

# SKRIPSI

**ESTIMASI EFEKTIF EFFISIENSI MOTOR INDUKSI 3  
PHASA BERDASAR DATA PENGUKURAN DI LAPANGAN  
MENGUNAKAN METODE EVOLUTIONARY  
PROGRAMMING**



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

*disusun Oleh :*

**IWAN KUNCAHYONO**  
Nim. 0112058

**JULI 2006**

# LEMBAR PENGESAHAN SKRIPSI

**ESTIMASI EFEKTIF EFFISIENSI MOTOR INDUKSI 3 PHASA  
BERDASAR DATA PENGUKURAN DI LAPANGAN MENGGUNAKAN  
METODE EVOLUTIONARY PROGRAMMING**

*Di susun guna melengkapi & memenuhi syarat-syarat dalam  
mencapai gelar sarjana teknik S-1*

di susun oleh :

**IWAN KUNCAHYONO  
Nim . 0112058**



Mengetahui  
Ketua Jurusan Teknik elektro itn  
Malang

Ir , F Yudi Limprapto , MT  
NIP . Y 103 9500 274

Menyetujui  
Dosen Pembimbing

Ir , Widodo Pudji Muljanto , MT  
NIP , Y 102 870 171

# ABSTRAKSI

## ESTIMASI EFEKTIF EFFISIENSI MOTOR INDUKSI 3 PHASA BERDASAR DATA PENGUKURAN DI LAPANGAN MENGGUNAKAN METODE EVOLUTIONARY PROGRAMMING

**Kata Kunci :** Motor Induksi Tiga Phasa, Efisiensi, Pengujian, Evolutionary Programming

Penggunaan motor induksi tiga phasa yang sangat luas untuk alat penggerak dalam kepentingan industri. Sekitar delapan puluh persen untuk motor listrik dan sekitar tiga puluh lima persen total konsumsi listrik dari motor induksi. Diperlukan pengujian-pengujian untuk mengetahui kelayakan atau efisiensi suatu motor induksi pada industri-industri.

Dengan semakin berkembangnya metode-metode dalam penentuan efisiensi saat ini, dilihat dari segi kerumitan prosedur, performa, dan kecocokan pada kondisi pabrik, maka industri-industri yang menggunakan aplikasi motor induksi berupaya untuk mencari metode yang dianggap mampu mengatasi masalah penentuan efisiensi dengan tingkat akurasi yang mendekati nilai sebenarnya.

Beberapa diantara metode tersebut membutuhkan pemutusan beban atau pengujian tanpa beban (*No-Load Test*) serta pengujian rotor tertahan (*Locked Rotor Test*) dari motor induksi, sehingga tidak memungkinkan untuk kondisi pabrik yang biasanya mengaplikasikan motor induksi dengan kapasitas daya yang besar.

Dengan metode yang telah dikembangkan saat ini, yaitu dengan metode *Evolutionary Programming* yang didasarkan pada mekanisme alam, mampu mendapatkan hasil yang mendekati nilai sebenarnya tanpa harus melakukan pengujian tanpa beban maupun rotor tertahan. Metode ini membutuhkan pengujian berbeban (*Load Test*), pengujian arus searah (*DC Test*) serta data papan-nama (*nameplate*) dari motor untuk mendapatkan parameter masukan program *Evolutionary Programming*.

## **KATA PENGANTAR**

Dengan Rahmat Tuhan dan mengucapkan syukur kehadirat-Nya atas karunia yang telah dilimpahkan kepada penulis sehingga dapat menyelesaikan skripsi yang berjudul “ **Estimasi efektif efisiensi motors induksi 3 phasa berdasar data pengukuran di lapangan menggunakan metode Evolutionary programming** “.

Skripsi ini bertujuan untuk memenuhi kurikulum akademik yang harus ditempuh oleh setiap mahasiswa ITN Malang dalam menempuh jenjang pendidikan strata satu pada Jurusan Teknik Elektro Konsentrasi Energi Listrik. Diharapkan dengan penulisan skripsi ini, dapat memperdalam dan sekaligus melatih mahasiswa agar dapat menerapkan ilmu yang didapat pada waktu kuliah dengan kondisi sesungguhnya.

Penulis menyadari bahwa skripsi ini jauh dari sempurna, karena itu saran dan kritik yang bersifat membangun penulis sangat harapkan. Atas tersusunnya skripsi ini tidak lupa penulis mengucapkan terima kasih kepada :

1. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. I Wayan Sudjana, MT, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
3. Ir. F. Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro, Institut Teknologi Nasional Malang.
4. Ir. Widodo Pudji Muljanto, MT, selaku Dosen Pembimbing yang memberikan ilmu, dorongan semangat dan bantuan.
5. Rekan-rekan Elektro angkatan 2001 khususnya kelas 2 atas segala kebersamaannya selama ini.

Akhirnya penyusun mohon maaf kepada semua pihak apabila ada kesalahan dan kekurangan selama penyusunan skripsi ini .

**Penyusun**



# DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>LEMBAR PERSETUJUAN .....</b>	<b>ii</b>
<b>ABSTRAKSI .....</b>	<b>iii</b>
<b>KATA PENGANTAR.....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>vi</b>
<b>DAFTAR GAMBAR .....</b>	<b>x</b>
<b>DAFTAR GRAFIK.....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan .....	2
1.4. Batasan Masalah .....	2
1.5. Metodologi Penelitian.....	3
1.6. Sistematika Penulisan .....	4
1.7. Relevansi .....	5
<b>BAB II KONSEP DASAR MOTOR INDUKSI TIGA PHASA .....</b>	<b>6</b>
2.1. Teori Dasar Motor Induksi Tiga Phasa .....	6
2.2. Konstruksi Motor Induksi.....	7
2.2.1. Stator .....	7
2.2.2. Rotor.....	8
2.3. Medan Magnet Putar.....	8
2.4. Prinsip Kerja Motor Induksi Tiga Phasa.....	10
2.4.1. Slip Dan Frekuensi Arus Rotor .....	12
2.5. Rangkaian Ekuivalen Motor Induksi .....	13
2.5.1. Rangkaian Ekuivalen .....	13
2.5.2. Rangkaian Ekuivalen Stator.....	13
2.5.3. Rangkaian Ekuivalen Rotor .....	14
2.5.4. Rangkaian Ekuivalen Motor Induksi.....	16
2.5.5. Rangkaian Ekuivalen Pada Penentuan Efisiensi Motor induksi Tiga Phasa .....	18

2.6. Pengujian Motor Induksi Tiga Phasa .....	23
2.6.1. Pengujian Arus Searah ( <i>DC Test</i> ) .....	23
2.6.2. Pengujian Tanpa Beban ( <i>No-Load Test</i> ) .....	24
2.6.3. Pengujian Rotor Tertahan ( <i>Locked Rotor Test</i> ).....	26
2.6.4. Pengujian Berbeban ( <i>Load Test</i> ).....	28

### **BAB III TEORI DASAR ALGORITMA GENETIKA DAN APLIKASI**

<b>PADA PENENTUAN EFISIENSI MOTOR INDUKSI</b> .....	<b>29</b>
3.1. Evolutionary Programming .....	29
3.1.1. Istilah-Istilah Evolutionary Programming .....	31
3.1.2. Proses Evolutionary Programming .....	34
3.1.3. Elitism.....	38
3.2. Kompetisi .....	38
3.3. Mutasi .....	40
3.4. Adaptasi Algoritma Evolusi Ke Masalah Estimasi	
Efisiensi Motor Induksi Tiga Phasa.....	42
3.4.1. Pengkodean (Representasi Solusi) .....	43
3.4.2. Populasi Awal.....	46
3.4.3. Reproduksi, Mutasi & Kompetisi .....	47
3.5. Objective Function .....	48
3.5.1. Objective Function untuk Permasalahan Estimasi	
Efisiensi Motor Induksi Tiga Phasa.....	48
3.6. Algoritma Program .....	50
3.6.1. Algoritma Program Pemecahan Masalah	
Secara Umum.....	50
3.6.2. Algoritma Program Pemecahan Masalah Penentuan	
Efisiensi Motor Induksi Tiga Phasa Dengan	
Metode Evolutionary Programming .....	51
3.6.3. Algoritma Program Fitness.....	52

<b>BAB IV PENGUJIAN DAN ANALISIS.....</b>	<b>53</b>
4.1. Pengujian Parameter Motor Induksi Tiga Fasa .....	53
4.1.1. Alat-Alat Yang Digunakan .....	53
4.1.2. Pengujian Arus Searah .....	54
4.1.2.1. Data Hasil Pengujian Arus Searah .....	54
4.1.2.2. Analisa Data Hasil Pengujian Arus Searah .....	55
4.1.3. Pengujian Berbeban .....	55
4.1.3.1. Data Hasil Pengujian Berbeban .....	56
4.1.3.2. Analisa Data Hasil Pengujian Berbeban .....	56
4.2. Data Masukan Untuk Program Evolutionary Programming .....	58
4.3. Program Komputer Metode Evolutionary Programming .....	58
4.4. Flowchart Algoritma Program .....	59
4.4.1. Flowchart Pemecahan Masalah Secara Umum .....	59
4.4.2. Flowchart Program Penentuan Efisiensi Motor Induksi Tiga Fasa Dengan Metode Evolutionary Programming .....	60
4.4.3. Flowchart Program Fitness .....	61
4.5. Tampilan Program .....	62
4.5.1. Satu Poin Pembebanan (OPT).....	62
4.5.2. Dua Poin Pembebanan (TPT) .....	69
4.6. Hasil Perhitungan Dengan Metode Evolutionary Programming ..	78
 <b>BAB V KESIMPULAN DAN SARAN .....</b>	 <b>81</b>
5.1. Kesimpulan .....	81
5.2. Saran.....	82

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

# DAFTAR GAMBAR

<b>Gambar :</b>	<b>Halaman</b>
2-1	Konstruksi Motor Induksi ..... 7
2-2	Stator Tiga Phasa Motor Induksi ..... 7
2-3	Rotor Motor Induksi Tiga Phasa..... 8
2-4	Medan Medan Putar Pada Motor Induksi ..... 9
2-5	Rangkaian Ekuivalen Stator..... 13
2-6	Rangkaian Ekuivalen Rotor ..... 15
2-7	Rangkaian Ekuivalen Motor Induksi ..... 16
2-8	Penyederhanaan Rangkaian Ekuivalen Motor Induksi..... 17
2-9	Rangkaian Ekuivalen Pada Estimasi Efisiensi Motor Induksi..... 18
2-10	Pengujian Arus Searah ( <i>DC Test</i> ) ..... 23
2-11	Pengujian Tanpa Beban ( <i>No-Load Test</i> ) ..... 24
2-12	Pengujian Rotor Tertahan ( <i>Blocked Rotor Test</i> ) ..... 26
2-13	Pengujian Berbeban ( <i>Load Test</i> )..... 28
3-1	Struktur Evolutionary Programming ..... 34
3-2	Roulette Wheel ..... 37
3-3	Ilustrasi Operator Kompetisi ..... 39
3-4	Ilustrasi Operator Kompetisi Dengan <i>Two Point Crossover</i> ..... 39
3-5	Ilustrasi Operator Kompetisi Dengan <i>Uniform Crossover</i> ..... 40
3-6	Ilustrasi Operator Kompetisi Untuk String Real ..... 41
3-7	Evolutionary Programming ..... 43
3-8	Pengkodean Untuk Parameter $R_2$ ..... 43
3-9	Pengkodean Untuk Parameter $R_m$ ..... 44
3-10	Pengkodean Untuk Parameter $X_1$ ..... 44
3-11	Pengkodean Untuk Parameter $X_m$ ..... 44
4-1	Pengujian Arus Searah..... 54
4-2	Flowchart Pemecahan Masalah Secara Umum ..... 59
4-3	Flowchart Program Evolutionary Programming ..... 60
4-4	Flowchart Program Fitness ..... 61
4-5	Tampilan Depan Program ..... 62
4-6	Tampilan Inputan Data Pada Beban 100% ..... 63

4-7	Tampilan Data Yang Sudah Tersimpan Pada Beban 100% .....	63
4-8	Tampilan Hasil Evolutionary Programming Pada Beban 100% .....	64
4-9	Tampilan Grafik Fitness - Generasi Pada Beban 100% .....	64
4-10	Tampilan Inputan Evolutionary Programming Pada Beban 75%.....	65
4-11	Tampilan Hasil Evolutionary Programming Pada Beban 75% .....	65
4-12	Tampilan Grafik Fitness - Generasi Pada Beban 75%.....	66
4-13	Tampilan Inputan Evolutionary Programming Pada Beban 50%.....	66
4-14	Tampilan Hasil Evolutionary Programming Pada Beban 50% .....	67
4-15	Tampilan Grafik Fitness - Generasi Pada Beban 50% .....	67
4-16	Tampilan Inputan Evolutionary Programming Pada Beban 25%.....	68
4-17	Tampilan Hasil Evolutionary Programming Pada Beban 25% .....	68
4-18	Tampilan Grafik Fitness - Generasi Pada Beban 25% .....	69
4-19	Inputan Evolutionary Programming Pada Beban 100% dan 75%.....	69
4-20	Hasil Evolutionary Programming Pada Beban 100% dan 75%.....	70
4-21	Tampilan Grafik Fitness - Generasi Pada 100% dan 75% .....	70
4-22	Inputan Evolutionary Programming Pada Beban 100% dan 50%.....	71
4-23	Hasil Evolutionary Programming Pada Beban 100% dan 50%.....	71
4-24	Tampilan Grafik Fitness - Generasi Beban 100% dan 50%.....	72
4-25	Inputan Evolutionary Programming Pada Beban 100% dan 25%.....	72
4-26	Hasil Evolutionary Programming Pada Beban 100% dan 25%.....	73
4-27	Tampilan Grafik Fitness - Generasi Beban 100% dan 25%.....	73
4-28	Inputan Evolutionary Programming Pada Beban 75% dan 50%.....	74
4-29	Hasil Evolutionary Programming Pada Beban 75% dan 50%.....	74
4-30	Tampilan Grafik Fitness -Generasi Pada Beban 75% dan 50% .....	75
4-31	Inputan Evolutionary Programming Pada Beban 75% dan 25%.....	75
4-32	Hasil Evolutionary Programming Pada Beban 75% dan 25%.....	76
4-33	Tampilan Grafik Fitness -Generasi Pada Beban 75% dan 25% .....	76
4-34	Inputan Evolutionary Programming Pada Beban 50% dan 25%.....	77
4-35	Hasil Evolutionary Programming Pada Beban 50% dan 25%.....	77
4-36	Tampilan Grafik Fitness -Generasi Pada Beban 50% dan 25% .....	78

## DAFTAR GRAFIK

<b>Grafik :</b>	<b>Halaman</b>
4-1 Efisiensi EP (OPT) – Efisiensi Uji.....	79
4-2 Efisiensi rata-rata EP (TPT) – Efisiensi Uji.....	80

## DAFTAR TABEL

<b>Grafik :</b>	<b>Halaman</b>
3-1 Istilah Yang Dipakai Dalam Evolutionary Programming.....	31
4-1 Data Pengujian Arus Searah .....	54
4-2 Data Pengujian Berbeban.....	56
4-3 Hasil Perhitungan Pengujian Berbeban.....	57
4-4 Inputan Data Untuk Program Evolutionary Programming .....	58
4-5 Perbandingan Hasil Evolutionary Programming (OPT) Dengan Hasil Pengujian.....	78
4-6 Perbandingan Hasil Evolutionary Programming (TPT) Dengan Hasil Pengujian.....	80

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang.

Motor induksi secara luas digunakan untuk berbagai kepentingan khususnya di bidang industri. Sekitar delapan puluh persen untuk motor listrik dan sekitar tiga puluh lima persen total konsumsi listrik dari motor induksi. Diperlukan pengujian-pengujian untuk mengetahui nilai efisiensi suatu motor induksi yang telah terhubung pada peralatan beban.

Metode penentuan efisiensi sangat bervariasi dipandang dari segi kerumitan, performa keseluruhan dan kecocokan pada kondisi pabrik. Beberapa diantaranya membutuhkan pengujian-pengujian (*No-Load Test*, *Blocked Rotor Test*) yang sangat sulit diterapkan pada industri-industri besar. Di lain pihak, evaluasi efisiensi motor berdasarkan papan-nama (*nameplate*) atau buku arahan (*manual book*) tidak menjamin penilaian efisiensi yang tepat dari motor induksi yang dipakai. Sehingga timbul permintaan dari industri-industri yang sebagian besar mesinnya menggunakan aplikasi motor induksi untuk mengembangkan dan meningkatkan metode penentuan efisiensi motor induksi dengan tingkat ketelitian yang mendekati nilai sebenarnya.<sup>[1,2,3]</sup>



## **1.2 Perumusan Masalah**

Mengacu pada latar belakang di atas maka permasalahan yang timbul untuk mendapatkan metode estimasi efisiensi yang efektif adalah Bagaimana penerapan aplikasi metode *Evolutionary Programming* motor induksi tiga fasa berdasarkan data pengukuran di lapangan ( Pengujian berbeban / *Load Test*, pengujian arus searah / *DC Test*, dan data papan-nama / *nameplate*) dari motor induksi sebagai solusi pemecahan masalah . Sehingga skripsi ini mengambil judul :

**“ESTIMASI EFEKTIF EFFISIENSI MOTOR INDUKSI TIGA PHASA  
BERDASAR DATA PENGUKURAN DI LAPANGAN MENGGUNAKAN  
METODE *EVOLUTIONARY PROGRAMMING* ”**

## **1.3 Tujuan Penelitian**

Tujuan pembahasan ini adalah untuk menganalisis aplikasi metode *Evolutionary Programming* pada penentuan efisiensi motor induksi tiga fasa berdasarkan data pengukuran di lapangan dari motor induksi.

#### **1.4 Batasan Masalah**

Agar pembahasan dalam skripsi ini lebih terarah sesuai dengan tujuan, maka permasalahan dibatasi oleh hal-hal sebagai berikut :

1. Analisis dilakukan pada Motor Induksi Tiga-Fasa berkapasitas 1,1 kW jenis Rotor Sangkar
2. Pembahasan ditekankan pada analisis aplikasi metode *Evolutionary Programming* pada estimasi efisiensi motor induksi tiga fasa.
3. Besaran-besaran rotor diasumsikan berpatokan pada besaran stator pada rangkaian ekivalen motor induksi.
4. Pengujian Berbeban (*Load Test*) dilakukan pada beban 25%, 50%, 75%, dan 100% dari beban penuh.
5. Motor beroperasi pada keadaan steady state.
6. Tidak membahas pengasutan motor induksi.
7. Tidak membahas sistem proteksi motor induksi.
8. Tidak membahas Perbaikan factor daya & pemakaian kapasitor .

## **1.5. Metodologi Penelitian**

Metodologi yang dipakai dalam penyusunan skripsi ini adalah :

1. Studi literatur, yaitu bahan untuk memahami prinsip kerja dari motor induksi tiga-fasa dan metode *Evolutionary Programming* .
2. Pengumpulan data melalui percobaan pada motor induksi dengan metode pengujian berbeban ( *Load Test* ) dan pengujian arus searah ( *DC Test* ).
3. Melakukan perhitungan untuk menentukan daya keluaran sehingga nantinya akan didapat nilai efisiensi untuk hasil sebenarnya (dari pengujian ).
4. Melakukan analisis penentuan efisiensi motor induksi menggunakan metode *Evolutionary programming* dengan bantuan program Delphi 7.0.
5. Membuat perbandingan antara efisiensi hasil pengujian dan efisiensi dari hasil metode *Evolutionary Programming* .
6. Menarik kesimpulan

## **1.6. Sistematika Penulisan**

Skripsi ini dibagi menjadi lima bab, yaitu :

### **BAB I : PENDAHULUAN**

Berisikan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi dan sistematik penulisan serta relevansinya.

### **BAB II : KONSEP DASAR MOTOR INDUKSI**

Membahas tentang teori dasar mengenai motor induksi, konstruksi, prinsip kerja motor induksi, rangkaian ekivalen dan pengujian motor induksi tiga fasa.

### **BAB III : TEORI DASAR EVOLUTIONARY PROGRAMMING DAN APLIKASI PADA ESTIMASI EFEKTIF EFISIENSI MOTOR INDUKSI TIGA PHASA**

Berisikan tentang teori dasar Evolutionary Programming , dan penerapannya pada estimasi efektif efisiensi motor induksi tiga fasa.

### **BAB IV : SIMULASI , ANALISIS DAN HASIL**

Membahas tentang analisa data pengujian yang digunakan dalam estimasi efektif efisiensi motor induksi tiga fasa, flowchart program, tampilan program, perbandingan hasil pengujian dengan program Evolutionary Programming .

### **BAB V : PENUTUP**

Berisikan kesimpulan dan saran sesuai dengan analisa yang dilakukan pada BAB IV.

## **1.7. Kontribusi & Relevansi**

Adalah penting bahwa efisiensi motor harus diperkirakan dengan cepat dengan gangguan minimum terhadap jasa pelayanannya. Dengan adanya analisis ini dapat memberikan suatu alternatif metode dalam penentuan efisiensi motor induksi tiga phasa yang dapat membantu pabrikan / industri untuk membuat keputusan yang tepat dalam menggantikan motor yang tidak efisien dengan motor lain yang lebih efisien dengan menggunakan hasil penilaian efektif dari efisiensi motor induksi di lapangan.

## **BAB II**

### **KONSEP DASAR MOTOR INDUKSI TIGA PHASA**

#### **2.1. Teori Dasar Motor Induksi.**

Motor arus bolak-balik ( Motor AC ) adalah suatu mesin yang berfungsi untuk mengubah tenaga listrik arus bolak-balik menjadi tenaga mekanik atau tenaga gerak, dimana tenaga gerak ini berupa perputaran pada motor. Salah satu jenis motor AC ini adalah motor induksi atau motor tak serempak.

Disebut motor tak serempak karena putaran motor tidak sama dengan putaran fluks magnet stator. Dengan kata lain, bahwa antara putaran rotor dan putaran fluks magnet terdapat selisih putaran yang disebut slip.<sup>[6]</sup>

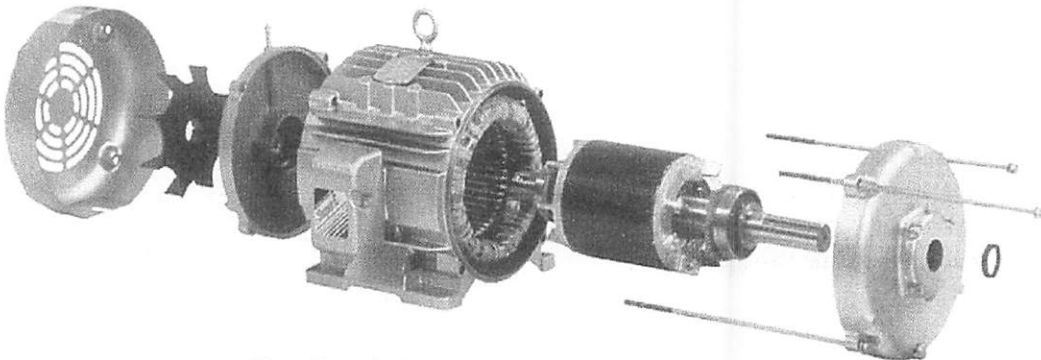
Motor induksi *polyphase* banyak dipakai dikalangan industri. Ini berkaitan dengan beberapa keuntungannya.

Keuntungan :

1. Sangat sederhana dan daya tahan kuat ( konstruksi hampir tak pernah mengalami kerusakan, khususnya tipe rotor sangkar bajing ).
2. Harga relatif murah dan perawatan mudah.
3. Efisiensi tinggi. Pada kondisi berputar normal, tidak dibutuhkan sikat dan karenanya rugi daya yang ditimbulkan dapat dikurangi (khususnya motor induksi rotor belitan).

## 2.2. Konstruksi Motor Induksi

Konstruksi motor induksi terdiri dari dua bagian utama yaitu stator dan rotor. Hal ini dapat dilihat pada gambar 2-1 di bawah ini :



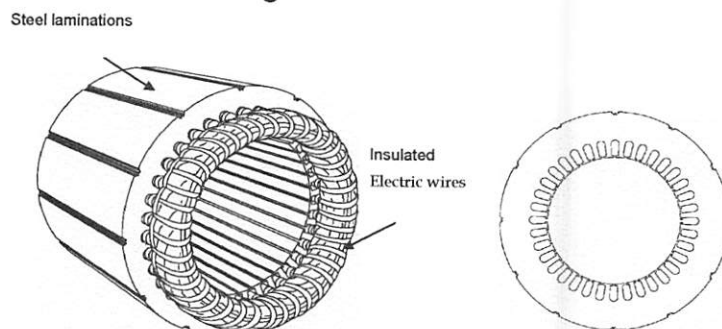
**Gambar 2-1 : Konstruksi Motor Induksi**

Sumber : <http://www.enm.com>, Dynamics of 3-phase Induction Motors, Part I.

### 2.2.1. Stator

Pada dasarnya konstruksi stator pada motor induksi mempunyai bentuk fisik yang sama dengan mesin sinkron, yang terdiri dari :

- Rumah stator terbuat dari besi tuang.
- Inti stator dari besi atau baja silikon.
- Alur dan gigi materialnya sama dengan inti, alur tempat meletakkan belitan.
- Belitan stator dari tembaga.



**Gambar 2-2: Stator Tiga-Phasa Motor Induksi**

Sumber : <http://www.enm.com>, Dynamics of 3-phase Induction Motors, Part I.



### 2.2.2. Rotor

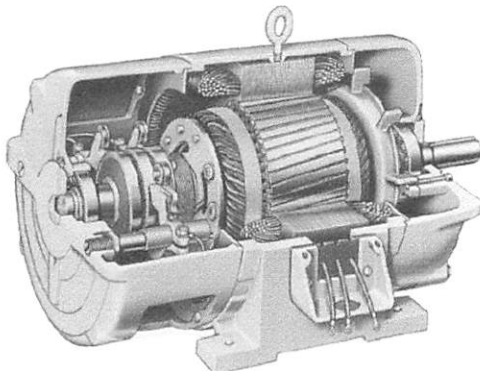
Konstruksi dari rotor motor induksi mempunyai dua bentuk, yaitu :

#### 1. Rotor Belitan (*wound rotor/ rotor slip ring*).

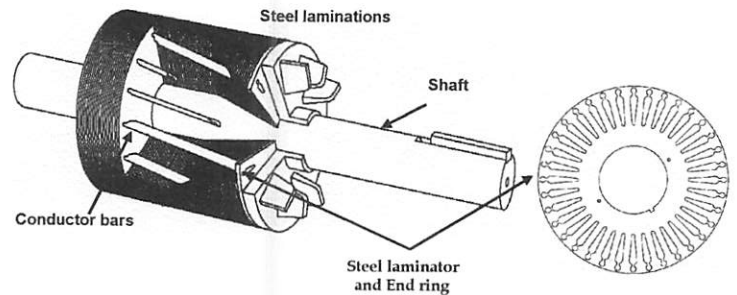
Motor induksi jenis ini mempunyai rotor dengan belitan kumparan tiga-fasa sama seperti kumparan stator. Kumparan stator dan rotor juga mempunyai jumlah kutub yang sama.

#### 2. Rotor sangkar (*squirrel cage rotor*).

Motor induksi jenis ini mempunyai rotor dengan kumparan yang terdiri atas beberapa batang konduktor yang disusun sedemikian rupa sehingga menyerupai sangkar tupai.



(a)



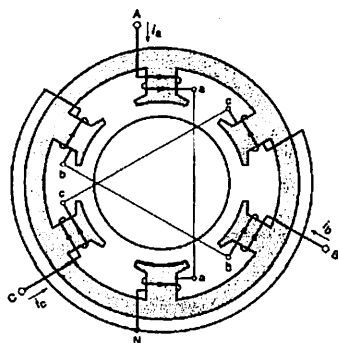
(b)

**Gambar 2-3 : a) Motor Induksi Rotor Belitan  
b) Rotor Sangkar Bajing**

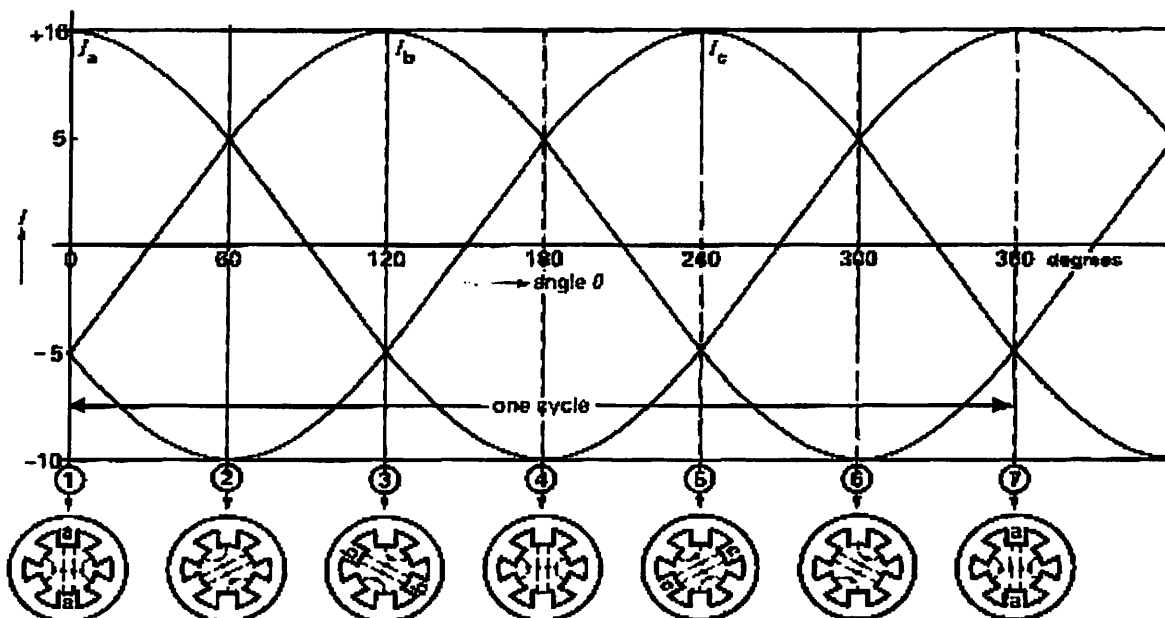
Sumber : <http://www.enm.com>, Dynamics of 3-phase Induction Motors, Part I.

### 2.3. Medan Magnet Putar<sup>[6]</sup>

Perputaran motor pada mesin arus bolak-balik ditimbulkan oleh adanya medan putar ( fluks yang berputar ) yang dihasilkan dalam kumparan statornya. Medan putar ini terjadi apabila kumparan stator dihubungkan dalam phasa banyak, umumnya tiga phasa. Hubungan dapat berupa hubungan bintang atau delta.



(a)



(b)

**Gambar 2-4 : Medan Putar Pada Motor Induksi**  
<http://www.enm.com>, Dynamics of 3-phase Induction Motors, Part I.

Medan putar terjadi apabila kumparan A-a, B-b, C-c dihungkan tiga fasa dengan beda fasa masing-masing  $120^{\circ}$  (hubungan bintang, Y) dan dialiri arus sinusoida. Distribusi  $i_a, i_b, i_c$  sebagai fungsi waktu adalah seperti gambar 2-4b. Pada keadaan  $t_1$  fluks resultan mempunyai arah yang sama dengan arah yang dihasilkan oleh kumparan A-a, sedangkan pada  $t_3$ , fluks resultannya dihasilkan oleh kumparan B-b. Untuk  $t_4$ , fluks resultannya berlawanan arah dengan fluks resultan yang dihasilkan pada  $t_1$ . Dari gambar 2-4b tersebut terlihat bahwa fluks resultan ini akan berputar satu kali.

#### 2.4. Prinsip Kerja Motor Induksi

Motor induksi tiga fasa dapat dibandingkan dengan transformator karena merupakan piranti yang melibatkan perubahan kebocoran fluks pada kumparan stator. Dalam hubungan ini diasumsikan bahwa rotor terdiri atas tipe lilitan dan hubungan bintang. Dengan lilitan rotor dalam keadaan rangkaian terbuka tidak ada torsi yang dibangkitkan. Dengan demikian pemberian tegangan tiga fasa pada kumparan stator tiga fasa menimbulkan medan magnet putar dan memotong kumparan rotor pada frekuensi  $f_1$ . Nilai rata-rata ggl induksi per fasa dari kumparan rotor dinyatakan dengan persamaan :

$$E_2 = 4,44 f_1 N_2 k_w \Phi_m \dots\dots\dots (2.1)$$

Perlu diketahui bahwa frekuensi stator  $f_1$  digunakan disini karena rotor tersebut dalam keadaan diam / berhenti. Dengan demikian  $E_2$  merupakan ggl frekuensi saluran . Fluks ( $\Phi_m$ ) tentu merupakan tiap elektroda (pole) dari kumparan stator dan rotor.

Rumus yang serupa menyatakan nilai rata-rata ggl induksi tiap fasa yang terjadi dari kumparan stator, yaitu :

$$E_1 = 4,44f_1 N_1 k_{w1} \Phi_m \dots\dots\dots (2.2)$$

Berdasarkan persamaan (2.1) dan (2.2) dapat dirumuskan rasio :

$$\frac{E_1}{E_2} = \frac{N_1 K_{w1}}{N_2 K_{w2}} \dots\dots\dots (2.3)$$

Pada dasarnya, motor induksi pada keadaan diam menyerupai karakteristik transformator dengan kumparan stator sebagai sisi primer dan kumparan rotor sebagai sisi sekundernya.

Untuk menghasilkan torsi mula (dan torsi penggerak berturut-turut) perlu arus yang mengalir dalam kumparan rotor. Mula-mula ggl induksi  $E_2$  mengakibatkan arus rotor  $I_2$  mengalir melalui rangkaian hubung-singkat, menghasilkan distribusi *ampere-conductor* yang berkerja dengan medan fluks untuk menghasilkan torsi mula. Pengaruh torsi ini selalu mengakibatkan rotor berputar dalam arah yang sama sebagaimana medan putar. Anggaplah bahwa medan fluks putar searah jarum jam pada kecepatan tertentu yang bergantung pada frukuensi stator dan banyaknya kutub dari kumparan stator. Kecepatan ini disebut “kecepatan sinkron” dan dinyatakan :

$$n_s = \frac{120f}{p} \text{ rpm} \dots\dots\dots (2.4)$$

karena rotor meningkat kecepatannya, laju yang mengijinkan medan stator memotong kumparan rotor menurun. Hal ini mengurangi ggl induksi resultan per phasa, pada gilirannya menurun magnitude distribusi *amper-conductor* dan menghasilkan torsi yang lebih kecil. Pada kenyataan proses ini berlanjut hingga kecepatan rotor mampu untuk menghasilkan ggl yang cukup agar arus yang diperlukan untuk membangkitkan torsi yang setara dengan torsi lawan.

### 2.4.1. Slip dan Frekuensi Arus Rotor

Slip diidentifikasi sebagai bagian Dari kecepatan sinkron  $n_s$  dan kecepatan aktual rotor  $n_r$ . Slip dirumuskan sebagai berikut :

$$s = \frac{n_s - n_r}{n_s} \dots\dots\dots (2.5)$$

Pada keadaan diam medan magnet putar yang dihasilkan oleh stator mempunyai kecepatan relatif yang sama dengan kumparan rotor. Pada saat ini frekuensi dari arus rotor sama dengan frekuensi stator ( $f_r = f_s$ ). Frekuensi rotor  $f_r$  adalah nol ketika motor berputar pada kecepatan sinkron. Pada saat tersebut tidak terdapat gerakan (putaran) relatif antara medan putar dan rotor. Pada kecepatan yang lain, frekuensi rotor proporsional dengan slip ( $s$ ). Hubungan antara slip dan frekuensi dapat dilihat dari persamaan berikut ini<sup>[6]</sup> :

$$n_s = \frac{120f_s}{p} \text{ atau } f_s = \frac{p.n_s}{120} \dots\dots\dots (2.6)$$

dimana :  $p$  = jumlah kutub

$f_s$  = frekuensi stator

Pada rotor berlaku hubungan :

$$f_r = \frac{(n_s - n_r)p}{120} = \frac{(n_s - n_r)n_s p}{n_s 120} \dots\dots\dots (2.8)$$

$$s = \frac{n_s - n_r}{n_s} \text{ dan } f_s = \frac{p.n_s}{120}$$

Maka :

$$f_r = s.f_s \dots\dots\dots (2.9)$$

## 2.5. Rangkaian Ekivalen Motor Induksi

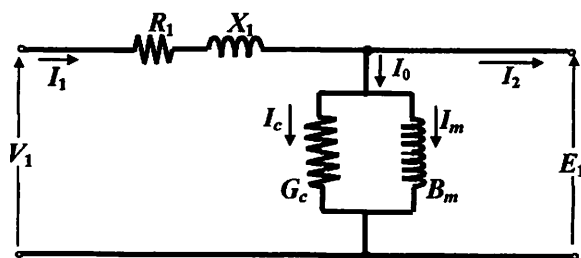
### 2.5.1. Rangkaian Ekivalen

Suatu rangkaian ekivalen motor induksi tiga fasa diperlukan untuk membantu analisis operasi dan untuk memudahkan penghitungan kinerja. Rangkaian ekivalen tersebut mengasumsikan suatu bentuk yang identik rangkaian ekivalen transformator. Proses penurunannya serupa dengan model dengan modifikasi-modifikasi baru seperlunya untuk menghitung kumparan sekunder (rotor) dalam hal ini berputar dan menghasilkan daya mekanik.

### 2.5.2. Rangkaian Ekivalen Stator

Apabila kumparan stator diberikan tegangan catu dari jala-jala sebesar  $V_L$ , maka akan mengalir arus putar tiga fasa pada kumparan stator yang membangkitkan medan magnet tiga fasa. Arus stator ( $I_1$ ) bercabang menjadi dua komponen arus yaitu :

1. Komponen arus beban ( $I_2$ )
2. Komponen arus eksitasi ( $I_0$ )



**Gambar 2-5 : Rangkaian Ekivalen Stator**

Sumber : A. E. Fitzgerald, Charles Kingsley, Jr., Stephen D. Umans, "Electric Machinery", McGraw-Hill International, fourth edition.

Dimana :

$V_1$  = tegangan terminal per-fasa

$R_1$  = resistansi kumparan stator per-fasa

$X_1$  = reaktansi bocor kumparan stator per-fasa

$E_1$  = tegangan induksi (ggl) per-fasa di dalam kumparan stator

$G_c$  = konduktansi rugi-rugi inti stator per-fasa

$B_m$  = suseptansi magnetisasi stator per-fasa

### 2.5.3. Rangkaian Ekuivalen Rotor

Pada saat rotor diam, medan putar stator akan memotong batang konduktor rotor dengan kecepatan putar sinkron ( $n_s$ ), sehingga frekuensi arus rotor sama dengan frekuensi arus stator ( $f_s = f_r$ ) dan slip sama dengan satu ( $s=1$ ). Dengan mengetahui bahwa frekuensi arus / tegangan rotor adalah frekuensi slip, maka reaktansi bocor rotor (*leakage reactance*) per fasa adalah :

$$X_2' = sX_2 \dots\dots\dots (2.10)$$

$$X_2 = 2\pi f_s L_2 \dots\dots\dots (2.11)$$

dimana  $X_2$  merupakan reaktansi rotor pada start atau diam.

Tegangan induksi pada rotor :

$$E_2 = 4,44f_2 N_2 \Phi_m \dots\dots\dots (2.12)$$

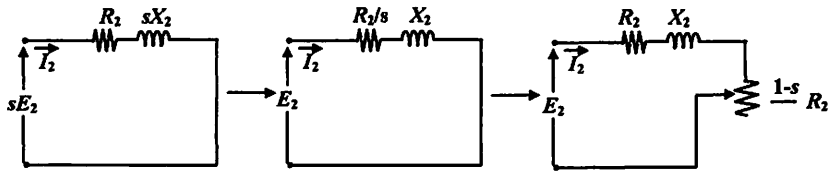
Pada slip,  $s$ , frekuensi rotor menjadi  $s f_s$ , maka tegangan induksi pada rotor ( $E_2'$ ) pada slip,  $s$ , adalah :

$$E_2' = 4,44s f_1 N_2 \Phi_m \dots\dots\dots (2.13)$$

Dengan memasukkan persamaan (2.12) ke (2.13) maka didapat persamaan :

$$E_2' = s E_2 \dots\dots\dots (2.14)$$





**Gambar 2-6: Rangkaian Ekivalen Rotor**

Sumber : A. E. Fitzgerald, Charles Kingsley, Jr., Stephen D. Umans, "Electric Machinery", McGraw-Hill International, fourth edition.

Dimana :

$S$  = Slip

$E_2$  = tegangan induksi per-fasa di dalam rotor pada keadaan diam

$R_2'$  = resistansi kumparan rotor per-fasa berpatokan pada stator

$X_2'$  = reaktansi bocor rotor per-fasa berpatokan pada stator

Berdasarkan persamaan (2.10) dan (2.14) maka diperoleh rangkaian ekivalen rotor seperti pada gambar 2-6.

Besar arus rotor ( $I_2$ ) saat berputar adalah :

$$I_2 = \frac{sE_2}{\sqrt{R_2'^2 + (sX_2')^2}} \dots\dots\dots (2.15)$$

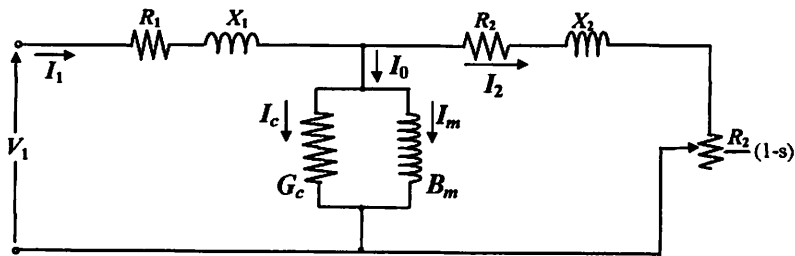
Atau

$$I_2 = \frac{E_2}{\sqrt{\left(\frac{R_2'}{s}\right)^2 + X_2'^2}} \dots\dots\dots (2.16)$$

### 2.5.4. Rangkaian Ekuivalen Motor Induksi

Kerja motor induksi seperti juga kerja pada transformator adalah berdasarkan prinsip induksi elektromagnetik. Oleh karena itu motor induksi dipandang sebagai transformator yang mempunyai ciri-ciri khusus, yaitu :

1. Stator sebagai sisi primer
2. Rotor sebagai sisi sekunder yang penghantar-penghantarnya dihubungkan singkat dan berputar
3. Kopling antara sisi primer dan sisi sekunder dipisahkan oleh celah udara (*air gap*).



**Gambar 2-7 : Rangkaian Ekuivalen Motor Induksi**

Sumber : A. E. Fitzgerald, Charles Kingsley, Jr., Stephen D. Umans, "Electric Machinery", McGraw-Hill International, fourth edition.

Rangkaian tersebut memperlihatkan bahwa daya keseluruhan yang dialihkan pada celah udara dari stator ( masukan daya ke rotor ) adalah :

$$P_2 = 3I_2^2 \frac{R_2}{s} \dots\dots\dots(2.17)$$

Dengan rugi tembaga rotor :

$$P_{cu2} = 3I_2^2 R_2 \dots\dots\dots(2.18)$$

Maka daya mekanis yang dibangkitkan oleh motor induksi adalah :

$$P_m = P_2 - P_{cu2} = 3I_2^2 \frac{R_2}{s} - 3I_2^2 R_2 = 3I_2^2 R_2 \frac{(1-s)}{s} \dots\dots\dots (2.19)$$

$$P_m = T \cdot \omega_r = T \cdot \omega_s(1-s) \dots\dots\dots (2.20)$$

dimana :

T = Torsi motor dalam N-m

$\omega_r$  = kecepatan rotor dalam rad/det

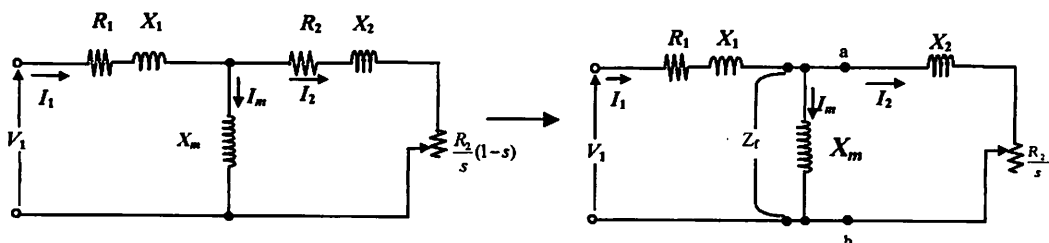
$\omega_s$  = kecepatan sinkron stator dalam rad/det

sehingga diperoleh :

$$T = \frac{3}{\omega_s} I_2^2 \frac{R_2}{s} \text{ Nm} \dots\dots\dots (2.21)$$

$$I_2 = \frac{V_1}{\sqrt{\left(R_1 + \frac{R_2}{s}\right)^2 + (X_1 + X_2)^2}} \dots\dots\dots (2.22)$$

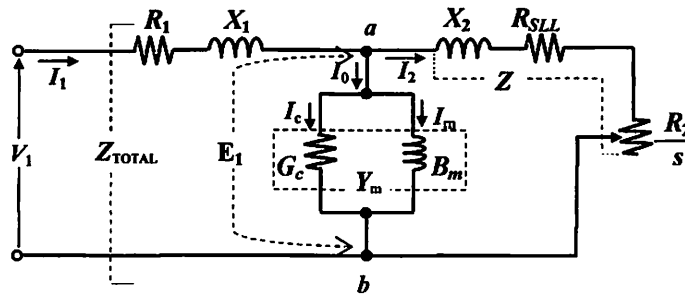
Dalam analisis rangkaian ekivalen, sering disederhanakan dengan menghilangkan konduktansi ( $R_c$ ), sehingga rangkaian ekivalen pada gambar (2.7) berubah menjadi :



**Gambar 2-8 : Penyederhanaan Rangkaian Ekivalen Motor Induksi**  
 Sumber : A. E. Fitzgerald, Charles Kingsley, Jr., Stephen D. Umans, "Electric Machinery", McGraw-Hill International, fourth edition.

### 2.5.5. Rangkaian Ekivalen Pada Penentuan Efisiensi Motor Induksi Tiga Fasa

Dengan menggunakan analisis rangkaian ekivalen eksak (*exact equivalent circuit*) dapat dihitung impedansi rangkaian sebagai berikut :



**Gambar 2-9 : Rangkaian Ekivalen Motor Induksi**

Sumber : T. Phumiphak and C. Chat-uthai, "Effective Estimation of Induction Motor Field Efficiency Based on On-Site Measurements", Proceedings of International Conference on Power System Technologi, Power Con, Kunming, China, 2003.

Untuk mencari tahanan stator ( $R_s$ ) pada hubungan bintang (Y) digunakan rumus :

$$R_1 = \frac{R_{1lin}}{2} \dots\dots\dots (2.23)$$

Power factor (pf) masukan dapat dicari dengan persamaan :

$$pf = \frac{P_{in}}{\sqrt{3}V_1I_1} \dots\dots\dots (2.24)$$

dimana :  $R_1$  = Resistansi stator(ohm/fasa)

$R_{1lin}$  = Resistansi line stator

$V_1$  = Tegangan line stator

$I_1$  = Arus stator

$P_{in}$  = Daya Masukan

Data yang digunakan untuk menentukan parameter pada rangkaian ekivalen dan efisiensi motor diukur dan dihitung pada 25%, 50%, 75% dan 100% dari beban penuh. Menurut prosedur IEEE untuk penentuan efisiensi dengan rugi beban tambahan(*stray load loss*) diasumsikan kerugian ini di bawah beban penuh sekitar 1.8% dari daya keluran ( $P_{out}$ ) dan kerugian dari beban manapun dapat dihitung dengan persamaan :

$$R_{SLL} = \frac{0.018 (1 - s_{fl}) R_2}{s_{fl}} \dots\dots\dots (2.25)$$

dimana :

- $R_2$  = Resistansi rotor berpatokan pada stator
- $s_{fl}$  = Slip beban penuh

Temperatur kumparan stator dan rotor diasumsikan sama dan dihitung dengan persamaan berikut :

$$T_t = \frac{I_1 - I_0}{I_{fl} - I_0} (T_r - T_s) + T_s \dots\dots\dots (2.26)$$

dimana :  $I_1$  = Arus stator yang diukur

$I_{fl}$  = Arus stator pada *nameplate*

$I_0$  = Arus stator pada pengukuran DC

$T_r$  = Temperatur pada isolasi(75°C)

$T_s$  = Temperatur lingkungan (25°C)

Nilai resistansi stator dan rotor dikoreksi (*corrected*) dengan data temperatur melalui persamaan :

$$R_{1c} = R_1 \frac{T_t + k_c}{T_s + k_c} \dots\dots\dots (2.27)$$

$$R_{2c} = R_2 \frac{T_t + k_a}{T_s + k_a} \dots\dots\dots (2.28)$$

dengan :  $R_{1c}$  = Resistansi stator *DC Test* yang dikoreksi

$R_{2c}$  = Resistansi rotor yang dikoreksi

$k_c$  = Faktor koreksi tembaga(234.5)

$k_a$  = Faktor koreksi alumunium(225)

Admitansi antara a dan b adalah<sup>[10]</sup> :

$$Y_{ab} = Y_m + \frac{1}{Z} \text{ mho/phase} \dots\dots\dots (2.29)$$

untuk nilai real(Re) dan Imajiner(Im) dari admitansi( $Y_{ab}$ ) :

$$\text{Re } Y_{ab} = \text{Re } Y_m + \left[ \frac{(R_{SLL} + R_{2c}/s)}{(R_{SLL} + R_{2c}/s)^2 + (X_2)^2} \right] \text{ mho/phase} \dots\dots\dots (2.30)$$

$$\text{Im } Y_{ab} = \text{Im } Y_m + \left[ \frac{-(X_2)}{(R_{SLL} + R_{2c}/s)^2 + (X_2)^2} \right] \text{ mho/phase} \dots\dots\dots (2.31)$$

sehingga impedansi pada a dan b ( $Z_{ab}$ ) adalah :

$$Z_{ab} = \frac{1}{Y_{ab}} \Omega/\text{phase} \dots\dots\dots (2.32)$$

untuk nilai real (Re) dan Imajiner (Im) dari Impedansi ( $Z_{ab}$ ) :

$$\text{Re}Z_{ab} = \frac{\text{Re} Y}{\text{Re}(Y)^2 + \text{Im}(Y)^2} \Omega/\text{phasa} \dots\dots\dots (2.33)$$

$$\text{Im}Z_{ab} = -\frac{\text{Im} Y}{\text{Re}(Y)^2 + \text{Im}(Y)^2} \Omega/\text{phasa} \dots\dots\dots (2.34)$$

$$|Z_{ab}| = \sqrt{\text{Re}(Z_{ab})^2 + \text{Im}(Z_{ab})^2} \Omega/\text{phasa} \dots\dots\dots (2.35)$$

Untuk nilai impedansi masukan ( $Z_{TOTAL}$ ) :

$$\text{Re}Z_{TOTAL} = R_{lc} + \text{Re}Z_{ab} \Omega/\text{phasa} \dots\dots\dots (2.36)$$

$$\text{Im}Z_{TOTAL} = X_1 + \text{Im}Z_{ab} \Omega/\text{phasa} \dots\dots\dots (2.37)$$

$$|Z_{TOTAL}| = \sqrt{\text{Re}(Z_{TOTAL})^2 + \text{Im}(Z_{TOTAL})^2} \Omega/\text{phasa} \dots\dots\dots (2.38)$$

Dengan menerapkan Hukum Ohm pada rangkaian arus bolak-balik (AC) maka diperoleh arus stator ( $I_1$ ) :

$$I_1 = \left| \frac{V_1}{Z_{TOTAL}} \right| \text{Ampere} \dots\dots\dots (2.39)$$

untuk nilai emf ( $E_1$ ) :

$$E_1 = \left| \frac{V_1 \cdot Z_{ab}}{Z_{TOTAL}} \right| \text{volt/phasa} \dots\dots\dots (2.40)$$

Arus rotor ( $I_2$ ) :

$$I_2 = \frac{E_1}{\sqrt{(R_{SLL} + R_{2c}/s)^2 + (X_2)^2}} \text{Ampere} \dots\dots\dots (2.41)$$



dan untuk arus yang melewati konduktansi rugi-rugi inti ( $I_c$ ) dan suseptansi magnetisasi ( $I_m$ ) dirumuskan masing-masing sebagai berikut :

$$I_c = E_1 \cdot \text{Re} Y_m \text{ Ampere} \dots\dots\dots (2.42)$$

$$I_m = E_1 \cdot \text{Im} Y_m \text{ Ampere} \dots\dots\dots (2.43)$$

Daya masukan ( $P_{in}$ ) pada rangkaian ekivalen adalah :

$$P_{in} = 3(I_1^2 R_{lc} + I_2^2 (R_{sll} + R_{2c} / s) + I_c^2 R_c) \text{ watt} \dots\dots\dots (2.44)$$

Daya Keluaran ( $P_{out}$ ) pada rangkaian ekivalen adalah :

$$P_{out} = T \frac{4\pi f}{p} (1 - s) \text{ watt} \dots\dots\dots (2.45)$$

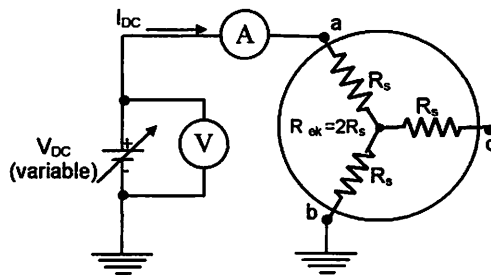
Efisiensi( $\eta$ ) didapat dari persamaan :

$$\eta = \frac{P_{out}}{P_{in}} \cdot 100\% \dots\dots\dots (2.46)$$

## 2.6. Pengujian Motor Induksi Tiga Phasa

### 2.6.1. Pengujian Arus Searah (DC Test)

Tujuan dari pengujian arus searah (*DC Test*) adalah untuk menentukan nilai resistansi stator. Diagram pengukuran ditunjukkan pada gambar 2.10.



**Gambar 2-10 : Pengujian Arus Searah (*DC Test*)**  
 Sumber : Electric Drive, An Integrative Systems Approach, EE 430.

Kumparan stator terhubung bintang (Y) dan bila sumber DC disuplai melalui dua kumparan (kumparan a dan b) , dengan kumparan ke tiga (kumparan c) dalam keadaan terbuka (*open circuit*), maka nilai dari resistansi ekivalen ( $R_{ek}$ ) :

$$R_{ek} = 2R_s \ \Omega \dots\dots\dots (2.47)$$

untuk nilai resistansi kumparan a dan b :

$$R_{ab} = R_{ek} = \frac{V_{DC}}{I_{DC}} \ \Omega \dots\dots\dots (2.48)$$

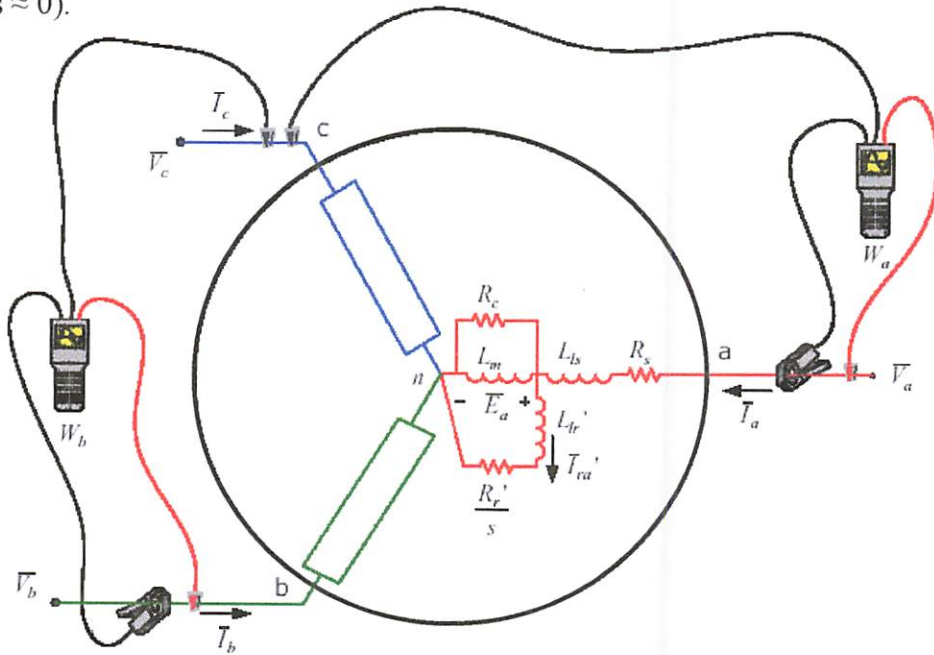
Dan nilai resistansi stator per phasa,  $R_s$  :

$$R_s = \frac{R_{ek}}{2} \ \Omega/\text{phasa} \dots\dots\dots (2.49)$$

Dalam pengujian arus searah dijaga agar arus DC ( $I_{DC}$ ) tidak melampaui nilai dari arus nominal motor induksi.

**2.6.2. Pengujian Tanpa Beban (No-Load Test)**

Pengujian Tanpa Beban (*No-Load Test*) bertujuan untuk menentukan nilai resistansi rugi-rugi inti ( $R_c$ ) dan reaktansi pemagnetan ( $X_m$ ). Pada pengujian ini motor induksi disuplai pada tegangan dan frekuensi nominalnya, serta rotor berputar tanpa terhubung dengan peralatan beban dimana harga slip mendekati nol ( $s \approx 0$ ).



**Gambar 2-11 : Diagram Pengujian Tanpa Beban**  
 Sumber : Electric Drive, An Integrative Systems Approach, EE 430.

$P_{3-\phi}$ , daya total yang terukur dari  $W_a$  dan  $W_b$  :

$$P_{3-\phi} = W_a + W_b \text{ watt} \dots\dots\dots (2.50)$$

$$P_{nl}, \text{ daya per fasa : } P_{nl} = \frac{P_{3-\phi}}{3} \text{ watt/fasa} \dots\dots\dots (2.51)$$

Dengan asumsi bahwa tegangan antar fasa stator seimbang, maka tegangan fasa stator :

$$V_{nl} = \frac{V_{ab}}{\sqrt{3}} \text{ Volt/fasa} \dots\dots\dots (2.52)$$

$I_{nl}$ , arus phasa stator :

$$I_{nl} = \frac{I_a + I_b + I_c}{3} \text{ Ampere} \dots \dots \dots (2.53)$$

$\theta_{nl}$ , Sudut phasa dimana  $I_{nl}$  lagging terhadap  $V_{nl}$  :

$$\theta_{nl} = \cos^{-1} \left( \frac{P_{nl}}{V_{nl} I_{nl}} \right) \dots \dots \dots (2.54)$$

dari gambar 2-11 dan dengan menerapkan hukum kirchhoff tegangan (KVL) maka tegangan terinduksi,  $E_a$  adalah :

$$E_a = V_{nl} \angle 0^\circ - (I_{nl} \angle -\theta_{nl}) (R_s + jX_s) \text{ volt/phasa} \dots \dots \dots (2.55)$$

$I_{ra}'$ , arus rotor per phasa :

$$I_{ra}' = \frac{E_a}{\left( \frac{R_r'}{s_{nl}} \right) + jX_r'} \dots \dots \dots (2.56)$$

dimana :  $s_{nl} = \frac{n_s - n_{r,nl}}{n_s} \dots \dots \dots (2.57)$

$P_c$ , rugi-rugi inti per phasa :

$$P_c = P_{nl} - (I_{nl})^2 R_s - (I_{ra}')^2 \frac{R_r'}{s_{nl}} \dots \dots \dots (2.58)$$

$R_c$ , resistansi rugi-rugi inti :

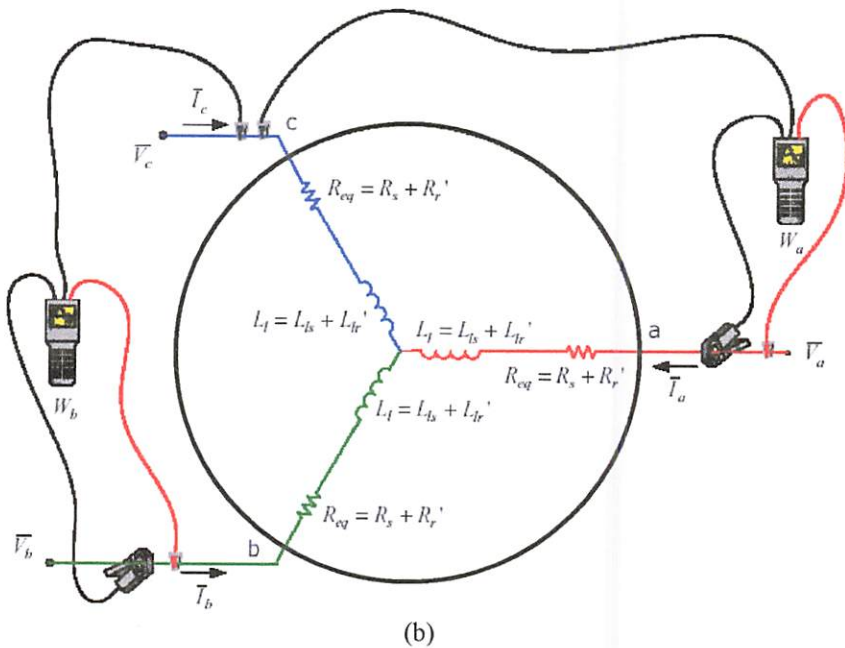
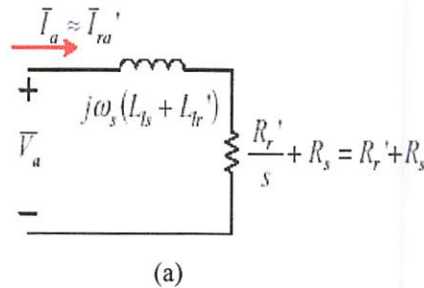
$$R_c = \frac{E_a^2}{P_c} \text{ ohm/phasa} \dots \dots \dots (2.59)$$

$Q_m$ , daya reaktif :  $Q_m = V_{nl} I_{nl} \sin(\theta_{nl}) - (I_{nl})^2 X_s - (I_{ra}')^2 X_r' \text{ VAR} \dots \dots \dots (2.60)$

$X_m$ , reaktansi pemagnetan :  $X_m = \frac{E_a^2}{Q_m} \text{ ohm/phasa} \dots \dots \dots (2.61)$

### 2.6.3. Pengujian Rotor Tertahan (*Blocked Rotor Test*)

Tujuan pengujian rotor tertahan adalah untuk menentukan resistansi rotor pada motor induksi. Pada saat pengujian ini perputaran rotor motor induksi dikunci/diblok sehingga slip ( $s$ ) sama dengan satu. Suplai tegangan tiga fasa motor induksi adalah tegangan yang nilainya di bawah tegangan nominalnya, yakni tegangan yang dapat menghasilkan arus nominalnya. Sebagai pendekatan, diasumsikan bahwa arus pemagnetan ( $I_m$ ) cukup kecil akibat penurunan suplai tegangan serta motor dalam keadaan tidak berputar ( $s = 1$ ) sehingga rugi-rugi inti dapat diabaikan.



**Gambar 2-12 : (a) Ragkaian Ekivalen Pengujian Rotor Tertahan  
(b) Ragkaian Pengujian Rotor Tertahan**

Sumber : Electric Drive, An Integrative Systems Approach, EE 430.

$P_{3-\phi}$  ,daya total yang terukur dari  $W_a$  dan  $W_b$  :

$$P_{3-\phi} = W_a + W_b \text{ watt} \dots\dots\dots (2.62)$$

Daya total tiga-fasa merupakan rugi-rugi tembaga stator dan rotor, karena motor tidak berputar maka rugi-rugi inti diabaikan.

$P_{br}$ , rugi-rugi daya per fasa :

$$P_{br} = \frac{P_{3-\phi}}{3} \text{ watt/fasa} \dots\dots\dots (2.63)$$

Dengan asumsi bahwa tegangan antar fasa stator seimbang, maka tegangan fasa stator :

$$V_{br} = \frac{V_{ab}}{\sqrt{3}} \text{ Volt/fasa} \dots\dots\dots (2.64)$$

$I_{br}$ , arus fasa stator :

$$I_{br} = \frac{I_a + I_b + I_c}{3} \text{ Ampere} \dots\dots\dots (2.65)$$

$R_{ek}$ , resistansi ekivalen :

$$R_{ek} = \frac{P_{br}}{I_{br}^2} \text{ ohm/fasa} \dots\dots\dots (2.66)$$

Untuk  $R_r'$ , resistansi rotor berpatokan pada stator :

$$R_r' = R_{ek} - R_s \text{ ohm/fasa} \dots\dots\dots (2.67)$$

$Z_{br}$ , Impedansi rotor tertahan :

$$Z_{br} = \frac{V_{br}}{I_{br}} \text{ ohm/fasa} \dots\dots\dots (2.68)$$

$X_{ek}$ , reaktansi ekivalen :

$$X_{ek} = \sqrt{(Z_{br}^2 - R_{ek}^2)} \text{ ohm/fasa} \dots\dots\dots (2.69)$$

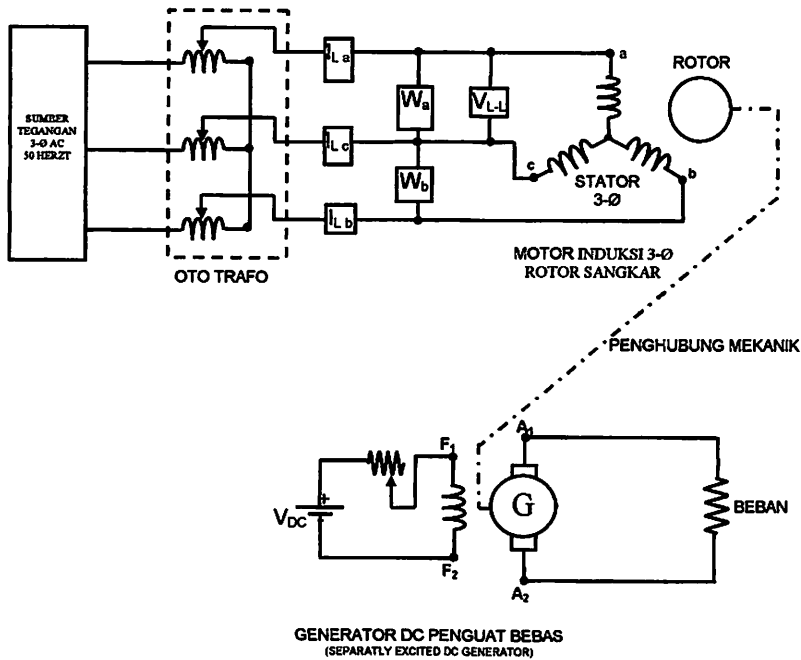
Dimana :

Motor induksi yang dipakai adalah motor induksi dengan rotor sangkar tunggal. Secara umum  $X_s$  dan  $X_r'$  diasumsikan sama<sup>[5]</sup>, sehingga ;

$$X_s = X_r' = 0.5 X_{ek} \text{ ohm/fasa}$$

### 2.6.4. Pengujian Berbeban (*Load Test*)

Tujuan dari pengujian berbeban adalah untuk mengetahui daya masukan ( $P_{in}$ ), Tegangan ( $V_l$ ), Arus ( $I_l$ ), Kecepatan rotor (rpm), Torsi (Nm) dari motor induksi pada kondisi-kondisi di bawah pembebanan (*under load*).



**Gambar 2-13 : Diagram Pengujian Berbeban**

Sumber : Laboratory #8, "Electric Machinery", Electrical Engineering Department, University of Alaska Fairbanks, EE 303.

## **BAB III**

### **TEORI DASAR *EVOLUTIONARY PROGRAMMING* DAN APLIKASI PADA ESTIMASI EFEKTIF EFISIENSI MOTOR INDUKSI**

#### **3.1. Evolutionary Programming [9]**

Evolutionary Programming merupakan metode adaptive yang bisa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Metode ini didasarkan pada proses evolusi yang ada dalam makhluk hidup, yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip seleksi alam. Dengan meniru proses ini, Evolutionary Programming dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata.

Evolutionary Programming ditemukan oleh Fogel pada awal tahun 1970 yang dilandasi oleh sifat-sifat evolusi alam. Fogel percaya bahwa ini sangat cocok digabungkan dalam sebuah algoritma komputer, menghasilkan sebuah teknik penyelesaian permasalahan-permasalahan yang sulit dengan langkah alamiah yaitu melalui evolusi. Fogel mulai bekerja dengan algoritma yang dibentuk dengan string-string bilangan real yang disebut *kromosom*. Seperti halnya alam, metode ini menyelesaikan permasalahan-permasalahan dengan menemukan kromosom-kromosom yang baik dengan memanipulasi materi dan sifat (*gene*) kromosom-kromosom. Algoritma ini mengetahui tipe permasalahan yang akan diselesaikan. Hanya informasi yang telah diberikan dari *evaluasi* berupa nilai fitness setiap kromosom yang dihasilkan digunakan untuk seleksi kromosom sehingga kromosom dengan nilai fitness terbaik yang bertahan hidup dan selalu diproduksi.



Sebelum Evolutionary Programming dijalankan, sebuah kode yang sesuai untuk persoalan harus dirancang. Titik solusi dalam ruang permasalahan dikodekan dalam bentuk kromosom / string yang terdiri dari komponen yang terkecil yaitu gen. Pemakaian bilangan real sebagai *allele* (nilai gen) memungkinkan penerapan operator evolutionary programming, yaitu : Reproduksi (*Reproduction*), Mutasi (*mutation*) dan kompetisi (*Competition*) untuk menciptakan himpunan titik-titik solusi. Untuk memeriksa hasil optimasi, kita membutuhkan fungsi fitness yang menandakan gambaran hasil yang sudah dikodekan. Selama proses, induk harus digunakan untuk reproduksi, mutasi dan kompetisi untuk menciptakan keturunan (*offspring*). Jika Evolutionary Programming didesain dengan baik, populasi akan mengalami konvergensi dan akan mendapatkan sebuah solusi yang optimum.

**Evolutionary Programming memiliki empat dasar kerja, yaitu :**

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

### 3.1.1. Istilah-Istilah Evolutionary Programming [9]

Evolutionary Programming menggunakan mekanisme Evolusi yang ada pada proses alami dan sistem buatan. Istilah-istilah yang digunakan adalah gabungan dari dua disiplin ilmu, yaitu ilmu Biologi dan ilmu Komputer. Mitsou Gen dan Runwei Cheng (1997) menjelaskan istilah-istilah yang digunakan didalam Evolutionary Programming sebagai berikut :

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

**Tabel 3-1**  
**Istilah Yang Digunakan Dalam Evolutionary Programming .**

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

#### 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 Evolutionary Programming ). Jumlah generasi yang besar dapat mengarah ke arah solusi yang optimal, namun akan membutuhkan waktu yang lama. Sedangkan jika jumlahnya terlalu sedikit maka solusi akan terjebak pada lokal optimum.

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

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

### **3.1.1.3 Probabilitas Kompetisi ( $P_c$ )**

Probabilitas kompetisi ini digunakan untuk mengendalikan frekuensi operator kompetisi . Dalam hal ini, dalam populasi terdapat  $P_c \times POPSIZE$  struktur (individu ) yang melakukan persaingan . Semakin besar nilai probabilitas kompetisi maka semakin cepat struktur baru diperkenalkan dalam populasi. Namun jika probabilitas kompetisi terlalu besar maka struktur dengan nilai fungsi obyektif yang baik dapat hilang dengan lebih cepat dari seleksi. Sebaliknya probabilitas kompetisi kecil akan menghalangi proses pencarian dalam proses Evolutionary Programming .

#### 3.1.1.4 Probabilitas Mutasi ( $P_m$ )

Mutasi digunakan untuk meningkatkan variasi populasi. Probabilitas mutasi ini digunakan untuk menentukan tingkat mutasi yang terjadi, karena frekuensi terjadinya mutasi tersebut menjadi  $P_m \times \text{POPSIZE} \times N$ , dimana  $N$  adalah panjang struktur / gen dalam suatu 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 Evolutionary Programming, mutasi menjalankan aturan yang penting, yaitu :

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

#### 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. Mekanisme secara umum dari Evolutionary Programming digambarkan oleh Fogel, proses kerjanya diawali oleh inisialisasi satu rangkaian nilai random yang disebut *populasi*. Setiap individu didalam populasi disebut *kromosom*. Sebuah kromosom dapat direpresentasikan dalam bentuk bilangan real. Kromosom-kromosom ini berkembang melalui beberapa iterasi yang disebut *generasi*. Setiap generasi, kromosom-kromosom ini dievaluasi dengan menggunakan ukuran *fitness* melalui fungsi tujuan (*Objective Function*) dan batasan-batasan fungsi kendala sehingga individu dengan solusi yang terbaik yang terpilih. Untuk menghasilkan generasi selanjutnya ( $t + 1$ ) sebagai kromosom baru yang disebut sebagai *offspring*, dibentuk melalui

penggabungan dua kromosom saat ini ( $t$ ) dengan menggunakan operator kompetisi dan memodifikasi sebuah kromosom menggunakan operator mutasi. Satu generasi baru dibentuk melalui proses seleksi sesuai dengan *fitness values* kromosom orang tua dan kromosom yang fit yang akan diturunkan. Kromosom dengan nilai fitness terbesar mempunyai probabilitas tertinggi untuk terpilih. Setelah beberapa generasi, kromosom dengan fitness terbaik konvergen, yang diharapkan sebagai solusi optimum atau solusi sub optimum permasalahan. Prosedur Evolutionary Programming dimodifikasi menurut versi Grenfenstette dan Baker's, digambarkan sebagai berikut :

```
Begin
     $t = 0$ ;
    Inisialisasi  $P(t)$ ;
    Evaluasi  $P(t)$ ;
    While ( not termination condition ) do
        Recombine  $P(t)$  ke  $C(t)$ ;
        Evaluasi  $C(t)$ ;
        Seleksi  $P(t+1)$  dari  $P(t)$  dan  $C(t)$ ;
     $t = t + 1$ ;
End;
End.
```

**Gambar 3-1 Struktur Evolutionary Programming**

### 3.1.2. Proses Evolutionary Programming [8]

Sangat perlu untuk mengetahui proses dalam Evolutionary Programming .  
Dibawah ini akan diuraikan mengenai hal itu, dimana uraian ini merupakan penjabaran dari mekanisme Evolutionary Programming seperti penjelasan pada bagian sebelumnya.

#### A. Pengkodean atau Representasi

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

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

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

## B. Fungsi Fitness (Fungsi Evaluasi )

Dalam Evolutionary Programming , 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  $f_k(x)$  dimana  $k = 1, 2, \dots, \text{POPSIZE}$ .

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

Permasalahan klasik dari Evolutionary Programming adalah beberapa kromosom dengan nilai fitness yang tertinggi (tetapi bukan nilai optimum) mendominasi populasi dan mengakibatkan Evolutionary programming konvergen pada lokal optimum. Ketika mencapai konvergen, kemampuan Evolutionary Programming untuk mencari solusi yang lebih baik menghilang. Kompetisi antara kromosom induk yang hampir identik menghasilkan keturunan (*offspring*) yang identik. Dalam hal ini hanya operasi mutasi yang mampu menghasilkan kromosom yang relatif baru dan merupakan cara untuk menghindari kromosom tertentu mendominasi populasi.

### C. Seleksi

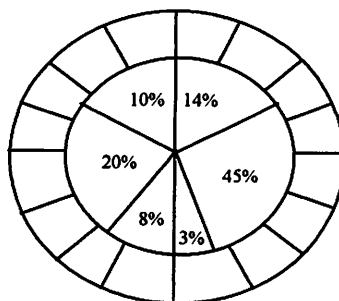
Pada Evolutionary Programming terdapat proses seleksi yaitu proses pemilihan kromosom yang akan di-kompetisikan -kan dengan kromosom dari individu lain. Masalah yang paling mendasar pada proses ini adalah bagaimana proses penyeleksiannya. Menurut teori evolusi Darwin proses seleksi individu adalah :” *individu terbaik akan tetap hidup dan menghasilkan keturunan*”. Pada proses seleksi ini dapat menggunakan banyak metode seperti *Roulette Wheel selection, Rank Selection, Elitism* dan sebagainya.

#### ○ Roulette Wheel Selection (Seleksi Roda Roulette)

Dimana setiap individual memiliki harga fitness sehingga didapatkan probabilitas individual ( $f(t) / \sum f(t)$ ) tersebut dicopykan pada populasi yang baru. Untuk individual yang memiliki probabilitas 20% untuk jumlah populasi 10 maka kemungkinan individual tersebut dapat terpilih sebanyak dua kali. Ilustrasi kerja dari operator ini digambarkan seperti pada gambar 3-2.

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 dalam populasi mulai dari 0 hingga total fitness lebih besar atau sama dengan nilai  $k$  lalu ambil kromosom tersebut.



**Gambar 3-2 :Roulette Wheel**



## ○ 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 meranking populasi dan kemudian tiap kromosom diberi fitness baru berdasarkan hasil ranking tersebut. Yang pertama akan mempunyai fitness 1, yang kedua akan mempunyai fitness 2, dan seterusnya sampai yang terakhir akan mempunyai fitness. Dengan demikian semua kromosom akan mempunyai peluang yang cukup besar untuk diseleksi.

### 3.1.3. Elitism

Selama membuat populasi baru dengan kompetisi dan mutasi, kemungkinan akan terjadi kehilangan kromosom terbaik (*best/few best*). Elitism adalah nama metode yang pertama kali meng-Copy-kan kromosom terbaik (*best/few best*) ke dalam populasi baru. Sisanya dikerjakan dengan cara yang biasa, yaitu melalui seleksi, mutasi dan kompetisi. Elitism dapat secara cepat meningkatkan performansi dari Evolutionary Programming karena elitism menghindarkan hilangnya solusi terbaik (*best/few best*) yang telah ditemukan.

### **3.2. Kompetisi ( persaingan )**

Fungsi dari kompetisi adalah menghasilkan kromosom keturunan dari kombinasi materi-materi gen dua kromosom induk. Cara kerjanya dengan membangkitkan sebuah nilai random  $r_k$  dimana  $k = 1, 2, \dots, \text{POPSIZE}$ . Probabilitas mutasi ( $P_m$ ) ditentukan dan digunakan untuk mengendalikan frekuensi kompetisi . Apabilanilai  $r_k < P_m$  maka kromosom ke- $k$  terpilih untuk mengalami kompetisi .

### **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  $k = 1, 2, \dots, \text{NVAR}$  (panjang kromosom). Probabilitas mutasi ( $P_m$ ) ditentukan dan digunakan untuk mengendalikan frekuensi mutasi. Apabila nilai random  $r_k < P_m$ , maka gen ke- $k$  kromosom tersebut terpilih untuk mengalami mutasi. Proses mutasi antar chomosom parent , biasanya disebut dengan proses flip yaitu dengan membalik nilai crhomosom yang satu dengan crhomosom lain . Ilustrasi kerja operator ini digambarkan seperti pada gambar 3-6.

Fungsi dari operator mutasi adalah untuk menghindari agar solusi permasalahan yang diperoleh bukan merupakan solusi optimum lokal. Seperti halnya pada operator mutasi , 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.

Perubahan parameter kontrol seperti besarnya populasi, probabilitas mutasi dan probabilitas kompetisi merupakan kemajuan dari pencarian di dalam metode Evolutionary programming . Variasi tersebut dapat memberikan beberapa hasil pencarian walaupun sangat sulit menentukan kombinasi yang optimal dari beberapa parameter pada tiap-tiap proses pencarian yang terjadi.

Proses optimasi dasar adalah mengkombinasi atau menggabungkan bagian populasi yang sehat untuk menghasilkan individu yang sehat sebagai kemajuan dari pencarian tersebut.

Langkah kerja di dalam penerapan metode Evolutionary Programming adalah :

1. Identifikasi parameter yang berpengaruh
2. Konversi parameter ke dalam substring
3. Menyusun substring tersebut (gen) menjadi string (kromosom, individu)
4. Melakukan ranking terhadap nilai kecocokan masing-masing individu
5. Melakukan reproduksi, mutasi dan kompetisi
6. Mengambil string yang terpilih menjadi parameter
7. Mengulangi proses tersebut sampai diperoleh suatu nilai parameter yang konvergen.

### 3.4. Adaptasi Evolutionary programming ke permasalahan Estimasi Efektif Efisiensi Motor Induksi Tiga-Phasa.

Evolutionary Programming adalah metode alternative yang bisa menentukan efisiensi motor induksi, sehingga diperoleh hasil yang mendekati nilai sebenarnya (hasil pengujian). Evolutionary programming pada mekanisme seleksi alam, individu dari sebuah populasi dikodekan secara real, populasi pertama dibangkitkan secara *random*. Generasi baru dibuat dengan mengaplikasikan 3 operator terhadap sebuah populasi yaitu : reproduksi, mutasi dan kompetisi dimana reproduksi adalah proses yang tergantung dari fungsi tujuan (*objective function*). Evolutionary programming menggunakan *Objective Function* yang didasarkan pada suatu kriteria kinerja untuk menentukan error.

#### 3.4.1. Pengkodean (Representasi)

Sebuah motor induksi tiga-fasa terdiri dari parameter yaitu  $R_2$ ,  $R_m$ ,  $X_1$  dan  $X_m$  mewakili satu (1) individu yang terdiri dari empat (4) string atau kromosom. Setiap kromosom ditentukan panjangnya 30, yang dibatasi nilai minimum dan nilai maksimum. Didalam string terdapat beberapa gen, dimana masing-masingnya adalah kode real.

Kromosom pertama adalah nilai  $R_2$  dimana dalam  $R_2$  ditentukan bahwa  $R_2$  minimum 2 dan maksimum 8. Maka arti dari pengkodean tersebut adalah<sup>[9]</sup> :

2	2,1	2,2	2,3	2,4	2,5	2,6	.....	.....	8
---	-----	-----	-----	-----	-----	-----	-------	-------	---

**Gambar 3 -8**  
**Pengkodean Untuk Parameter  $\kappa_2$**

Untuk mencari  $R_2$  dengan melalui persamaan sebagai berikut :

$$R_2 = R_{2\min} + \left[ \frac{(R_{2\max} - R_{2\min})x}{R_{2\max} - 1} \right] \dots\dots\dots (3.1)$$

Kromosom ke dua adalah nilai  $R_m$  dimana dalam  $R_m$  ditentukan bahwa  $R_m$  minimum 372 dan maksimum 900. Maka arti dari pengkodean tersebut adalah :

372	372.1	372.2	372.3	372.4	372.5	372.6	.....	.....	900
-----	-------	-------	-------	-------	-------	-------	-------	-------	-----

**Gambar-39**  
**Pengkodean Untuk Parameter  $R_m$**

Untuk mencari  $R_m$  dengan melalui persamaan sebagai berikut :

$$R_m = R_{m\min} + \left[ \frac{(R_{m\max} - R_{m\min})x}{R_{mmaks} - 1} \right] \dots\dots\dots (3.2)$$

Kromosom ke tiga adalah nilai  $X_1$  dimana dalam  $X_1$  ditentukan bahwa  $X_1$  minimum 2 dan  $X_1$  maksimum 8. Maka arti dari pengkodean tersebut adalah :

2	2.1	2.2	2.3	2.4	2.5	2.6	.....	.....	8
---	-----	-----	-----	-----	-----	-----	-------	-------	---

**Gambar 3 -10**  
**Pengkodean Untuk Parameter  $X_1$**

Untuk mencari  $X_1$  dengan melalui persamaan sebagai berikut :

$$X_1 = X_{1\min} + \left[ \frac{(X_{1\max} - X_{1\min})x}{X_{1maks} - 1} \right] \dots\dots\dots (3.3)$$

Kromosom ke empat adalah nilai  $X_m$  dimana dalam  $X_m$  ditentukan bahwa  $X_m$  minimum 40 dan  $X_m$  maksimum 200. Maka arti dari pengkodean tersebut adalah :

40	40.1	40.2	40.3	40.4	40.5	40.6	.....	.....	200
----	------	------	------	------	------	------	-------	-------	-----

**Gambar 3 -11**  
**Pengkodean Untuk Parameter  $X_m$**

Untuk mencari  $X_m$  dengan melalui persamaan sebagai berikut :

$$X_m = X_{m \min} + \left[ \frac{(X_{m \max} - X_{m \min})x}{X_{m \max} - 1} \right] \dots\dots\dots (3.4)$$

Setelah nilai  $R_2$ ,  $R_m$ ,  $X_1$ , dan  $X_m$  diperoleh maka dimasukkan ke persamaan-persamaan rangkaian ekuivalen. Dari hasil perhitungan dengan menggunakan persamaan rangkaian ekuivalen maka akan diperoleh nilai arus stator dan daya masukan yang kemudian dimasukkan kedalam fungsi evaluasi (*Objective Function*) sebagai berikut<sup>[1]</sup> :

$$F_{objective} = \sum_{i=1}^n \left| \frac{I_{i, hitung}}{I_{i, ukur}} - 1 \right|^2 + \sum_{i=1}^n \left| \frac{P_{input i, hitung}}{P_{input i, ukur}} - 1 \right|^2 \dots\dots\dots (3.5)$$

Selanjutnya dicari error sebagai berikut :

$$\varepsilon = F_{Objective} \dots\dots\dots (3.6)$$

dan dicari *fitnessnya* adalah :

$$Fitness = \frac{100}{100 + F_{objective}} \dots\dots\dots (3.7)$$

Perhitungan di atas untuk satu individu sedangkan untuk individu lainnya dilakukan dengan cara yang sama dengan catatan masing-masing kromosom nilainya random. Pada perhitungan tersebut merupakan proses dari satu poin pembebanan (*one operating point test/OPT*).

Untuk dua poin pembebanan (*two operating point test/TPT*) terdiri dari delapan (8) parameter, yaitu  $R_{2\_1}$ ,  $R_{m\_1}$ ,  $X_{1\_1}$ ,  $X_{m\_1}$  dan  $R_{2\_2}$ ,  $R_{m\_2}$ ,  $X_{1\_2}$ ,  $X_{m\_2}$  yang mewakili satu individu. Untuk empat parameter pertama mewakili nilai pembebanan pada poin pertama, sedangkan empat parameter berikutnya mewakili nilai pembebanan pada poin kedua; dimana nilai pembebanan yang berbeda untuk masing-masing poin. Sedangkan prosesnya dilakukan secara bersamaan untuk ke dua nilai pembebanan tersebut, dan urutan kerjanya seperti pada OPT.

Setelah semua individu dihitung *fitness*nya maka dicari prosentase fitness untuk masing-masing individu yaitu :

$$Prosentase = (Fitness_k / \text{sum fitness}) \cdot 100\% \dots\dots\dots (3.8)$$

Dimana :

$Fitness_k$  = fitness individu ke-k

Sum Fitness = jumlah fitness seluruh individu

$I_{i, \text{hitung}}$  = Hasil perhitungan dengan menggunakan persamaan (2.39)

$P_{\text{input } i, \text{hitung}}$  = Hasil perhitungan dengan menggunakan persamaan (2.44)

$I_{i, \text{ukur}}$  = Hasil pengukuran yang diperoleh dari test di lapangan.

$P_{\text{input } i, \text{ukur}}$  = Hasil pengukuran yang diperoleh dari test di lapangan.

Dimana:

$n = 1$  untuk metode OPT dan  $n = 2$  untuk metode TPT

Tujuan dari optimasi evolusi adalah untuk meminimalisasi kesalahan (Fungsi objektif/ $F_{\text{objective}}$ ) antara pengukuran dan perhitungan parameter.

### 3.4.2. Populasi Awal

Untuk sebuah motor induksi tiga fasa, populasi awal yang dibangkitkan secara random (acak). Dalam pembahasan kali ini populasi dibangkitkan sebanyak 50 populasi.

**Populasi pada ke dua metode tersebut adalah sebagai berikut :**

- Satu poin pembebanan (OPT). Untuk membuat satu individu misalnya untuk individu pertama dibuat 4 kromosom yang masing-masing adalah  $R_2$ ,  $R_m$ ,  $X_1$  dan  $X_m$ . Di dalam ke empat unsur tersebut dibangkitkan string real secara acak. Selanjutnya proses di atas diulang sampai  $n$  kali, untuk masing-masing nilainya dipilih secara acak.
- Dua poin pembebanan (TPT). Untuk membuat satu individu misalnya untuk individu pertama dibuat 8 kromosom yang masing-masing adalah  $R_{2\_1}$ ,  $R_{m\_1}$ ,  $X_{1\_1}$ ,  $X_{m\_1}$  dan  $R_{2\_2}$ ,  $R_{m\_2}$ ,  $X_{1\_2}$ ,  $X_{m\_2}$ . Di dalam ke delapan unsur tersebut dibangkitkan string real secara acak. Selanjutnya proses di atas diulang sampai  $n$  kali, untuk masing-masing nilainya dipilih secara acak.

### 3.4.3. Reproduksi, Mutasi dan kompetisi

Generasi baru dibuat dari hasil yang diperoleh dari generasi sebelumnya. Untuk setiap individu, pertama kita menghitung dan menormalisasikan nilai *objective function*. Semakin rendah nilai *objective function*-nya maka individu akan semakin baik. Reproduksi, mutasi dan kompetisi dilakukan secara bergantian. Kemudian 2 individu diambil secara acak, probabilitas pengambilan individu ini berhubungan langsung dengan nilai *objective function*-nya. Penarikan ini dilakukan dengan metode *biased roulette wheel* seperti pada gambar 3-2.

Proses kompetisi dapat dilakukan dengan probabilitas kompetisi ( $P_c$ ). Jika nilai yang dibangkitkan kurang dari  $P_c$  maka tidak perlu diadakan kompetisi . Pada proses ini, posisi untuk memotong keempat string ( $R_2$ ,  $R_m$ ,  $X_1$  dan  $X_m$ ) untuk kedua individu dipilih secara acak, begitupun pada proses TPT.



### 3.5. Objective Function

*Objective function* adalah parameter yang penting dalam Evolutionary Programming . Optimasi yang kita inginkan harus direpresentasikan secara matematis. *Objective function* yang buruk tidak dapat menghasilkan individu yang baik, dan tidak dapat mencapai optimasi yang kita inginkan.

#### 3.5.1. Objective Function Untuk Permasalahan Estimasi Efektif Efisiensi Efisiensi Motor Induksi Tiga Phasa<sup>[1]</sup>

Tujuan dari optimasi evolusi adalah untuk meminimalisasi kesalahan (*error*) antara pengukuran dan perhitungan parameter. Parameter input tersebut akan dimasukkan ke dalam *Objective function* yang dirumuskan sebagai berikut :

$$F_{objective} = \sum_{i=1}^n \left| \frac{I_{1i,hitung}}{I_{1i,ukur}} - 1 \right|^2 + \sum_{i=1}^n \left| \frac{P_{input\ i,hitung}}{P_{input\ i,ukur}} - 1 \right|^2 \dots\dots\dots (3.9)$$

Selanjutnya dicari error sebagai berikut :

$$\epsilon = F_{Objective} \dots\dots\dots (3.10)$$

dan dicari *fitness*nya adalah :

$$Fitness = \frac{100}{100 + F_{objective}} \dots\dots\dots (3.11)$$

Perhitungan di atas untuk satu individu sedangkan untuk individu lainnya dilakukan dengan cara yang sama dengan catatan masing-masing kromosom nilainya random. Proses ini berulang untuk setiap string baru (parameter motor yang baru) sampai menghasilkan 150 nilai fungsi *fitness*.

Hasil evaluasi pada proses Evolutionary Programming digunakan untuk mencari nilai *error* terkecil atau nilai *fitness* terbesar. Nilai *error* yang diperoleh digunakan untuk menentukan parameter motor induksi tersebut. Parameter motor tersebut didapat dari rangkaian ekivalen motor induksi.

### **3.6 Algoritma Program**

#### **3.6.1 Algoritma program Estimasi Efektif Efisiensi motor Induksi tiga phasa menggunakan *Evolutionary Programming* .**

Urutan langkah-langkah dalam program komputer yang menggunakan metode *Evolutionary Programming* adalah sebagai berikut :

1. Menentukan input parameter perhitungan motor induksi .
2. Menentukan parameter inputan *Evolutionary Programming* yang meliputi jumlah populasi , maksimum generasi , Probabilitas Kompetisi , probabilitas mutasi dan panjang kromosom tiap-tiap individu .
3. Generasi = 0 ; Populasi = 0
4. Inisialisasi awal Populasi
5. Melakukan proses statistik.
6. Melakukan proses seleksi.
7. Melakukan proses mutasi.
8. Apakah *offspring* sama dengan jumlah Populasi maksimum .
  - a. Jika “Ya” maka dilanjutkan ke langkah 9
  - b. Jika “Tidak” maka kembali ke langkah 10
9. Melakukan proses kompetisi.
10. Populasi = Populasi + 1 , kembali ke proses 5
11. Di pertanyakan , apakah sudah Konvergen ? jika tidak , kembali ke proses 6 (melakukan proses seleksi) .
12. Jika ya , di pertanyakan lagi apakah jumlah Generasi = Generasi Maksimum .
13. Jika tidak , Gen = Gen + 1 , kembali ke Proses 5
14. Jika ya , Stop

### 3.6.2 Algoritma Program Pemecahan Masalah Secara Umum.

1. Memasukkan input data berupa informasi data umum dari motor induksi yang merupakan general data (data umum) :
  - Daya Keluaran ( $W$ )
  - Tegangan nominal ( $V_{in}$ ) dalam Volt
  - Arus nominal ( $I$ ) dalam Ampere
  - Kecepatan putaran rotor nominal (rpm)
  - Resistansi stator ( $R_s$ )
2. Menentukan metode yang di gunakan : One operating test ( OPT ) atau two operating test ( TPT ) .
3. Bila menggunakan One operating One operating test ( OPT ) maka input datanya untuk satu point berupa :
  - a. Tegangan input ( $V_{in}$ ) dalam Volt .
  - b. Daya input ( $P_{in}$ ) dalam watt .
  - c. Power factor (pf)
  - d. Arus input ( $I_{in}$ ) dalam ampere
  - e. Kecepatan putaran motor (rpm0).
4. Proses evaluasi parameter dengan menggunakan Evolutionary Programming.
5. Dengan data parameter hasil Evolutionary Programming di gunakan untuk menentukan efisiensi motor induksi pada satu point pembebanan .
6. Cetak hasil .
7. Bila menggunakan Two Operating test ( TPT) , maka input datanya unuk point pertama dan point kedua sama seperti OPT , tetapi untuk masing-masing point dengan nilai pembebanan yang berbeda .
8. Proses evaluasi parameter motor dengan Evolutionary Programming .
9. Dengan data parameter hasil Evolutionary Programming digunakan untuk menentukan efisiensi motor induksi pada dua point pembebanan .
10. Cetak hasil .

### **3.6.3. Algoritma Program Fitness.**

1. Memasukkan input data parameter
2. Memasukkan datanya ke persamaan Rangkaian Ekuivalen
3. Memasukkan datanya ke persamaan Fungsi Objektif / $F_{\text{objective}}$
4. Memasukkan nilai fungsi objektif ke persamaan *fitness*
5. Cetak hasil

## BAB IV

### PENGUJIAN DAN ANALISIS

#### 4.1. Pengujian Parameter Motor Induksi Tiga Phasa.

##### 4.1.1. Alat-alat Yang Digunakan

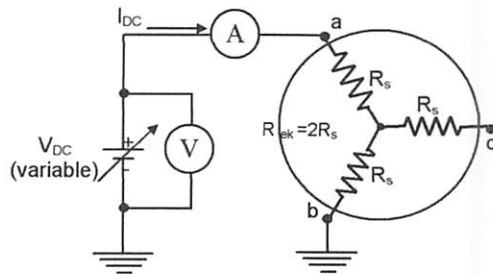
1. 1 Motor Induksi Tiga-Phasa 1,1 kW

Data papan-nama (*nameplate*) :

TEGANGAN	: 220/380 ( $\Delta$ / Y ) VOLT
ARUS	: 4.3/2.5 ( $\Delta$ / Y ) AMPERE
COS $\phi$	: 0,83
FREKUENSI	: 50 HERTZ
DAYA	: 1.1 kW
PUTARAN	: 2820 rpm
KUTUP	: 2

2. 1 Generator DC penguat bebas
3. 1 Wattmeter + 1 Ampmeter + 1 Voltmeter
4. 1 AC/DC Voltage Regulator
5. 1 Tachometer
6. 1 Torsimeter
7. Beban ( Lampu Pijar / 4 x 100 Watt )

### 4.1.2. Pengujian Arus Searah (DC Test)



**Gambar 4-1 : Pengujian Arus Searah (DC Test)**

Sumber : Electric Drive, An Integrative Systems Approach, EE 430.

#### 4.1.2.1. Data Pengujian DC Kumparan Stator

**Tabel 4-1**  
**Data Pengujian DC Kumparan Stator**

No	V <sub>DC</sub> (Volt)	I <sub>DC</sub> (Ampere)
1	6.27	0.5
2	12.02	1
3	18.1	1.5
4	24.2	2
5	29.9	2.5

#### 4.1.2.2. Analisa Data Pengujian Arus Searah

$$R_{ek(1)} = \frac{6.27}{0.5} = 12.54\Omega$$

$$R_{ek(2)} = \frac{12.02}{1.0} = 12.02\Omega$$

$$R_{ek(3)} = \frac{18.1}{1.5} = 12.07\Omega$$

$$R_{ek(4)} = \frac{24.2}{2.0} = 12.1\Omega$$

$$R_{ek(5)} = \frac{29.9}{2.5} = 11.96\Omega$$

$$R_{ek \text{ rata-rata}} = \frac{12.54 + 12.02 + 12.07 + 12.1 + 11.96}{5} = 12.138\Omega$$

$$R_S = \frac{12.138}{2} = 6.069 \text{ [ohm/Phasa]}$$

#### 4.1.3. Pengujian Berbeban ( Load Test )

Pada pengujian ini motor diberi beban 25%, 50%, 75% dan 100% dari beban penuh. Pengujian ini dimaksudkan untuk memperoleh nilai arus stator ( $I_1$ ), daya input ( $P_{input}$ ), kecepatan rotor ( $n_r$ ), dan torsi beban ( $T$ ). Sehingga nantinya didapat nilai efisiensi di setiap beban. Pemakaian torsi pada perhitungan untuk mencari nilai daya keluaran, sesuai dengan IEEE lewat rekomendasinya (*IEEE Std. 112-B*) dan standar dari negara Kanada (*CSA C390*).

### 4.1.3.1 DATA hasil PENGUJIAN BERBEBAN

Tabel 4-2  
Data Pengujian Berbeban

% Beban	V <sub>L-L</sub> (Volt)	I <sub>1</sub> (Ampere)	P <sub>Input</sub> (Watt)	Kec. Rotor (rpm)	Torsi Beban (Nm)
100	380	2.3	1051	2800	3.23
75	380	1.96	810	2917	2.39
50	380	1.62	560	2955	1.57
25	380	1.48	320	2984	0.71

### 4.1.3.1 Analisa Data Hasil Pengujian Berbeban

#### 1. Pembebanan 100%

$$s = \frac{n_s - n_r}{n_s} = \frac{3000 - 2800}{3000} = 0,06$$

$$pf = \frac{P_{Input}}{\sqrt{3}V_{L-L}I_1} = \frac{1051}{(1.732)(380)(2,3)} = 0,69$$

$$P_{Out} = T \cdot \frac{4\pi f}{p} (1-s) = 3,23 \cdot \frac{(4)(3,14)(50)}{2} (1-0,06) = 953.3668 \text{ Watt}$$

$$\eta_{100} = \frac{P_{Out}}{P_{Input}} \cdot 100\% = \frac{953.3668}{1051} \cdot 100\% = 90.7104\%$$

#### 2. Pembebanan 75%

$$s = \frac{n_s - n_r}{n_s} = \frac{3000 - 2917}{3000} = 0,028$$

$$pf = \frac{P_{Input}}{\sqrt{3}V_{L-L}I_1} = \frac{810}{(1.732)(380)(1,96)} = 0,62$$

$$P_{Out} = T \cdot \frac{4\pi f}{p} (1-s) = 2,39 \cdot \frac{(4)(3,14)(50)}{2} (1-0,028) = 729.4471 \text{ Watt}$$

$$\eta_{75} = \frac{P_{Out}}{P_{Input}} \cdot 100\% = \frac{729.4471}{810} \cdot 100\% = 90,0552\%$$



### 3. Beban 50%

$$s = \frac{n_s - n_r}{n_s} = \frac{3000 - 2955}{3000} = 0,015$$

$$pf = \frac{P_{Input}}{\sqrt{3}V_{L-L}I_1} = \frac{560}{(1.732)(380)(1,62)} = 0,52$$

$$P_{Out} = T \cdot \frac{4\pi f}{p} (1-s) = 1,57 \cdot \frac{(4)(3,14)(50)}{2} (1-0,015) = 485,5853 \text{ Watt}$$

$$\eta_{50} = \frac{P_{Out}}{P_{Input}} \cdot 100\% = \frac{485,5853}{560} \cdot 100\% = 86,7117\%$$

### 4. Beban 25%

$$s = \frac{n_s - n_r}{n_s} = \frac{3000 - 2984}{3000} = 0,005$$

$$pf = \frac{P_{Input}}{\sqrt{3}V_{L-L}I_1} = \frac{320}{(1.732)(380)(1,48)} = 0,33$$

$$P_{Out} = T \cdot \frac{4\pi f}{p} (1-s) = 0,71 \cdot \frac{(4)(3,14)(50)}{2} (1-0,005) = 221,8253 \text{ Watt}$$

$$\eta_{25} = \frac{P_{Out}}{P_{Input}} \cdot 100\% = \frac{221,8253}{320} \cdot 100\% = 69,3204\%$$

**Tabel 4-3**  
**Data Hasil Perhitungan Pengujian Berbeban**

<b>% Pembebanan</b>	<b>Slip (s)</b>	<b>Faktor Daya (pf)</b>	<b>Daya Output (Watt)</b>	<b>Effisiensi (%)</b>
<b>100%</b>	0.06	0.69	953.3668	90.71
<b>75%</b>	0.028	0.62	729.4471	90.055
<b>50%</b>	0.015	0.52	485.5853	86.712
<b>25%</b>	0.005	0.33	221.8253	69.32

#### 4.2. Data Inputan Untuk Program Evolutionary Programming .

Tabel 4-4  
Input Data Untuk Program Evolutionary Programming .

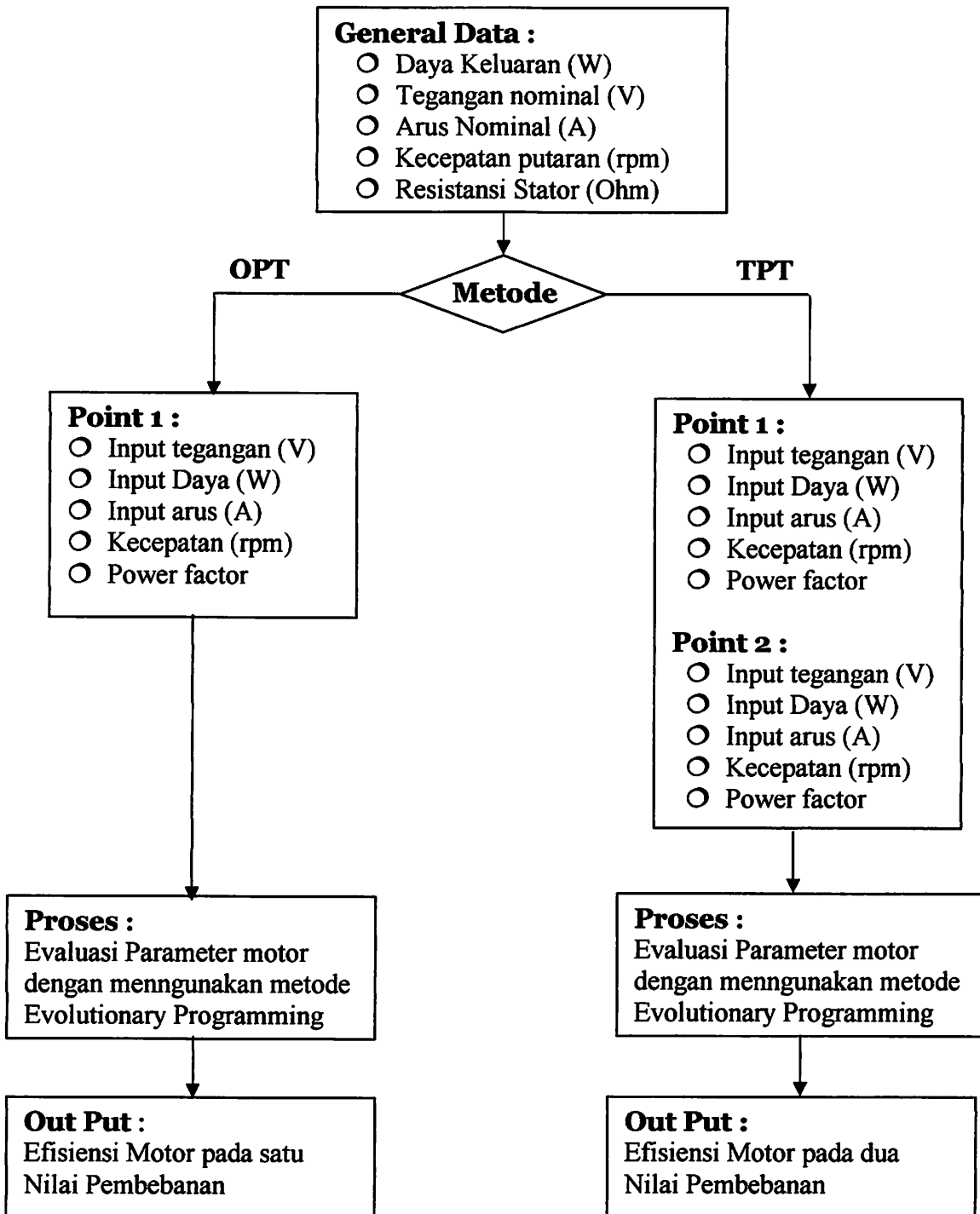
<b>%BEBAN</b>	<b>Tegangan (V)</b>	<b>Arus (A)</b>	<b>Pinput (Watt)</b>	<b>Kec.rotor (rpm)</b>	<b>Power Factor</b>
100%	380	2.3	1051	2800	0.69
75%	380	1.96	810	2917	0.62
50%	380	1.62	5460	2955	0.52
25%	380	1.48	320	2984	0.33

#### 4.3. Program Komputer untuk Metode Evolutionary Programming .

Untuk pemecahan masalah pada metode Evolutionary Programming digunakan bantuan komputer. Program ini sangat berguna untuk mempercepat proses perhitungan yang membutuhkan ketelitian tinggi dan proses pemilihan, seleksi dan statistik pada metode Evolutionary Programming serta tidak dimungkinkannya melakukan perhitungan secara manual. Program komputer ini menggunakan bahasa pemrograman Borland Delphi versi 7.

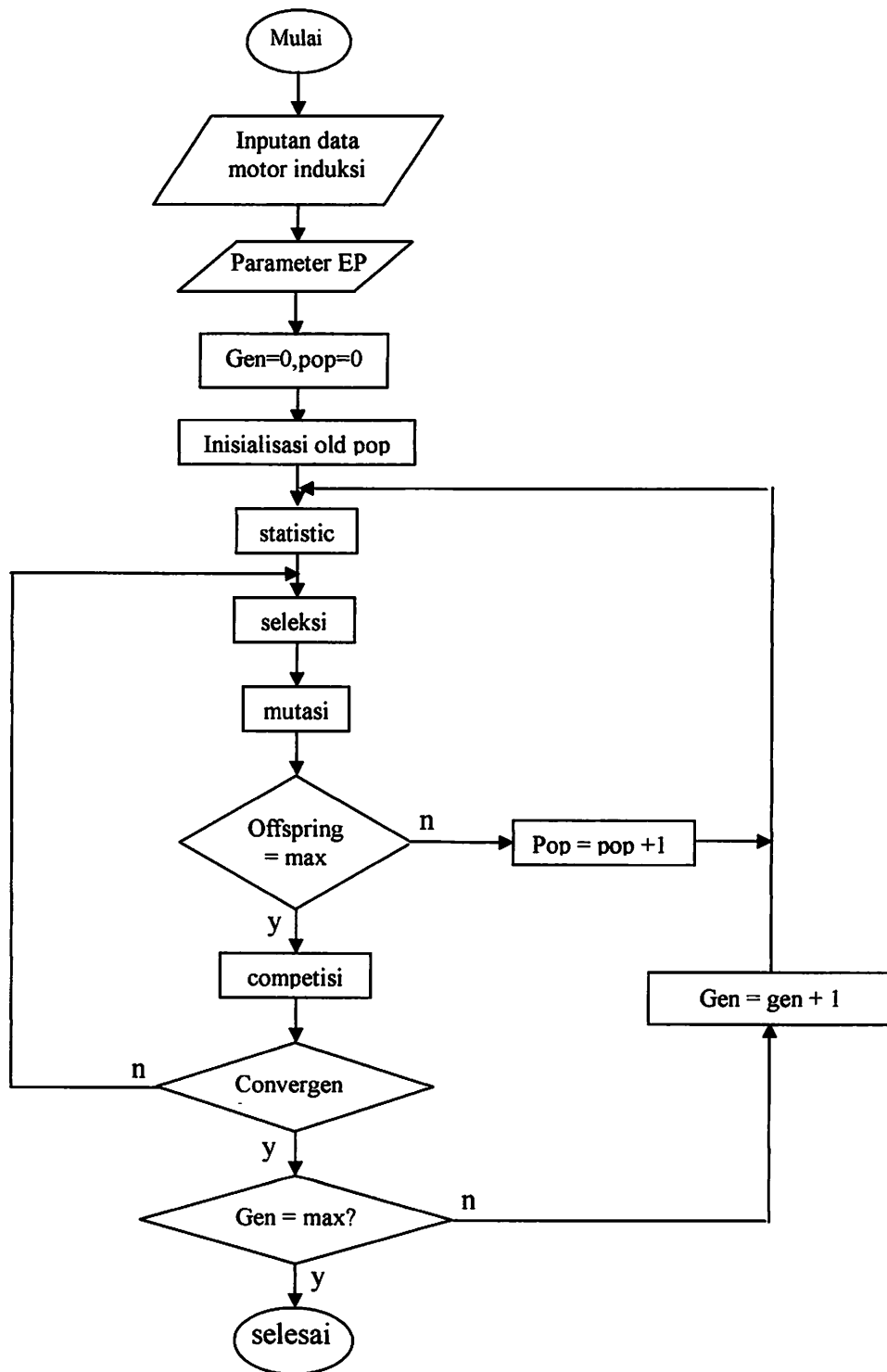
#### 4.4. Flowchart Algoritma Program

##### 4.4.1. Flowchart Pemecahan Masalah Secara Umum



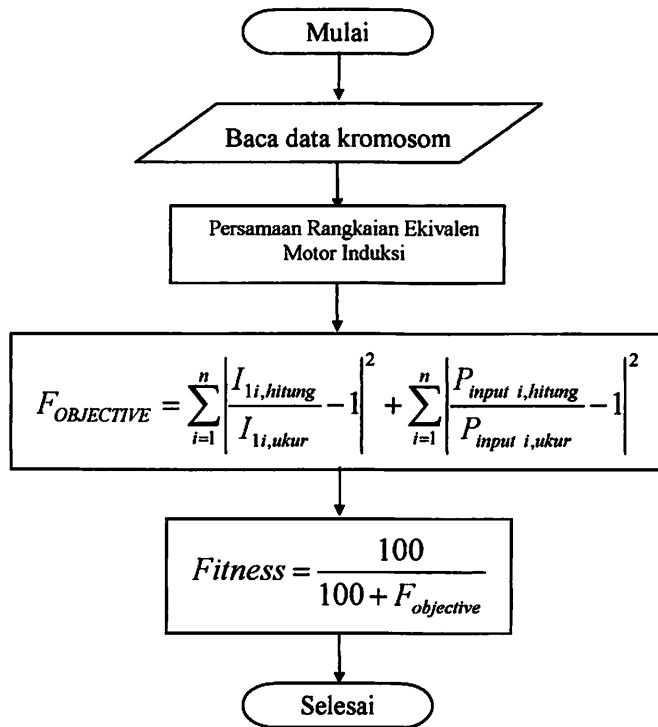
Gambar 4.2 Flowchart Pemecahan Masalah Secara Umum<sup>[1]</sup>

**4.4.2. Flow Chart Estimasi Efektif Efisiensi efisiensi menggunakan metode *Evolutionary Programming*..**



**Gambar 4-3 : Flowchart Program Evolutionary Programming .**

#### 4.4.3. Flowchart Program Fitness.



**Gambar 4-4**  
**Flowchart Program Fitness**

## 4.5. Tampilan Program

Program dalam skripsi ini dijalankan dengan menggunakan bahasa pemrograman Borland Delphi versi 7.0 dan diaplikasikan pada komputer berprosesor Pentium III 8000MHz, dengan memori 256 Mb.

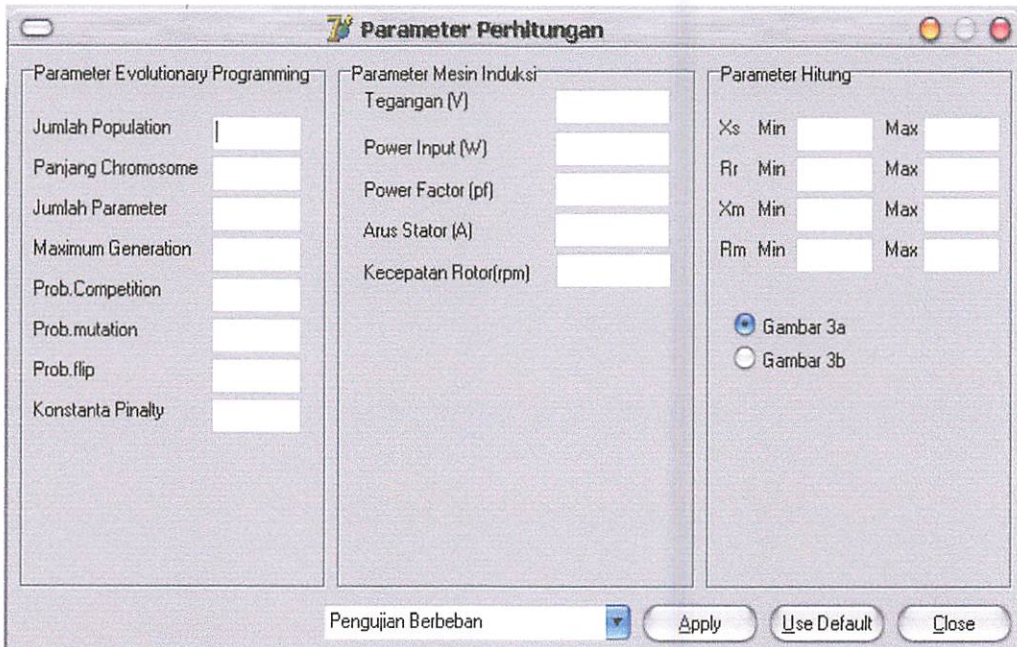
### 4.5.1. TAMPILAN UTAMA DARI PROGRAM

#### 4.5.1.1 DATA INPUTAN SATU POINT PEMBEBANAN (OPT) PADA PEMBEBANAN 100%



**Gambar 4-5**  
**Tampilan Depan Program**

1. Tekan Tombol Run OPT untuk membuka file yang sudah tersimpan



**Gambar 4-6**  
**Tampilan Inputan Data Pada Beban 100%**

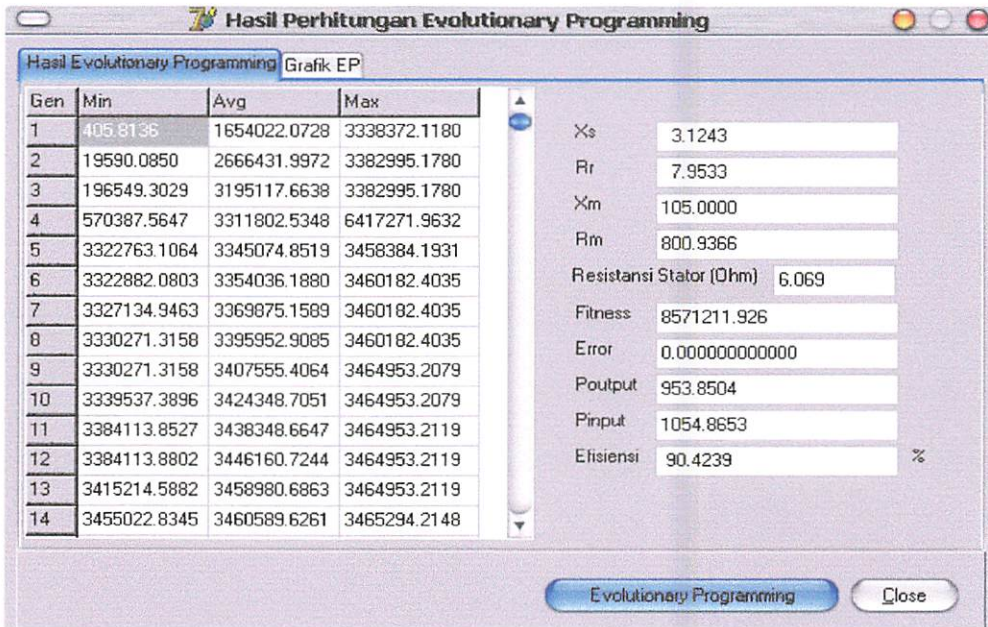
2. Tekan tombol pengujian berbeban untuk menampilkan data 100% .



**Gambar 4-7**  
**Tampilan Data Yang Sudah Tersimpan Pada Beban 100%**

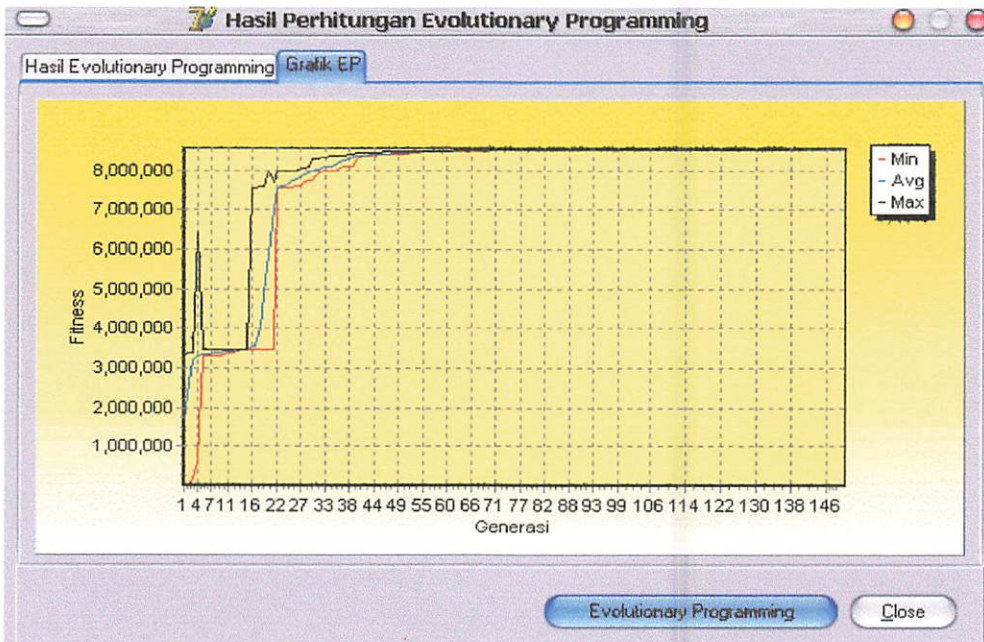


3. Kemudian pilih tombol Apply sehingga tampilan seperti berikut :



Gambar 4-8  
Tampilan Hasil Evolutionary Programming Pada Beban 100%

5. Jika ingin melihat grafik tekan tombol Grafik EP :



gambar 4-9  
Tampilan Grafik Fitness Function-Generasi Pada Beban 100%



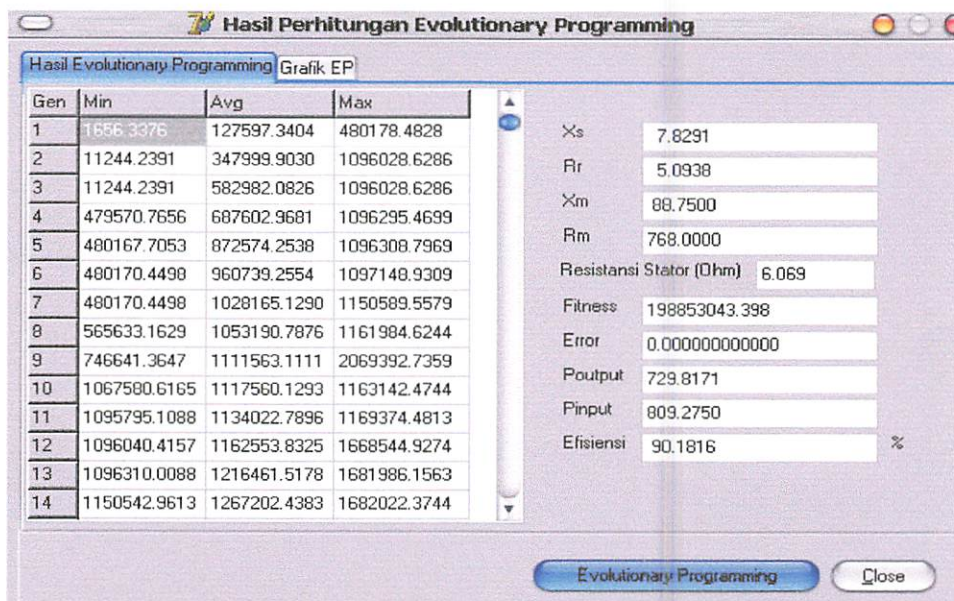
### 4.5.1.2 DATA INPUTAN PADA PEMBEBANAN 75%

#### 1. Data Inputan Evolutionary Programming



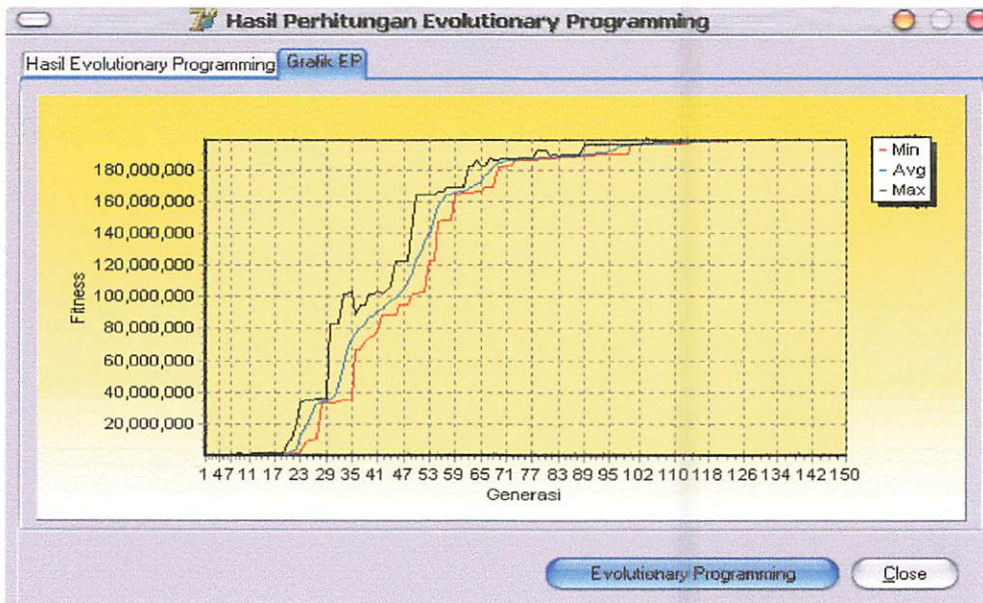
Gambar 4-10  
Data Inputan Evolutionary Programming Pada Beban 75%

#### 2. Hasil Program Evolutionary Programming



Gambar 4-11  
Tampilan Hasil Evolutionary Programming Pada Beban 75%

### 3. Grafik Proses Evolutionary Programming

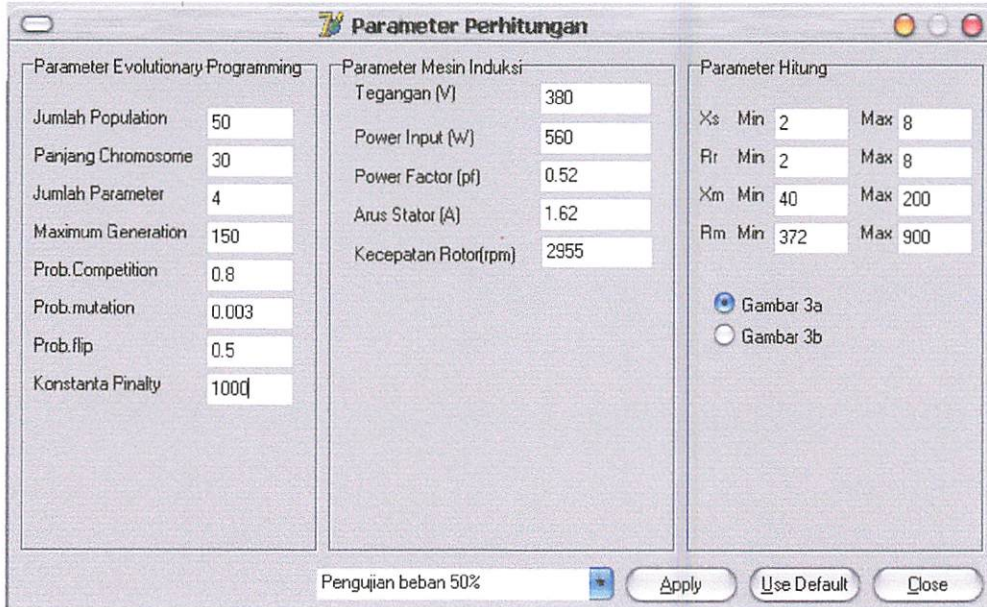


**Gambar 4-12**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 75%**



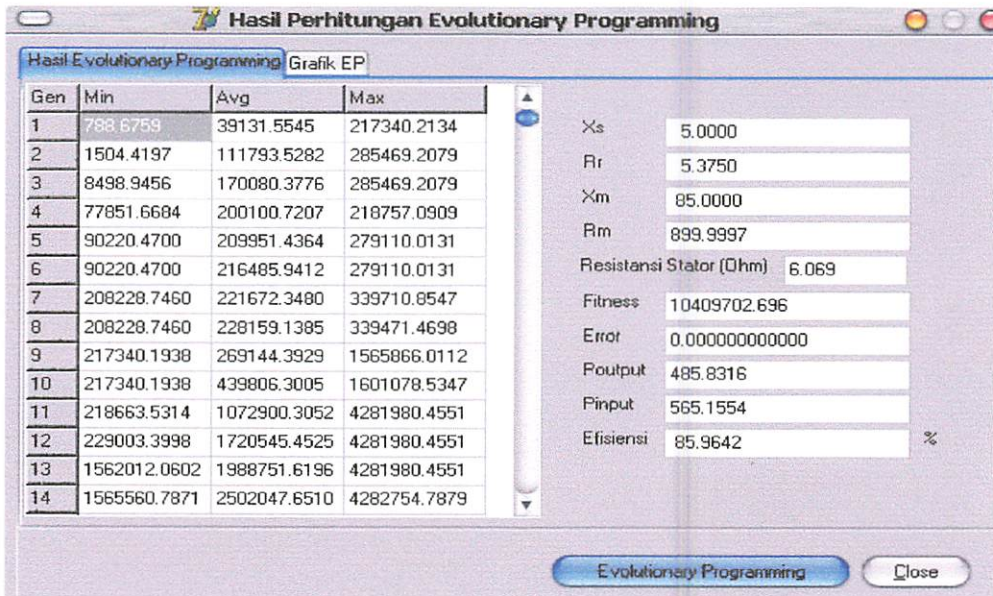
### 4.5.1.3 DATA INPUTAN PADA PEMBEBANAN 50%

#### 1. Data Inputan Evolutionary Programming .



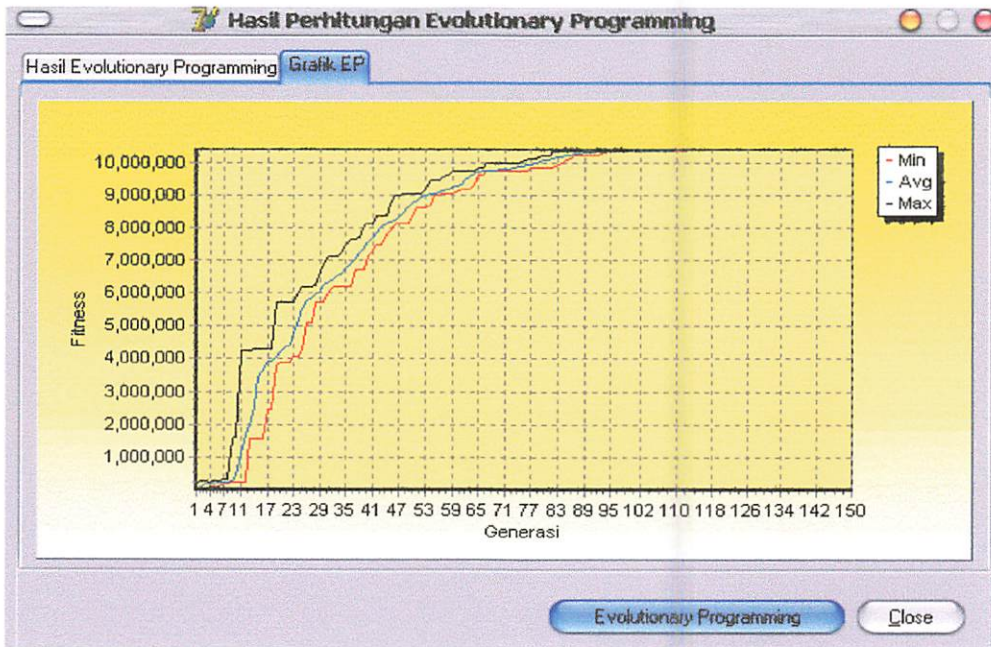
Gambar 4-13  
Data Inputan Evolutionary Programming Pada Beban 50%

#### 2. Hasil Program Evolutionary Programming .



Gambar 4-14  
Tampilan Hasil Evolutionary Programming Pada Beban 50%

### 3. Grafik Proses Evolutionary Programming

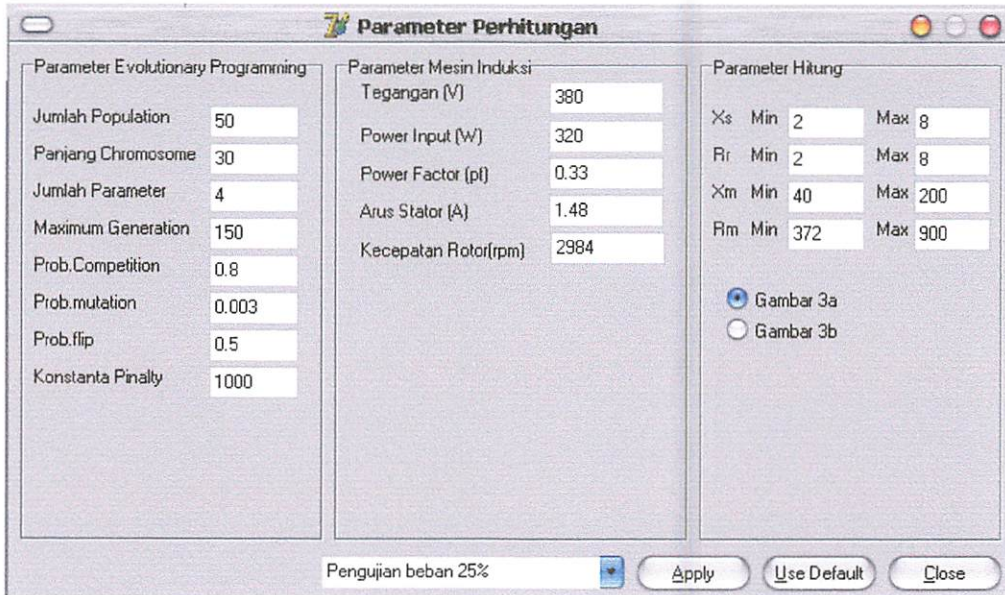


**Gambar 4-15**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 50%**



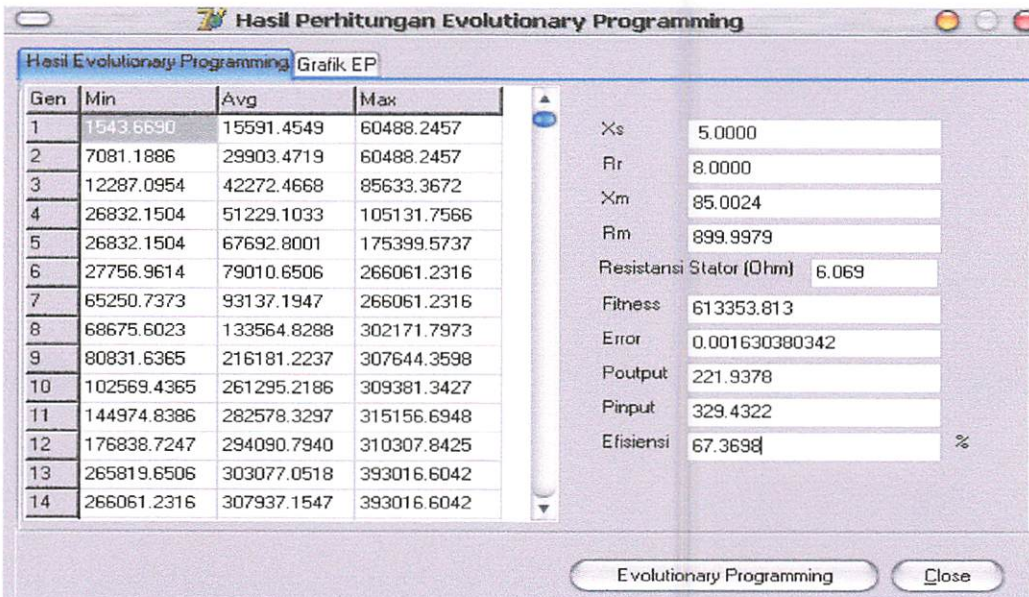
### 4.5.1.4 DATA INPUTAN PADA PEMBEBANAN 25%

#### 1. Data Inputan Evolutionary Programming



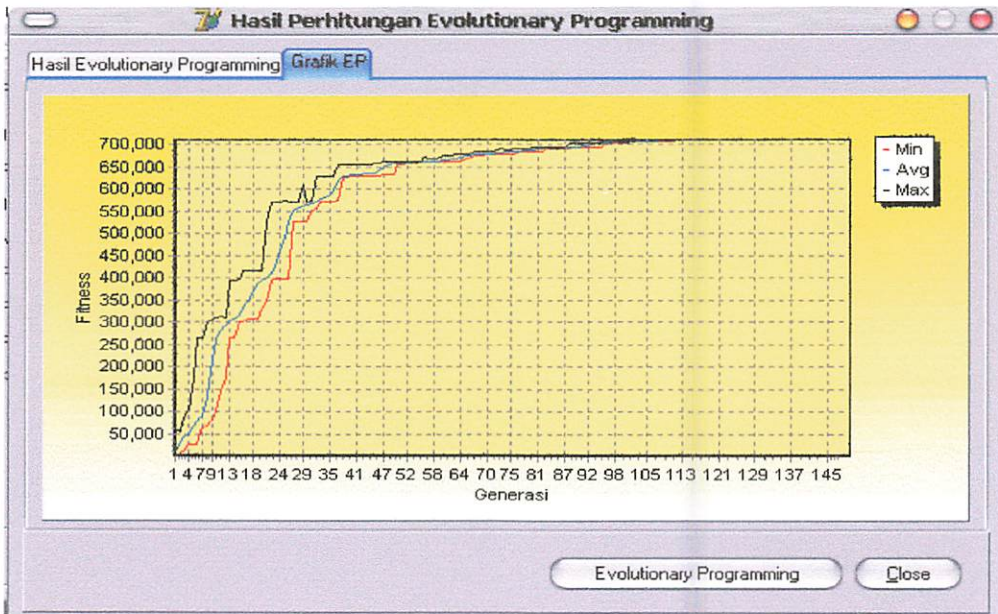
Gambar 4-16  
Data Inputan Evolutionary Programming Pada Beban 25%

#### 2. Hasil Program Evolutionary Programming



Gambar 4-17 : Tampilan Hasil Evolutionary Programming Pada Beban 25%

### 3. Grafik Proses Evolutionary Programming



**Gambar 4-18 :**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 25%**



## 4.5.2. DUA POINT PEMBEBANAN (TPT)

### 4.5.2.1 DATA INPUTAN PADA PEMBEBANAN 100% & 75%

#### 1. Data Inputan Evolutionary Programming .

The screenshot shows a software window titled "Form Set Parameter" with three main sections of input fields:

- Parameter Evolutionary Programming:**
  - Jumlah Population: 50
  - Panjang Chromosome: 30
  - Jumlah Parameter: 4
  - Maximum Generation: 150
  - Prob.Competition: 0.8
  - Prob.Mutation: 0.003
  - Prob.Flip: 0.5
  - Konstanta Pinalty: 1000
- Parameter Mesin Induksi Poin 1:**
  - Tegangan (V): 380
  - Power Input (W): 1051
  - Power Factor (pf): 0.69
  - Arus Stator (A): 2.3
  - Kec. Putar Rotor(rpm): 2800
- Parameter Mesin Induksi Poin 2:**
  - Tegangan (V): 380
  - Power Input (W): 810
  - Power Factor (pf): 0.62
  - Arus Stator (A): 1.96
  - Kec. Putar Rotor(rpm): 2917

At the bottom, there is a dropdown menu set to "Pengujian beban 100% - 75%", and buttons for "Apply" and "Use Default".

**Parameter Hitung:**

Xs	Min 2	Max 8
Rr	Min 2	Max 8
Xm	Min 40	Max 200
Rm	Min 372	Max 900

Radio buttons for "Gambar 3a" (selected) and "Gambar 3b" are also present.

Gambar 4-19  
Data Inputan Evolutionary Programming Pada Beban 100% dan 75%

#### 2. Hasil Program Evolutionary Programming

The screenshot shows a software window titled "Form Hasil" with two main sections:

- Hasil Evolutionary Programming Grafik EP:** A table showing the results for 17 generations.
- Parameter Hasil:** Final values for various parameters.

Gen	Min	Average	Max
1	405.8136	1654022.0728	3338372.1180
2	19590.0850	2666431.9972	3382995.1780
3	196549.3029	3195117.6638	3382995.1780
4	570387.5647	3311802.5348	6417271.9632
5	3322763.1064	3345074.8519	3458384.1931
6	3322882.0803	3354036.1880	3460182.4035
7	3327134.9463	3369875.1589	3460182.4035
8	3330271.3158	3395952.9085	3460182.4035
9	3330271.3158	3407555.4064	3464953.2079
10	3339537.3896	3424348.7051	3464953.2079
11	3384113.8527	3438348.6647	3464953.2119
12	3384113.8802	3446160.7244	3464953.2119
13	3415214.5882	3458980.6863	3464953.2119
14	3455022.8345	3460589.6261	3465294.2148
15	3455022.8345	3461655.1655	3465287.8463
16	3460179.8811	3544247.1559	7551852.9685
17	3460182.3882	3545286.8385	7551853.0471

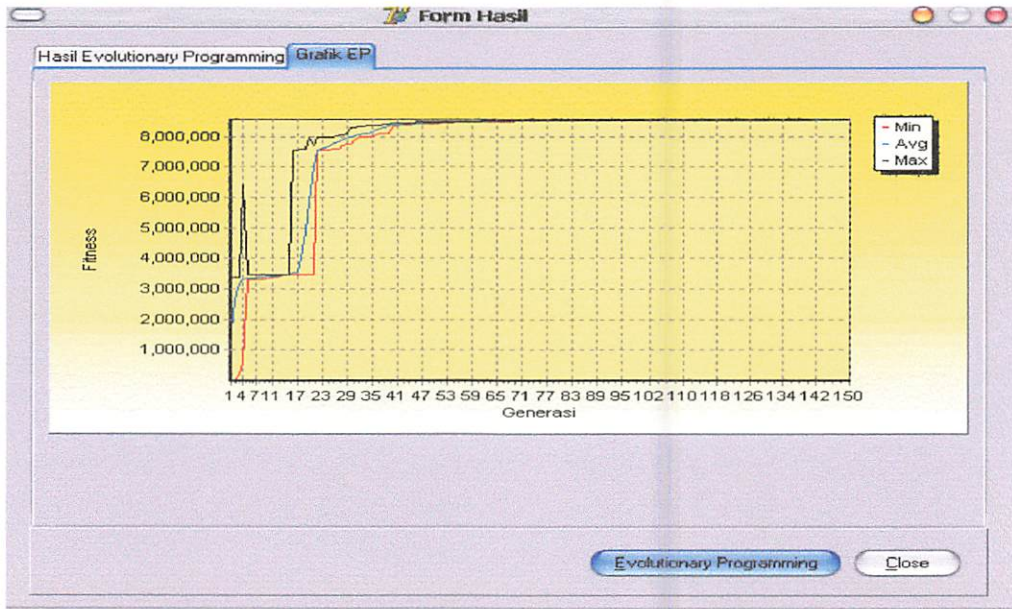
**Parameter Hasil:**

- Xs: 3.1243
- Rr: 7.9533
- Xm: 105.0000
- Rm: 800.9366
- R. Stator1 (Ohm): 6.069
- R. Stator2 (Ohm): 6.069
- Fitness: 8571211.926
- Error: 0.000000000000
- Poutput 1: 953.8504
- Pinput 1: 1054.8653
- Efisiensi 1: 90.4239 %
- Poutput 2: 729.8171
- Pinput 2: 809.2750
- Efisiensi 2: 90.1816 %

Buttons for "Evolutionary Programming" and "Close" are at the bottom.

Gambar 4-20  
Tampilan Hasil Evolutionary Programming Pada Beban 100% dan 75%

### 3. Grafik Proses Evolutionary Programming



**Gambar 4-21**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 100% dan 75%**



## 4.5.2.2 DATA INPUTAN PADA PEMBEBANAN 100% & 50%

### 1. Data Inputan Evolutionary Programming

The screenshot shows a software window titled "Form Set Parameter" with three main sections:

- Parameter Evolutionary Programming:**
  - Jumlah Population: 50
  - Penjang Chromosome: 30
  - Jumlah Parameter: 4
  - Maximum Generation: 150
  - Prob.Competition: 0.8
  - Prob.Mutation: 0.003
  - Prob.Flip: 0.5
  - Konstanta Pinsky: 1000
- Parameter Mesin Induksi Poin 1:**
  - Tegangan (V): 380
  - Power Input (W): 1051
  - Power Factor (pf): 0.69
  - Arus Stator (A): 2.3
  - Kec. Putar Rotor(rpm): 2800
- Parameter Mesin Induksi Poin 2:**
  - Tegangan (V): 380
  - Power Input (W): 560
  - Power Factor (pf): 0.52
  - Arus Stator (A): 1.62
  - Kec. Putar Rotor(rpm): 2955

At the bottom, there is a dropdown menu set to "Pengujian beban 100% - 50%", and "Apply" and "Use Default" buttons.

**Parameter Hitung:**

- Xs Min: 2, Max: 8
- Rr Min: 2, Max: 8
- Xm Min: 40, Max: 200
- Rm Min: 372, Max: 900

Radio buttons for "Gambar 3a" (selected) and "Gambar 3b" are also present.

Gambar 4-22  
Data Inputan Evolutionary Programming Pada Beban 100% dan 50%

### 2. Hasil Program Evolutionary Programming

The screenshot shows a software window titled "Form Hasil" with the following content:

**Hasil Evolutionary Programming Grafik EP**

Gen	Min	Average	Max
1	405.8136	1654022.0728	3338372.1180
2	19590.0850	2666431.9972	3382995.1780
3	196549.3029	3195117.6638	3382995.1780
4	570387.5647	3311802.5348	6417271.9632
5	3322763.1064	3345074.8519	3458384.1931
6	3322882.0803	3354036.1880	3460182.4035
7	3327134.9463	3369875.1589	3460182.4035
8	3330271.3158	3395952.9085	3460182.4035
9	3330271.3158	3407555.4064	3464953.2079
10	3339537.3896	3424348.7051	3464953.2079
11	3384113.8527	3438348.6647	3464953.2119
12	3384113.8802	3446160.7244	3464953.2119
13	3415214.5882	3458980.6863	3464953.2119
14	3455022.8345	3460589.6261	3465294.2148
15	3455022.8345	3461655.1655	3465297.8463
16	3460179.8811	3544247.1559	7551852.9685
17	3460182.3882	3545286.8385	7551853.0471

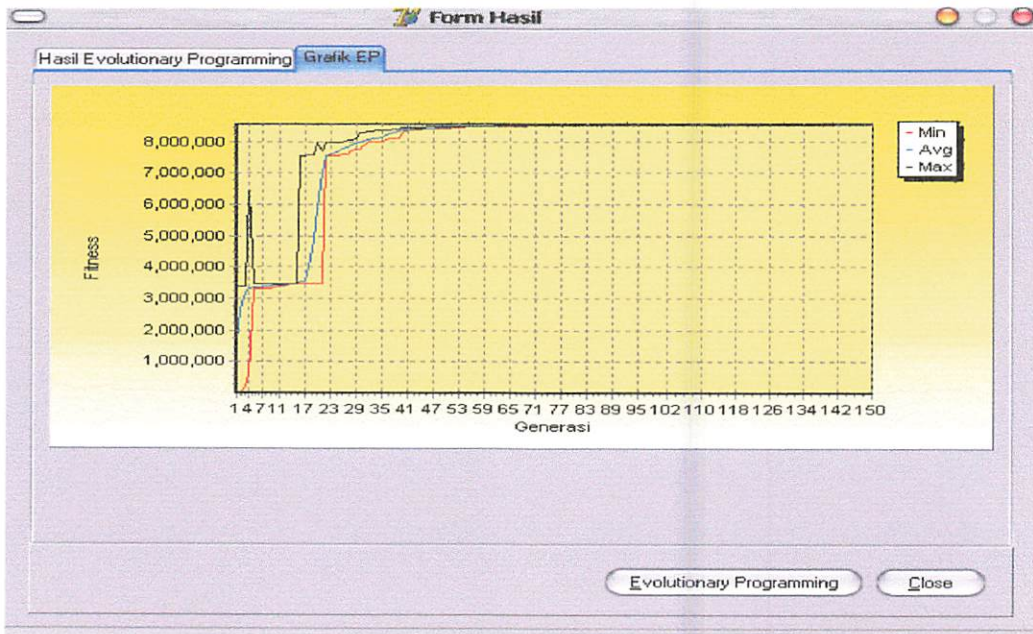
**Final Parameters:**

- Xs: 3.1243
- Rr: 7.9533
- Xm: 105.0000
- Rm: 800.9366
- R.Stator1 (Ohm): 6.069
- R.Stator2 (Ohm): 6.069
- Fitness: 8571211.926
- Error: 0.000000000000
- Pouput 1: 953.8504
- Pinput 1: 1054.8653
- Efisiensi 1: 90.4239 %
- Pouput 2: 485.8316
- Pinput 2: 561.7382
- Efisiensi 2: 86.4872 %

Buttons for "Evolutionary Programming" and "Close" are at the bottom.

Gambar 4-23  
Tampilan Hasil Evolutionary Programming Pada Beban 100% dan 50%

### 3. Grafik Proses Evolutionary Programming .



**Gambar 4-24**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 100% dan 50%**



### 4.5.2.3 DATA INPUTAN PADA PEMBEBANAN 100% & 25%

#### 1. Data Inputan Evolutionary Programming

The screenshot shows a software window titled "Form Set Parameter" with three main sections:

- Parameter Evolutionary Programming:**
  - Jumlah Population: 50
  - Panjang Chromosome: 30
  - Jumlah Parameter: 4
  - Maximum Generation: 150
  - Prob.Competition: 0.8
  - Prob.Mutation: 0.003
  - Prob.Flip: 0.5
  - Konstanta Pinalty: 1000
- Parameter Mesin Induksi Poin 1:**
  - Tegangan (V): 380
  - Power Input (W): 1051
  - Power Factor (pf): 0.69
  - Arus Stator (A): 2.3
  - Kec. Putar Rotor(rpm): 2800
- Parameter Mesin Induksi Poin 2:**
  - Tegangan (V): 380
  - Power Input (W): 320
  - Power Factor (pf): 0.33
  - Arus Stator (A): 1.48
  - Kec. Putar Rotor(rpm): 2984

At the bottom, there is a dropdown menu for "Pengujian beban 100% - 25%", and "Apply" and "Use Default" buttons.

**Parameter Hitung:**

Xs	Min 2	Max 8
Rr	Min 2	Max 8
Xm	Min 40	Max 200
Rm	Min 372	Max 900

Radio buttons for "Gambar 3a" (selected) and "Gambar 3b" are also present.

Gambar 4-25  
Data Inputan Evolutionary Programming Pada Beban 100% dan 25%

#### 2. Hasil Program Evolutionary Programming

The screenshot shows a software window titled "Form Hasil" with a table of evolutionary programming results and a list of final parameters.

**Hasil Evolutionary Programming Grafik EP**

Gen	Min	Average	Max
1	405.8136	1654022.0728	3338372.1180
2	19590.0850	2666431.9972	3382995.1780
3	196549.3029	3195117.6638	3382995.1780
4	570387.5647	3311802.5348	6417271.9632
5	3322763.1064	3345074.8519	3458384.1931
6	3322882.0803	3354036.1880	3460182.4035
7	3327134.9463	3369875.1589	3460182.4035
8	3330271.3158	3395952.9085	3460182.4035
9	3330271.3158	3407555.4064	3464953.2079
10	3339537.3896	3424348.7051	3464953.2079
11	3384113.8527	3438348.6647	3464953.2119
12	3384113.8802	3446160.7244	3464953.2119
13	3415214.5882	3458980.6863	3464953.2119
14	3455022.8345	3460589.6261	3465294.2148
15	3455022.8345	3461655.1655	3465287.8463
16	3460179.8811	3544247.1559	7551852.9685
17	3460182.3882	3545286.8385	7551853.0471

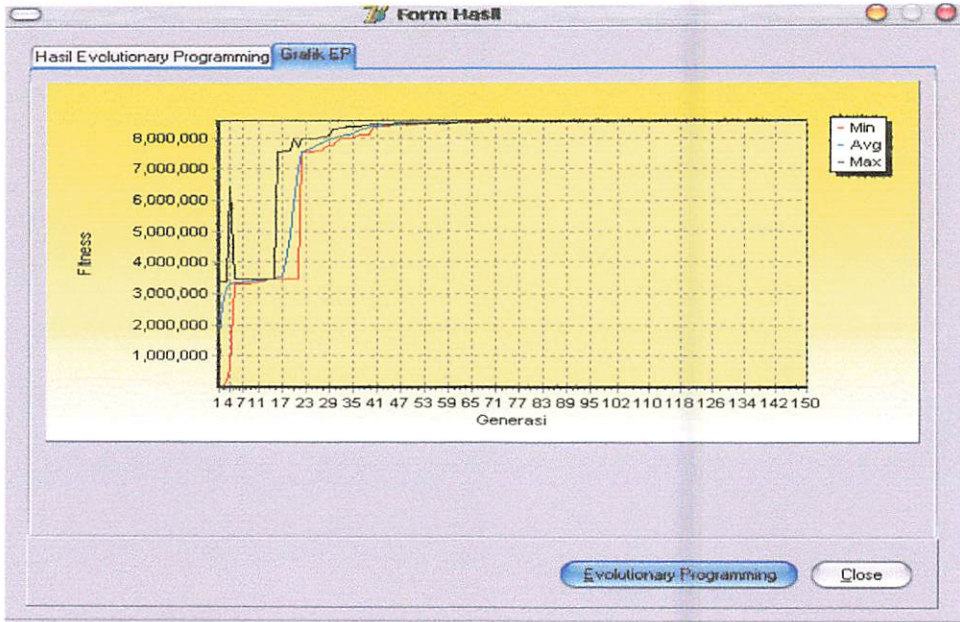
**Final Parameters:**

- Xs: 3.1243
- Rr: 7.9533
- Xm: 105.0000
- Rm: 800.9366
- R. Stator1 (Ohm): 6.069
- R. Stator2 (Ohm): 6.069
- Fitness: 8571211.926
- Error: 0.000000000000
- Poutput 1: 953.8504
- Pinput 1: 1054.8653
- Efisiensi 1: 90.4239 %
- Poutput 2: 221.9378
- Pinput 2: 324.5089
- Efisiensi 2: 68.3919 %

Buttons for "Evolutionary Programming" and "Close" are at the bottom.

Gambar 4-26  
Tampilan Hasil Evolutionary Programming Pada Beban 100% dan 25%

### 3. Grafik Proses Evolutionary Programming



**Gambar 4-27**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 100% dan 25%**

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melakukan analisis dari uji coba program dengan menggunakan metode *Evolutionary Programming* pada penentuan efisiensi motor induksi 3 $\phi$ , maka dapat ditarik kesimpulan sebagai berikut :

1. Metode penentuan efisiensi pada motor induksi dengan menggunakan metode *Evolutionary Programming* ternyata mempunyai akurasi yang hampir sama dengan hasil pengujian. Dari hasil pengujian dan test program *Evolutionary Programming* pada 25%, 50%, 75% dan 100% dari beban penuh didapat efisiensi dengan prosentase sebesar :

	25%	50%	75%	100%
<b>Pengujian</b>	<b>69.3204</b>	<b>86.7117</b>	<b>90.0552</b>	<b>90.71</b>
<b>OPT</b>	<b>67.3698</b>	<b>85.9642</b>	<b>90.1816</b>	<b>90.424</b>
<b>TPT</b>	<b>68.0512</b>	<b>86.5697</b>	<b>90.1254</b>	<b>90.424</b>

2. Dari hasil simulasi program komputer diperoleh selisih antara efisiensi pengujian dan hasil test program *Evolutionary Programming* untuk pembebanan 25%, 50%, 75% dan 100% dengan prosentase sebesar :

<b>METODE</b>	<b>25%</b>	<b>50%</b>	<b>75%</b>	<b>100%</b>
<b>OPT</b>	<b>1.9506</b>	<b>0.7475</b>	<b>0.1264</b>	<b>0.2865</b>
<b>TPT</b>	<b>1.2692</b>	<b>0.142</b>	<b>0.0702</b>	<b>0.2865</b>

3. Dari hasil simulasi dengan metode *Evolutionary Programming* untuk OPT dan TPT, tingkat akurasi dalam penilaian efisiensi untuk :
  - Pembebanan 25%, 50%, 75% dan 100% ; TPT lebih mendekati nilai sebenarnya ( hasil pengujian).
  - Pada pembebanan 100%; OPT dan TPT mempunyai akurasi yang sama dalam penilaian efisiensi motor induksi.
4. Metode *Evolutionary Programming* merupakan suatu metode alternatif untuk penentuan efisiensi motor induksi tiga fasa tanpa melakukan pemutusan beban . Dengan menerapkan metode ini hanya memerlukan data pengujian berbeban (*Load Test*), pengujian arus searah (*DC Test* untuk mengetahui resistansi stator ) dan papan-nama (*nameplate*) dari motor induksi.

## 5.2. Saran

Pengembangan lebih lanjut dari metode *Evolutionary Programming* untuk memperoleh nilai parameter ( $R_2$ ,  $R_m$ ,  $X_1$  dan  $X_m$ ) yang optimal, agar dimasukkan fungsi obyektif (*Objective Function* ) untuk nilai daya keluaran ( $P_{out}$ ) dari hasil perhitungan dan pengujian.

## DAFTAR PUSTAKA

- [1] T. Phumiphak and C. Chat-uthai, “ **Effective Estimation of Induction Motor Field Efficiency Based on On-Site Measurements**”, Proceedings of International Conference on Power System Technologi, Power Con, Kunming, China, 2003.
- [2] P. Pillay, V. Levin, P. Otaduy, and J. Kueck, “**In-situ Induction Motor Efficiency Determination Using The Genetic Algorithm**”, IEEE Transaction on EnergyConversion, Vol.13,pp.326-333, Dec.1998.
- [3] A. E. Fitzgerald, Charles Kingsley, Jr., Stephen D. Umans, “ **Electric Machinery**”, McGraw-Hill International, Fourth Editio
- [4] Mitsuo Gen, Runwei Cheng, “ **Evolutionary Programming and Engineering Design**”, (john Wiley & Son, Inc., 1997), p.7.
- [5] C. Indulkar@ieee. Org, “**Induction Motor Problems with solution**”, chapter 3.



#### 4.5.2.4 DATA INPUTAN PADA PEMBEBANAN 75% & 50%

##### 1. Data Inputan Evolutionary Programming .

The screenshot shows a software window titled "Form Set Parameter" with three main sections:

- Parameter Evolutionary Programming:**
  - Jumlah Population: 50
  - Panjang Chromosome: 30
  - Jumlah Parameter: 4
  - Maximum Generation: 150
  - Prob.Competition: 0.8
  - Prob.Mutation: 0.003
  - Prob.Flip: 0.5
  - Konstanta Pinalty: 1000
- Parameter Mesin Induksi Poin 1:**
  - Tegangan (V): 380
  - Power Input (W): 810
  - Power Factor (pf): 0.62
  - Arus Stator (A): 1.96
  - Kec. Putar Rotor(rpm): 2917
- Parameter Mesin Induksi Poin 2:**
  - Tegangan (V): 380
  - Power Input (W): 560
  - Power Factor (pf): 0.52
  - Arus Stator (A): 1.62
  - Kec. Putar Rotor(rpm): 2955

At the bottom, there is a dropdown menu set to "Pengujian beban 75% - 50%", and "Apply" and "Use Default" buttons.

**Parameter Hitung:**

Xs	Min 2	Max 8
Rr	Min 2	Max 8
Xm	Min 40	Max 200
Rm	Min 372	Max 900

Radio buttons for "Gambar 3a" (selected) and "Gambar 3b" are also present.

**Gambar 4-28**  
Data Inputan Evolutionary Programming Pada Beban 75% dan 50%

##### 3. Hasil Program Evolutionary Programming

The screenshot shows a software window titled "Form Hasil" with two main sections:

- Hasil Evolutionary Programming Grafik EP:** A table showing the results over 17 generations.
- Calculated Parameters:** A list of final values for various parameters.

Gen	Min	Average	Max
1	405.8136	1654022.0728	3338372.1180
2	19590.0850	2666431.9972	3382995.1780
3	196549.3029	3195117.6638	3382995.1780
4	570387.5647	3311802.5348	6417271.9632
5	3322763.1064	3345074.8519	3458384.1931
6	3322882.0803	3354036.1880	3460182.4035
7	3327134.9463	3369875.1589	3460182.4035
8	3330271.3158	3395952.9085	3460182.4035
9	3330271.3158	3407555.4064	3464953.2079
10	3339537.3896	3424348.7051	3464953.2079
11	3384113.8527	3438348.6647	3464953.2119
12	3384113.8802	3446160.7244	3464953.2119
13	3415214.5882	3458980.6863	3464953.2119
14	3455022.8345	3460589.6261	3465294.2148
15	3455022.8345	3461655.1655	3465287.8463
16	3460179.8811	3544247.1559	7551852.9685
17	3460182.3882	3545286.8385	7551853.0471

**Calculated Parameters:**

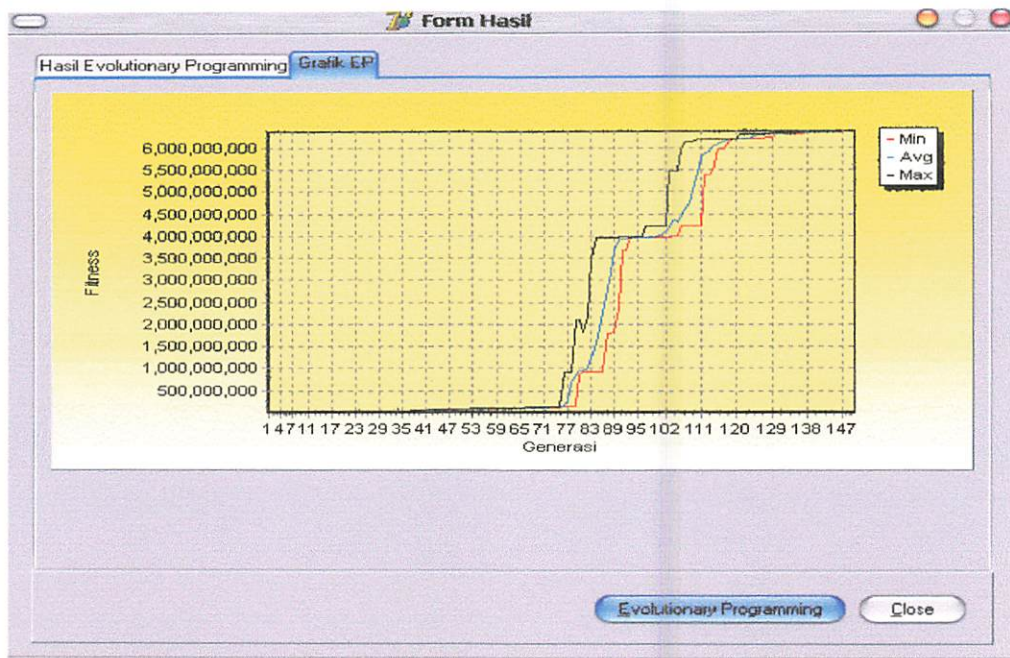
- Xs: 3.1243
- Rr: 7.9533
- Xm: 105.0000
- Rm: 800.9365
- R. Stator1 (Ohm): 6.069
- R. Stator2 (Ohm): 6.069
- Fitness: 8571211.926
- Error: 0.000000000000
- Poutput 1: 953.8504
- Pinput 1: 1054.8653
- Efisiensi 1: 90.4239 %
- Poutput 2: 485.8316
- Pinput 2: 561.7382
- Efisiensi 2: 86.4872 %

Buttons for "Evolutionary Programming" and "Close" are at the bottom.

**Gambar 4-29 .** Tampilan Hasil Evolutionary Programming Pada Beban 75% dan 50%



#### 4. Grafik Proses Evolutionary Programming



**Gambar 4-30**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 75% dan 50%**

### 4.5.2.5 DATA INPUTAN PADA PEMBEBANAN 75% & 25%

#### 1. Data Inputan Evolutionary Programming

The screenshot shows a software window titled "Form Set Parameter" with three main sections:

- Parameter Evolutionary Programming:**
  - Jumlah Population: 50
  - Panjang Chromosome: 30
  - Jumlah Parameter: 4
  - Maximum Generation: 150
  - Prob.Competition: 0.8
  - Prob.Mutation: 0.003
  - Prob.Flip: 0.5
  - Konstanta Pinalty: 1000
- Parameter Mesin Induksi Poin 1:**
  - Tegangan (V): 380
  - Power Input (W): 810
  - Power Factor (pf): 0.62
  - Arus Stator (A): 1.96
  - Kec. Putar Rotor(rpm): 2917
- Parameter Mesin Induksi Poin 2:**
  - Tegangan (V): 380
  - Power Input (W): 320
  - Power Factor (pf): 0.33
  - Arus Stator (A): 1.48
  - Kec. Putar Rotor(rpm): 2984

At the bottom, there is a dropdown menu for "Pengujian beban 75% - 25%", an "Apply" button, and a "Use Default" button. A "Parameter Hitung" section contains min/max values for Xs, Rr, Xm, and Rm, and radio buttons for "Gambar 3a" (selected) and "Gambar 3b".

Gambar 4-31  
Data Inputan Evolutionary Programming Pada Beban 75% dan 25%

#### 2. Hasil Program Evolutionary Programming

The screenshot shows a software window titled "Form Hasil" with the following data:

**Hasil Evolutionary Programming Grafik EP**

Gen	Min	Average	Max
1	1806.5725	25690.3690	127378.4512
2	9886.6020	54364.5796	127378.4512
3	21736.1948	86036.3625	128739.9302
4	34326.2878	118864.6006	358120.8585
5	55574.4942	139096.8286	454313.3941
6	100111.1066	157973.9463	454315.2039
7	127275.7650	223738.7151	615322.0933
8	127376.4303	352193.1663	615322.0933
9	127382.9722	443720.3319	661364.8084
10	265349.2437	530120.3015	661368.6857
11	367944.5924	597133.6928	1183904.8432
12	367944.5924	693680.4872	1183904.8432
13	551997.3837	790032.8532	1183904.8432
14	615103.5898	896352.9584	1183912.8529
15	615322.0933	958248.2099	1285652.2345
16	615322.0933	1026151.6686	1285652.2345
17	633766.0010	1141265.8055	1769073.7937

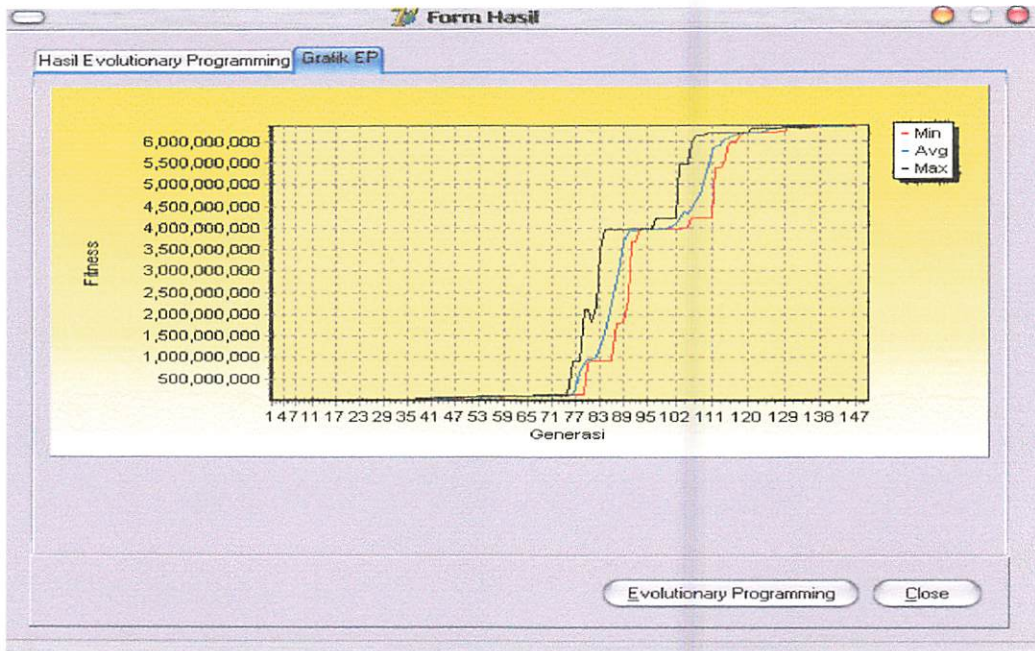
**Final Results:**

- Xs: 6.6186
- Rr: 5.2373
- Xm: 89.6680
- Rm: 784.6297
- R. Stator1 (Ohm): 6.069
- R. Stator2 (Ohm): 6.069
- Fitness: 6373706934.930
- Error: 0.000000000000
- Poutput 1: 729.8171
- Pinput 1: 809.9885
- Efisiensi 1: 90.1022 %
- Poutput 2: 221.9378
- Pinput 2: 329.4322
- Efisiensi 2: 67.3698 %

Buttons: Evolutionary Programming, Close

Gambar 4-32  
Tampilan Hasil Evolutionary Programming Pada Beban 75% dan 25%

### 3. Grafik Proses Evolutionary Programming



**Gambar 4-33**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 75% dan 25%**



## 4.5.2.6 DATA INPUTAN PADA PEMBEBANAN 50% & 25%

### 1. Data Inputan Evolutionary Programming

The screenshot shows a software window titled "Form Set Parameter" with three main sections:

- Parameter Evolutionary Programming:**
  - Jumlah Population: 50
  - Panjang Chromosome: 30
  - Jumlah Parameter: 4
  - Maximum Generation: 150
  - Prob. Competition: 0.8
  - Prob. Mutation: 0.003
  - Prob. Flip: 0.5
  - Konstanta Pinalty: 1000
- Parameter Mesin Induksi Poin 1:**
  - Tegangan (V): 380
  - Power Input (W): 560
  - Power Factor (pf): 0.52
  - Arus Stator (A): 1.62
  - Kec. Putar Rotor(rpm): 2955
- Parameter Mesin Induksi Poin 2:**
  - Tegangan (V): 380
  - Power Input (W): 320
  - Power Factor (pf): 0.33
  - Arus Stator (A): 1.48
  - Kec. Putar Rotor(rpm): 2984

At the bottom, there is a dropdown menu for "Pengujian beban 50% - 25%", and "Apply" and "Use Default" buttons.

**Parameter Hitung:**

- $X_s$  Min: 2, Max: 8
- $R_r$  Min: 2, Max: 8
- $X_m$  Min: 40, Max: 200
- $R_m$  Min: 372, Max: 900

Radio buttons for "Gambar 3a" (selected) and "Gambar 3b" are also present.

Gambar 4-34

Data Inputan Evolutionary Programming Pada Beban 50% dan 25%

### 2. Hasil Program Evolutionary Programming .

The screenshot shows a software window titled "Form Hasil" with a table of results and a list of final parameters.

Gen	Min	Average	Max
1	2462.7029	39051.4482	399362.0070
2	11485.0307	206540.1409	1354798.3393
3	31702.2157	446021.0580	1354798.3445
4	42744.3860	814212.3732	4635656.2726
5	98902.9593	1423819.2110	4635656.2726
6	433203.9560	2540580.2837	4729252.7084
7	1354800.7563	4100561.7293	4729252.7084
8	4573384.7722	4695338.5775	5898093.6041
9	4625047.5674	4793674.6056	5898269.3973
10	4636637.8680	4883902.2102	5898271.1571
11	4659240.6668	5226627.9685	9766492.3745
12	4728732.5906	5594418.8603	9777104.2664
13	4728832.4210	6148149.6817	9777104.2664
14	5582005.3204	6743008.0542	9785384.5827
15	5898087.1129	7221866.4422	10118323.6135
16	5898093.9904	8778190.4131	17368963.8816
17	5898261.4474	10039155.9238	27767896.8871

Final parameters on the right:

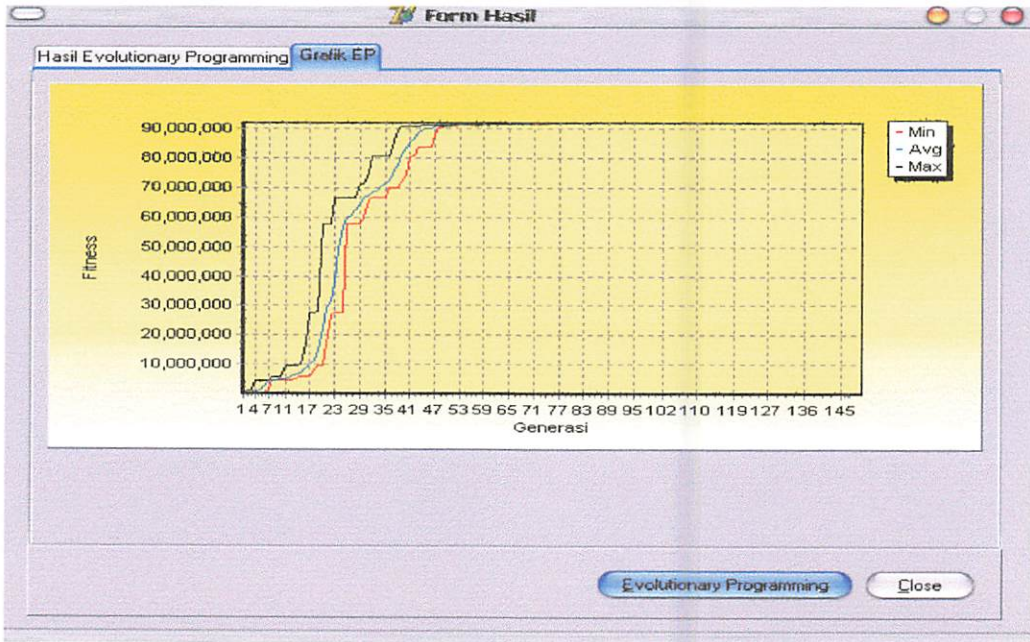
- $X_s$ : 2.0000
- $R_r$ : 6.8538
- $X_m$ : 87.5000
- $R_m$ : 734.4198
- R. Stator1 (Ohm): 6.069
- R. Stator2 (Ohm): 6.069
- Fitness: 92151572.812
- Error: 0.000000000000
- Poutput 1: 485.8316
- Pinput 1: 560.1347
- Efisiensi 1: 86.7348 %
- Poutput 2: 221.9378
- Pinput 2: 324.5089
- Efisiensi 2: 68.3919 %

Buttons for "Evolutionary Programming" and "Close" are at the bottom.

Gambar 4-35

Tampilan Hasil Evolutionary Programming Pada Beban 50% dan 25%

### 3. Grafik Proses Evolutionary Programming



**Gambar 4-36**  
**Tampilan Grafik Fitness Function-Generasi Pada Beban 50% dan 25%**

#### 4.6. Perbandingan Efisiensi Perhitungan Evolutionary Programming (OPT) VS Efisiensi Pengujian

Metode OPT	Pembebanan 100%		Pembebanan 75%		Pembebanan 50%		Pembebanan 25%	
	UJI	EP	UJI	EP	UJI	EP	UJI	EP
$P_{out}$ (W)	953.367	953.85	729.45	729.817	485.585	485.832	221.825	221.94
$P_{input}$ (W)	1051	1054.87	810	809.275	560	565.155	320	329.43
Effisiensi (%)	90.7104	90.4239	90.055	90.1816	86.7117	85.9642	69.3204	67.37

Tabel 4-5 Perbandingan hasil program Evolutionary Programming (EP-OPT) dengan hasil pengujian

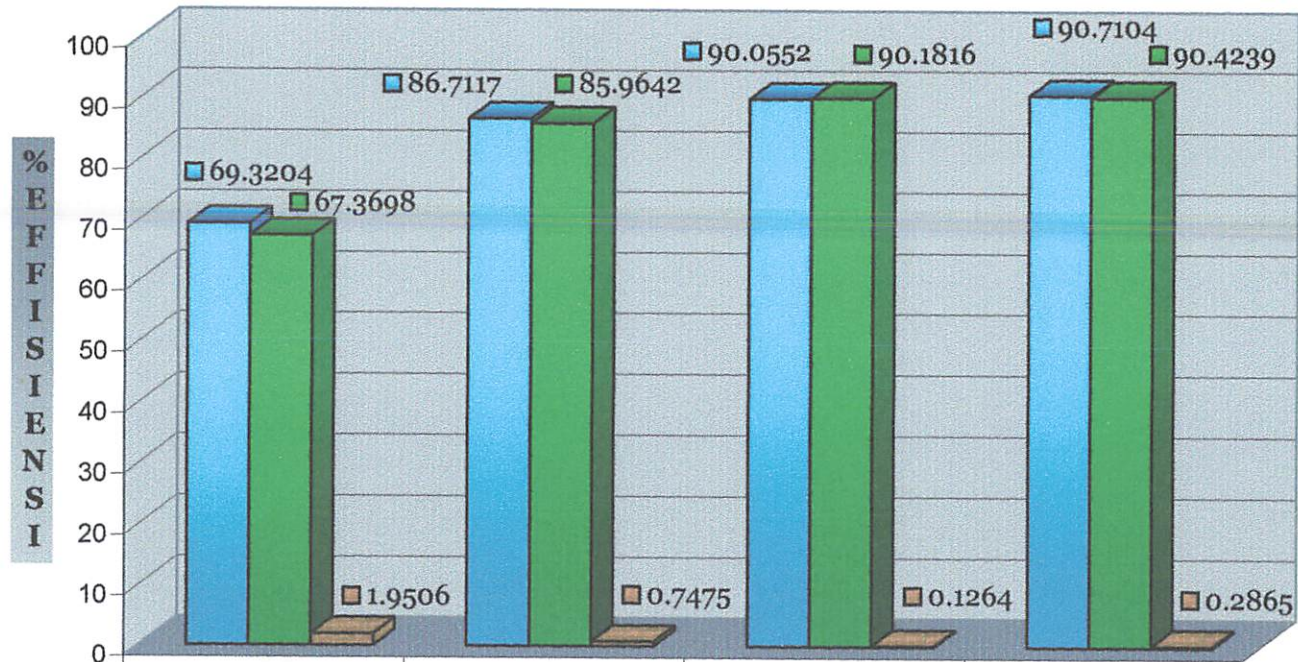
#### 4.7 Perbandingan efisiensi rata-rata Perhitungan Evolutionary Programming (TPT) VS Efisiensi Pengujian

Metode TPT	% Pembebanan			
	100%	75%	50%	25%
100% & 75%	90.4239	90.182	-	-
100% & 50%	90.4239	-	86.4872	-
100% & 25%	90.4239	-	-	68.3919
75% & 50%	-	90.092	86.4872	-
75% & 25%	-	90.102	-	67.3698
50% & 25%	-	-	86.7348	68.3919
Nilai Rata-Rata	90.4239	90.125	86.5697	68.0512
pENGUJIAN	90.7104	90.0552	86.7117	69.3204

Tabel 4-6 Perbandingan hasil program rata-rata Evolutionary Programming (EP-TPT) dengan hasil pengujian



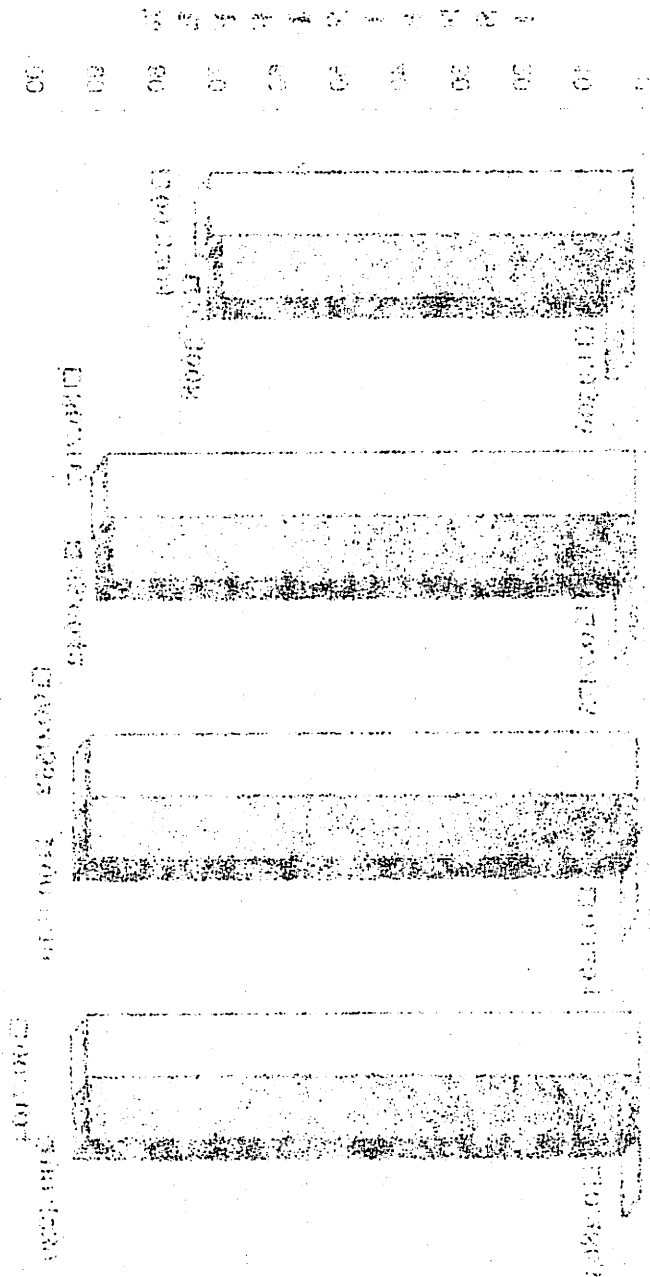
### EFFISIENSI EP VS PENGUJIAN (OPT)



	25%	50%	75%	100%
■ EFFISIENSI PENGUJIAN	69.3204	86.7117	90.0552	90.7104
■ EFFISIENSI EP	67.3698	85.9642	90.1816	90.4239
■ SELISIH EFFISIENSI EP VS PENGUJIAN	1.9506	0.7475	0.1264	0.2865

%BEBAN

STATISTICAL INFORMATION



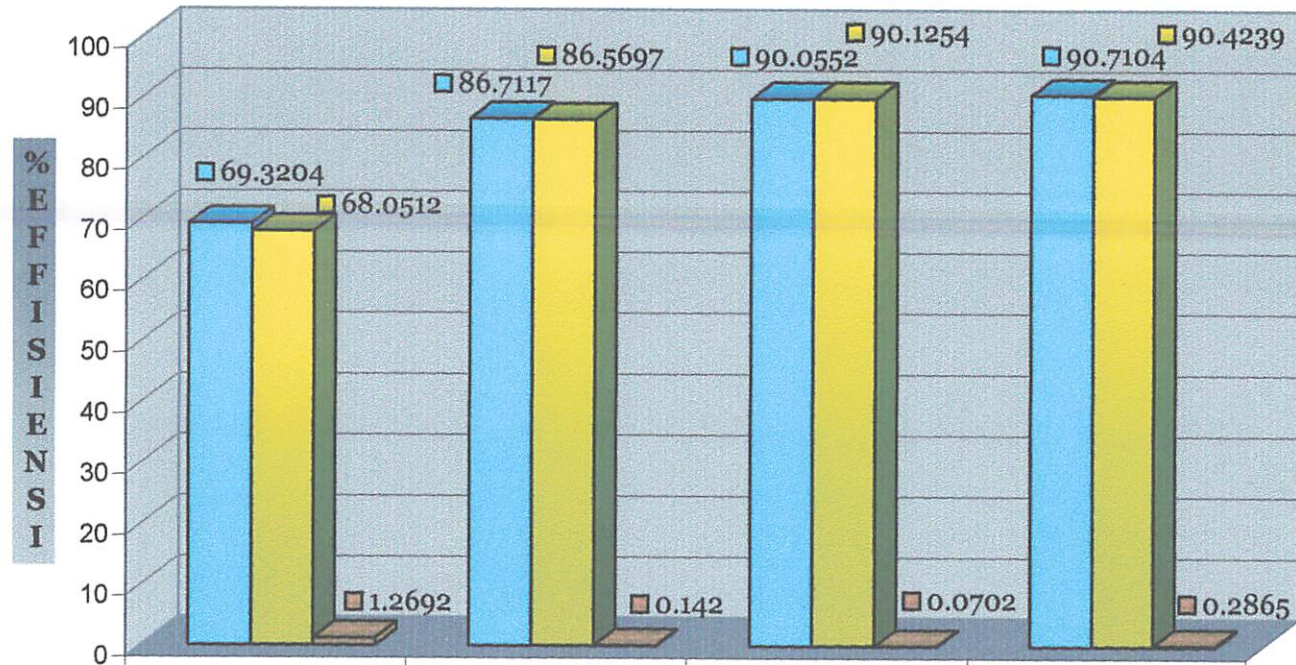
STATISTICAL INFORMATION  
 STATISTICAL INFORMATION  
 STATISTICAL INFORMATION

STATISTICAL INFORMATION  
 STATISTICAL INFORMATION  
 STATISTICAL INFORMATION  
 STATISTICAL INFORMATION

STATISTICAL INFORMATION



**PERBANDINGAN EFFISIENSI EP VS PENGUJIAN (TPT)**



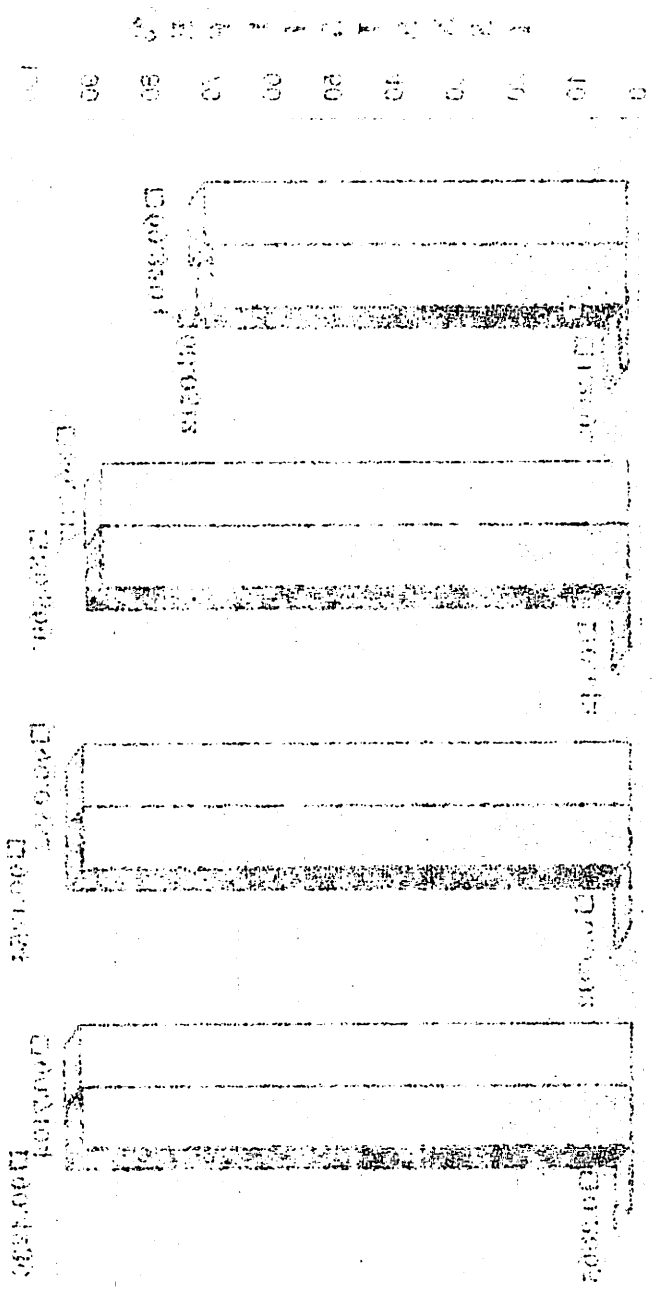
	25%	50%	75%	100%
■ EFFISIENSI PENGUJIAN	69.3204	86.7117	90.0552	90.7104
■ EFFISIENSI EP	68.0512	86.5697	90.1254	90.4239
■ SELISIH EFFISIENSI PENGUJIAN VS EP (TPT)	1.2692	0.142	0.0702	0.2865

**%BEBAN**

ЗАДАНИЕ

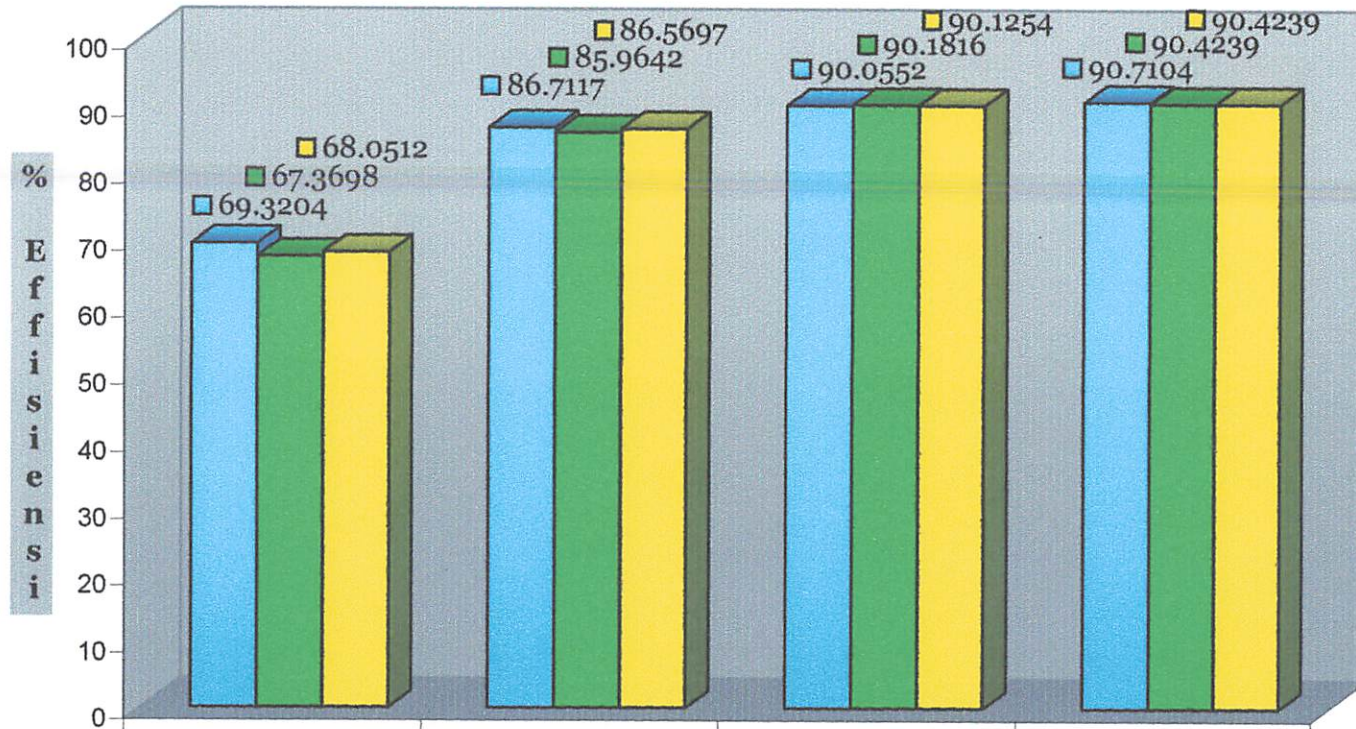
АВТОР (ИМЯ)  
 ПОДПИСЬ  
 ПОДПИСЬ

15000	20000	30000	40000
000000	000000	000000	000000
000000	000000	000000	000000



ЗАДАНИЕ НА РАБОТУ

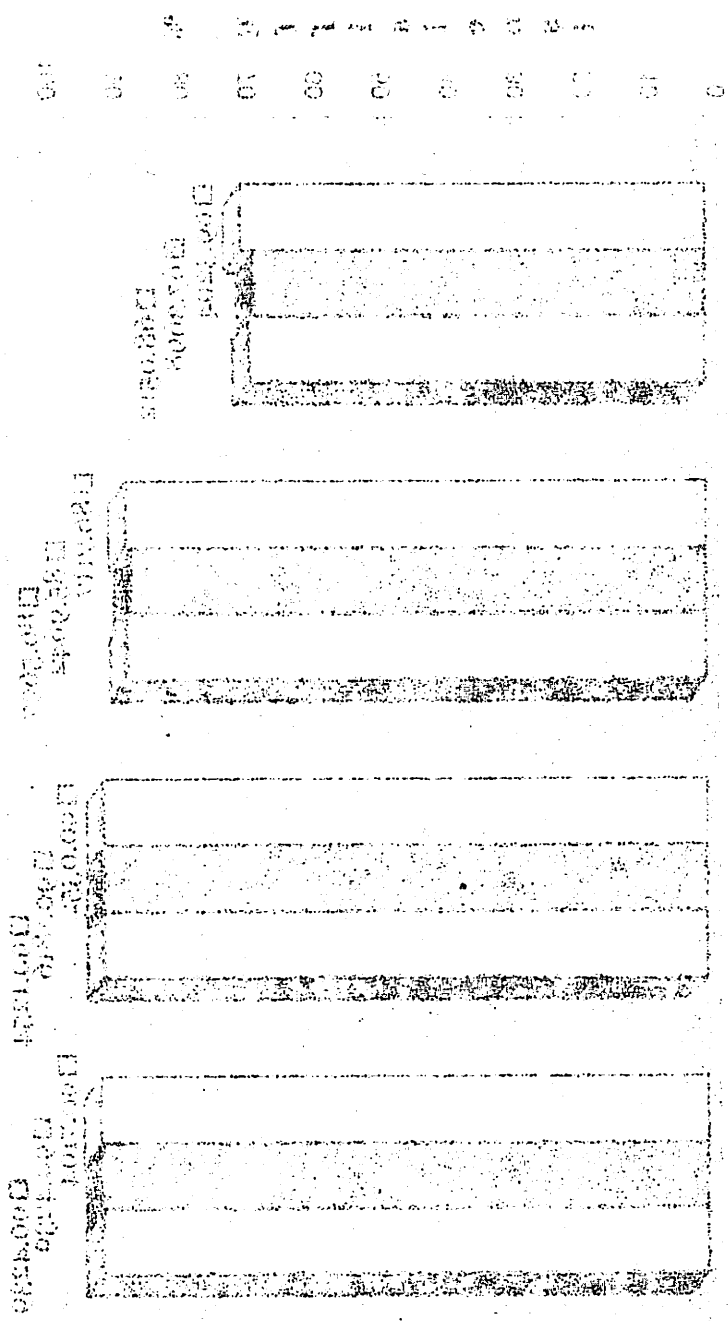
**perbandingan Effisiensi Metode Evolutionary Programming (OPT & TPT)  
VS Effisiensi Pengujian**



	25%	50%	75%	100%
■ Effisiensi Pengujian	69.3204	86.7117	90.0552	90.7104
■ Effisiensi OPT	67.3698	85.9642	90.1816	90.4239
■ Effisiensi TPT	68.0512	86.5697	90.1254	90.4239

**% Beban**

DESCRIPTION	QTY	UNIT	AMOUNT	TOTAL
DESCRIPTION	1000	EA	1000.00	1000.00
DESCRIPTION	1000	EA	1000.00	2000.00
DESCRIPTION	1000	EA	1000.00	3000.00
DESCRIPTION	1000	EA	1000.00	4000.00



DESCRIPTION QTY UNIT AMOUNT TOTAL