

BAB IV

ANALISIS

A. Data

Pada tugas akhir ini sistem yang diuji adalah sistem kelistrikan industri PT. Semen Gresik Pabrik Tuban I dan II Tbk. *Single line diagram* dari sistem kelistrikan industri PT. Semen Gresik Pabrik Tuban I dan II Tbk. dapat dilihat pada lampiran.

Data bus, data trafo dan data jaringan yang digunakan adalah data dari sistem kelistrikan PT. Semen Gresik Pabrik Tuban I dan II Tbk. tahun 2005. *Single line diagram* terdiri dari 207 bus, 51 Trafo dan 206 jaringan dengan 1 swing bus yaitu dari PLN. Pada sistem terdiri dari 3 level tegangan yaitu Tegangan Tinggi (150 kV), Tegangan Menengah (20 kV dan 6.3 kV) dan Tegangan Rendah (0.4 kV).

Dan hal-hal yang ditentukan sebelum perhitungan aliran daya adalah sebagai berikut :

KVA Dasar	: 100.000 KVA
Toleransi Error	: 0.00001
Maksimum Iterasi	: 100
Toleransi Arus	: 10 %
Toleransi Tegangan	: 5 %

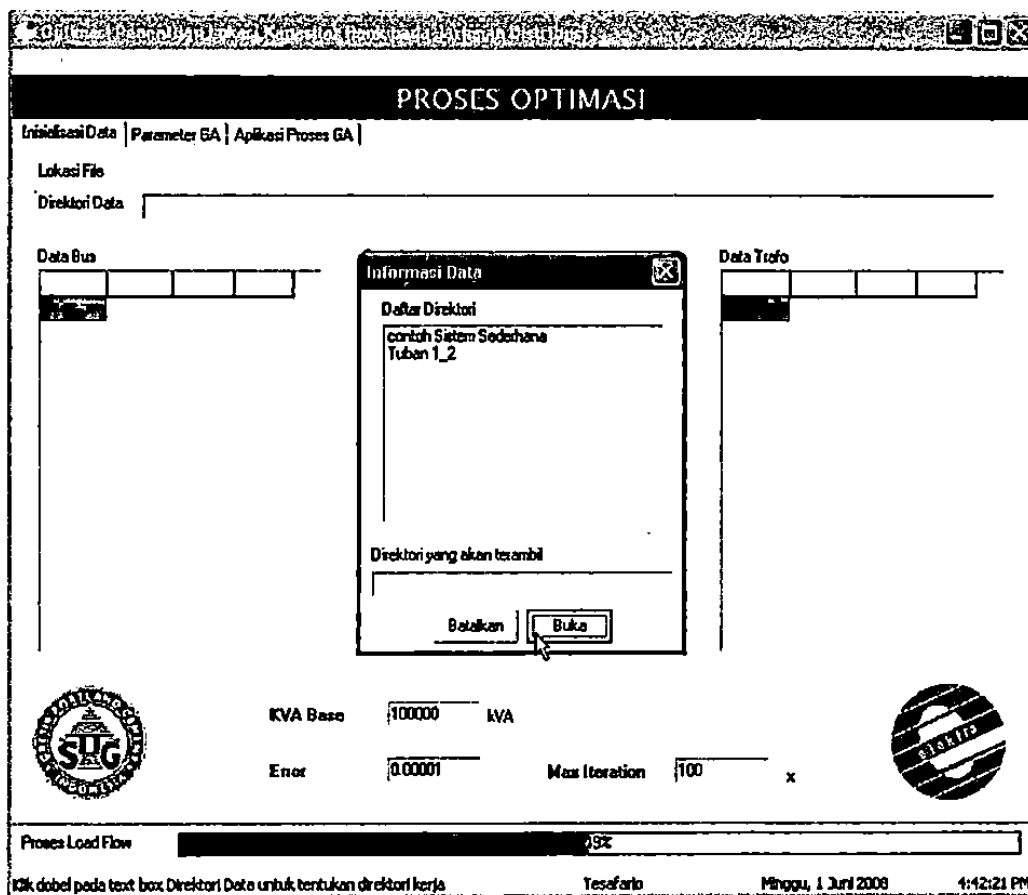
Data bus awal, data trafo dan data bus jaringan awal dapat dilihat pada lampiran. Dari data-data tersebut dilakukan studi aliran daya untuk

membandingkan antara sistem kelistrikan industri tanpa pemasangan kapasitor dengan sistem kelistrikan industri yang diinjeksi atau adanya pemasangan kapasitor untuk injeksi daya reaktif yang lokasi dan kapasitas dari kapasitor bank ditentukan dengan menggunakan metode optimasi *Genetic Algorithms (GA)*.

B. Analisa Program GA

1. Proses memasukkan data

Pada program GA input program berupa data file delimiter .xls, tampilan proses inputnya dapat dilihat dibawah ini.



Gambar 4.1 Tampilan form menu input data sistem kelistrikan

Dari tampilan di atas user harus memasukkan data input ke dalam program GA, yang sebelumnya user sudah memodifikasi data-data yang diperlukan sebagai input yang berupa file delimiter .xls dari data sistem kelistrikan menggunakan ETAP berupa file .lfr. Data-data yang diperlukan sebagai input yang berupa file delimiter yaitu meliputi daftar kapasitor.xls, data beban.xls, data bus.xls, data jaringan.xls, data trafo.xls, data umum.xls, dan data-data tersebut diambil dari data kelistrikan Pabrik Tuban I dan II PT. Semen Gresik Tbk. Dengan menggunakan program ETAP. Routines program untuk proses input data sebagai berikut:

```

procedure TfrmMain.PesanTempatGA (Target : String);
var
  DirKerja, TargetUtama : String;
Begin
  TargetUtama := Target;
  DirKerja := ExtractFilePath(Application.ExeName)+'Input\' +
TargetUtama;
  GandakanFile (frmMain.edtDir.Text + '\Data Beban.xls',
  DirKerja+'\Data Beban.xls');
  GandakanFile (frmMain.edtDir.Text + '\Data Bus GA.xls',
  DirKerja+'\Data Bus GA.xls');
  GandakanFile (frmMain.edtDir.Text + '\Data Jaringan GA.xls',
  DirKerja+'\Data Jaringan GA.xls');
  GandakanFile (frmMain.edtDir.Text + '\Data Swing Bus GA.xls',
  DirKerja+'\Data Swing Bus GA.xls');
  GandakanFile (frmMain.edtDir.Text + '\Data Trafo.xls',
  DirKerja+'\Data Trafo.xls');
  GandakanFile (frmMain.edtDir.Text + '\Data Umum.xls',
  DirKerja+'\Data Umum.xls');
  GandakanFile (frmMain.edtDir.Text + '\GA Parameter.txt',
  DirKerja+'\GA Parameter.txt');
  end;

```

2. Informasi parameter-parameter GA

Parameter parameter GA yang digunakan dapat dilihat pada gambar

Optimasi Penentuan Lokasi Kapasitor Dink pada Jaringan Distribusi

File Bantuan Keluar

PARAMETER GENETIK ALGORITMA

Instalisasi Data [Parameter GA] Aplikasi Proses GA

Data Bus		Daftar Kapasitor		Informasi Proses GA	
Bus		No.	Kvar		
705	↑	1	0	Jumlah Kromosom	35
716		2	100	Jumlah Seleksi	7
725		3	200	Max Generasi (Iterasi)	7
728		4	300		
732		5	400		
726		6	500	Toleransi Kesempatan	
717		7	600	Mutasi	0.1
713		8	700	Perukaran Gen	0.9
714		9	800	Max Perukaran Gen	0.01
723	↓	10	900		
		11	1000		
		12	1100	Jumlah Gen per Kromosom	35
		13	1200		
		14	1300		
		15	1400		
		16	1500		

Pembobot Rugi P (0..100)	30
Pembobot Rugi Q (0..100)	30
Penyetaraan PF (0..100)	30
Pembobot Harga (0..100)	30

Proses Load Flow ON

Data telah siap di proses ... III

Tesafarib Minggu, 1 Juni 2008 4:43:36 PM

Gambar 4.2 Parameter *Genetic Algorithm*

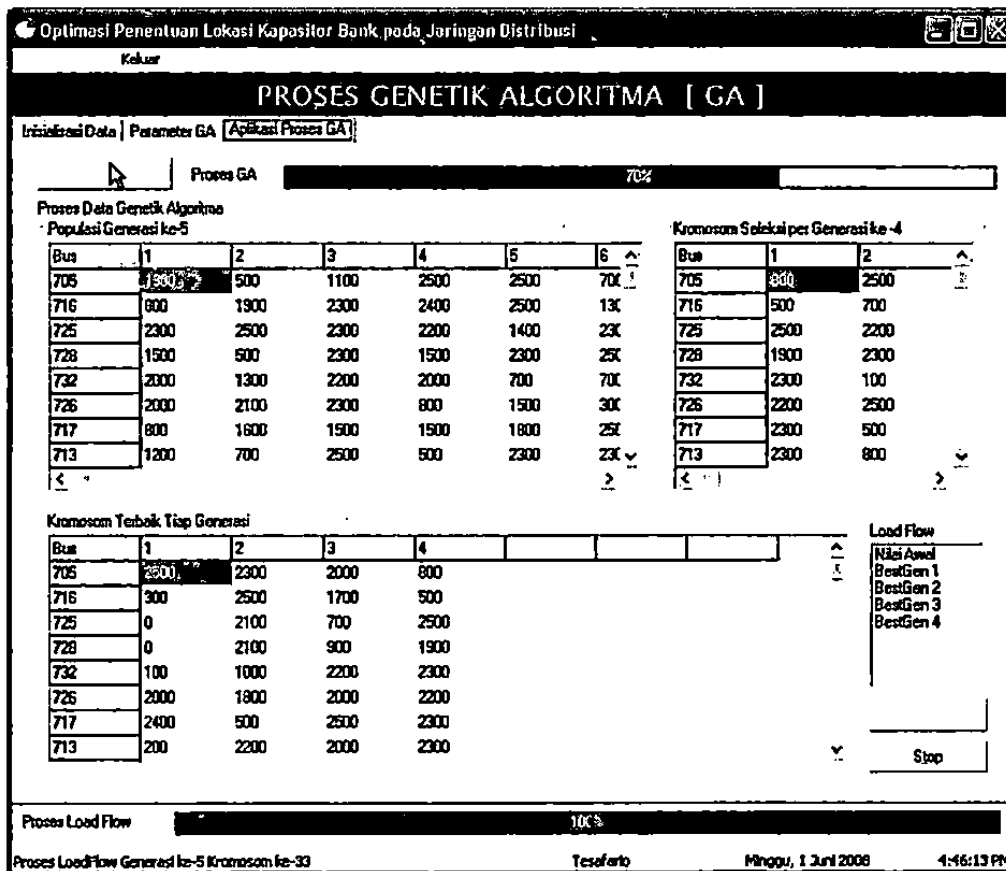
Parameter GA merupakan data yang mempengaruhi proses GA. Parameter yang ditentukan meliputi jumlah kromosom, jumlah seleksi, max generasi (iterasi), toleransi kesempatan, jumlah gen per kromosom, pembobot rugi P, pembobot rugi Q, penyetaraan PF, pembobot harga.

Pembobot rugi-rugi P, pembobot rugi-rugi Q, penyetara PF, dan pembobot harga merupakan faktor yang mempengaruhi dalam menentukan nilai *fitness*.

3. Proses GA

Setelah menentukan jumlah kromosom, jumlah seleksi, maksimal generasi (iterasi), toleransi kesempatan, jumlah gen per kromosom, dan

pembobot-pembobot-nya. Proses selanjutnya yaitu Aplikasi Proses GA. Pada tahap ini dilakukan proses seleksi, kawin silang, mutasi, dan regenerasi. Tampilan dari proses dapat dilihat pada gambar 4.5 dibawah ini.



Gambar 4.3 Proses perhitungan menggunakan GA

Tampilan di atas merupakan proses perhitungan menggunakan GA, mulai dari populasi generasi ke-n, kromosom seleksi per generasi ke-n, kromosom terbaik tiap generasi. Proses yang terjadi yaitu:

a. Inisialisasi kromosom awal

i. Input data – data kapasitor (proses inisialisasi)

```
for jProses := 1 to frmMain.GrdBus.RowCount - 1 do
```

```
  Begin
```

```

if HasilRandom = 0 then
  frmMain.grdPopulasi.Cells [iProses, jProses] := '0'
else
  begin
    frmMain.grdPopulasi.Cells [iProses, jProses] :=
frmMain.grdDaftarKapasitor.Cells [1, HasilRandom];
    Val (frmMain.grdDaftarKapasitor.Cells [1, HasilRandom],
CalonHarga, Errornya);
    JumHarga := JumHarga + CalonHarga;
  end;
  // apply langsung ke data pusat
  for nDum := 1 to Form1.StringGrid2.RowCount - 1 do
    begin
      T1 := Form1.StringGrid2.Cells [0, nDum];
      T2 := frmMain.grdPopulasi.Cells [0, jProses];
      if T1 = T2 then
        //Proses pemasukan kap ke sistem --> nilai jadi minus
        begin
          Form1.StringGrid2.Cells [2, nDum] := '-' +
frmMain.grdPopulasi.Cells [iProses, jProses];
          frmMain.StringGrid1.Cells [2, nDum] := '-' +
frmMain.grdPopulasi.Cells [iProses, jProses];
        end;
      end;
    // end apply langsung ke data pusat
  End;

```

- ii. Random kromosom sejumlah seleksi yang ditentukan, berupa nilai variabel sesuai dengan parameter.

```

//PEMBUATAN KROMOSOM BARU DALAM 1 POP

JumProses := StrToInt (frmMain.edtNPop.Text);
for iProses := 1 to JumProses do
  if frmMain.Selesai then Break
  else
    Begin
      JumHarga := 0;
      frmMain.grdPopulasi.ColCount :=
frmMain.grdPopulasi.ColCount + 1;
      If iProses = 1 Then
        frmMain.grdPopulasi.FixedCols := 1;

      // ---- random dari parent

      If iProses = 1 Then
        frmMain.grdPopulasi.FixedCols := 1;
        for jProses := 1 to frmMain.GrdBus.RowCount - 1 do
          Begin
            frmMain.grdPopulasi.Cells [iProses, jProses] :=
            frmMain.grdNBest.Cells [Random
(frmMain.grdNBest.ColCount - 2) + 1,
            Random (frmMain.GrdBus.RowCount - 2) + 1];

```

```

        JumHarga := JumHarga + CalonHarga;
    end; //for jProses := 1 to frmMain.grdPopulasi.RowCount - 7 do
        // ----- end of random dari parent

```

b. Generasi ke-1

- i. Terbentuk populasi pada tiap generasi, sejumlah kromosom yang telah ditentukan berdasarkan parameter GA, dari kromosom hasil inialisasi awal dengan berdasarkan toleransi kesempatan :

- Mutasi
- Pertukaran gen
- Maks pertukaran gen

```

// ----- kesempatan mutasi
    Val (frmMain.edtMutChance.Text, CalonHarga, Errornya);
    if Random <= CalonHarga then
        for jProses := 1 to Random (frmMain.GrdBus.RowCount - 2)
+ 1 do
            begin
                RCol := iProses;
                RRow := Random (frmMain.GrdBus.RowCount - 2) + 1;
                Val (frmMain.grdPopulasi.Cells [RCol, RRow], CalonHarga, Errornya);
                JumHarga := JumHarga - CalonHarga;
                HasilRandom := Random (frmMain.grdDaftarKapasitor.RowCount);
                if HasilRandom = 0 then
                    frmMain.grdPopulasi.Cells [RCol, RRow] := '0'
                else
                    frmMain.grdPopulasi.Cells [RCol, RRow] :=
frmMain.grdDaftarKapasitor.Cells [1, HasilRandom];
                    Val (frmMain.grdDaftarKapasitor.Cells [1,
HasilRandom], CalonHarga, Errornya);
                    JumHarga := JumHarga + CalonHarga;
                end; //for jProses := 1 to Random
(frmMain.grdPopulasi.RowCount - 8) + 1 do
                    // ----- end of kesempatan mutasi

                // ----- kesempatan pertukaran
                Val (frmMain.edtCreepChance.Text, CalonHarga, Errornya);
                if Random <= CalonHarga then
                    begin
                        Val (frmMain.edtCreepAmmount.Text, CalonHarga, Errornya);
                        for jProses := 1 to Round(CalonHarga *
frmMain.GrdBus.RowCount) do
                            begin
                                RCol := Random (frmMain.GrdBus.RowCount - 2) + 1;
                                RRow := Random (frmMain.GrdBus.RowCount - 2) + 1;
                                CalonHargaS := frmMain.grdPopulasi.Cells [iProses, RCol];
                                frmMain.grdPopulasi.Cells [iProses, RCol] :=
frmMain.grdPopulasi.Cells [iProses, RRow];
                                frmMain.grdPopulasi.Cells [iProses, RRow] :=

```

```

end; //for jProses := 1 to Round(CalonHarga) do
end; //if Random <= CalonHarga then
// ----- end of kesempatan pertukaran

```

- ii. Seleksi kromosom terbaik berdasarkan fitness terbaik.
- iii. Menyimpan satu kromosom terbaik dalam tabel Kromosom Terbaik Tiap Generasi.

```

// ----- mengurutkan populasi
//----PROSES SELEKSI AWAL --- kromosom tiap populasi ke data best
kromosom
    frmMain.grdNBest.ColCount := StrToInt (frmMain.edtNBest.Text) + 1;
    frmMain.grdNBest.RowCount := frmMain.grdPopulasi.RowCount;
    frmMain.grpBestKromosom.Visible := True;
    for iProses := 0 to frmMain.grdNBest.ColCount - 1 do
        for jProses := 0 to frmMain.grdPopulasi.RowCount - 1 do
            frmMain.grdNBest.Cells [iProses, jProses] :=
frmMain.grdPopulasi.Cells [iProses, jProses];
            frmMain.grdBestKromosom.ColCount := StrToInt
(frmMain.edtNGen.Text) + 1;
            frmMain.grdBesKromosom.RowCount := frmMain.grdPopulasi.RowCount;
            frmMain.grpBesGenerasi.Visible := True;
// ---- END OF SELEKSI AWAL kromosom tiap populasi ke data best kromosom
end; //itProses := 2 to jGenerasi do
    for iProses := 0 to frmMain.grdNBest.ColCount - 1 do
        for jProses := 0 to frmMain.grdPopulasi.RowCount - 1 do
            frmMain.grdNBest.Cells [iProses, jProses] :=
frmMain.grdPopulasi.Cells [iProses, jProses];

```

c. Generasi ke-2

- i. Terbentuk 35 populasi pada tiap generasi, sejumlah kromosom yang ditentukan (35 kromosom) dari kromosom hasil seleksi kromosom terbaik dari generasi sebelumnya dengan berdasarkan toleransi kesempatan :

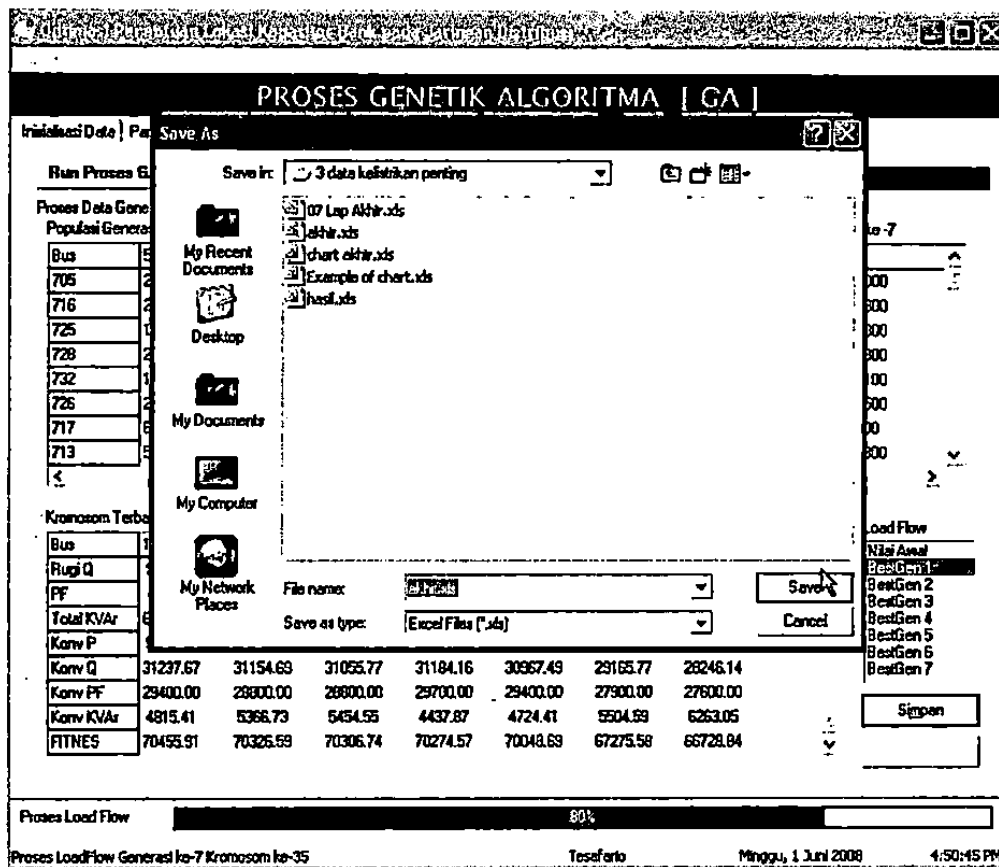
- Mutasi
- Pertukaran gen
- Maks pertukaran gen

- ii. Seleksi kromosom terbaik berdasarkan fitness terbaik sejumlah 7

- iii. Menyimpan satu kromosom terbaik dalam tabel Kromosom Terbaik Tiap Generasi.
- d. Generasi ke-3 dan seterusnya sampai max generasi yang ditentukan (7 generasi) sama dengan generasi ke-2.

Rutin proses program GA untuk generasi ke-2 dan seterusnya dapat dilihat pada lampiran program GA.

4. Proses menyimpan data hasil perhitungan GA



Gambar 4.4 Proses penyimpanan file berbentuk .xls

Pada *tab page* diatas proses penyimpanan file berformat .xls dan disimpan pada direktori pada komputer sesuai keinginan user. Rutin program untuk proses penyimpanan data bisa dilihat sebagai berikut

```

n.memPenolong.Lines.Clear;
  for iii := 0 to Form1.StringGrid1.RowCount - 1 do
  begin
    sss := Form1.StringGrid1.Cells [0, iii];
    for jjj := 1 to Form1.StringGrid1.ColCount - 1 do
      sss := sss + #9 + Form1.StringGrid1.Cells [jjj, iii];
      frmMain.memPenolong.Lines.Add(sss);
    end;
    frmMain.memPenolong.Lines.SaveToFile(frmMain.edtDir.Text+'\Data
Jaringan.xls');
    frmMain.memPenolong.Lines.Clear;
    for iii := 0 to Form1.StringGrid2.RowCount - 1 do
    begin
      sss := Form1.StringGrid2.Cells [0, iii];
      for jjj := 1 to Form1.StringGrid2.ColCount - 1 do
        sss := sss + #9 + Form1.StringGrid2.Cells [jjj, iii];
        frmMain.memPenolong.Lines.Add(sss);
      end;
      frmMain.memPenolong.Lines.SaveToFile(frmMain.edtDir.Text+'\Data
Bus.xls');

      frmMain.ListView1.Clear;
      frmMain.ListView2.Clear;

      frmMain.isInisial := True;
      frmMain.isKonfergen := True;
      Form5.Edit1.Text := frmMain.FileSimpan;
      Form5.Button1.Caption := 'Buka';
      Form5.Button1.Click;
      Form1.Hitung1.Click;

      GandakanFile(frmMain.edtDir.Text + '\Data Bus Temp.xls'
, frmMain.edtDir.Text + '\Data Bus.xls');
      GandakanFile(frmMain.edtDir.Text + '\Data Jaringan Temp.xls'
, frmMain.edtDir.Text + '\Data Jaringan.xls');

      If not HasError or frmMain.isKonfergen then
      begin
        frmMain.isSavePusat := True;
        frmMain.FileNya := frmMain.SaveDialog1.FileName;
        Form1.perProfill.Click ();
        frmMain.isSavePusat := False;
        Application.MessageBox(PChar ('Data telah sukses disimpan ... !'
+ Chr (13) + 'Lokasi File: '
+ frmMain.SaveDialog1.FileName), 'Simpan Data',
MB_ICONINFORMATION)
      end
      else //If not HasError or frmMain.isKonfergen then
      begin
        Application.MessageBox('Proses tidak bisa disimpan,
karena data tidak konfergen'
, 'Proses Fatal', MB_ICONERROR + MB_OK)
      end; //If not HasError or frmMain.isKonfergen then

      frmMain.isProsesGA := False;
      frmMain.isInisial := False;

// -----
// ----- end of simpan load flow dengan nilai awal

```

C. Analisa Data

Berdasarkan perhitungan aliran daya Fast Decoupled pada sistem jaringan distribusi radial PT. Semen Gresik Tbk. Pabrik Tuban I dan II yang telah dilakukan maka dapat dilihat hasil dari masing-masing perhitungan pada tabel 4.1 dibawah ini :

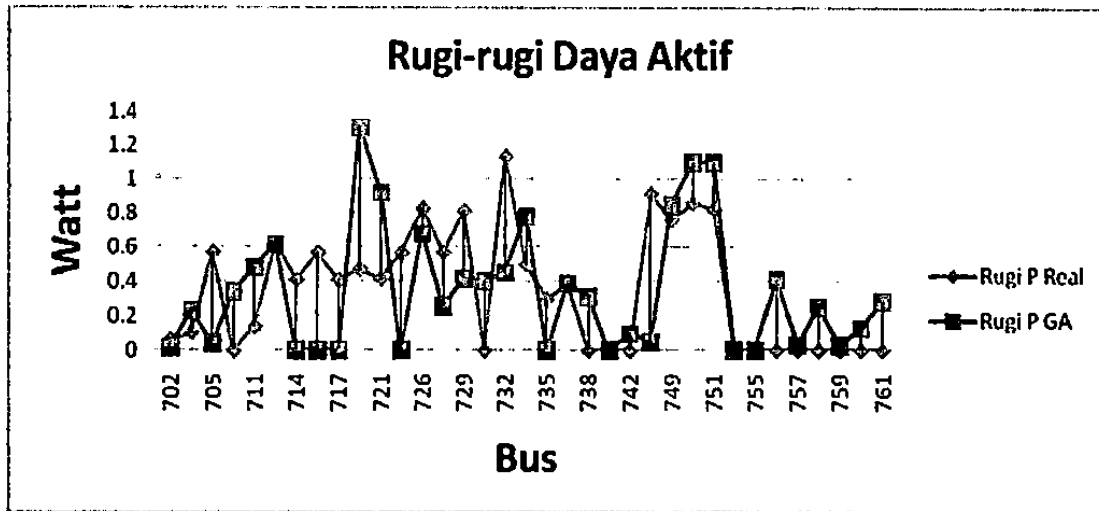
Tabel 4.1. Data perbandingan hasil perhitungan aliran daya sistem distribusi PT.

Semen Gresik Tbk. Pabrik Tuban I dan II.

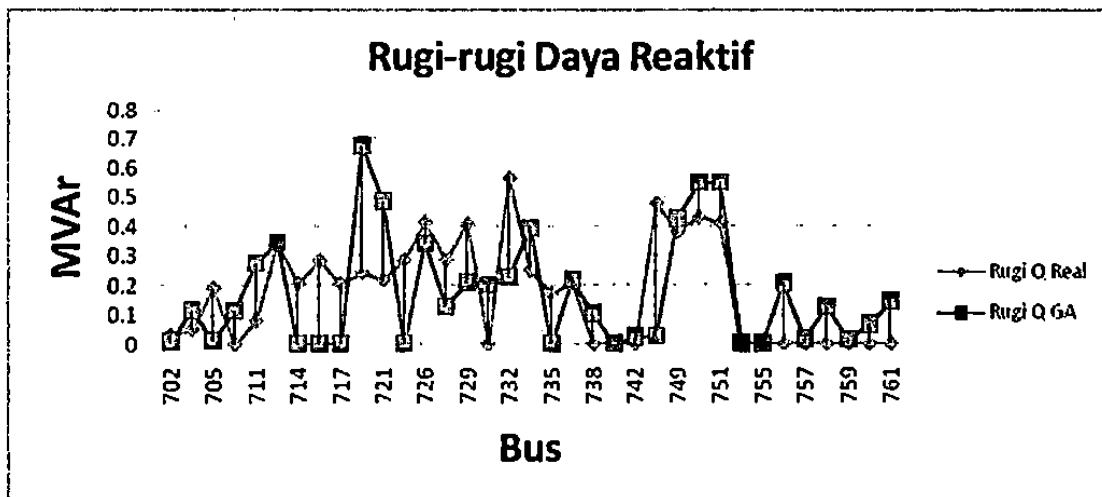
	Sebelum Optimasi		Optimasi GA		Optimasi <i>Reality</i>	
	P (KW)	Q (Kvar)	P (KW)	Q (Kvar)	P (KW)	Q (Kvar)
Total Rugi - Rugi (KVA)	1,029.719	19,194.742	599.66	9,603.79	606.327	9,633.787
Power Factor	0.658		0.98		0.958	
Total KVAr	0.000		62,300		50,166	

Berdasarkan perhitungan aliran daya Fast Decoupled pada sistem jaringan distribusi radial PT. Semen Gresik Pabrik Tuban I dan II Tbk., dengan adanya penambahan kapasitor bank, pada sistem jaringan distribusi radial dapat mengurangi rugi-rugi daya yang semula sebesar $1,029.72 + j 19,194.74$ kVA menjadi $606.327 + j 9,633.78$ KVA, dan dengan menggunakan perhitungan GA dalam menentukan letak dan ukuran bank kapasitor yang optimal dapat memperbaiki rugi-rugi daya yang semula $606.327 + j 9,633.787$ KVA menjadi $599.66 + 9,603.79$ KVA, serta dapat memperbaiki faktor daya yang semula $\cos \theta = 0.658$ menjadi $\cos \theta = 0.958$ dengan menggunakan perhitungan GA menjadi $\cos \theta = 0.98$

Adapun perbandingan hasil sebelum dan sesudah penggunaan GA dapat dilihat pada grafik perbandingan antara daya aktif dan daya reaktif dibawah ini.



Gambar 4.5 Grafik Perbandingan Rugi-rugi Daya Aktif sebelum dan sesudah menggunakan GA



Gambar 4.6 Grafik Perbandingan Rugi-rugi Daya Reaktif sebelum dan sesudah menggunakan GA

Gambar 4.5 dan 4.6 grafik diatas berdasarkan data pada tabel 4.2 dibawah

ini dan lebih lengkapnya terdapat pada lampiran

Tabel 4.2 Data perbandingan daya aktif dan reaktif sebelum dan sesudah menggunakan GA

Bus	Rugi P Real	Rugi P GA	Rugi Q Real	Rugi Q GA
702	0.062552	0.016542	0.031276	0.008271
703	0.104962	0.231508	0.052481	0.115754
705	0.568243	0.040947	0.189414	0.013649
708	0	0.339674	0	0.113225
711	0.138533	0.479256	0.079162	0.273861
713	0.619007	0.618905	0.343893	0.343836
714	0.410051	0	0.205026	0
716	0.567983	0	0.283991	0
717	0.408772	0	0.204386	0
719	0.472232	1.304936	0.244861	0.676633
721	0.415926	0.921091	0.217866	0.482476
725	0.569014	0	0.284507	0
726	0.827281	0.679686	0.413641	0.339843
728	0.568955	0.256945	0.284478	0.128473
729	0.813031	0.416209	0.406516	0.208104
731	0	0.399598	0	0.199799
732	1.129618	0.451541	0.564809	0.225771
733	0.502116	0.781695	0.251058	0.390848
735	0.297505	0	0.170003	0
737	0.382707	0.389677	0.212615	0.216487
738	0	0.310163	0	0.103388
740	0	0	0	0
742	0	0.09466	0	0.023665
745	0.911953	0.051533	0.47769	0.026993
749	0.760755	0.850426	0.380378	0.425213
750	0.859158	1.098116	0.429579	0.549058
751	0.823556	1.098116	0.411778	0.549058
752	0	0	0	0
755	0	0	0	0
756	0	0.410726	0	0.205363
757	0	0.031454	0	0.015727
758	0	0.249899	0	0.12495
759	0	0.028665	0	0.014333
760	0	0.128922	0	0.064461
761	0	0.283817	0	0.141909