]:=RealToStr(Arus[ia,ja].real,3);
  fgBranch2.Cells[6,i+1]:=RealToStr(Arus[ia,ja].imag,3);
  fgBranch2.Cells[7,i+1]:=IntToStr(ja+1);
  fgBranch2.Cells[8,i+1]:=IntToStr(ia+1);
  fgBranch2.Cells[9,i+1]:=RealToStr(Alir[ja,ia].real,3);
  fgBranch2.Cells[10,i+1]:=RealToStr(Alir[ja,ia].imag,3);
  fgBranch2.Cells[11,i+1]:=RealToStr(Arus[ja,ia].real,3);
  fgBranch2.Cells[12,i+1]:=RealToStr(Arus[ja,ia].imag,3);
end;
edtSumGen2.Text:=toStringJ(gParamLF.SumGen,3);
edtSumLoad2.Text:=toStringJ(gParamLF.SumLoad,3);
edtSumLoss2.Text:=toStringJ(gParamLF.SumLoss,3);
edtIterasi2.Text:='3';
DecodeTime(selang,jam,menit,detik,mdetik);
edtTime2.Text:=IntToStr(jam)+':' + IntToStr(menit)+':'+
  IntToStr(detik)+':' + IntToStr(mdetik);
CostSesudah:=gParamLF.SumLoss.real;
edtSebelum.Text:=FormatFloat('#,##0.000',CostSebelum);
edtSesudah.Text:=FormatFloat('#,##0.000',CostSesudah);
for i:=0 to 3 do
begin
  fgOS.Cells[0,i+1]:=IntToStr(i+1);
  fgOS.Cells[1,i+1]:=IntToStr(sw[i].dari+1);
  fgOS.Cells[2,i+1]:=IntToStr(sw[i].ke+1);
end;
end;

procedure TfrmHasil.btnUseDefaultClick(Sender: TObject);
begin
  edtNLoop.Text:='4';
  fgOS.RowCount:=StrToInt(edtNLoop.Text)+1;
  edtmaxgen.Text:='50';
  edtpopsize.Text:='10';
  edtpcross.Text:='0.75';
  edtpmutasi.Text:='0.001';
  edtKa.Text:='10';
  edtPinality.Text:='1000';
end;

```

```

procedure TfrmHasil.FormCreate(Sender: TObject);
begin
  fgOS.Cells[0,0]:='No';
  fgOS.Cells[1,0]:='Asal';
  fgOS.Cells[2,0]:='Tujuan';
end;

procedure TfrmHasil.edtNLoopChange(Sender: TObject);
begin
  if edtNLoop.Text="" then
  begin
    fgOS.RowCount:=2;
  end
  else
  begin
    fgOS.RowCount:=StrToInt(edtNLoop.Text)+1;
  end;
end;

procedure TfrmHasil.btnSendParamClick(Sender: TObject);
var str:TStringList;
  strData,strDataAkhir:string;
  MaxGen,PopSize,Length,NLoop,NFault:integer;
  pcross,pmutasi,Ka:double;
begin
  statPC1:=false;
  statPC2:=false;
  statPC3:=false;
  statPC4:=false;
  str:=TStringList.Create;
  NLoop:=StrToInt(edtNLoop.Text);
  NFault:=isDataOnList(cmbFault.Text,cmbFault.Items);
  MaxGen:=StrToInt(edtMaxGen.Text);
  PopSize:=StrToInt(edtPopSize.Text);
  Length:=high(gBranch)+1;
  pcross:=StrToFloat(edtPCross.Text);
  pmutasi:=StrToFloat(edtPMutasi.Text);
  Ka:=StrToFloat(edtKa.Text);
  SetLength(Data,NLoop);
  SetLength(DataAkhir,NLoop);
  Data[0]:=0;
  Data[1]:=0;
  Data[2]:=0;
  Data[3]:=0;
  DataAkhir[0]:=5;
  DataAkhir[1]:=25;
  DataAkhir[2]:=25;
  DataAkhir[3]:=25;
  strData:=DecodeDataToStr(Data);
  strDataAkhir:=DecodeDataToStr(DataAkhir);
  str.Add('Param');
  str.Add(IntToStr(NLoop));
  str.Add(IntToStr(NFault));
  str.Add(IntToStr(MaxGen));
  str.Add(IntToStr(PopSize));
  str.Add(IntToStr(Length));

```

```
str.Add(FloatToStr(pcross));
str.Add(FloatToStr(pmutasi));
str.Add(FloatToStr(Ka));
str.Add(strData);
str.Add(strDataAkhir);
frmSetting.ClientSocket1.Socket.SendText(str.Text);
frmSetting.ClientSocket2.Socket.SendText(str.Text);
frmSetting.ClientSocket3.Socket.SendText(str.Text);
frmSetting.ClientSocket4.Socket.SendText(str.Text);
str.Free;
btnCreate.Enabled:=true;
end;

procedure TfrmHasil.cmbFaultClick(Sender: TObject);
begin
  btnSendParam.Enabled:=true;
end;

end.
```