

18.477/ITS/2003



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

OPTIMASI UNIT PEMBANGKIT LISTRIK MENGUNAKAN ALGORITMA REFINED GENETIK

TUGAS AKHIR



R.51f
005.1
Lak
0-1

2000

Disusun Oleh :

PRADIPTO LAKSONO

NRP. 2691.100.020

| PERPUSTAKAAN ITS | |
|---------------------|-----------|
| Tgl. Terima | 16-7-2003 |
| Terima dari | A1 |
| No. Agenda Frp. | 208352 |

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2000

**OPTIMASI UNIT PEMBANGKIT LISTRIK
MENGUNAKAN ALGORITMA REFINED GENETIK**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer**

Pada

Jurusan Teknik Informatika

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing



Drs.Ec. Ir. RIYANARTO SARNO, M.Sc., Ph.D.

NIP. 131 570 363

SURABAYA

Februari, 2000

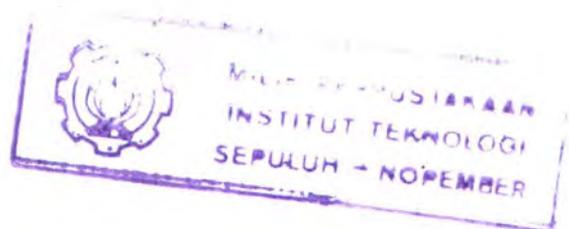
.....Karena sesungguhnya sesudah kesulitan itu ada kemudahan (5)

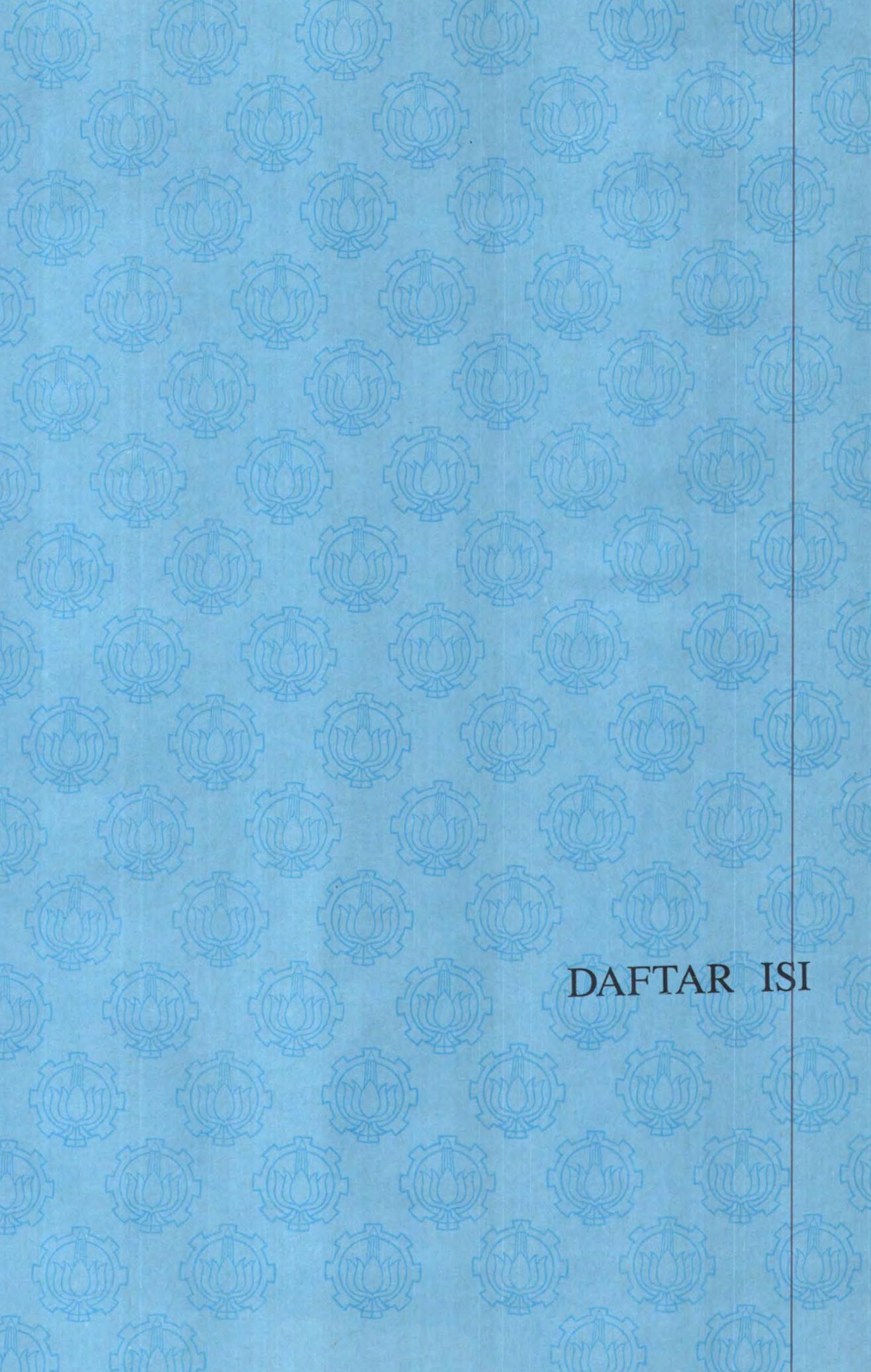
.....Sesungguhnya sesudah kesulitan itu ada kemudahan (6)

(QS Alam Nasyrah 30 : 94)

Untuk Ayah dan Ibu Yang Tercinta

Dari Pradipto, putramu





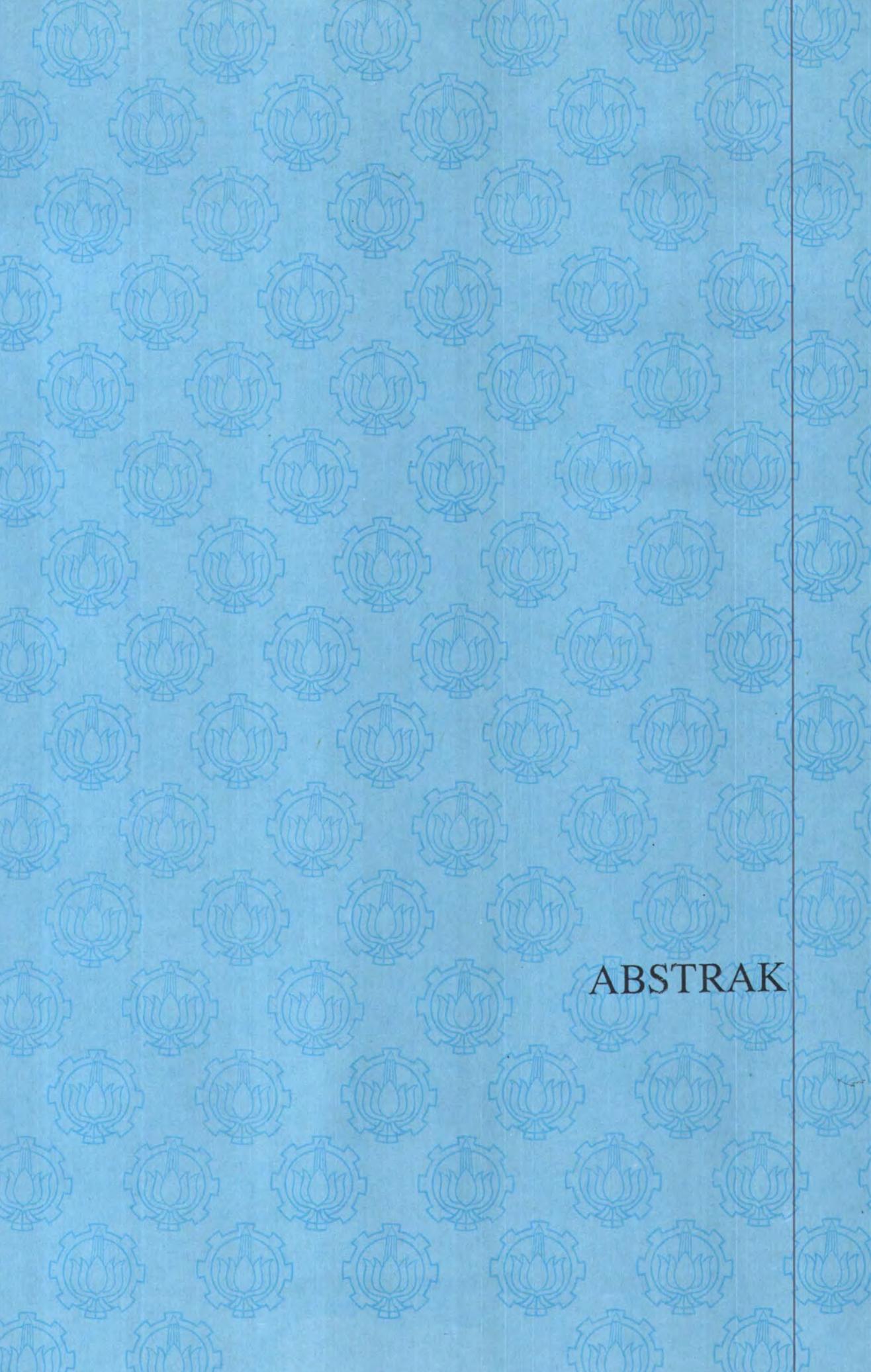
DAFTAR ISI

DAFTAR ISI

| | |
|-------------------------------------------------|-----------|
| DAFTAR ISI | 1 |
| ABSTRAK | 4 |
| KATA PENGANTAR | 5 |
| BAB I PENDAHULUAN | 7 |
| 1.1. Latar Belakang | 7 |
| 1.2. Perumusan Masalah | 8 |
| 1.3. Batasan Masalah | 9 |
| 1.4. Tujuan | 10 |
| 1.5. Manfaat | 10 |
| 1.6. Metodologi | 10 |
| 1.7. Sistematika Pembahasan | 11 |
| BAB II SISTEM PEMBANGKIT | 12 |
| 2.1. Konsep Umum Optimasi Unit Pembangkit | 12 |
| 2.2. Sistem Catu Daya | 13 |
| 2.3. Karakteristik Unit Pembangkit..... | 13 |
| • Karakteristik Input-Output | 14 |
| • Karakteristik Incremental Heat Rate..... | 16 |
| 2.4. Perkiraan Jangka Waktu Beban Sistem | 16 |
| 2.5. Unit Commitment | 18 |
| 2.6. Batasan Unit Commitment..... | 19 |
| 2.6.1. Spinning Reserve | 19 |
| 2.6.2. Thermal Unit Constraint | 20 |
| 2.6.3. Biaya Start (Start Up Cost) | 20 |

| | |
|------------------------------------------------------------|-----------|
| BAB III ALGORITMA REFINED GENETIK | 22 |
| 3.1. Struktur dasar algoritma genetik..... | 22 |
| 3.2. Representasi Problem..... | 23 |
| 3.3. Pemetaan..... | 23 |
| 3.4. Seleksi | 24 |
| 3.4.1. Daerah sampling..... | 25 |
| Daerah sampling tetap..... | 25 |
| Daerah sampling yang diperbesar | 26 |
| 3.4.2. Mekanisme sampling | 26 |
| Sampling stokastik | 27 |
| Sampling deterministik | 27 |
| Sampling campuran..... | 27 |
| 3.4.3. Probabilitas Seleksi | 28 |
| 3.4.4. Skala Fitness (Fitness Scaling)..... | 28 |
| Skala Linier (Linear scaling)..... | 29 |
| Skala linier dinamik (Dynamic Linear Scaling)..... | 29 |
| Windowing..... | 30 |
| Normalizing..... | 30 |
| Sigma Truncation..... | 31 |
| Power Law Scaling | 32 |
| Logarithmic Scaling..... | 33 |
| 3.5. Problem Optimasi Genetik..... | 33 |
| • Nonlinear Programming (constrained optimation) | 34 |
| 3.6. Teorema Skema..... | 35 |
| 3.7. Parameter dan Operator Algoritma Refined Genetik..... | 36 |
| 3.7.1. Seleksi | 37 |
| 3.7.2. Pindah Silang (Crossover)..... | 38 |
| 3.7.3. Mutasi (Mutation) | 40 |

| | |
|-----------------------------------------------------------------|-----------|
| 3.7.4. Elitism | 40 |
| 3.7.5. Probabilitas Crossover dan Probabilitas Mutation..... | 40 |
| 3.8. Karakteristik Algoritma Genetik..... | 41 |
| 3.8.1. Eksplorasi dan Eksploitasi | 41 |
| 3.9. Proses Generasi | 43 |
| BAB IV PERANCANGAN SISTEM..... | 44 |
| 4.1. Fungsi Obyektif Economic Dispatch (Objectif function)..... | 44 |
| 4.2. Representasi Kromosom Dalam Economic Dispatch | 48 |
| 4.2.1. Jumlah Kromosom (Population Size) | 49 |
| 4.2.2. Jumlah Generasi (Number of Generation)..... | 50 |
| 4.3. Data Masukan | 52 |
| 4.4. Deskripsi dan Implementasi Obyek | 53 |
| 4.4.1. Obyek Tchromosome..... | 53 |
| 4.4.2. Obyek Tpopulation | 55 |
| 4.4.3. Obyek TCommitment dan TPowerDispatchRGA | 56 |
| 4.5. Fungsi Obyektif Unit Commitment | 57 |
| 4.6. Penjadwalan Unit Commitment..... | 58 |
| 4.7. Jenis Distribusi Random | 59 |
| BAB V HASIL UJI COBA APLIKASI PROGRAM | 61 |
| 5.1. Uji Coba Unit Commitment..... | 61 |
| 5.1.1. Data Input Percobaan..... | 61 |
| 5.1.2. Hasil Uji Coba (perubahan populasi)..... | 62 |
| 5.1.3. Hasil Uji Coba (perubahan probabilitas crossover)..... | 65 |
| BAB VI KESIMPULAN DAN SARAN | 67 |
| Daftar Pustaka..... | 68 |



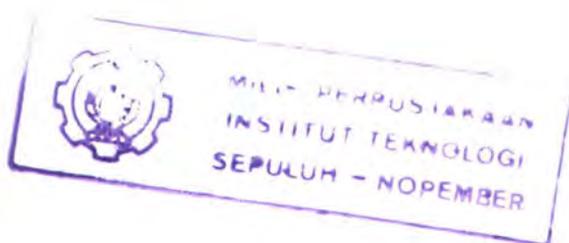
ABSTRAK

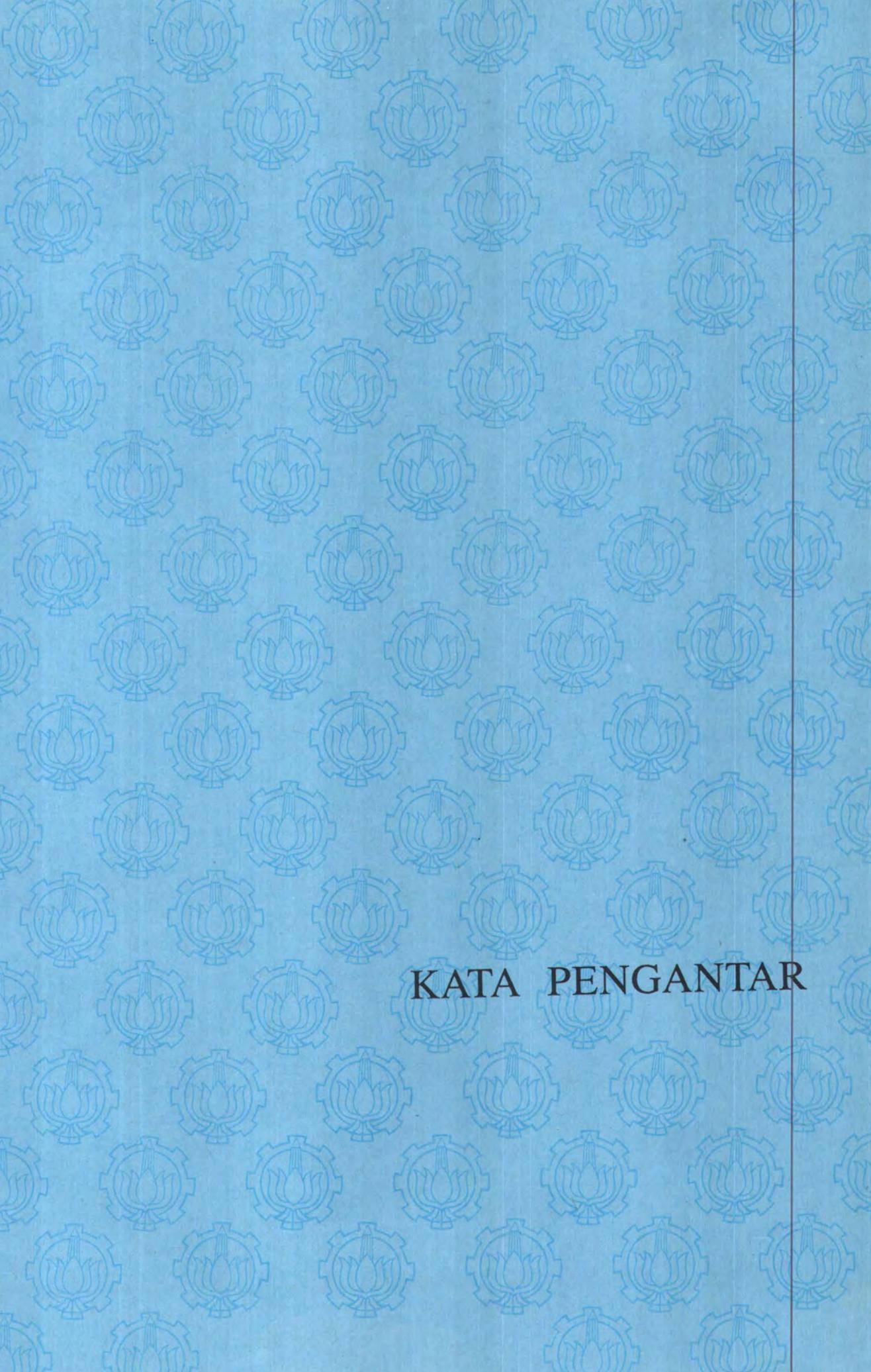
ABSTRAK

Dalam tugas akhir ini, diterapkan algoritma refined genetik untuk mengoptimalkan daya yang dibangkitkan oleh unit-unit pembangkit listrik. Daya yang dihasilkan oleh unit-unit listrik tersebut disesuaikan dengan permintaan beban (demand) ditambah dengan biaya rugi-rugi disertai pertimbangan pemakaian bahan bakar (input) yang seefisien mungkin.

Permasalahan optimasi unit pembangkit umumnya melibatkan persamaan nonlinear dan fungsi yang tidak kontinu, sehingga sulit penyelesaian komputasinya. Berdasarkan hal tersebut maka algoritma refined genetik diharapkan dapat digunakan untuk mencari solusi yang optimal.

Analisa tersebut diharapkan dapat memberikan gambaran pembebanan serta penjadwalan unit-unit pembangkit listrik dalam memenuhi kebutuhan listrik yang ada serta menjaga kontinuitas kebutuhan daya listrik.





KATA PENGANTAR

KATA PENGANTAR

Puji Syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, taufiq serta hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul :

“Optimasi Unit Pembangkit Listrik Menggunakan Algoritma Refined Genetik”

Sebagaimana peraturan akademik di jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya, penulisan Tugas Akhir ini adalah merupakan salah satu syarat kelulusan untuk mendapatkan gelar kesarjanaan pada jenjang pendidikan Strata satu (S1).

Pada kesempatan ini penulis tak lupa mengucapkan terima kasih kepada :

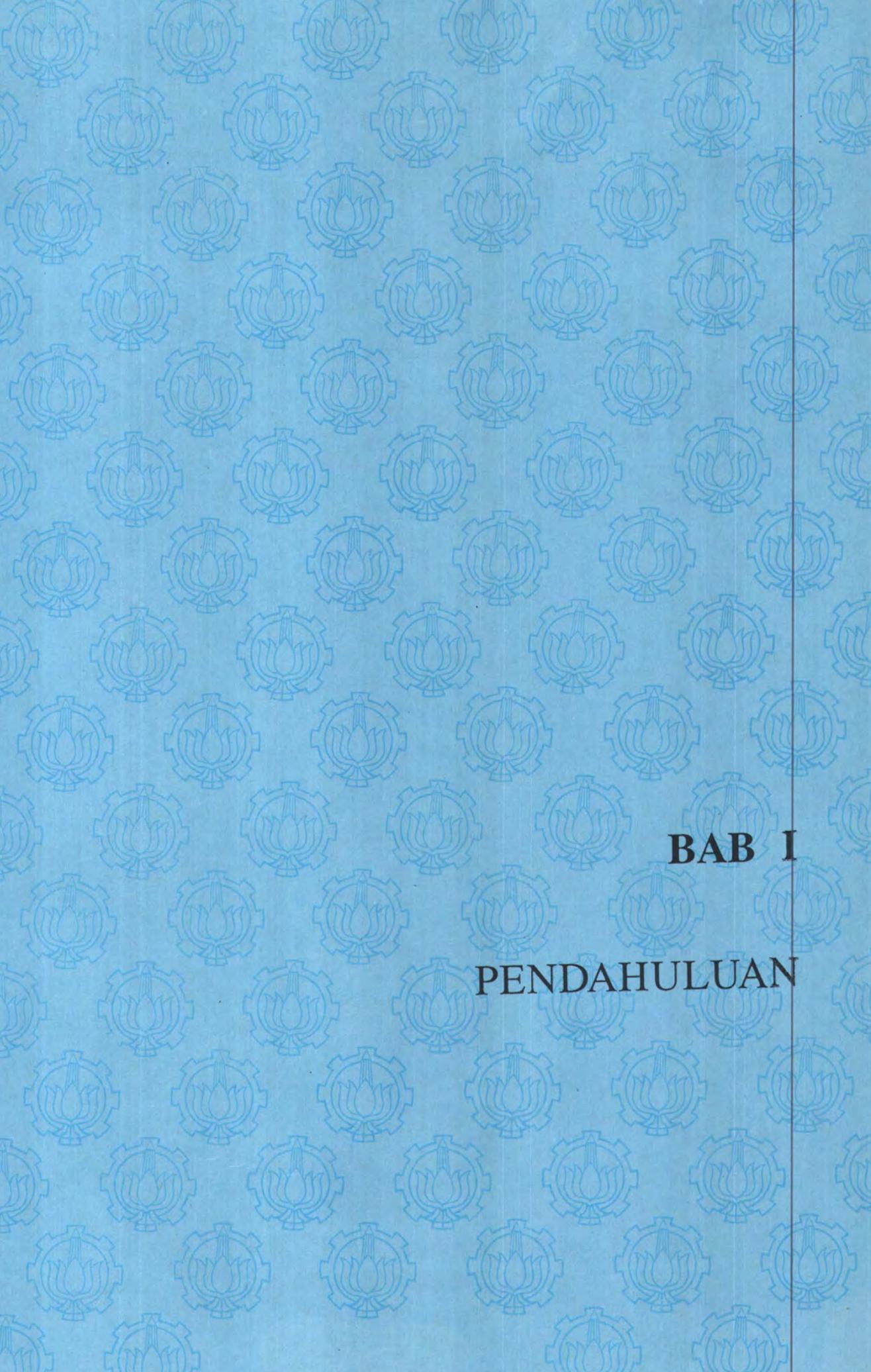
- Bapak Drs. Ec. Ir. Riyanarto Sarno MSc, PhD, selaku Dosen Pembimbing Tugas Akhir.
- Bapak Ir. Arif Djunaedi MSc, PhD, selaku Ketua Jurusan Teknik Informatika ITS
- Ayah dan Ibu serta saudara kandungku yang tak bosan mendorong terselesaikannya tugas akhir ini.
- Ibu Ir. Ester Hanaya MSc, selaku Koordinator Tugas Akhir.
- Ibu Ir. Handayani Tjandrasa MSc, PhD, selaku Dosen Wali.
- Segenap Staf Pengajar khususnya Pak Yudi Mulyono dan Karyawan Jurusan Teknik Informatika ITS.

- Rekan-rekan mahasiswa Teknik Informatika khususnya Edi Setiawan, Sucahyo Aji Condro, dan rekan-rekan kost yang selalu mendorong terselesaikannya Tugas akhir ini.
- Sahabatku “Ular Naga” dan “Burung Gagak” yang telah menemani selama perjalanan kehidupan ini.

Akhirnya semoga tulisan ini bermanfaat dan atas segala kekurangan yang ada penulis mohon maaf yang sebesar-besarnya.

Surabaya, Januari 2000

Pradipto Laksono



BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kebutuhan akan adanya penyediaan listrik telah menjadi penentu dalam berbagai sektor baik sektor jasa, industri, sektor rumah tangga dan lain-lain. Hal tersebut menunjukkan bahwa listrik merupakan komponen utama dalam menggerakkan berbagai sektor didalam kehidupan. Untuk itu diperlukan pelayanan karakteristik dasar yang baik yaitu mutu dan harga dapat terjangkau sehingga memberikan kepuasan pada konsumen. Mutu yang baik adalah tegangan dan frekuensi dari listrik yang disalurkan sesuai standar serta kontinuitas dalam penyediaannya terjamin. Oleh karena itu penyediaan energi listrik harus bersifat efisien dan bisa diandalkan, hal ini berarti pembangkitan dan penyaluran energi listrik itu harus dilakukan secara ekonomis dan rasional. Berhubungan dengan hal tersebut maka diperlukan suatu metode optimasi yang berguna untuk mengefisienkan penyediaan listrik.

Pemakaian energi listrik yang efisien berarti dalam penyaluran energi tersebut, daya yang disalurkan untuk memenuhi beban diusahakan agar sebesar-besarnya dengan pemakaian bahan bakar seminim mungkin untuk membangkitkan unit-unit listrik yang ada

Untuk mencapai pemecahan masalah seperti diatas maka perlu dilakukan suatu proses perencanaan operasi dari unit-unit listrik tersebut secara baik dan akurat. Oleh karena biaya operasi dari seluruh sistem mencapai persentase sekitar 70% dari seluruh biaya maka perencanaan operasi harus dilakukan menggunakan teknik optimasi yang tepat agar bisa menekan biaya operasi seminim mungkin.

Algoritma refined genetik diharapkan dapat digunakan untuk menentukan biaya minimum dalam suatu proses produksi energi listrik dengan berbagai pilihan untuk menuju kondisi akhir. Dimana dalam algoritma refined genetik tersebut terdapat persamaan apabila dibandingkan dengan algoritma simple genetik, yaitu dalam beberapa operatornya seperti seleksi, mutasi. Akan tetapi terdapat beberapa perbedaan yang tidak terdapat dalam algoritma simple genetik yaitu teknik crossover, elitism, teknik perhitungan probabilitas mutasi dan probabilitas crossover. Sehingga dengan menggunakan algoritma refined genetik maka diharapkan lebih cepat dalam mencapai hasil yang optimal.

1.2. Perumusan Masalah

Unit-unit pembangkit listrik membutuhkan suatu metode yang dapat digunakan untuk melayani seluruh permintaan beban. Pada kenyataannya kemampuan daya yang dihasilkan atau output tidak dapat dihitung dengan hanya menjumlah kapasitas (MW) seluruh unit-unit pembangkit listrik yang ada. Perhitungan penyediaan bahan-bakar (input) untuk menghasilkan output daya dari unit-unit pembangkit listrik dipengaruhi

beberapa faktor antara lain adalah pemakaian daya yang digunakan untuk membangkitkan unit pembangkit itu sendiri, faktor error dalam memperhitungkan daya (output) yang dihasilkan oleh unit-unit pembangkit listrik. Pemecahan masalah ini menyangkut koordinasi dalam sistem pembangkitan yang menyangkut penjadwalan pada siklus waktu tertentu selama periode waktu tertentu.

Dalam tugas akhir ini algoritma refined genetik merupakan suatu cara untuk mengoptimalkan pembagian penyediaan daya serta koordinasi dari unit-unit pembangkit listrik dengan memperhatikan faktor-faktor seperti yang disebutkan diatas.

1.3. Batasan Masalah

Dalam optimasi penyediaan energi listrik ini diasumsikan bahwa :

- Pemecahan permasalahan sampai pada unit commitment.
- Biaya transmision losses tidak diperhitungkan.
- Pada penjadwalan didalam unit commitment tidak diperhitungkan minimum down time dan minimum up time serta spinning reserves.

1.4. Tujuan

1. Mengoptimalkan unjuk kerja dari unit pembangkit listrik.
2. Mengefisienkan bahan-bakar yang digunakan oleh unit-unit pembangkit listrik dalam memenuhi permintaan beban.
3. Penerapan algoritma refined genetik yang diharapkan dapat mencapai solusi yang optimal.

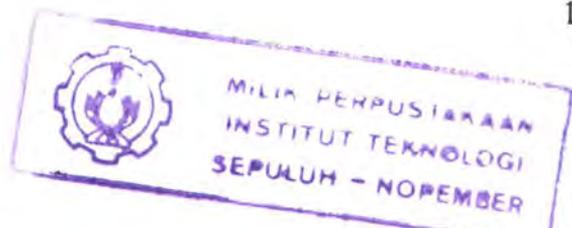
1.5. Manfaat

1. Sebagai bahan acuan didalam mengembangkan algoritma genetik, sehingga dicapai suatu model algoritma genetik yang lebih bagus daripada generasi sebelumnya.
2. Tugas akhir ini diharapkan dapat memberikan suatu laporan untuk perencanaan optimasi terhadap penyediaan energi listrik yang memberikan investasi rendah dan menghasilkan daya (output) optimal.

1.6. Metodologi

Dalam memecahkan permasalahan optimasi unit-unit pembangkit listrik ini menggunakan metodologi sebagai berikut :

1. Studi pustaka.
2. Penyusunan algoritma dan desain sistem.

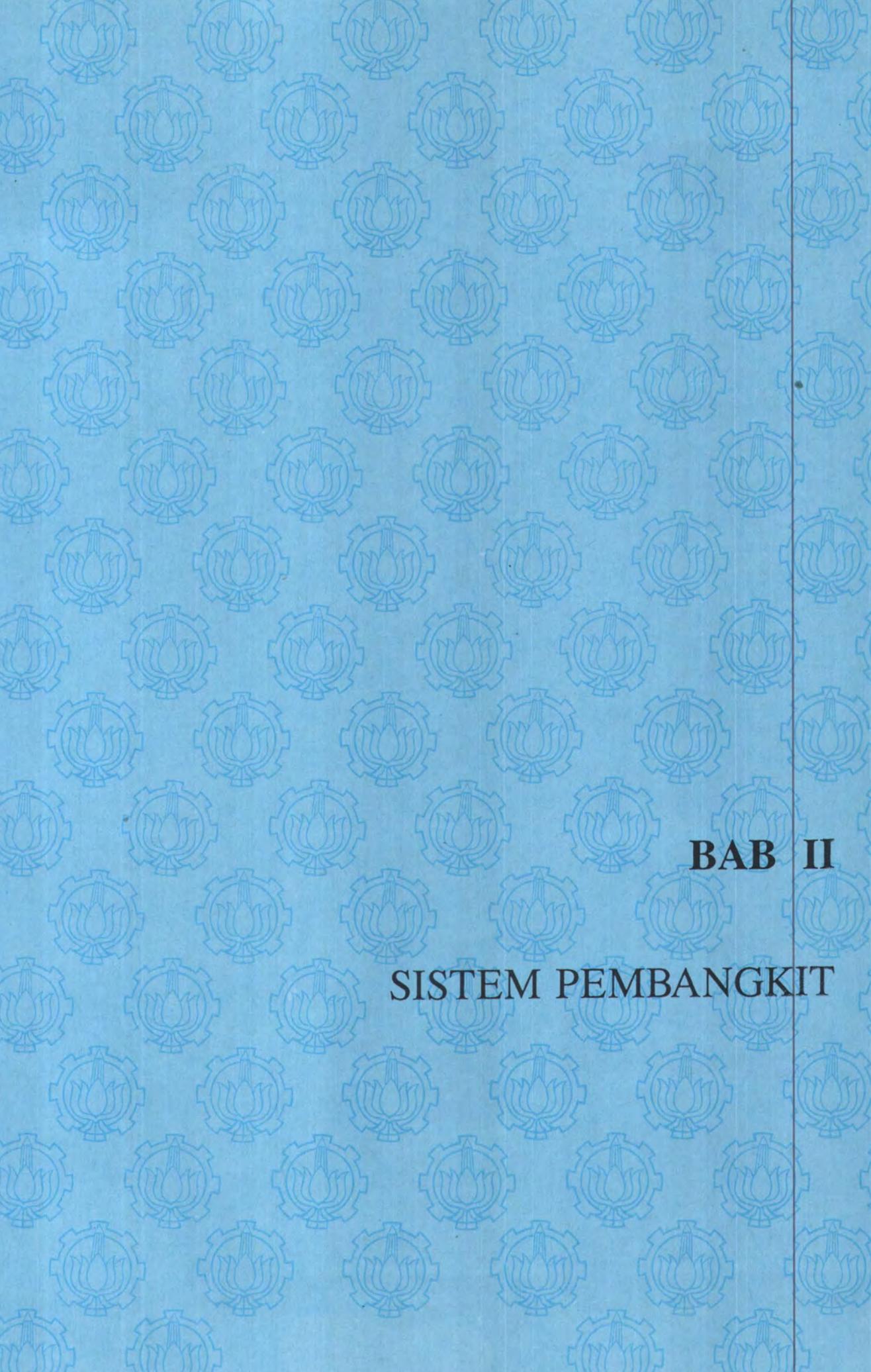


3. Pengujian algoritma.
4. Perancangan program dan struktur data.
5. Implementasi
6. Evaluasi dan revisi.
7. Penyusunan laporan.

1.7. Sistematika Pembahasan

Agar pembaca lebih mudah dalam memahami tujuan dan manfaat penelitian ini maka sistematika pembahasan tugas akhir ini disusun sebagai berikut :

- Bab I. Pendahuluan, yang menjelaskan tentang latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat, metodologi serta sistematika pembahasan.
- Bab II. Pembahasan tentang unit pembangkit listrik yang meliputi karakteristik pembangkit serta permasalahan economic dispatch dan unit commitment.
- Bab III. Pembahasan dasar teori dari algoritma refined genetik.
- Bab IV. Menjelaskan perancangan desain sistem, struktur data program refined genetik.
- Bab V. Penjelasan hasil uji coba aplikasi program refined genetik pada unit commitment.
- Bab VI. Merupakan bab terakhir yang berisi beberapa kesimpulan dan saran untuk program refined genetik.



BAB II

SISTEM PEMBANGKIT

BAB II

SISTEM PEMBANGKIT

Dalam bab ini dijelaskan tentang konsep sistem pembangkit. Kajian sistem pembangkit akan meliputi diantaranya konsep umum unit pembangkit, karakteristik pembangkit, tinjauan ekonomis dari unit pembangkit (economic dispatch), sistem catu daya, unit commitment.

2.1. Konsep Umum Optimasi Unit Pembangkit

Energi listrik merupakan kebutuhan yang memegang peranan penting didalam menjalankan roda kehidupan. Energi listrik telah lama menjadi penggerak yang potensial di bidang industri, jasa dan lain sebagainya. Dengan makin bertambahnya permintaan pemakaian energi listrik maka sangatlah perlu adanya pengendalian secara optimum energi listrik yang ada, dengan cara membentuk sistem distribusi yang andal dan efektif. Sejalan dengan kenaikan harga bahan bakar yang dipakai oleh unit pembangkit, apabila ditinjau dari segi ekonomi maka efisiensi dan optimasi dalam hal ini tentunya mempunyai kaitan yang erat dan memegang peranan penting.

2.2. Sistem Catu Daya

Sistem catu daya terdiri dari tiga bagian utama yaitu : unit/stasiun pembangkit, saluran transmisi dan sistem distribusi. Berikut ini uraian dari masing-masing bagian :

- Unit/stasiun pembangkit

Merupakan pusat pembangkitan daya.

- Saluran Transmisi

Menghubungkan unit/stasiun pembangkit dengan sistem distribusi serta sistem-sistem pembangkitan yang lain.

- Sistem distribusi

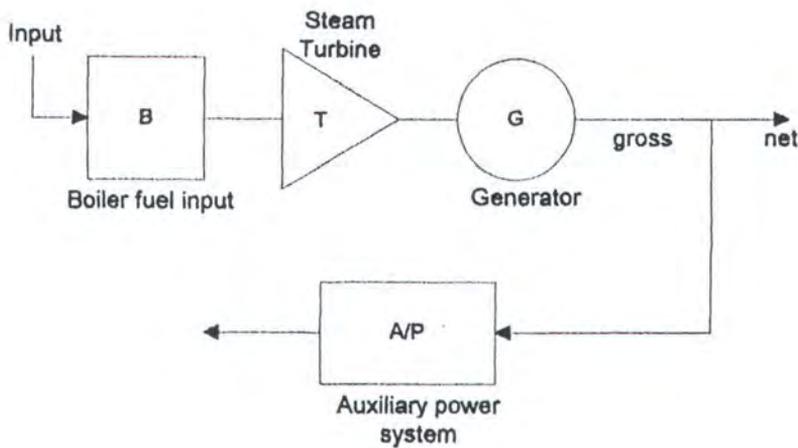
Merupakan penghubung antara saluran transmisi dengan semua beban pada daerah tertentu

2.3. Karakteristik Unit Pembangkit

Dalam menganalisa suatu permasalahan yang berhubungan dengan operasi pengontrolan suatu sistem pembangkit terdapat banyak parameter yang dipakai. Hal yang paling mendasar dalam pengoperasian sistem pembangkit apabila dilihat dari segi ekonomi adalah karakteristik input-output, karakteristik heat rate, karakteristik biaya tambahan (incremental cost) dari unit pembangkit tersebut. Untuk lebih jelasnya berikut ini uraian dari masing-masing karakteristik seperti tersebut diatas.

- **Karakteristik Input-Output**

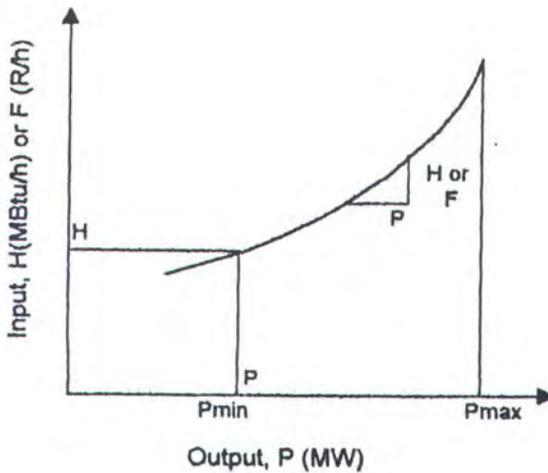
Karakteristik input-output menggambarkan hubungan antara gross input dengan net output dari unit pembangkit. Dimana gross input adalah merupakan total input / jumlah panas dalam satuan kalori / jam atau MBtu / jam misalkan, yang harus dimasukkan kedalam boiler sebagai bahan bakar. Sedangkan net output adalah merupakan daya listrik yang dihasilkan oleh pembangkit diluar keperluan pembangkit itu sendiri.



Gambar 2.1 Power Generator

Dalam mendefinisikan karakteristik input-output, dapat dilihat dengan membandingkan gros input dengan net output dimana selanjutnya dapat digambarkan dalam bentuk lengkung / kurva input-output. Lengkung input-output tersebut untuk lebih jauh dimaksudkan memperlihatkan efisiensi termis pusat pembangkit tersebut. Dalam lengkung input-output, koordinat y mewakili

gross input yang dinotasikan sebagai H (MBtu/jam) dan koordinat x adalah net output yang dinyatakan dalam notasi P (MegaWatt).



Gb. 2.2 Lengkung Input-Output

Lengkung Input-output (gb 2.2) tersebut dapat diterjemahkan ke dalam persamaan polinomial sebagai berikut :

$$H_i(P_i) = a_i P_i^2 + b_i P_i + c_i ; \quad P_{\min} \leq P \leq P_{\max} \quad (2.1)$$

dengan :

a, b, c : konstanta unit ke-i.

P_i : output daya unit ke-i.

Dimana persamaan (2.1) apabila dikalikan dengan fuel cost akan menghasilkan fungsi obyektif cost sebagai berikut :

$$F_i(P_i) = (a_i P_i^2 + b_i P_i + c_i) * \text{Fuel cost unit-i} \quad (2.2)$$

- **Karakteristik Incremental Heat Rate**

Incremental heat rate didapat dari penurunan fungsi persamaan 2.2 yaitu sebagai berikut :

$$\frac{dP_i}{dF_i} = 2a_i P_i + b_i \quad (2.3)$$

dimana apabila persamaan (2.3) dikalikan fuel cost masing-masing unit maka akan didapatkan fungsi incremental cost.

2.4. Perkiraan Jangka Waktu Beban Sistem

Beban sistem tenaga listrik merupakan besarnya daya yang dibutuhkan oleh pelanggan. Permintaan beban selalu berubah setiap siklus waktu tertentu, dimana tidak ada perhitungan eksak tentang perubahan beban pada siklus waktu tertentu. Oleh karena itu diperlukan suatu perkiraan siklus beban dalam Koordinasi unit pembangkit dalam suatu sistem pembangkitan untuk mencapai keadaan optimum.

Maka perkiraan perubahan beban pada siklus waktu tertentu merupakan salah satu faktor yang turut menentukan dalam perencanaan dan koordinasi unit-unit pembangkit dalam suatu sistem pembangkitan.

Perkiraan beban dapat dibagi dalam tiga kategori yaitu :

1. Perkiraan beban jangka panjang.

Perhitungan perkiraan beban jangka panjang menyangkut masalah-masalah makro ekonomi yang merupakan masalah eksternal perusahaan listrik.

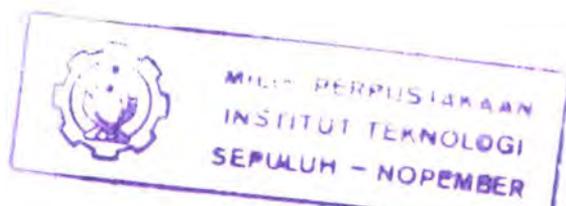
Masalah-masalah makro ekonomi tersebut antara lain adalah pendapatan perkapita penduduk Indonesia. Perkiraan beban jangka panjang untuk jangka waktu diatas satu tahun, dan perhitungan ini hanya diperkirakan beban puncak tertinggi yang akan terjadi dalam sistem tenaga listrik. Perkiraan beban jangka panjang tersebut diperlukan untuk perencanaan pengembangan sistem. Dan masalah perkiraan beban jangka panjang tersebut banyak menyangkut faktor-faktor eksternal dari perusahaan listrik maka penyusunannya memerlukan koordinasi dari pemerintah.

2. Perkiraan beban jangka menengah.

Perkiraan jangka panjang adalah untuk jangka waktu satu bulan sampai dengan satu tahun dimana hal ini diambil dari perkiraan jangka panjang. Sehingga perkiraan jangka panjang tersebut tidak menyimpang dari perkiraan beban untuk jangka panjang. Masalah perkiraan jangka menengah tersebut faktor penentunya adalah dalam hal manajerial perusahaan. Misalkan kemampuan teknis saat penyelesaian proyek pembangunan jaringan listrik yang baru, termasuk kemampuan teknis penyelesaian saluran transmisi dan sistem distribusi.

Perkiraan beban jangka menengah ini ditentukan juga berdasarkan faktor-faktor pemeliharaan unit-unit pembangkit, penggantian unit pembangkit yang telah mencapai batas waktu pemakaian.

3. Perkiraan beban jangka pendek.



Perkiraan beban jangka pendek adalah untuk tenggang waktu dari satu hari sampai satu bulan. Perkiraan beban jangka pendek ini dipengaruhi beberapa faktor misalkan perubahan permintaan daya yang dibutuhkan oleh daerah-daerah tertentu. Dan hal tersebut menyangkut proses penjadwalan dari unit-unit pembangkit yang ada dalam suatu sistem pembangkitan.

2.5. Unit Commitment

Dalam operasi unit-unit pembangkit maka untuk interval waktu tertentu didalam suatu siklus waktu tertentu pula, terjadi koordinasi didalam pembagian beban secara optimum ekonomi. Siklus waktu tertentu ini yang dimaksud ialah, misalkan dalam siklus waktu 24 jam dimana tiap jam, beban listrik dalam sistem tenaga selalu berubah.

Oleh karena itu dalam operasi koordinasi pembagian beban unit-unit pembangkit pada suatu siklus waktu tertentu diperlukan penjadwalan pada operasi dari unit-unit tersebut.

Sedangkan pada penjadwalan ini ditentukan mana unit pembangkit yang *commit* atau 'on' dan mana yang 'off' dalam rangka koordinasi beban pada siklus waktu tertentu.

Penjadwalan tersebut menggunakan suatu pertimbangan teknis dan pertimbangan secara ekonomis. Penjadwalan ini dikenal dengan istilah 'Unit Commitment'.

2.6. Batasan Unit Commitment

Dalam melakukan penjadwalan perlu dibuat suatu batasan-batasan. Batasan-batasan tersebut antara lain adalah :

- Spinning Reserve
- Thermal Unit Constraint
- Biaya Start (Start Up Cost)

Untuk masing-masing batasan yang ada ikuti penjelasan pada halaman berikutnya

2.6.1. Spinning Reserve

Batasan ini membahas tentang cadangan daya yang harus disediakan apabila terdapat unit pembangkit yang mengalami kegagalan saat operasi. Bila beban naik maka frekuensi akan turun yang mengakibatkan terjadinya drop tegangan (pusat pembangkitan mati/trip). Sehingga apabila terjadi unit pembangkit yang gagal saat operasi dimana mengakibatkan turunnya penyediaan beban oleh pusat pembangkitan maka hal tersebut dapat digantikan oleh daya cadangan yang telah dipersiapkan.

Umumnya cadangan daya yang dipergunakan untuk mengganti apabila ada kegagalan operasi besarnya adalah sesuai dengan daya yang dapat dihasilkan oleh unit pembangkit yang besar.

2.6.2. Thermal Unit Constraint

Untuk batasan thermal unit constraint tersebut adalah berupa batas waktu dimana unit baru terhubung 'on' dan unit pembangkit yang baru 'off' tidak dilepas kembali sebelum batas waktunya.

Terdapat dua kategori dari batasan thermal constraint ini yaitu :

1. Minimum up time

Merupakan batasan waktu minimum dimana unit pembangkit yang baru diaktifkan tidak bisa dimatikan sebelum unit tersebut melewati batas 'up time'.

2. Minimum down time

Merupakan batasan waktu minimum dimana unit pembangkit yang baru dimatikan tidak bisa dihubungkan kembali ke sistem / 'on' , sebelum unit tersebut melewati 'minimum down time' yang telah ditentukan.

2.6.3. Biaya Start (Start Up Cost)

Biaya yang diperlukan oleh unit pembangkit dari keadaan tidak beroperasi sampai pembangkit beroperasi.

Terdapat 2 kategori biaya start yaitu :

1. Biaya start pada kondisi dingin

Kondisi ini terjadi karena saat pembangkit dilepas dari sistem (tidak beroperasi) temperatur boiler dibiarkan turun dari temperatur kerjanya,

sehingga pada saat akan beroperasi kembali baru dilakukan pemanasan kembali.

Berikut ini perumusan matematis dari biaya start pada kondisi dingin.

$$\text{Biaya start} = C_c (1 - e^{-t/\alpha}) F + C_f$$

Dengan :

C_c = cold start cost (MBTu)

F = Fuel Cost

C_f = fixed Cost

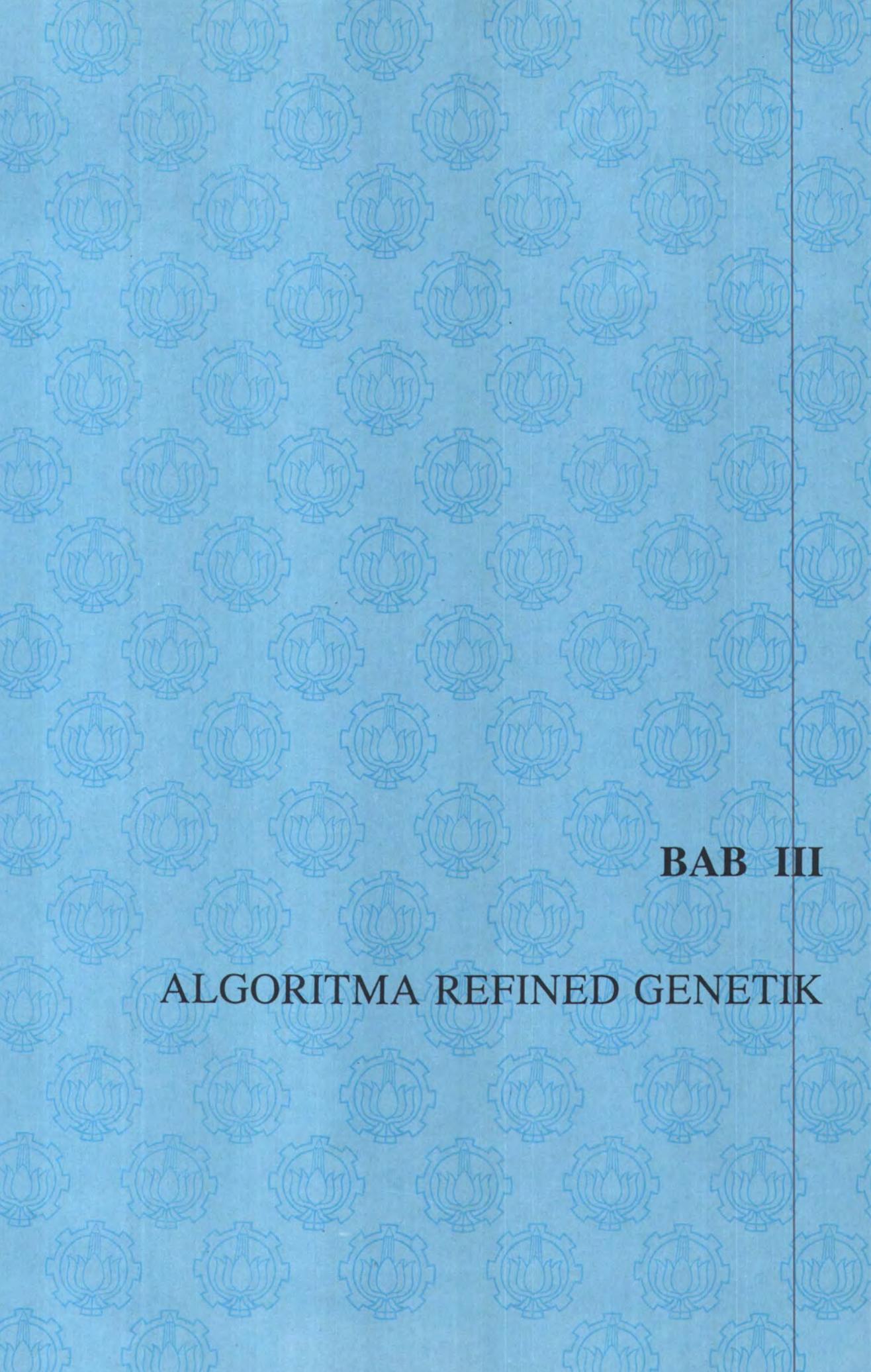
α = thermal time constant

2. Biaya start pada kondisi pada temperatur kerja

Kondisi ini terjadi karena saat pembangkit dilepas dari sistem, temperatur boiler tetap dijaga pada temperatur kerja.

$$\text{Biaya start} = C_t \times t \times F + C_f$$

C_t = biaya untuk mempertahankan temperatur kerja (Mbtu / jam)



BAB III

ALGORITMA REFINED GENETIK

BAB III

ALGORITMA REFINED GENETIK

Bab III ini menjelaskan tentang dasar teori dari algoritma genetik yang selanjutnya diteruskan dengan penjelasan algoritma refined genetik (Refined Genetic Algorithm). Pembahasan mengenai algoritma refined genetik meliputi struktur dasar dari algoritma genetik, representasi problem, problem optimasi genetik, teorema skema, operator algoritma refined genetik yang meliputi tukar silang, pindah silang (crossover), mutasi, serta parameter elitism, probabilitas pindah silang (crossover) dan probabilitas mutasi.

3.1. Struktur dasar algoritma genetik

Dalam memecahkan suatu permasalahan, algoritma genetik berangkat dari himpunan solusi yang dihasilkan secara random. Himpunan solusi yang ada disebut sebagai populasi. Individu-individu yang ada dalam sebuah populasi disebut kromosom yang merupakan bentuk representasi dari solusi. Kromosom-kromosom yang ada mengalami suatu proses evolusi yang berulang. Tiap-tiap proses evolusi yang terjadi pada kromosom tersebut disebut sebagai satu generasi. Pada tiap-tiap generasi, masing-masing kromosom yang ada diproses berdasarkan nilai evaluasi (fitness function). Tiap-tiap kromosom mempunyai fitness function dan kromosom

yang memiliki fitness function yang lebih tinggi dari yang lain mempunyai peluang yang lebih besar untuk dipilih sebagai parents dalam proses berikutnya.

3.2. Representasi Problem

Representasi solusi dari sebuah problem yang ada ke dalam kromosom merupakan persoalan tersendiri dari algoritma genetik. Representasi solusi yang dipakai adalah bergantung pada problem optimasi yang akan diselesaikan. Berbagai macam problem optimasi yang ada dapat direpresentasikan dengan cara *biner*, *real number coding* dan *integer coding*. Pada Tugas akhir ini, representasi solusi dari problem ke dalam kromosom adalah dengan cara biner.

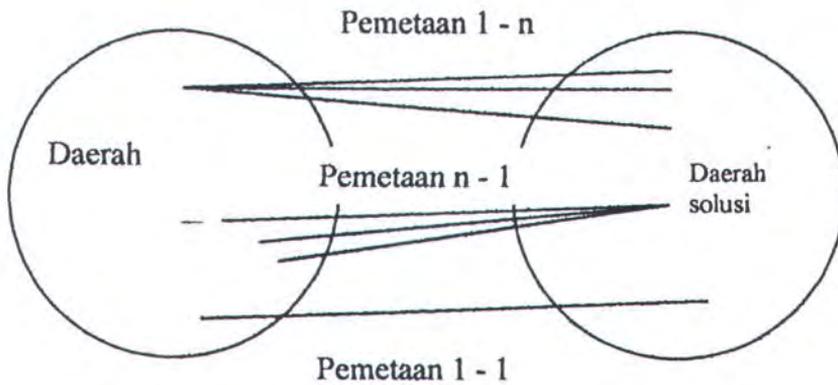
3.3. Pemetaan

Pemetaan dari kromosom ke dalam solusi dapat dikelompokkan menjadi :

- Pemetaan 1 – 1.
- Pemetaan n – 1.
- Pemetaan 1 – n.

Seperti yang terlihat dalam gambar. Pemetaan 1 – 1 adalah merupakan yang terbaik diantara 3 pemetaan yang ada karena untuk satu kromosom tepat dipasangkan dengan satu solusi. Sedangkan pemetaan 1 – n adalah yang pemetaan yang terjelek dikarenakan pemetaan ini merelasikan satu kromosom

dengan beberapa solusi, hal tersebut menyebabkan ketidakjelasan solusi mana yang sebenarnya dirujuk.



Gambar 3.1

3.4. Seleksi

Perlu diingat bahwa seleksi merupakan proses yang bertanggung jawab atas pemilihan kromosom. Pada proses seleksi, bobot seleksi atau yang diistilahkan sebagai *selective pressure* memegang peranan yang sangat penting, sebagai akibat dari bobot seleksi yang kurang tepat maka proses pencarian akan memberikan hasil yang tidak optimal atau premature dan program akan berjalan lambat dari yang seharusnya. Bobot seleksi yang rendah menjadikan proses pencarian menjadi kurang efisien dan sebaliknya yang terjadi jika bobot seleksi meningkat hal tersebut akan diiringi dengan makin kecilnya keragaman dalam populasi yang ada.

Terdapat 3 hal penting yang terdapat dalam proses seleksi :

- Daerah sampling
- Mekanisme sampling
- Probabilitas sampling

Kesemuanya sangat mempengaruhi bobot seleksi dan kinerja dari algoritma genetik.

3.4.1. Daerah sampling

Pada proses seleksi populasi yang dihasilkan untuk generasi berikutnya mungkin berasal dari semua parent dari generasi sebelumnya dan offspring atau sebagian dari keduanya. Asal mula kromosom inilah yang disebut sebagai persoalan daerah sampling.

Sebagai contoh jika terdapat `pop_size` yang merupakan banyaknya kromosom dalam suatu populasi dan `off_size` yang merupakan lebar offspring

Daerah sampling tetap

Holland menyatakan bahwa parent langsung digantikan oleh offspring segera setelah offspring tersebut muncul, hal inilah yang disebut sebagai operasi pergantian generasi (*generational replacement*). Pada daerah sampling tetap ini terjadi operasi seleksi yang tidak pandang bulu sehingga berakibat offspring yang muncul adalah lebih jelek daripada parent.

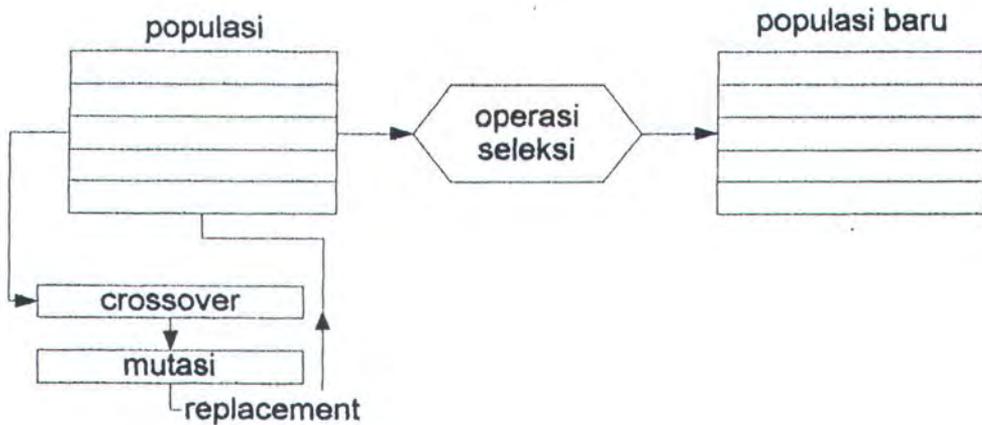
Daerah sampling yang diperbesar

Pada daerah sampling yang diperbesar, parent maupun offspring mempunyai kesempatan yang sama untuk terpilih masuk pada populasi yang baru. Sehingga populasi baru terbentuk berdasarkan seleksi terhadap semua parent dan semua offspring.

3.4.2. Mekanisme sampling

Mekanisme sampling memecahkan permasalahan pemilihan kromosom dari daerah sampling. Terdapat 3 pendekatan yang dipakai yaitu :

- Sampling stokastik
- Sampling deterministik
- Sampling mixed



Gambar 3.2 Proses seleksi pada daerah seleksi tetap

Sampling stokastik

Proses seleksi menentukan banyaknya kromosom yang mempunyai kemungkinan untuk tetap bertahan. Metode yang paling bagus didalam proses seleksi ini ialah *roulette wheel selection*, yang ide dasarnya adalah dengan menentukan besarnya probabilitas seleksi untuk tiap kromosom berdasarkan nilai fitness atau fitness value. Untuk nilai k dengan nilai fitness f_k maka nilai probabilitas seleksi p_k adalah :

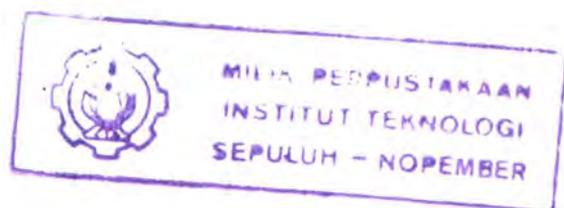
$$p_k = \frac{f_k}{\sum_{j=1}^{popsize} f_j}$$

Sampling deterministik

Pada sampling deterministik, memilih kromosom terbaik pada daerah sampling. Kromosom yang terbaik dalam sebuah populasi dimasukkan pada populasi baru pada generasi berikutnya. Salah satu metode dari sampling deterministik ini ialah metode seleksi elitis atau *elitism (elite copy)*.

Sampling campuran

Menggunakan pendekatan gabungan antara sampling deterministik dan sampling stokastik. Salah satu metode dalam sampling campuran adalah metode seleksi stokastik turnamen (*stochastic tournamen selection*). Dimana seleksi stokastik turnamen ini, sekelompok kromosom dipilih secara random dan kromosom dengan



nilai fitness tertinggi dimasukkan pada populasi berikutnya. Proses ini berlanjut hingga jumlah kromosom dalam populasi baru terpenuhi.

3.4.3. Probabilitas Seleksi

Pada prosedur seleksi, kemungkinan terpilihnya sebuah kromosom adalah sebanding dengan nilai fitness kromosom tersebut. Hal tersebut merupakan kondisi yang tidak tepat. Sebagai contoh, pada awal generasi, terdapat kecenderungan dimana kromosom-kromosom yang mempunyai nilai fitness yang tinggi mempunyai kemungkinan terpilih lebih besar. Dan pada generasi yang mendekati akhir, populasi yang ada telah mempunyai konvergensi yang tinggi sehingga persaingan antar kromosom untuk terpilih menjadi tidak ketat yang akhirnya terjadi pola random dalam pemilihan kromosom-kromosom tersebut.

3.4.4. Skala Fitness (Fitness Scaling)

Secara umum skala fitness (fitness scaling) dapat dituliskan sebagai berikut :

$$f'_k = g(f_k)$$

dengan f'_k merupakan skala fitness dan f_k merupakan fitness yang semula dan fungsi $g(.)$ merupakan mentransformasikan dari nilai fitness f_k menjadi skala fitness.

Dalam probabilitas seleksi terdapat dua macam kategori yaitu :

- Skala Statik.
- Skala dinamis

Lebih lanjut skala dinamis dibagi menjadi beberapa bagian yaitu :

1. Skala parameter yang diadaptasikan sesuai dengan keadaan dari nilai fitness dalam tiap generasi untuk menjaga agar bobot seleksi tetap konstan.
2. Skala parameter yang secara dinamik berubah sesuai dengan penambahan jumlah generasi untuk meningkatkan bobot seleksi.

Berikut ini merupakan contoh dari beberapa skala fitness yaitu :

Skala Linier (Linear scaling)

Linear scaling menyesuaikan nilai fitness terhadap semua kromosom sehingga kromosom yang terbaik menghasilkan nilai yang tepat dari populasi yang ada.

Fungsi $g(.)$ merupakan bentuk transformasi secara linear. Dan perumusan secara matematis adalah sebagai berikut :

$$F'_k = a \times f_k + b$$

Dimana parameter a dan b dipilih sehingga rata-rata kromosom menerima satu populasi dari keseluruhan.

Skala linier dinamik (Dynamic Linear Scaling)

Parameter b bervariasi dalam tiap generasi, dengan demikian skala dinamik mengikuti aturan sebagai berikut :

$$F'_k = a \times f_k + b$$

Dimana b_1 mengambil nilai fitness yang minimal dari populasi $b_1 = -f_{\min}$

Berikut ini beberapa contoh dari metode skala :

Windowing

Jika terdapat 2 kromosom dengan nilai fitness 2 dan 1, maka kromosom pertama akan mendapatkan nilai 3 dan kromosom yang kedua akan mendapatkan nilai 2.

Teknik windowing ini memperkenalkan sebuah konsep yang dinamakan moving baseline dalam seleksi yang proporsional yang menjaga agar bobot seleksi yang ada menjadi konstan. Dan metode windowing ini masih termasuk dalam jenis :

$$F'_k = f_k - f_w$$

Dimana w merupakan ukuran dari *window* dan biasanya berkisar antara nilai 2 sampai 10.

Normalizing

Merupakan salah satu jenis skala dinamis yang diperkenalkan oleh Cheng dan Gen. Untuk *problem maksimisasi* maka memakai bentuk persamaan sebagai berikut :

$$f'_k = \frac{f_k - f_{\min} + \gamma}{f_{\max} - f_{\min} + \gamma}$$

dimana :

f_{\max} : nilai fitness (fitness asal) terbaik dalam satu populasi.

f_{\min} : nilai fitness (fitness asal) terburuk dalam satu populasi.

γ : bilangan real positif yang kecil antara 0 dan 1.

Metode ini merupakan teknik spesial dari metode windowing dimana ukuran window dalam normalizing ini adalah 1.

Tujuan dari metode ini adalah :

1. Mencegah terjadinya bilangan pembagi yang bernilai nol.
2. Memungkinkan terjadinya seleksi fitness yang proporsional untuk menghasilkan seleksi random yang murni.

Untuk *problem minimisasi* maka dipakai persamaan matematis sebagai berikut :

$$f'_k = \frac{f_{\max} - f_k + \gamma}{f_{\max} - f_{\min} + \gamma}$$

Sigma Truncation

Forrest memperkenalkan metode ini untuk mengembangkan linear scaling agar dicapai kesesuaian antara nilai negatif dan problem yang berhubungan dengan informasi pada pemetaan.

Metode ini kemudian di formulasikan kembali oleh Goldberg sebagai berikut :

$$f'_k = f_k - (\bar{f} - c \times \sigma)$$

dimana :

c = bilangan bulat yang kecil

σ = nilai standar deviasi dari populasi

\bar{f} = rata-rata nilai fitness asli

dengan syarat jika nilai fitness terskala f'_k yang negatif diisi dengan nol.

Power Law Scaling

Metode ini diperkenalkan oleh Gillies, dimana fungsi $g(.)$ merupakan bentuk khusus, pangkat dari nilai fitness yang asli. Berikut ini bentuk perumusan dari metode Power Law Scaling :

$$f'_k = f_k^\alpha$$

secara umum, nilai α bergantung pada masalah yang ada, Gillies menentukan bahwa nilai α adalah 1,005.

Dalam skala fitness jarak antara kromosom yang terbaik dan kromosom yang terburuk bertambah sesuai dengan pertambahan nilai α . Jika α mendekati nilai nol maka gap atau jarak juga mendekati nilai nol dan sampling menjadi random. Jika $\alpha > 1$ maka jarak akan diperbesar dan sampling akan diarahkan pada kromosom yang lebih bagus. Saat dijalankan, nilai α dapat disesuaikan secara dinamis dengan melebarkan atau menyusutkan range sesuai dengan keperluan.

Versi lain dari power law scaling adalah sebagai berikut :

$$f'_k = (a \times f_k + b)^\alpha$$

Logarithmic Scaling

Metode ini diperkenalkan oleh Fitzpatrick dan Grefenstette. Tujuannya adalah untuk pemetaan dari fungsi obyektif pada problem minimisasi dimana fungsi $g(\cdot)$ menerapkan bentuk logarithmic sebagai berikut :

$$f'_k = b - \log(f_k)$$

dimana nilai b dipilih lebih besar dari nilai $\log(f_k)$ yang lain.

Tekanan seleksi berhubungan dengan penyebaran dari nilai fitness dalam populasi. Sigma scalling membeberkan pengamatan tersebut dan menset nilai faktor skala sebagai beda antara mean dan standar deviasi. Kromosom dibawah nilai ini diberi nilai fitness nol. Metode ini membantu mengatasi kemungkinan masalah yang timbul oleh kromosom yang jelek yang menyebabkan penyebaran dari fitness menjadi sangat besar, sehingga mengurangi keakuratan seleksi.

3.5. Problem Optimasi Genetik

Secara umum algoritma genetik merupakan sebuah teknik optimasi yang potensial dalam memecahkan berbagai persoalan optimasi yang ada. Beberapa jenis persoalan optimasi yang ada adalah optimasi unconstrained, nonlinear programming, stochastic programming, goal programming dan interval programming. Dibawah ini hanya akan

diuraikan beberapa diantaranya yang berhubungan dengan tugas akhir ini, yaitu nonlinear programming.

- **Nonlinear Programming (constrained optimization)**

Secara umum nonlinear programming dapat dituliskan secara matematis sebagai berikut :

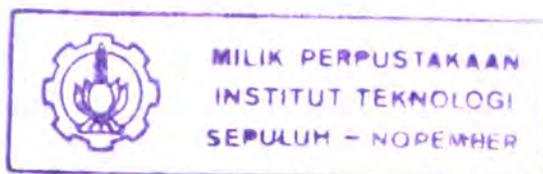
$$\text{Max } f(x)$$

$$\text{Dengan } g_i(x) \leq 0, \quad i = 1, 2, \dots, m_1$$

$$h_i(x) = 0, \quad i = m_1 + 1, \dots, m (= m_1 + m_2)$$

$$x \in X$$

dimana $f, g_1, g_2, \dots, g_{m_1}, h_{m_1+1}, h_{m_1+2}$ adalah fungsi dengan nilai real yang didefinisikan pada E^n , X adalah sebuah subset of E^n dan x adalah merupakan n dimensi vektor bertipe real dengan komponen X_1, X_2, \dots, X_n . $F(x)$ adalah suatu fungsi yang biasa disebut *objective function* atau fungsi obyektif yang akan diminimalkan, dan tiap $g_i(x) \leq 0$ disebut *inequality constraint* sedangkan untuk tiap $h_i(x) = 0$ disebut *equality constraint*. Dan variabel X merupakan batas bawah dan atas dari variabel dimana biasa disebut *domain constraint*. Sedangkan untuk variabel $x \in X$ yang merupakan nilai dari semua constraint yang ada disebut sebagai *feasible solution*. Dan selanjutnya nonlinear programming dimaksudkan untuk mencari titik x sehingga $f(x) \leq f(x)$ untuk tiap feasible point x . Yang akhirnya point tersebut disebut sebagai solusi optimal atau *optimal solution*.



3.6. Teorema Skema

Dasar teori dari algoritma genetik bermula dari representasi string biner. Hal tersebut dijelaskan dalam teori skema yang diperkenalkan oleh Holland J.H (1975).

Sebuah skema dibentuk dari simbol $\{0, 1, \#\}$. Dan skema tersebut merepresentasikan semua string yang mempunyai posisi dan nilai yang sama kecuali untuk simbol # (don't care). Sebagai contoh, skema dengan panjang 10 yaitu $(\# 111100100)$ sesuai dengan 2 string sebagai berikut :

$$\{(0111100100), (1111100100)\},$$

sebuah contoh berikutnya misal skema dengan nilai $(\#1\#1100100)$ sesuai dengan 4 string sebagai berikut :

$$\{(0101100100), (0111100100), (1101100100), (1111100100)\}$$

Dengan demikian jika terdapat sebanyak n simbol don't care dalam sebuah skema maka akan ada 2^n string yang cocok / sesuai. Selain itu, string dengan panjang m cocok dengan 2^m skema.

Banyaknya bit 0 dan 1 yang terdapat dalam sebuah skema dinyatakan dengan notasi $o(S)$, misal skema $S_1 = (\###001\#110)$ maka $o(S_1) = 6$ dan skema $S_2 = (11101\##001)$ mempunyai $o(S_2) = 8$.

Panjang sebuah skema dinyatakan dengan sebuah notasi $\delta(S)$ yang merupakan jarak antara string (0 atau 1) yang pertama dengan string yang terakhir, dapat disimpulkan bahwa $\delta(S_1) = 6$ dan $\delta(S_2) = 9$.

Nilai fitness saat waktu t dinotasikan sebagai $eval(S,t)$. Dan didefinisikan sebagai nilai rata-rata fitness untuk semua string dalam suatu populasi yang sesuai dengan skema S . Dengan asumsi bahwa terdapat p string $\{V_1, \dots, V_p\}$ dalam sebuah populasi yang cocok dengan skema S pada waktu t maka :

$$Eval(S,t) = \frac{\sum_{j=1}^p eval(V_{ij})}{p}$$

3.7. Parameter dan Operator Algoritma Refined Genetik

Algoritma refined genetik menggunakan beberapa operator yaitu :

- Seleksi (Selection).
- Pindah silang (Crossover).
- Mutasi (Mutation).

Selain itu terdapat parameter sebagai berikut

- Elitism.
- Probabilitas Seleksi.
- Probabilitas mutasi (mutation).
- Probability pindah silang (crossover).

3.7.1. Seleksi

Proses seleksi terjadi didahului oleh adanya sebuah proses awalan yang dinamakan roulette wheel sebagai berikut :

Perhitungan fitness value / eval(V_i) dari tiap-tiap kromosom yang ada V_i

Dimana ($i = 1, \dots, \text{pop_size}$).

- Tentukan nilai total fitness dari keseluruhan populasi dengan rumus :

$$F = \sum \text{eval}(V_i). \text{ Dengan } V_i (i=1, \dots, \text{pop_size}).$$

- Hitung probabilitas seleksi P_i untuk tiap-tiap kromosom V_i ($i = 1, \dots, \text{pop_size}$) :

$$P_i = \text{eval}(V_i) / F$$

- Hitung nilai probabilitas kumulatif Q_i untuk tiap-tiap kromosom V_i ($i = 1, \dots, \text{pop_size}$) :

$$Q_i = \sum P_j. \text{ Dengan } P_j (j = 1, \dots, i).$$

Setelah proses perhitungan diatas selesai maka kita selanjutnya dapat melakukan proses seleksi sebanyak pop_size yang ada dan tiap kali dilakukan pemilihan kromosom untuk membentuk suatu populasi yang baru, mengikuti aturan sebagai berikut :

- Hasilkan bilangan random (float) r sebanyak pop_size dengan nilai antara $[0 \dots 1]$.
- Jika $r < q_1$ maka pilih kromosom (V_1). Begitu juga pilih kromosom ke- i (V_i) dimana ($2 \leq i \leq \text{pop_size}$) sehingga $q_{i-1} < r \leq q_i$.

Dapat terjadi bahwa beberapa kromosom terpilih lebih dari satu kali. Secara tidak langsung kromosom yang baik akan terpilih pada populasi berikutnya.

3.7.2. Pindah Silang (Crossover)

Populasi yang dihasilkan setelah proses seleksi diatas, selanjutnya mengalami proses pindah silang (crossover). Dalam proses ini dipakai sebuah parameter yaitu probabilitas crossover (P_c). Jumlah kromosom yang mengalami crossover dapat ditentukan dari $P_c \cdot \text{pop_size}$.

Untuk tiap pasang kromosom yang ada dalam populasi diproses sebagai berikut :

- Hasilkan bilangan secara random sebanyak pop_size yang mempunyai nilai antara $[0 \dots 1]$.
- Jika $r_i < P_c$, pilih kromosom yang berindeks i (V_i) untuk proses crossover.

Setelah proses diatas selesai maka kromosom yang terpilih dipasangkan, berikutnya dilakukan proses crossover terhadap kromosom-kromosom yang terpilih dengan mengikuti aturan sebagai berikut :

- Hasilkan random bit (R_i) sejumlah ukuran kromosom.
- Untuk child yang pertama, jika nilai random bit adalah '1' maka parent pertama memberikan bit tersebut pada child pertama, dan jika random bit adalah '0' maka parent kedua pada posisi bit tersebut akan memberikan nilai

bit itu pada child pertama. Proses ini berlanjut terus untuk setiap bit yang ada sampai sejumlah ukuran kromosom.

- Untuk child yang kedua, proses yang dialami merupakan komplemen dari proses child yang pertama. Jika random bit adalah '1' maka child yang kedua pada posisi tersebut akan mendapatkan nilai bit dari parent yang kedua (pada posisi yang sama). Sebaliknya jika random bit bernilai '0' maka child kedua mendapatkan nilai '0' tersebut dari parent yang pertama.

Sebagai ilustrasi maka dapat dituliskan sebagai contoh seperti dibawah ini :

Random bit yang membentuk patern adalah :

101001

Parent 1 : 000010

Parent 2 : 111001

Child 1 : 010000

Child 2 : 101011

Jika jumlah kromosom (yang terpilih) yang dihasilkan dari perkalian $P_c \cdot pop_size$ menghasilkan bilangan genap maka tiap-tiap kromosom mendapat pasangan untuk melakukan crossover, jika ganjil maka kita dapat menambah satu kromosom extra atau menghilangkan salah satu kromosom (yang terpilih).

Dengan demikian proses crossover sudah terpenuhi dan menghasilkan populasi yang baru untuk selanjutnya masuk pada proses mutasi.

3.7.3. Mutasi (Mutation)

Pada proses mutasi digunakan parameter probabilitas mutasi (P_m) dan proses mutasi tersebut terjadi pada tiap bit untuk kromosom yang terpilih.

Untuk tiap kromosom pada populasi yang dihasilkan (setelah proses pindah silang), tiap bit yang ada melalui proses sebagai berikut :

- Hasilkan bilangan random (float) r sebanyak $\text{chrom_size} * \text{pop_size}$ yang mempunyai nilai antara $[0... 1]$.
- Jika $r_i < P_m$ maka mutasikan bit pada indeks tersebut dengan cara lakukan komplemen pada bit tersebut.

3.7.4. Elitism

Dalam mekanisme elitism, kromosom dengan nilai terbaik dari generasi sebelumnya dimasukkan menjadi kromosom dalam generasi berikutnya. Mekanisme ini menjamin bahwa kromosom yang terbaik tidak akan hilang selama proses iterasi dari operator yang ada, dan diakhir program akan muncul sebagai solusi yang terbaik.

3.7.5. Probabilitas Crossover dan Probabilitas Mutation

Pada algoritma simple genetik, nilai probabilitas crossover dan probabilitas mutasi besarnya tetap. Sedangkan untuk algoritma refined genetik besar nilai probabilitas crossover dan probabilitas mutasi berubah untuk tiap generasi, pada

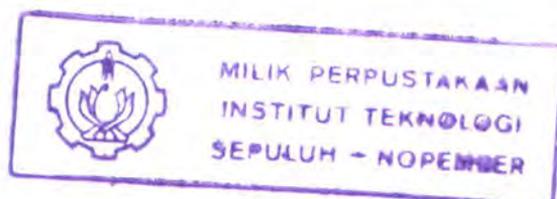
generasi pertama nilai tersebut sesuai dengan input pertama kali dan untuk generasi selanjutnya nilai probabilitas crossover secara eksponensial berkurang sedangkan untuk probabilitas mutasi secara eksponensial nilainya bertambah. Untuk nilai pembatas dari keduanya diatur agar tidak melebihi standar.

3.8. Karakteristik Algoritma Genetik

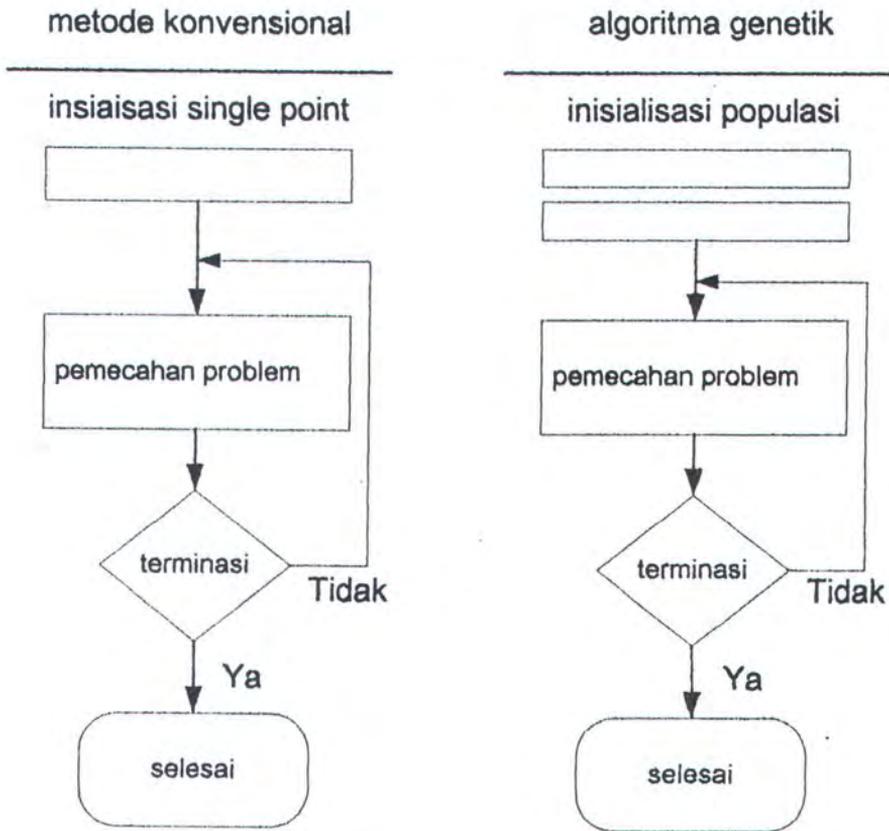
3.8.1. Eksplorasi dan Eksploitasi

Pada problem optimasi terdapat 2 strategi yang perlu diperhatikan yaitu eksplorasi daerah solusi dan eksploitasi solusi yang terbaik. Algoritma genetik merupakan metode pencarian yang menggabungkan teknik pencarian directed dan stochastic dimana terjadi keseimbangan antara eksplorasi daerah pencarian dan eksploitasi solusi yang terbaik.

Secara umum, algoritma yang dipakai untuk memecahkan permasalahan optimasi menggunakan komputasi yang berurutan sehingga mencapai suatu pemecahan yang optimal. Pada metode pencarian klasik digunakan suatu pendekatan perhitungan secara point to point yang dapat terjebak pada solusi yang disebut *local optima*. Hal tersebut terjadi karena metode pencarian klasik menggunakan pendekatan single point pada daerah pencarian. Lain halnya apabila dibandingkan dengan algoritma genetik yang menggunakan pendekatan daerah solusi dengan cara membentuk suatu populasi dari potensial solusi yang ada. Pada tiap generasi terjadi



suatu mekanisme untuk mempertahankan solusi yang terbaik dengan suatu pola pendekatan secara probabilistik.



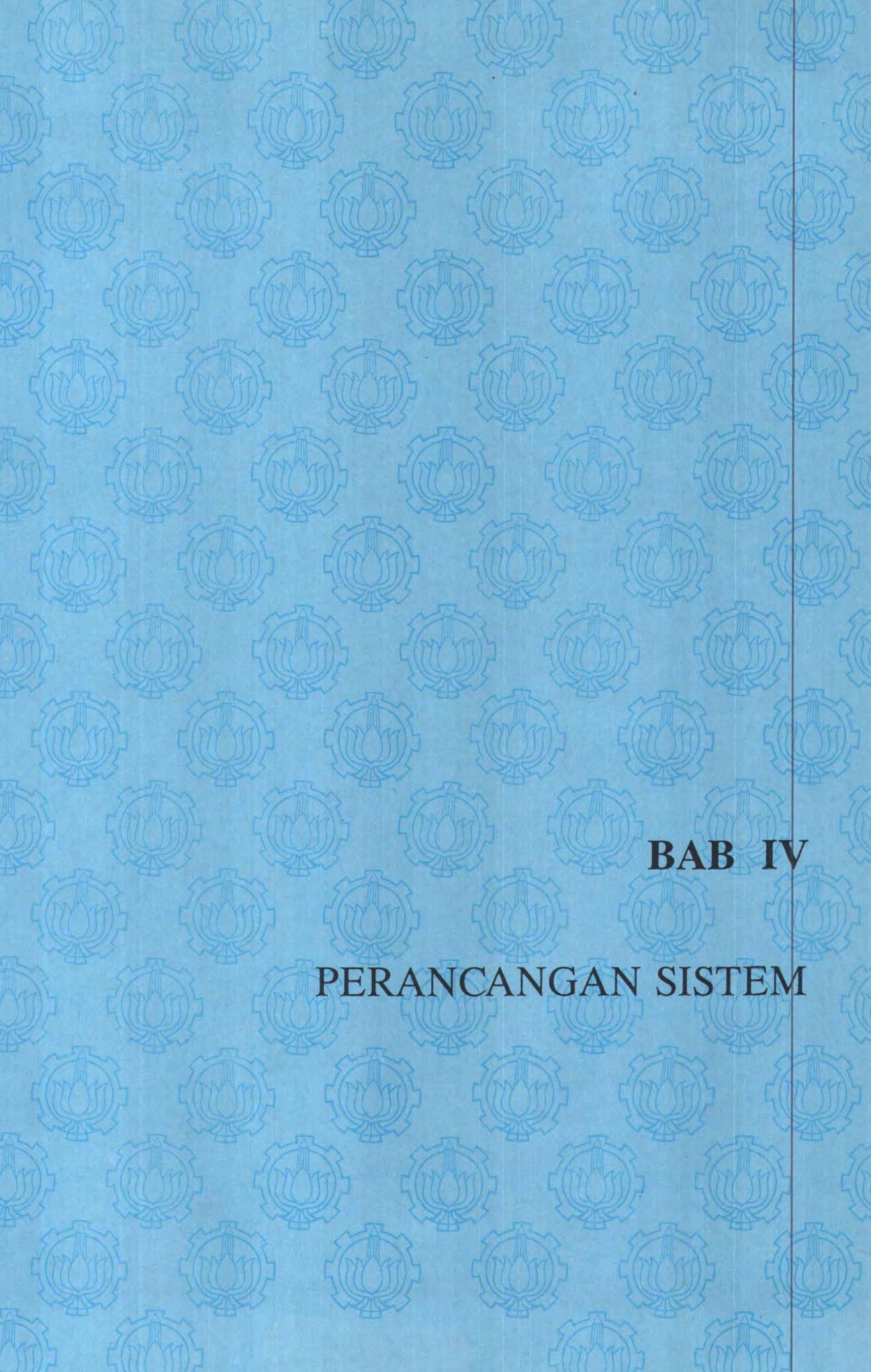
Gambar 3.3 Perbandingan Algoritma

3.9. Proses Generasi

Dari populasi yang ada, operator reproduksi adalah yang memegang pokok bahasan. Dan operator-operator tersebut tersebut merupakan pencerminan dari teori Darwin tentang seleksi alam. Oleh karena itu peluang dari masing-masing kromosom untuk terseleksi hanya bergantung dari nilai fitness masing-masing kromosom tersebut.

Proses seleksi mempromosikan kromosom-kromosom yang baik, dan untuk menemukan kromosom-kromosom yang lebih baik kualitasnya maka kromosom-kromosom yang baru dan lebih bagus harus dibuat dan hal tersebut merupakan tugas dari operator pindah silang (crossover). Proses seleksi dan crossover tersebut bersama-sama membuat agar generasi yang sedang diproses menghasilkan generasi yang terdiri dari kromosom-kromosom dengan nilai yang lebih baik.

Mutasi meningkatkan keragaman dalam populasi. Dengan maksud sebagai berikut, jika sebuah kromosom yang mencerminkan suatu pemecahan yang suboptimal, jumlahnya telah mencapai sebagian dari jumlah populasi yang ada maka dapat diprediksi bahwa akan terjadi *local optima*. Sehingga dengan adanya mutasi diharapkan muncul kromosom baru sebagai alternatif dari dominasi kromosom-kromosom super.



BAB IV

PERANCANGAN SISTEM

BAB IV

PERANCANGAN SISTEM

Pada tugas akhir ini disusun sebuah program yang digunakan menyelesaikan persoalan unit commitment menggunakan algoritma refined genetik. Pada intinya unit commitment adalah koordinasi antara unit pembangkit dalam suatu sistem pembangkitan yang secara ekonomis mencapai kondisi optimum dalam suatu interval waktu tertentu pada siklus waktu tertentu. Dan pengertian kondisi optimum secara ekonomis ini ialah cost yang minimum dapat dicapai dalam penyediaan daya, hasil dari koordinasi antar unit pembangkit sebagai akibat dari perubahan permintaan daya atau load demand dalam siklus waktu tertentu. Sehingga cost yang dicapai adalah optimum dengan penyediaan daya yang sesuai dengan permintaan daya atau load demand.

4.1. Fungsi Obyektif Economic Dispatch (Objectif function)

Implementasi fungsi obyektif dalam program refined genetik direalisasikan ke dalam fungsi fitness (fitness function). Fungsi fitness merupakan fungsi yang sesungguhnya yang akan diproses kedalam operator-operator refined genetik. Dan fungsi inilah yang merupakan penentu dalam mencapai solusi optimal. Oleh karena itu batasan-batasan yang ada dalam fungsi obyektif dari economic dispatch dengan sendirinya masuk kedalam fungsi fitness.

Untuk memecahkan permasalahan optimasi unit pembangkit dengan menggunakan algoritma refined genetik tersebut maka pada tugas akhir ini digunakan pendekatan karakteristik input output.

Dengan pendekatan karakteristik input-output tersebut maka fungsi obyektif yang digunakan adalah sebagai berikut :

$$OBJ = \text{MIN} \left[\sum_{i=1}^n F_i(P_i) \right] \quad (4.1)$$

dan batasan untuk persamaan diatas adalah sebagai berikut :

$$\sum_{i=1}^n P_i = P_L + P_{TL} \quad (4.2)$$

dan

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (4.3)$$

Dengan menggabungkan fungsi obyektif yang ada pada persamaan (4.1) dan batasan-batasan yang ada pada persamaan (4.2) dan (4.3) ke dalam fungsi fitness, dan nilai inilah yang diminimalkan dengan meminimalkan pula batasan-batasan yang ada. Batasan-batasan yang ada pada persamaan (4.2) dapat dituliskan kembali sebagai berikut :

$$f_{err} = \left| \sum_{i=1}^n P_i - P_L - P_{TL} \right| \quad (4.4)$$

Nilai fitness yang ada didalam kromosom akan dibandingkan dengan nilai fitness dari kromosom yang lain dalam populasi yang sama, selanjutnya nilai obyektif yang

diwakili oleh fungsi fitness tersebut dibandingkan dengan nilai obyektif dalam satu populasi dan nilai pembatas pada persamaan (4.2) dan (4.3) juga dibandingkan. Dan nilai fitness berdasarkan pada persentase sebagai berikut :

$$\%obj = \frac{stringobj - minobj}{maxobj - minobj} \quad (4.5)$$

dan

$$\%lerr = \frac{stringerr - minerr}{maxerr - minerr} \quad (4.6)$$

dimana :

$\%obj$ = persentase objective function masing-masing kromosom.

$Stringobj$ = string dari objective function pada pers. (1) dari masing-masing kromosom

$minobj$ = nilai objective function yang minimum dalam satu populasi dari keseluruhan kromosom yang ada

$maxobj$ = nilai objective function yang maksimum dalam satu populasi dari keseluruhan kromosom yang ada

$\%lerr$ = persentase constrain error

$stringlerr$ = string error dari persamaan masing-masing kromosom (4)

$minerr$ = constraint error yang minimum dalam satu populasi dari keseluruhan kromosom yang ada

maxerr = constraint error yang maksimum dalam satu populasi

Agar jumlah daya yang dihasilkan saat proses optimasi diluar kapasitas maksimum dan minimum dari masing-masing unit maka dapat ditambahkan range error untuk masing-masing kromosom dalam satu populasi yaitu sebagai berikut :

$$\text{Range Error} = \frac{\text{RErr} - \text{MinRErr}}{\text{MaxRErr} - \text{MinRErr}} \quad (4.7)$$

dengan :

MaxRErr = kromosom yang mempunyai error maksimum daya dalam satu populasi

MinRErr = kromosom yang mempunyai nilai error minimum (daya) dalam satu populasi

RErr = error pada masing-masing kromosom
misal diketahui 3 buah unit pembangkit

| | | |
|--------------|--------|-----------------------|
| 100 – 600 MW | unit 1 | } dalam satu kromosom |
| 100 – 400 MW | unit 2 | |
| 50 – 200 MW | unit 3 | |

untuk masing-masing unit jika terdapat kelebihan daya atau kekurangan daya pada proses komputasi maka kesemuanya dijumlahkan (absolut) dan menjadi nilai variabel RErr.

selanjutnya nilai fitness yang merupakan penterjemahan dari persamaan (4.1) dapat dituliskan sebagai berikut :

$$\text{fitness} = \text{MIN} [\%obj + \%lerr + \%range\ error] \quad (4.8)$$

karena parameter yang dipakai adalah dalam bentuk persentase maka, solusi minimum dicapai dengan mengurangi persamaan (4.7) dengan 1 sebagai berikut :

$$\text{Fitnes} = (1 - \%obj) + (1 - \%lerr) + (1 - \%range\ error) \quad (4.9)$$

4.2. Representasi Kromosom Dalam Economic Dispatch

Implementasi dari optimasi unit pembangkit dengan menggunakan algoritma refined genetik diwujudkan dalam sebuah fitness function, dimana fitness function tersebut merupakan nilai solusi yang ada.

Berikut ini penjelasan secara urut representasi kromosom :

Misalkan :

Diketahui terdapat n buah unit pembangkit dimana **output** dari n buah unit pembangkit yang ada diwakili oleh / sebagai 1 kromosom dengan panjang kromosom adalah $n \times B$ dimana B adalah 2^B yang merupakan total demand dari keseluruhan unit pembangkit. Jika terdapat $n = 3$ (unit pembangkit) dan permintaan (total demand) beban adalah 1024 MW maka dapat dituliskan dengan 2^{10} sehingga panjang kromosom adalah $10 \times n$. Nilai tiap bit dalam kromosom-kromosom tersebut pada awalnya dihasilkan secara random dan banyaknya kromosom tersebut sesuai dengan input ukuran dari populasi yang diminta (misalkan dalam satu populasi terdapat 100 kromosom).



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



Gambar 4.1 Representasi kromosom

4.2.1. Jumlah Kromosom (Population Size)

Jumlah kromosom dalam populasi adalah merupakan banyaknya kromosom dalam satu populasi dimana masing-masing kromosom tersebut mewakili output dari unit pembangkit (gambar 4.1). Kromosom-kromosom tersebut terdiri dari beberapa alela yang bernilai boolean (0 atau 1). Nilai pada masing-masing alela yang membentuk sebuah kromosom pada mulanya dihasilkan secara random. Proses inialisasi pada tiap alela dilakukan secara iterasi pada awal program. Panjang kromosom bergantung dari data dari unit pembangkit yang akan dioptimalkan. Agar lebih terperinci hal tersebut secara alur dapat dituliskan sebagai berikut :

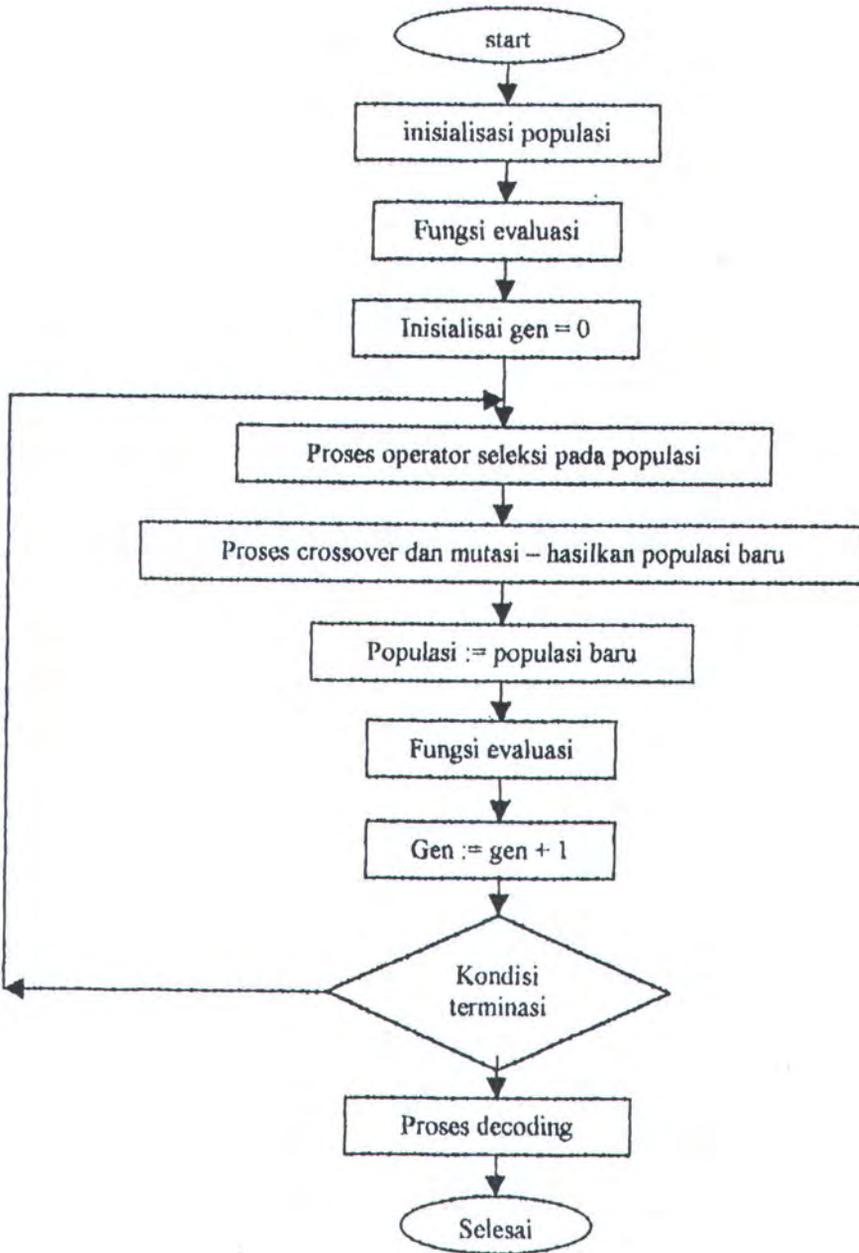
1. Tentukan jumlah unit pembangkit (input).
2. Tentukan daya maksimum dan minimum dari masing-masing unit pembangkit.
3. Lakukan proses pencarian kapasitas daya maksimum dan kapasitas daya minimum dari keseluruhan (total) unit pembangkit yang aktif.

4. Tentukan selisih antara kapasitas daya maksimum dan kapasitas daya minimum dari keseluruhan unit pembangkit yang aktif .
5. Konversikan nilai yang didapat dari no 4, ke dalam basis biner (2^n).
6. Dapatkan panjang kromosom.

Panjang kromosom = $n \times$ jumlah unit pembangkit

4.2.2. Jumlah Generasi (Number of Generation)

Kromosom-kromosom dalam satu populasi selanjutnya melalui beberapa proses yaitu pindah silang, mutasi, replacement dan proses evaluasi. Jumlah generasi menentukan berapa kali proses yang harus dilalui oleh populasi untuk lebih jelasnya lihat bagan alur Gambar 4.2.



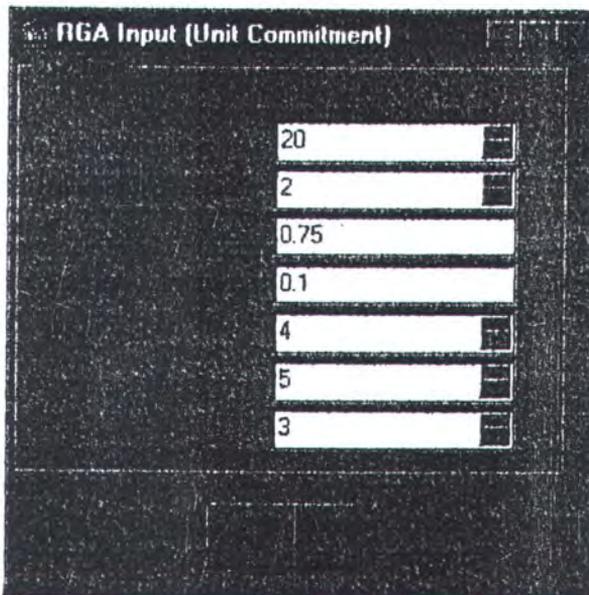
Gambar 4.2 Bagan Alur Program

4.3. Data Masukan

Data masukan yang dipergunakan adalah sebagai berikut :

1. Jumlah kromosom dalam populasi (Population size).
2. Jumlah generasi (Number of Generation).
3. Probabilitas Pindah Silang (Probabilitas Crossover).
4. Probabilitas mutasi (Probabilitas Mutasi).
5. Akurasi (Acuracy).

Berikut ini merupakan contoh dari tampilan data input (gambar 4.3) :



The image shows a software window titled "RGA Input (Unit Commitment)". It contains seven input fields, each with a numerical value and a small arrow icon on the right side. The values are: 20, 2, 0.75, 0.1, 4, 5, and 3.

| Parameter | Value |
|------------------------|-------|
| Population size | 20 |
| Number of Generation | 2 |
| Probabilitas Crossover | 0.75 |
| Probabilitas Mutasi | 0.1 |
| Acuracy | 4 |
| | 5 |
| | 3 |

Gambar 4.3 Tampilan Input

4.4. Deskripsi dan Implementasi Obyek

Terdapat 3 obyek dalam program refined genetik tersebut yaitu :

- Obyek TChromosome.
- Obyek TPopulation.
- Obyek TPowerDispatchRGA dan TCommitment.

Adapun pemakaian obyek-obyek tersebut diatas dalam implementasi adalah disebabkan karena beberapa faktor sebagai berikut :

- Kemudahan dalam *coding* disebabkan karena antara lain terdapat enkapsulasi, inheritance, hal ini mendukung pada implementasi program-program yang secara teknis membutuhkan pembagian proses.
- Pelacakan kesalahan yang terdapat dalam program menjadi lebih mudah karena pengecekan menjadi lebih terarah pada tiap-tiap obyek.

4.4.1. Obyek Tchromosome

Untuk lebih jelasnya representasi obyek pada TChromosome adalah :

```
TChromosome = class(Tobject)
```

```
private
```

```
pBits : TByteList;
```

```
Procedure WAlella(x : longint; Value : Boolean);
```

```
Function RAlella(x : longint) : Boolean;
```

```
Public
```

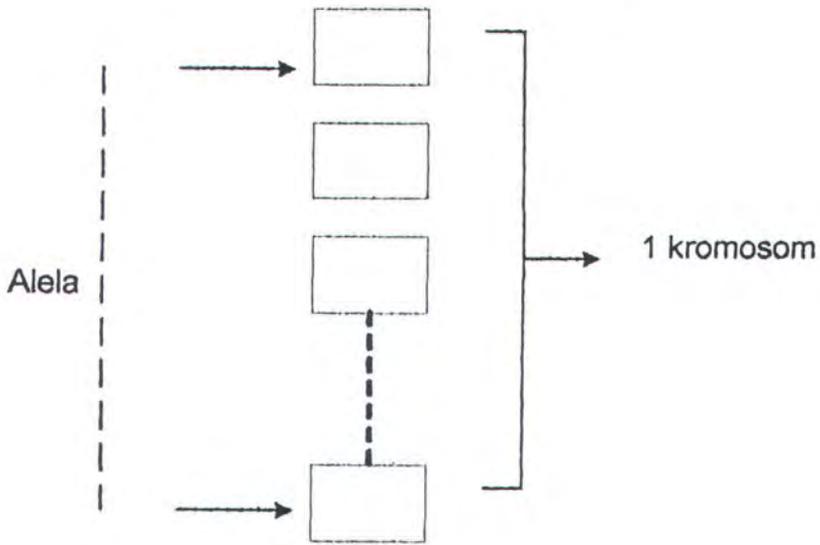
```
Property Alella[x : TIndex] : Byte Read RAlella Write WAlella; Default;
```



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

1. Proses *Write Chromosome*.

Chromosome terdiri dari sejumlah alela. Alela tersebut ditampung dalam suatu variabel yang dinamakan Tbits yang bernilai boolean. Gambar 4.4 merupakan ilustrasi kromosom



Gambar 4.4 Ilustrasi kromosom

2. Proses *Read Chromosome*.

Untuk pembacaan data yang telah tertulis dalam masing-masing alela maka nilai tiap-tiap alela dalam kromosom dibaca melalui sebuah fungsi yang menghasilkan nilai bertipe boolean.

4.4.2. Obyek Tpopulation

Populasi terdiri dari beberapa kromosom. Kromosom-kromosom dalam populasi tersebut disimpan dalam array dinamis yang dinamakan TstringList dengan tipe data string.

Representasi obyek untuk Tpopulation adalah sebagai berikut :

TPopulation ⇒ Class

Private

iPopulation : Array [0..MaxPopulasi] Of TChromosome;

iCount : TIndex;

Procedure WChromosome(x : TIndex; Value : TChromosome);

Function RChromosome(x : TIndex) : TChromosome;

Public

iPopulationSize : TIndex;

Constructor Create(PopulationSize, ChromosomeSize : TIndex);

Destructor Destroy; Override;

Procedure Free;

Property Chromosomes[x : TIndex] : TChromosome Read RChromosome Write WChromosome;

Property Count : TIndex Read iCount;

- Proses Tulis (Tpopulation)

Dilakukan oleh sebuah prosedur *WChromosome*. Masing-masing kromosom yang sudah terbentuk pada obyek TChromosome selanjutnya ditulis kedalam obyek Tpopulation. Turunan dari obyek tersebut adalah obyek Tchromosome, selanjutnya obyek inilah yang dituliskan kedalam obyek TPopulation dengan array dinamis berupa TstringList. Disinilah pentingnya pembuatan obyek dalam implementasi program dari suatu permasalahan

sehingga proses-proses yang ada dapat terbagi sesuai dengan porsi dari masing-masing obyek

- Proses Baca (*Tpopulation*)

Langkah pada proses baca kromosom dimulai dengan tiap bit yang ada pada obyek *TChromosome* kemudian setelah satu kromosom selesai dilanjutkan dengan kromosom lain sampai pembacaan kromosom dalam satu populasi terpenuhi.

4.4.3. Obyek *TCommitment* dan *TPowerDispatchRGA*

Dalam Obyek *TPowerDispatchRGA* terdapat prosedur dan fungsi dari operator-operator refined genetik algoritma yaitu :

1. Selection
2. CrossOver
3. Mutation
4. Replacement
5. Evaluation

4.5. Fungsi Obyektif Unit Commitment

Fungsi Obyektif pada unit commitment secara matematis dapat dituliskan -

sebagai berikut :

$$\text{OBJCost UC} = \sum_{i=1}^n \left[\sum_{j=1}^{n = \text{jumlah generator}} \text{Run}_{\text{Cost}} \right] + \sum_{j=1}^{n = \text{jumlah generator}} \text{StartUp}_{\text{Cost}} + \sum_{L=1}^{n = \text{jumlah generator}} \text{ShutDown}_{\text{Cost}} \left. \right]$$

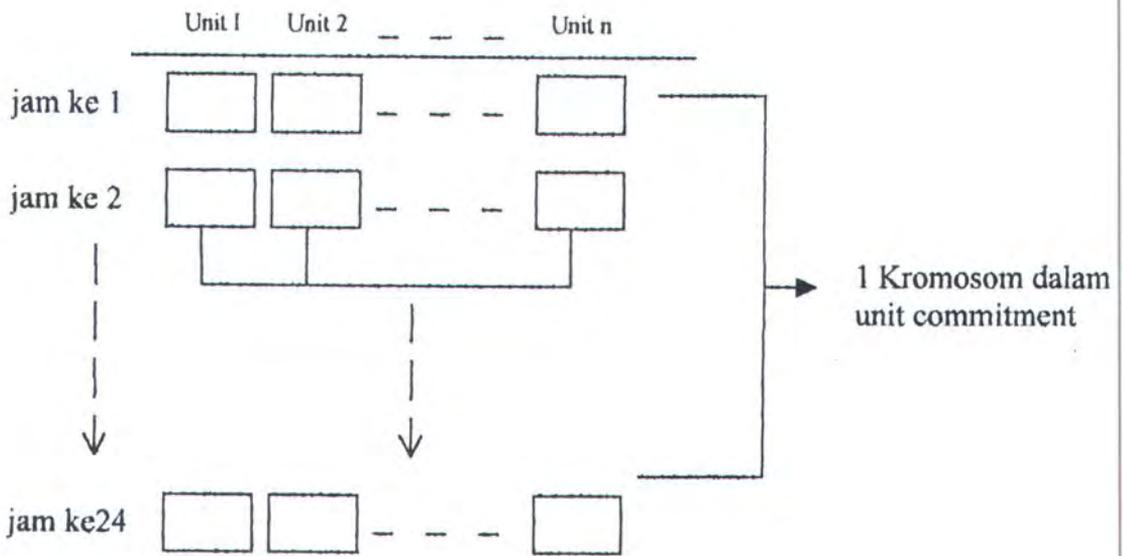
dimana :

RunCost : merupakan OBJ cost function dari economic dispatch

StartUpcost : merupakan biaya yang dibutuhkan saat unit pembangkit dinyalakan
startupcost tersebut merupakan input langsung dalam satuan biaya.

ShutdownCost : merupakan biaya yang dibutuhkan saat unit pembangkit dari kondisi kerja hingga unit pembangkit mencapai kondisi off/mati. Dan shutdown cost ini diasumsikan merupakan input langsung dalam satuan biaya.

t : merupakan banyaknya jam penjadwalan.



Gambar 4.5 Penjadwalan Unit Commitment

4.6. Penjadwalan Unit Commitment

Pada penjadwalan unit commitment ini dibagi dalam tiap-tiap jam sehingga mencapai batasan siklus waktu 24 jam. Dalam proses penjadwalan tersebut perhitungan *minimum down time* dan *minimum up time* diasumsikan selama 1 jam.

Design dari persoalan penjadwalan tersebut seperti pada gambar 4.5

4.7 Jenis Distribusi Random

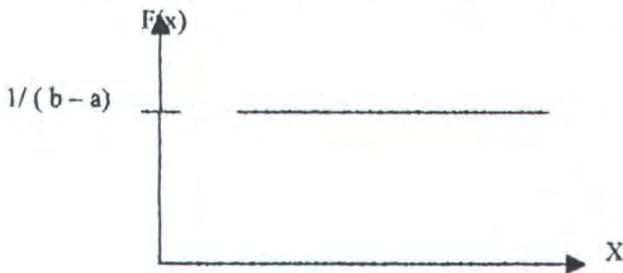
Pada tugas akhir ini digunakan dua jenis distribusi yaitu :

1. Distribusi Uniform.

Fungsi distribusi $F(x)$ dari bilangan acak U dengan pembatas dari 0 sampai dengan 1 dapat dihasilkan dari 2 langkah berikut ini

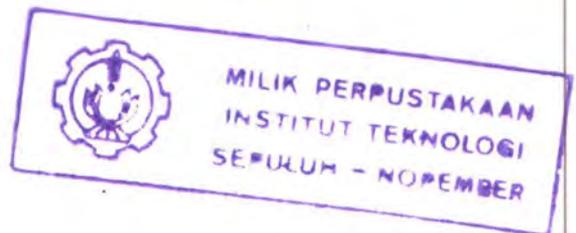
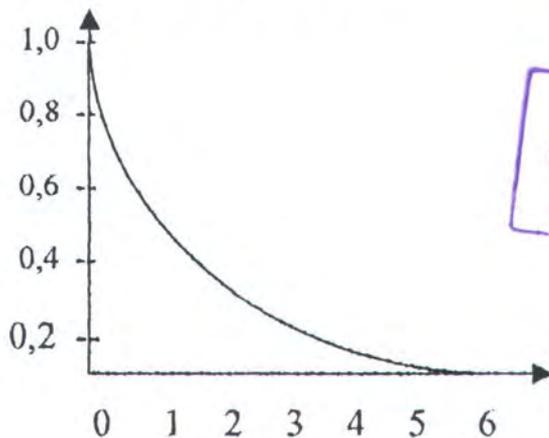
- hasilkan bilangan random U antara 0 dan 1
- maka $F(x) = a + (b - a) * U$

berikut ini grafik dari distribusi Uniform



2. Distribusi Exponential

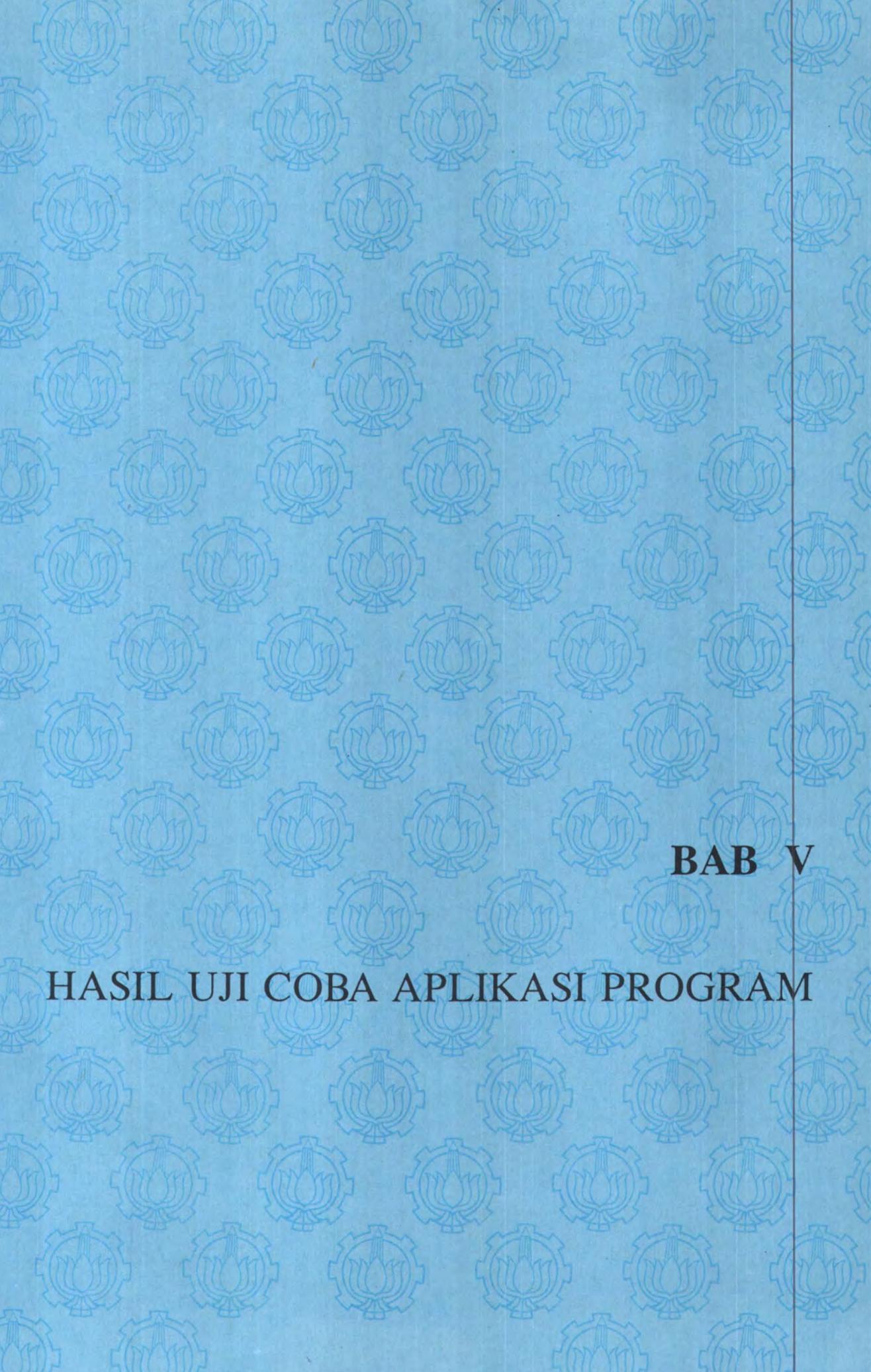
- Hasilkan bilangan random antara 0 sampai dengan 1
- Maka $F(X) = -\beta * \ln U$



4.8. Data Keluaran

Data keluaran yang ditampilkan berupa hasil dari nilai kromosom terbaik yang ada dalam tiap-tiap generasi dari unit commitment. Nilai yang dimaksud tersebut adalah nilai biaya (cost) dan daya dari masing-masing pembangkit.

Tiap-tiap kromosom terbaik dalam masing-masing generasi didapatkan dari proses pencarian nilai evaluasi (evaluation function) yang optimal. Dan proses pencarian nilai evaluasi yang optimal ini terletak pada fungsi evaluasi Seperti pada gambar (4.2).



BAB V

HASIL UJI COBA APLIKASI PROGRAM

BAB V

HASIL UJI COBA APLIKASI PROGRAM

5.1. Uji Coba Unit Commitment

5.1.1. Data Input Percobaan

Input Parameter Refined Genetik

Population Size : 20 - 40

Number of Generation : 2

Probability of Crossover : 75 %

Probability of Mutation : 1%

Elite Copy : 4

Akurasi : 3

- Input Unit Pembangkit

| Unit Pembangkit | Min Power | Maks Power | Fuel Cost | Const A | Const B | Const C |
|-----------------|-----------|------------|-----------|---------|---------|---------|
| Unit 1 | 100 | 600 | 1,1 | 510 | 7,2 | 0,00142 |
| Unit 2 | 100 | 400 | 1 | 310 | 7,85 | 0,00194 |
| Unit 3 | 50 | 200 | 1,2 | 78 | 7,97 | 0,00482 |

ShutDown Cost Untuk masing-masing unit = 1100

StartUp Cost untuk masing-masing unit = 1000

- Input load demand yang digunakan sebagai berikut :

| Waktu penjadwalan | Load Demand |
|-------------------|-------------|
| Jam 1 | 750 |
| Jam 2 | 850 |
| Jam 3 | 950 |
| Jam 4 | 1000 |

5.1.2. Hasil Uji Coba (perubahan populasi)

- Jumlah generasi yang digunakan adalah 2 generasi dengan 20 populasi dalam unit commitment dan probabilitas crossover menurun hingga mencapai nol serta probabilitas mutasi naik hingga mencapai batas 10%, maka berikut ini adalah hasil uji coba :

| | Unit 1 | | Unit 2 | | Unit 3 | | Total Power | Total Cost |
|-------|---------|----------|---------|----------|---------|----------|-------------|------------|
| | Daya | Cost | Daya | Cost | Daya | Cost | | |
| Jam 1 | 467,068 | 4600,931 | 277,881 | 2641,173 | - | - | 744,949 | 7242,104 |
| Jam 2 | 492,477 | 4840,259 | 363,907 | 3423,584 | - | - | 856,385 | 15505,947 |
| Jam 3 | 467,837 | 4608,145 | 383,147 | 3602,502 | - | - | 850,984 | 23716,594 |
| Jam 4 | 527,081 | 5169,429 | 333,161 | 3140,650 | 134,844 | 1488,415 | 995,086 | 33515,088 |

- Jumlah generasi yang digunakan adalah 2 generasi dengan 30 populasi dalam unit commitment adalah hasil uji coba :

| | Unit 1 | | Unit 2 | | Unit 3 | | Total Power | Total Cost |
|-------|---------|----------|---------|----------|---------|----------|-------------|------------|
| | Daya | Cost | Daya | Cost | Daya | Cost | | |
| Jam 1 | 593,402 | 5810,762 | 365,689 | 3440,093 | 105,425 | 1166,173 | 698,827 | 6976,935 |
| Jam2 | 337,390 | 3410,935 | 368,035 | 3461,849 | 144,721 | 1598,857 | 850,147 | 15448,576 |
| Jam 3 | 490,469 | 4821,271 | 351,320 | 3307,305 | 103,079 | 1140,906 | 944,868 | 24718,858 |
| Jam 4 | 459,238 | 4527,568 | 381,525 | 3587,360 | 157,918 | 1748,168 | 998,680 | 34581,172 |

- Generasi = 2 dan populasi = 40

maka berikut ini adalah hasil uji coba :

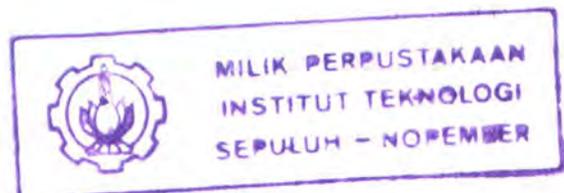
| | Unit 1 | | Unit 2 | | Unit 3 | | Total Power | Total Cost |
|-------|---------|----------|---------|----------|---------|----------|-------------|------------|
| | Daya | Cost | Daya | Cost | Daya | Cost | | |
| Jam 1 | 419,208 | 4155,628 | 212,903 | 2069,226 | 116,422 | 1285,460 | 748,534 | 7510,314 |
| Jam2 | 430,645 | 4261,391 | 300,587 | 2844,887 | 119,795 | 1322,322 | 851,026 | 15938,914 |
| Jam 3 | 498,827 | 4900,380 | 392,375 | 3688,826 | 59,091 | 678,942 | 950,293 | 25207,061 |
| Jam 4 | 501,026 | 4921,234 | 392,082 | 3686,078 | 114,018 | 1259,268 | 1007,126 | 33814,373 |

Dari tiga kali percobaan yang ada diatas (dengan nilai semua parameter tetap kecuali jumlah populasi diubah) terlihat dengan semakin bertambahnya jumlah populasi maka terbentuk kecenderungan total power perjam yang dihasilkan semakin mendekati load demand perjam pada tabel input load demand.

Hal tersebut terjadi karena bobot *scaling factor* power lebih besar dibandingkan bobot *scaling factor* cost dalam kromosom yang ada pada Unit Commitment program tersebut.

Scaling Factor adalah bobot skala (daya dan cost) dimana user dapat memilih dengan tujuan sebagai kontrol terhadap kecenderungan pemenuhan daya atau biaya.

Kondisi optimal dicapai saat jumlah populasi mencapai 40 dalam tiap generasi pada unit commitment.



5.1.3. Hasil Uji Coba (perubahan probabilitas crossover)

Dengan parameter sebagai berikut :

- Population Size : 40
- Number of Generation : 2
- Probability of Crossover : 50 % s/d 30%
- Probability of Mutation : 1%
- Elite Copy : 4
- Akurasi : 3

- Probabilitas crossover = 50%

| | Unit 1 | | Unit 2 | | Unit 3 | | Total Power | Total Cost |
|-------|---------|----------|---------|----------|---------|----------|-------------|------------|
| | Daya | Cost | Daya | Cost | Daya | Cost | | |
| Jam 1 | 425,367 | 4212,520 | 214,663 | 2048,498 | 105,279 | 1164,592 | 745,308 | 7461,616 |
| Jam 2 | 552,933 | 5417,783 | 243,695 | 2338,217 | 52,199 | 608,595 | 848,827 | 15826,212 |
| Jam 3 | 598,680 | 5862,398 | 370,674 | 3486,350 | - | - | 969,355 | 25174,959 |
| Jam 4 | 570,528 | 5588,015 | 377,713 | 3551,818 | 62317 | 712,058 | 1010,557 | 35026,850 |

- Probabilitas Crossover = 40%

| | Unit 1 | | Unit 2 | | Unit 3 | | Total Power | Total Cost |
|-------|---------|----------|---------|----------|---------|----------|-------------|------------|
| | Daya | Cost | Daya | Cost | Daya | Cost | | |
| Jam 1 | 376,979 | 3768,659 | 312,023 | 2948,260 | 65,103 | 740,756 | 754,106 | 7457,675 |
| Jam 2 | 536,217 | 5256,959 | 122,581 | 1301,409 | 189,443 | 2113,011 | 848,240 | 16129,053 |
| Jam 3 | 440,762 | 4355,291 | 346,041 | 3258,726 | 152,786 | 1689,864 | 939,589 | 25432,934 |
| Jam 4 | 594,282 | 5816,362 | 319,355 | 3014,791 | 84,164 | 939,518 | 997,801 | 35206,605 |

- Probabilitas Crossover = 30%

| | Unit 1 | | Unit 2 | | Unit 3 | | Total Power | Total Cost |
|-------|---------|----------|---------|----------|---------|----------|-------------|------------|
| | Daya | Cost | Daya | Cost | Daya | Cost | | |
| Jam 1 | 545,015 | 5341,494 | 202,639 | 1980,380 | 112,463 | 1242,355 | 747,654 | 7321,874 |
| Jam 2 | 335,630 | 3395,149 | 395,601 | 3719,080 | 122,727 | 1354,482 | 853,959 | 15790,586 |
| Jam 3 | 566,569 | 5549,628 | 357,771 | 3366,825 | 137,390 | 1516,777 | 924,340 | 24707,039 |
| Jam 4 | 592,522 | 5802,165 | 244,868 | 2348,537 | 166,716 | 1848,828 | 1004,106 | 34706,569 |

Pada percobaan dengan perubahan pada probabilitas crossover maka dapat dilihat bahwa nilai probabilitas crossover yang paling optimal dicapai pada nilai 30%. Hal tersebut berkaitan dengan nilai probabilitas mutasi yang makin meningkat pada tiap iterasi dari unit commitment dengan tujuan menghilangkan dominasi kromosom super pada akhir generasi.



BAB VI

KESIMPULAN DAN SARAN

BAB VI

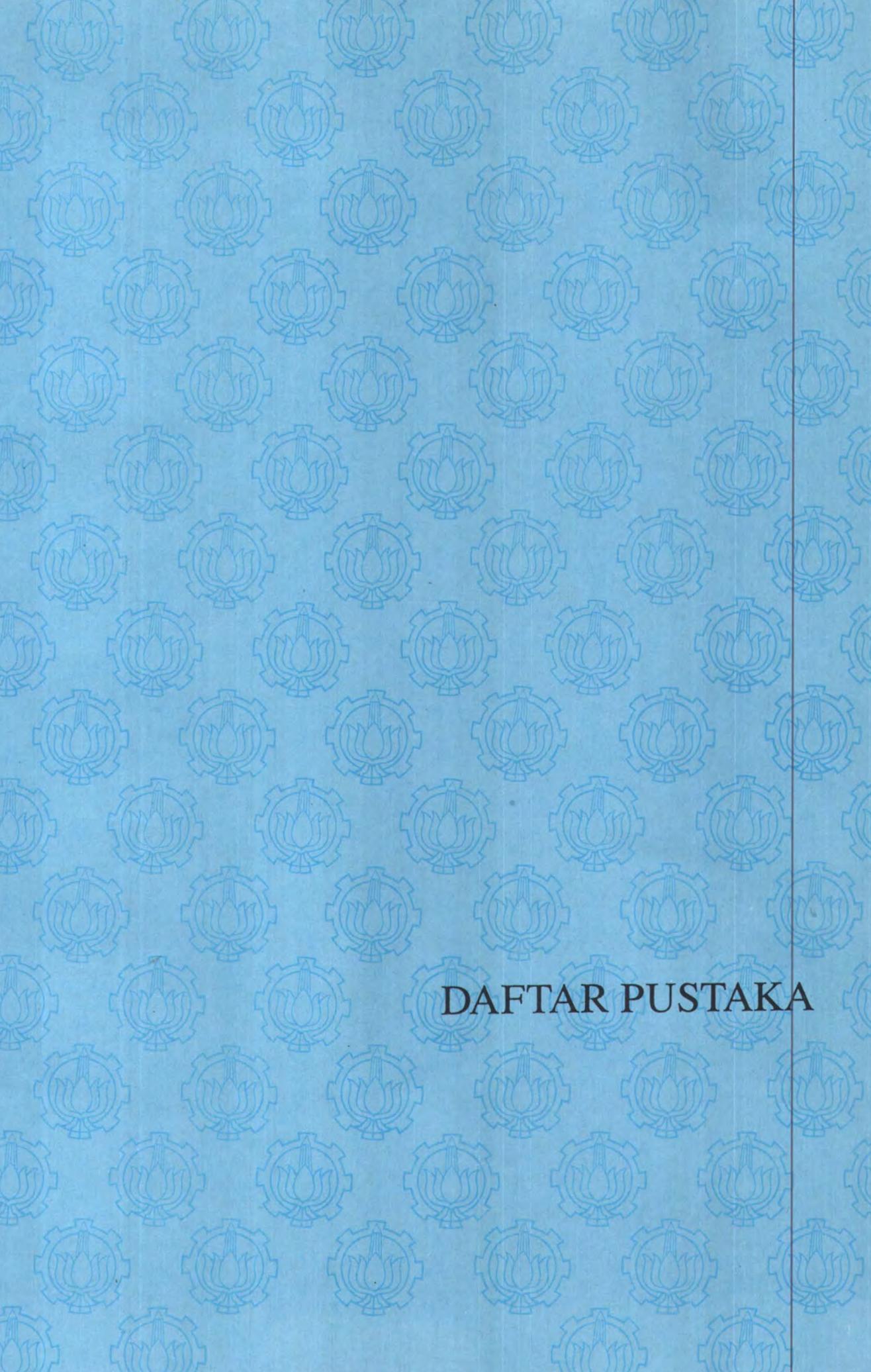
KESIMPULAN DAN SARAN

Dalam menentukan solusi yang tepat pada implementasi algoritma refined genetik banyak bergantung pada pengalaman. Penyelesaian komputasi optimasi yang tepat dengan menggunakan algoritma diatas sangat dipengaruhi oleh banyak faktor yang ada pada algoritma tersebut. Sebagai contoh penentuan nilai crossover serta mutasi, penerapan probabilitas mutasi dan pindah silang secara eksponensial dan penentuan distribusi random pada bilangan random. Juga penerapan fitness scaling serta scaling factor. Penentuan kondisi terminasi dapat dengan cara toleransi error (persentase error yang diinginkan user) yaitu kondisi terminasi yang terjadi jika terdapat evaluation function yang telah mencapai :

$$\text{evaluation function} = \text{maks. eval} - (\text{maks. eval} \times \text{toleransi error})$$

Selain itu kondisi terminasi dapat dicapai dengan cara mengecek nilai eval yang tercatat dalam parameter elitism jika sampai beberapa generasi nilai elitism yang disimpan tidak berubah maka dapat ditetapkan bahwa telah terjadi kondisi terminasi dan jumlah generasi yang ditentukan pada kalimat diatas dapat diambil menurut perhitungan dari designer program.

Sejalan dengan iterasi generasi maka kromosom-kromosom yang ada makin membaik, oleh karena itu nilai probabilitas crossover pada tiap iterasi generasi dalam unit commitment dikurangi (secara eksponensial).



DAFTAR PUSTAKA

Daftar Pustaka

1. Wood, A.J. dan Wollenberg, B.F., "Power Generation, Operation, and Control", John Wiley & Sons, New York, NY 1984.
2. Gerald B. Sheble dan Kristin Brittig, "Refined Genetic Algorithm – Economic Dispatch Example", Iowa State University, 1995.
3. Mitsuo Gen dan Runwei Cheng, "Genetic Algorithms And Engineering Design", Ashikaga Institute of Technology, 1996.
4. Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Department of Computer Science, University of North Carolina, 1996.
5. Orero Mr. S.O. dan Irving M.R. DR. , "A Genetic Algorithm for Generator Scheduling in Power System", Brunel Institut of Power System, Dept of Electrical Eng. & Electronics, Brunel University.
6. Walters, David C., "An Application of Genetic Algorithms to the Economic Dispatch of Power Generation, MS Thesis, Auburn University, May, 1991
7. Strunz Kai, "Object-Oriented Implementation of a Genetic Algorithm Based Approach to Economic Dispatch of Power Systems", Brunel Institute of Power System, Brunel University.

