

21.219/112/H/05



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

**PENGGUNAAN ALGORITMA GENETIKA PADA
DATA MINING UNTUK PENCARIAN
POLA-POLA IMPLIKASI DALAM
SUATU BASIS DATA**

TUGAS AKHIR



RSIF
005.1
Nas
p-1
waw

PERPUSTAKAAN ITS	
Tgl. Terima	9-7-2003
Terima Dari	H
No. Agenda Prp.	217565

Oleh :

WAWAN TAUFIQ NASICH
NRP. 2695 100 039

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

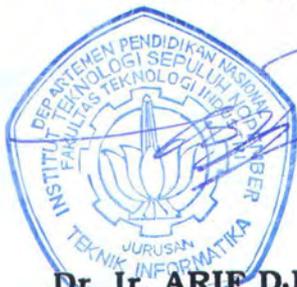
**PENGGUNAAN ALGORITMA GENETIKA PADA
DATA MINING UNTUK PENCARIAN
POLA-POLA IMPLIKASI DALAM
SUATU BASIS DATA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

Mengetahui / Menyetujui

Dosen Pembimbing I



Dr. Ir. ARIF DJUNAIDY, MSc
NIP. 131 633 403

Dosen Pembimbing II



RULLY SOELAIMAN, S. Kom
NIP. 132 085 802

**SURABAYA
AGUSTUS, 2000**



Katakanlah : “tiap-tiap orang bertindak menurut keadaanya masing-masing”. Maka tuhanmu lebih mengetahui mana jalan yang lebih benar.

(QS 17 : 84)

Percayalah padaku : rahasia untuk mencapai buah terbesar dan kenikmatan tertinggi dari manusia adalah hidup dengan berbahaya!. Dirikanlah kota-kotamu di lereng gunung vesuvius, kirimkanlah kapal-kapalmu ke samudra yang belum dipetakan, hiduplah dalam perang melawan sesamamu dan dirimu sendiri.

(Friederich Nietzsche)



ABSTRAK

Pengertian Data Mining digunakan untuk mendefinisikan suatu proses pencarian otomatis terhadap informasi yang menarik dan berguna dalam suatu basis data yang dititikberatkan pada penemuan pola yang sulit atau bahkan tidak mungkin untuk dideteksi menggunakan mekanisme query standar dan teknik statistik klasik. Secara tradisional, Data Mining dibagi dalam tiga tingkatan, yaitu terarah (*directed*), tidak terarah (*undirected*), dan penyempurnaan hipotesa (*hypotesis refinement*). Sebagian besar Data Mining pada saat ini menggunakan variasi dari algoritma pembelajaran mesin tradisional untuk menyelesaikan proses pencarian pengetahuan secara terarah (*directed data mining*).

Algoritma Genetika telah secara luas digunakan dalam beragam masalah optimasi dan pencarian. Namun selama ini penerapannya terbatas pada permasalahan numerik saja. Dalam Tugas Akhir ini disajikan suatu pendekatan yang tidak hanya berguna untuk Data Mining yang tidak terarah, tapi diharapkan dapat juga diterapkan untuk Data Mining yang terarah. Pendekatan yang dimaksud menggunakan Algoritma Genetika sebagai suatu mekanisme pencarian untuk secara perlahan mendapatkan aturan-aturan (*rules*) yang optimal. Penerapan Algoritma Genetika pada Data Mining membutuhkan pengkodean yang kompleks yang dilengkapi dengan mekanisme pengendalian agar aturan-aturan tidak keluar dari domain solusi yang dibolehkan, desain operator operator genetika yang mampu menghasilkan aturan aturan yang berkualitas, fungsi evaluasi yang tepat dan mekanisme seleksi yang mampu menjaga kelebaran ruang pencarian sehingga tidak terjadi konvergensi dini dan dominasi individu tertentu dalam populasi.

Dari hasil uji coba didapatkan bahwa Algoritma genetika telah berhasil diterapkan pada Data Mining untuk pencarian pola implikasi dalam suatu Basis Data. Keberhasilan ini ditunjukkan dengan kemampuan Perangkat Lunak yang dibuat untuk menemukan pola-pola yang menarik dan berkualitas dari beberapa Basis Data yang diujicobakan.



KATA PENGANTAR

Segala puji bagi Allah SWT, dzat dimana kita bermula dan akan kembali. Yang mengarahkan kita pada kesadaran tertinggi melalui proses perkembangan yang terus menerus dan usaha untuk memecahkan misteri alam. Semua tidak lepas dari curahan kasih sayang yang agung, sehingga penulis berhasil menyelesaikan Tugas akhir ini.

Penulis memberi judul tugas akhir ini dengan :

**PENGGUNAAN ALGORITMA GENETIKA PADA DATA MINING
UNTUK PENCARIAN POLA-POLA IMPLIKASI
DALAM SUATU BASIS DATA**

Tugas akhir ini disusun guna memenuhi sebagian persyaratan untuk memperoleh gelar sarjana pada jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Melalui kesempatan ini penulis menyampaikan terima kasih dan penghargaan sebesar-besarnya kepada :

1. Bapak Dr. Ir. Arif Djunaidy, MSc., selaku ketua Jurusan Teknik Informatika FTI-ITS, sekaligus sebagai dosen pembimbing pertama dalam penyusunan tugas akhir ini.
2. Bapak Rully Soelaiman, S. Kom., selaku dosen wali selama masa

perkuliahan, sekaligus sebagai dosen pembimbing kedua dalam penyusunan tugas akhir ini.

3. Seluruh staf pengajar Jurusan Teknik Informatika FTI-ITS yang tidak bisa disebutkan satu persatu, atas ilmu yang diberikan selama masa perkuliahan.
4. Segenap staf pegawai dan tata usaha Jurusan Teknik Informatika; Bapak Mu'in, Mas Yudi, Mas Khodir, Mas Soleh dan staf lainnya yang juga telah banyak membantu penulis selama di kampus.
5. Seluruh Personil C-0B; Heri *Guk Guk*, Roy *Embek*, Sudito, Pram, Salim, Ahmed, Eko *Nyambik*, Doddy, Damang, Fauzi, Panji *Milenium*, Ade, Ronny dan seluruh personil C-0B yang lain terima kasih atas semua waktu yang kita habiskan bersama.
6. Penghuni Lab. SI; mas Akaton, Husein, Anton, Bowo, Andre, Fauzi, Pram dan penghuni penghuni Lab. SI yang lain. Terima kasih atas hiburan dan gannguannya selama proses pengerjaan tugas akhir ini.
7. Teman teman di GL 57; Topeng, Koller, Selebor, Thole , Doel, Kepik dan anggota GL 57 yang lain. Terima kasih atas guyon, ledekan dan kebersamaannya, semoga kalian juga cepat-cepat minggat dari sini.
8. Tak lupa terima kasih buat teman teman di GL 32C; Tono, Joko, Deni, Gatot, Yanto dan Rio yang rajin membangunkanku dipagi hari.
9. Teman teman seperjuangnan; mas Akaton, Yanuar, Sonny, Ronny,

Yoga, Daniel dan teman teman seperjuangan yang lain. kalian adalah orang hebat.

10. Adik adik C0-F, C0-E, C0-D dan C0-C; terutama Hera dan Daning, terima kasih atas support dan kehadirannya. Juga buat si banyak makan Silvi Cenora, terima kasih atas supportnya.

Penulis menyadari, masih banyak kekurangan dalam penyusunan tugas akhir ini. Namun penulis tetap berharap, karya sederhana ini dapat memberi manfaat sebesar-besarnya bagi generasi sekarang dan generasi yang akan datang. Pintu kritik dan saran membangun, tetap terbuka.

Penulis

DAFTAR ISI

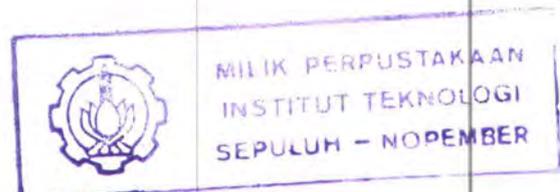
ABSTRAK.....	I
KATA PENGANTAR.....	II
DAFTAR ISI	V
DAFTAR GAMBAR.....	IX
DAFTAR TABEL	XI
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH	2
1.3 TUJUAN DAN MANFAAT.....	3
1.4 BATASAN MASALAH.....	4
1.5 METODOLOGI PELAKSANAAN TUGAS AKHIR	4
1.6 SISTEMATIKA PEMBAHASAN	5
BAB II DASAR TEORI.....	7
2.1 DATA MINING.....	7
2.1.1 Pengertian Data Mining	7
2.1.2 Peranan Data Warehouse	10
2.1.3 Proses Penyiapan Data.....	11
2.1.4 Model Model Data Mining.....	14
2.2 ALGORITMA GENETIKA	15
2.2.1 Umum.....	16
2.2.2 Sejarah Algoritma Genetika.....	17
2.2.3 Teori Dasar Algoritma Genetika.....	20

2.2.3.1	Beberapa Istilah dalam Algoritma Genetika	20
2.2.3.2	Parameter Parameter Algoritma Genetika	21
2.2.3.2	Mekanisme Algoritma Genetika.....	22

BAB III ALGORITMA GENETIKA UNTUK PENCARIAN POLA

IMPLIKASI DALAM BASIS DATA	25
3.1 POLA IMPLIKASI.....	26
3.2 PENGKODEAN	29
3.2.1 Pengkodean Aturan.....	30
3.2.2 Mekanisme Kontrol	32
3.3 OPERATOR GENETIKA.....	34
3.3.1 Pra Pencarian.	34
3.3.2 Algoritma Pembangkitan populasi	34
3.3.2.1 Algoritma Pembangkitan Term	35
3.3.2.2 Algoritma Pembangkitan Klausa	35
3.3.2.3 Algoritma Pembangkitan Aturan.....	35
3.3.3 Pindah Silang	36
3.3.3.1 Pindah Silang Term.....	36
3.3.3.2 Pindah Silang Klausa	37
3.3.3.3 Pindah Silang Aturan	38
3.3.4 Mutasi	39
3.3.4.1 Mutasi Nilai Tunggal	39
3.3.4.2 Mutasi Range.....	39
3.3.4.3 Mutasi Nilai Katagori.....	40
3.3.4.5 Mutasi Term	40
3.3.4.4 Mutasi Klausa.....	41
3.4 FUNGSI EVALUASI.....	42
3.5 MEKANISME SELEKSI DAN PENGGANTIAN	42
3.6 SIKLUS GENERASIONAL	43

BAB IV PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK	44
4.1 DESKRIPSI SISTEM	44
4.2 PERANCANGAN PERANGKAT LUNAK	46
4.3.1 Perancangan Data	46
4.3.1.1 Perancangan Data Masukan	46
4.3.1.2 Perancangan Data Proses	48
4.3.1.3 Perancangan Data Keluaran	56
4.3.2 Perancangan Proses.....	56
4.3.3 Perancangan Antar Muka.....	62
4.3 PEMBUATAN PERANGKAT LUNAK.....	64
4.4.1 Lingkungan Implementasi.....	64
4.4.2 Implementasi Struktur Data	66
4.4.2.1 Kelas RefUtama.....	66
4.4.2.2 Kelas InfoField	67
4.4.2.3 Kelas Katagori.....	68
4.4.2.4 Kelas Numerik.....	68
4.4.2.5 Kelas KlausaVar	68
4.4.2.6 Kelas Populasi	69
4.4.2.7 Kelas Rule	70
4.4.2.8 Kelas Term	71
4.4.3 Implementasi Proses	71
4.4.3.1 Pembangkitan Populasi Awal.....	72
4.4.3.2 Pindah Silang.....	75
4.4.3.3 Mutasi	76
4.4.3.4 Evaluasi.....	77
4.4.3.5 Seleksi.....	78
 BAB V UJI COBA DAN EVALUASI PERANGKAT LUNAK	 80
5.1 DATA UNTUK UJI COBA.....	80
5.2 LINGKUNGAN UJI COBA.....	81



5.3	EVALUASI TERHADAP KINERJA PERANGKAT LUNAK.....	83
5.3.1	Parameter-Parameter Yang Diamati.....	83
5.3.2	Uji Coba Pada Beberapa Lingkungan	84
5.3.3	Uji Coba Dengan Data Berbeda.....	86
5.4	INTERPRETASI TERHADAP ATURAN YANG DIHASILKAN.....	88
5.4.1	Interpretasi Hasil pada Basis Data Market	89
5.4.2	Interpretasi Hasil pada Basis Data Dermatologi.....	91
5.4.3	Interpretasi Hasil pada <i>German Credit Data</i>	93
BAB VI KESIMPULAN DAN SARAN		95
6.1	KESIMPULAN.....	95
6.2	KEMUNGKINAN PENGEMBANGAN	96
DAFTAR PUSTAKA.....		97
LAMPIRAN A DESKRIPSI BASIS DATA MARKET.....		99
LAMPIRAN B DESKRIPSI BASIS DATA DERMATOLOGI.....		101
LAMPIRAN C DESKRIPSI BASIS DATA <i>GERMAN CREDIT</i>		104
LAMPIRAN D PETUNJUK PENGGUNAAN PERANGKAT LUNAK		108
LAMPIRAN E CONTOH KELUARAN		104

DAFTAR GAMBAR

GAMBAR 2.1 SIKLUS GENERASIONAL DALAM GA.....	17
GAMBAR 2.2 MEKANISME ALGORITMA GENETIKA	23
GAMBAR 3.1 ILUSTRASI PENYELEKSIAN RECORD RECORD BASIS DATA.....	27
GAMBAR 3.2 ILUSTRASI PENGKODEAN DALAM GA	30
GAMBAR 3.3 TIPE VARIABEL, TANDA DAN NILAI.....	31
GAMBAR 3.4 FEASIBILITAS DAN LEGALITAS SEBUAH INDIVIDU.	33
GAMBAR 4.1 ILUSTRASI SISTEM SECARA FISIK	45
GAMBAR 4.2 HUBUNGAN DATA PROSES.....	48
GAMBAR 4.3 DAD LEVEL 0, SISTEM GA-MINER	56
GAMBAR 4.4 DAD LEVEL 1, GA-MINER	57
GAMBAR 4.5 DAD LEVEL 2, INISIALISASI & DECODING.....	58
GAMBAR 4.6 DAD LEVEL 2, SIKLUS GENERASIONAL	59
GAMBAR 4.7 DAD LEVEL 2, DECODING & INTERPRETASI.....	60
GAMBAR 4.8 HIRARKI PROSES.....	61
GAMBAR 4.9 HIRARKI MENU.....	64
GAMBAR 4.10 ILUSTRASI CARA KERJA THIN DRIVER	65
GAMBAR 4.11 ILUSTRASI CARA KERJA OCI DRIVER.....	66
GAMBAR 5.1 ATURAN 1, CONTOH HASIL DARI TABEL MARKET2	89
GAMBAR 5.2 ATURAN 2, CONTOH HASIL DARI TABEL MARKET2	89
GAMBAR 5.3 ATURAN 3, CONTOH HASIL DARI TABEL MARKET2	90
GAMBAR 5.4 ATURAN 1, CONTOH HASIL DARI TABEL DERMATOLOGI.....	91
GAMBAR 5.5 ATURAN 2, CONTOH HASIL DARI TABEL DERMATOLOGI.....	92
GAMBAR 5.6 ATURAN 3, CONTOH HASIL DARI TABEL DERMATOLOGI.....	92
GAMBAR 5.7 ATURAN 1, CONTOH HASIL DARI TABEL <i>GERMAN CREDIT DATA</i>	93
GAMBAR 5.8 ATURAN 2, CONTOH HASIL DARI TABEL <i>GERMAN CREDIT DATA</i>	93

GAMBAR 5.9 ATURAN 3, CONTOH HASIL DARI TABEL <i>GERMAN CREDIT DATA</i>	94
GAMBAR D.1 TAMPILAN UTAMA.....	108
GAMBAR D.2 DIALOG KONEKSI	109
GAMBAR D.3 DIALOG LIHAT KONEKSI.....	110
GAMBAR D.4 DIALOG PEMUTUSAN KONEKSI.....	110
GAMBAR D.5 DIALOG PEMILIHAN TABEL	111
GAMBAR D.6 DIALOG PEMILIHAN VARIABEL.....	111
GAMBAR D.7 DIALOG DIRECTED.....	112
GAMBAR D.8 DIALOG PARAMETER GENETIKA.....	112
GAMBAR D.9 DIALOG PROSES MINING	113
GAMBAR D.10 DIALOG ABOUT	114
GAMBAR E.1 CONTOH KELUARAN PADA BASIS DATA MARKET	115
GAMBAR E.2 CONTOH KELUARAN PADA BASIS DATA DERMATOLOGI.....	116
GAMBAR E.3 CONTOH KELUARAN PADA BASIS DATA <i>GERMAN CREDIT</i>	117

DAFTAR TABEL

TABEL 2.1 BEBERAPA ISTILAH DALAM ALGORITMA GENETIKA.....	20
TABEL 3.1 TABEL KONTINGENSI 2 X 2.....	28
TABEL 3.2 CONTOH SEBUAH TABEL DALAM BASIS DATA.....	31
TABEL 4.1 DATA MASUKAN.....	46
TABEL 4.2 DATA AGREGASI INFOFIELD.....	49
TABEL 4.3 DATA AGREGASI KLAUSA VAR.....	49
TABEL 4.4 DATA AGREGASI KATAGORI.....	50
TABEL 4.5 DATA AGREGASI NUMERIK.....	50
TABEL 4.6 DATA AGREGASI REFERENSI UTAMA.....	51
TABEL 4.7 OPERASI OPERASI PRIMITIF REFERENSI UTAMA.....	52
TABEL 4.8 DATA AGREGASI TERM.....	52
TABEL 4.9 OPERASI OPERASI PRIMITIF TERM.....	53
TABEL 4.10 DATA AGREGASI RULE.....	53
TABEL 4.11 OPERASI OPERASI PRIMITIF RULE.....	54
TABEL 4.12 DATA AGREGASI POPULASI.....	54
TABEL 4.13 OPERASI OPERASI PRIMITIF POPULASI.....	55
TABEL 5.1 DATA UJI COBA PADA BEBERAPA LINGKUNGAN.....	84
TABEL 5.2 DATA UJI COBA PADA DATA MARKET.....	86
TABEL 5.3 DATA HASIL UJI COBA DENGAN DATA DERMATOLOGI.....	87
TABEL 5.4 DATA HASIL UJI COBA DENGAN DATA <i>GERMAN CREDIT</i>	87



BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Dengan semakin canggihnya teknologi pada saat ini, banyak organisasi yang setiap harinya mengumpulkan data dalam jumlah yang besar dan hal ini dapat dilakukan dengan sangat mudah dan cepat. Data data ini disimpan sebagai data perusahaan yang pada saat ini jumlahnya sudah sangat banyak, seperti yang dapat ditemui pada perusahaan perusahaan retail, perbankan, asuransi dan sebagainya.

Seringkali data data ini hanya dibiarkan begitu saja dan pemanfaatannya juga kurang optimal, seperti yang telah banyak dilakukan adalah kebanyakan data data ini dipakai sebagai sumber data untuk peramalan (*Forecasting*). Padahal masih banyak yang dapat dilakukan untuk mendapatkan informasi yang beragam dari data yang telah terkumpul.

Dalam hal ini, pengertian Data Mining digunakan untuk mendefinisikan suatu proses pencarian otomatis terhadap informasi yang menarik dan berguna dalam suatu basis data. Terutama dititik beratkan pada penemuan pola yang sulit atau bahkan tidak mungkin untuk dideteksi menggunakan mekanisme query standar dan teknik statistik klasik.

Salah satu yang dapat dilakukan dalam Data Mining dan juga menjadi topik dalam penyusunan Tugas Akhir ini adalah usaha untuk melakukan pencarian

aturan aturan implikasi dari suatu basis data. Aturan aturan ini akan sangat berguna untuk memahami hubungan hubungan dan pola pola dalam basis data atau sebagai model data untuk memprediksi trend di masa datang. Hal ini penting sebagai bahan pengambilan keputusan bisnis yang penting.

Algoritma Genetika telah berhasil digunakan dalam beragam masalah pencarian dan optimasi. Namun, selama ini penerapannya masih terbatas pada optimasi permasalahan numerik, Padahal teknik ini dapat digunakan untuk memanipulasi beragam struktur data, dan telah lama dieksploitasi untuk memanipulasi aturan (*rule*). Karena beberapa hal tersebut, maka dalam Tugas Akhir ini kami akan mencoba mengeksploitasi Algoritma Genetika sebagai sebuah proses pencarian untuk diimplementasikan dalam Data Mining.

1.2 PERUMUSAN MASALAH

Dari latar belakang yang telah diuraikan di atas timbul permasalahan permasalahan berikut sehingga perlu usaha untuk menggunakan Algoritma Genetika dalam Data Mining untuk pencarian pola pola Implikasi dalam suatu basis data.

- Bagaimana mendapatkan informasi yang berharga dari suatu basis data, dalam hal ini informasi tersebut disajikan dalam format aturan implikasi IF-Then, yang sangat berguna dalam memahami hubungan hubungan dan pola pola dalam basis data sebagai bahan penunjang pengambilan keputusan.
- Bagaimana menerapkan Algoritma Genetika kedalam permasalahan Data Mining yang membutuhkan pengkodean yang kompleks, desain operator

operator genetika yang tepat serta mekanisme pengendalian untuk mencegah solusi keluar dari domain permasalahan.

- Bagaimana melakukan koneksi ke beberapa format file basis data yang berbeda, mengingat ada beberapa standar yang sering dipakai dalam penyimpanan file basis data relasional. Koneksi tersebut diusahakan agar disamping mempunyai kemampuan multi platform juga mempunyai kinerja yang tinggi.

1.3 TUJUAN DAN MANFAAT

Tujuan yang hendak dicapai dalam Tugas Akhir ini adalah:

- Membangun sebuah sistem Data Mining yang mampu menemukan pola-pola Implikasi dalam suatu basis data. Dan kemudian menyajikan pola-pola tersebut kedalam bentuk yang mudah dimengeti oleh pengguna.
- Menerapkan Algoritma Genetika dalam Data Mining, Hal ini membutuhkan desain dan penerapan yang baik dari operator rekombinasi dan mutasi, demikian pula dengan operator untuk pengumpulan aturan aturan, dan untuk penyederhanaan aturan.

Sistem ini nantinya diharapkan dapat dimanfaatkan sebagai sebuah alat Pencarian informasi dalam basis data yang dapat berperan sebagai sistem penunjang pengambilan keputusan dalam bisnis, ataupun pencarian pengetahuan dalam basis data untuk kepentingan kedokteran, teknik dsb.

1.4 BATASAN MASALAH

Dalam Tugas Akhir ini diberikan beberapa pembatasan, hal ini dimaksudkan agar penyusunan Tugas Akhir tetap sesuai dengan tujuan yang akan dicapai .

- Data yang dijadikan input adalah file file dari basis data relasional yang sudah diproses melalui proses penyiapan data dan sudah dalam bentuk yang siap untuk digali oleh algoritma.
- Sistem dibuat dengan menggunakan teknologi yang berbasis Java dan hanya bisa melakukan koneksi dengan basis data melalui JDBC.
- Pola pola yang akan dicari dari basis data adalah pola implikasi.

1.5 METODOLOGI PELAKSANAAN TUGAS AKHIR

Langkah langkah yang dilakukan dalam pembuatan Tugas Akhir ini adalah :

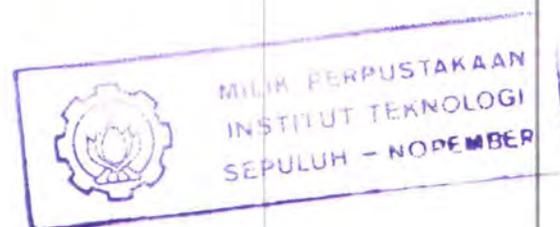
- Studi kepustakaan. Selama kegiatan ini dilakukan studi literatur meliputi Konsep data mining, Algoritma Genetika, bahasa pemrograman Oracle JDeveloper 2.0 , JDBC dan pengumpulan data data yang diperlukan.
- Penyusunan Algoritma. Dalam fase ini dilakukan analisa terhadap algoritma yang akan digunakan.
- Desain Perangkat Lunak. Selama fase ini perangkat lunak didesain dengan menerapkan algoritma yang sudah di tentukan sebelumnya.

- Implementasi. Dalam tahap ini desain Perangkat Lunak yang sudah jadi, diimplementasikan dalam lingkungan dan dengan menggunakan bahasa pemrograman yang telah dipilih.
- Tes dan Evaluasi. Dalam tahapan ini dilakukan tes dan evaluasi terhadap program yang telah dibuat dan kemudian dilakukan pembenahan seperlunya.
- Pembuatan Dokumentasi. Kegiatan ini berupa pembuatan dokumentasi untuk setiap tahap yang telah dilakukan dan nantinya akan disusun dalam sebuah buku.

1.6 SISTEMATIKA PEMBAHASAN

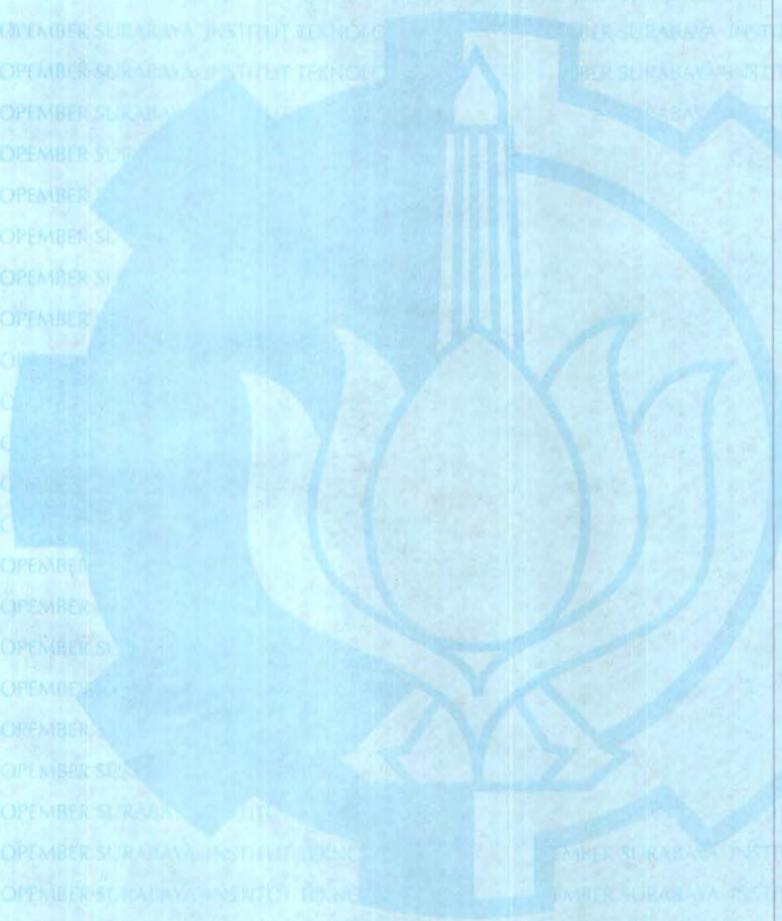
Pembahasan mengenai perangkat lunak yang dibuat dalam Tugas Akhir ini disusun dalam beberapa Bab.

- **Bab I** berisi Pendahuluan yang memuat latar belakang pembuatan perangkat lunak, tujuan dan manfaat, permasalahan, batasan masalah, langkah langkah pembuatan tugas Akhir ini dan sistematika penulisan buku Tugas Akhir.
- **Bab II** berisikan landasan teori yang di bagi kedalam dua bagian. Bagian pertama berkaitan dengan masalah Data Mining, yang meliputi pengertian Data Mining, peranan data warehouse, proses penyiapan data, model model data mining, dan contoh aplikasinya. Sedangkan bagian kedua membahas algoritma genetika.
- **Bab III** membahas secara lengkap Algoritma Genetika yang digunakan dalam Data Mining. Meliputi semua aspek dari algoritma yang akan digunakan mulai



dari pengkodean, algoritma pembangkitan populasi awal, operator operator crossover dan mutasi, metode seleksi, sampai dengan fungsi evaluasinya.

- **Bab IV** berisi rancangan dan implementasi perangkat lunak, untuk mengimplementasikan uraian dari Bab III, yang meliputi deskripsi sistem, perancangan dan pembuatan perangkat lunak meliputi perancangan data, perancangan proses, perancangan antar muka dan implementasinya dalam bahasa pemrograman.
- **Bab V** mengulas tentang hasil uji coba dan analisa unjuk kerja perangkat lunak lunak yang dihasilkan.
- **Bab VI** merupakan penutup yang berisi kesimpulan serta saran saran untuk pengembangan lebih lanjut guna memperoleh hasil yang lebih sempurna.



BAB II

DASAR TEORI

Dalam bab ini dibahas dasar teori yang menjadi landasan tugas akhir ini. Ada dua bidang utama yang melandasi tugas akhir ini, yakni Data Mining dan Algoritma Genetika. Pembahasan yang dilakukan dalam bab ini bersifat teoritis, sedangkan untuk pembahasan algoritma yang bersifat implementatif akan dijelaskan secara rinci dalam Bab III..

2.1 DATA MINING

Dalam bagian pertama bab ini dibahas dasar teori yang berkaitan dengan data mining. Diantaranya adalah pengertian data mining, proses penyiapan data, peranan data warehouse, model model data mining, interpretasi hasil dan contoh aplikasi data mining. Bahan untuk pembahasan Data Mining ini disarikan dari [SKA-98].

2.1.1 Pengertian Data Mining

Data mining, yang juga dikenal sebagai KDD (Knowledge discovery in Database) adalah pencarian terhadap informasi yang bernilai dan tersembunyi dalam suatu basis data yang sulit atau bahkan tidak mungkin untuk ditemukan dengan menggunakan mekanisme query standar atau teknik statistik klasik. Data mining adalah merupakan bagian dari apa yang disebut oleh industri sebagai

mining adalah merupakan bagian dari apa yang disebut oleh industri sebagai *business intelligence* (BI). Tujuan dari software BI adalah untuk membantu perusahaan tetap kompetitif, mendapatkan penguasaan pasar, dan secara cepat dapat melihat trend pasar. Software DSS, *Data Warehousing*, OLAP, dan Data Mining adalah bagian dari *business intelligence* (BI).

Teknik dalam Data Mining datang dari basis data, *machine learning*, dan statistik. Elemen elemen kunci untuk Data Mining ini telah dibuat dalam beberapa tahun terakhir. Secara umum Data Mining dapat dibagi kedalam dua tipe, *Predictive Data Mining* dan *knowledge discovery*.

Predictive Data Mining adalah tipe yang paling sering diaplikasikan dan mempunyai potensi untuk mendatangkan keuntungan yang besar. *Predictive mining* mempunyai tujuan khusus yang akan dicari jawabanya. Yang termasuk dalam aplikasi tipe ini adalah antara lain : klasifikasi, regresi, dan masalah *time series*.

Knowledge discovery diaplikasikan dalam beberapa aplikasi antara lain : deteksi deviasi, segmentasi basis data, clustering, aturan asosiasi, dan sumerisasi. *Knowledge discovery* kadang disebut pula sebagai pencarian pola (*pattern discovery*). Sistem dapat menemukan pola yang tersembunyi dalam data. Pola ini dapat ditemukan oleh manusia jika jumlah data dan jumlah fieldnya kecil. Tapi jika jumlah fieldnya mencapai ratusan dengan record yang mencapai jutaan, diperlukan waktu bertahun tahun untuk mendapatkan pola pola yang berguna

dalam data. Disinilah peran teknologi komputer yang cepat dan sistem Data Mining yang powerful sangat membantu dalam permasalahan ini.

Sebelum sebuah sistem Data Mining dapat membantu menyelesaikan permasalahan dalam bisnis, data harus dipersiapkan terlebih dahulu. Data harus diringkas dan dibersihkan dari kesalahan. Maksudnya data harus terlebih dulu dirangkum dan record record yang mengandung kesalahan dibuang. Seringkali bentuk akhirnya adalah sebuah spreadsheet. Dan biasanya ini digunakan sebagai bentuk standard. Setelah data sudah berada dalam form standar algoritma AI atau metode prediksi, barulah proses Mining dapat dijalankan. Proses ini bisa berupa iterasi. Pemilihan algoritma untuk prediksi dapat tergantung pada aplikasi.

Salah satu langkah penting dalam Data Mining adalah interpretasi terhadap hasil. Software atau algoritma akan memberikan informasi yang berharga tentang data yang ada, tetapi sistem mungkin dapat juga memberikan informasi yang tidak membantu.

Data Mining dapat membantu perusahaan lebih mengerti pelanggan dan klien mereka. Salah satu contoh informasi yang didapat dari Data Mining adalah trend pembelian pelanggan. Hal ini dapat membuat perusahaan merubah strateginya untuk meningkatkan keuntungan. Mempunyai informasi yang berharga dapat merubah cara sebuah perusahaan dioperasikan, karena dapat meningkatkan keuntungan dan mengurangi kerugian. Data Mining dapat membuat sebuah perusahaan lebih kompetitif.

Masa depan Data Mining dan teknologi yang berhubungan seperti Data Marts, Data Warehousing, dan Decision Support sangat cerah. *Business Intelligence* (BI) masih sangat jauh dari titik jenuh dalam industri. Teknologi ini masih sangat baru dan masih dalam tahap pengembangan.

Grup Meta memprediksikan Data Mining akan berkembang dari 100 juta dolar pada tahun 1996 menjadi lebih dari 800 juta dolar pada tahun 2000. Software BI yang lain berkembang dalam tingkat pertumbuhan yang kurang lebih sama. Hal ini berarti akan lebih banyak produk yang tersedia dalam waktu dekat ini. Produk produk baru ini akan mempunyai algoritma baru, feature baru, dan fungsi fungsi baru. Dan akan mempunyai interface yang lebih user friendly.

2.1.2 Peranan Data Warehouse

Sebuah data warehouse adalah sebuah resource terpusat yang digunakan sebagai penunjang keputusan dan proses analisis. Sistem penunjang keputusan adalah sebuah teknologi yang membantu perusahaan untuk dapat membuat keputusan pada level eksekutif berdasarkan pada informasi yang ada pada basis data mereka. Data warehouse membantu bisnis untuk melakukan query pada basis data mereka, kemudian menganalisisnya. Data mining dalam hal ini merupakan langkah selanjutnya, dapat membantu perusahaan untuk menanyakan pertanyaan yang bersifat prediktif. Sebuah perusahaan Cola mungkin bertanya : “berapa banyak Minuman jenis X yang telah terjual di daerah indonesia timur selama triwulan terakhir ?”, atau jika dengan menggunakan teknik Data Mining pertanyaan

itu akan berupa : “berapa banyak Minuman jenis X yang akan terjual di daerah indonesia timur selama triwulan mendatang ?”.

Data warehouse dapat mempunyai peranan yang sangat penting dalam proses data mining. Hampir semua Algoritma data mining yang ada cenderung untuk tidak bekerja dengan baik pada data yang tersimpan dalam basis data operasional perusahaan. Karenanya dengan bekerja pada data yang tersimpan pada data warehouse dapat meringankan beberapa masalah. Yang pertama adalah data yang terdapat dalam data warehouse biasanya sudah siap dimining. Data itu sudah tersentralisasi dan tidak berada pada mesin operasional. Bahkan mungkin sudah dibersihkan (*cleansing*) dan diringkas secara ringan. Banyak ahli data mining yang memperhatikan proses penyiapan data sebagai proses yang paling kritis dalam keseluruhan proses. Bahkan Mr. Aaron Zornes, wakil presiden eksekutif Meta Group mengatakan bahwa : “60% sampai 80% investasi dalam data mining biasanya digunakan dalam penyiapan data”.

Data mining mungkin saja dilakukan tanpa data warehouse ataupun data mart. Namun demikian, perusahaan perusahaan besar yang siap melakukan data mining biasanya sudah mempunyai data ware house. Perusahaan perusahaan ini melihat kecerdasan bisnis (*bussines technology*) sebagai bagian dari masa depan mereka, dan untuk itu mereka sudah siap dengan data warehouse, data mining hanyalah satu langkah lebih lanjut.

2.1.3 Proses Penyiapan Data

Sebelum sebuah sistem data mining dijalankan, data terlebih dulu harus disiapkan agar hasil yang didapatkan lebih optimal, dan waktu komputasinya lebih minimal. Proses penyiapan data ini bisa sampai mencapai 60% investasi yang harus ditanam untuk data mining. Tahapan tahapan tersebut antara lain:

- Basis Data Relasional

Pada saat ini, hampir semua data data bisnis disimpan dalam basis data relasional. Sebuah model basis data relasional dibangun dari serangkaian tabel, setiap tabel disimpan sebagai sebuah file. Sebuah tabel relasional terdiri dari baris dan kolom. Kebanyakan model basis data relasional saat ini dibangun diatas lingkungan OLTP. OLTP (Online Transaction Processing) adalah tipe akses yang digunakan oleh bisnis yang membutuhkan transaksi konkuren dalam jumlah besar.

- Ekstraksi Data

Data dikumpulkan dengan cepat dalam sebuah basis data bisnis. Jika data tersebut disimpan dalam kantor regional, seringkali data tersebut di upload ke sebuah server yang lebih terpusat. Ini bisa dilakukan secara harian, mingguan, atau mingguan tergantung jumlah data, keamanan dan biaya. Data dapat diringkas dulu sebelum dikirimkan ke tempat penyimpanan pusat. Sebagai contoh, sebuah toko perangkat keras mungkin mengirim data yang

menunjukkan bahwa 18 palu telah terjual pada hari ini oleh karyawan nomer 26 dibanding pengiriman data detail transaksi.

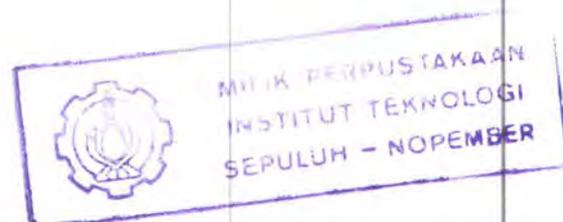
- Transformasi Data

Transformasi data melakukan peringkasan data dengan mengasumsikan bahwa data telah tersimpan dalam tempat penyimpanan tunggal. Pada langkah terakhir, data telah di ekstrak dari banyak basis data ke dalam basis data tunggal. Tipe peringkasan yang dikerjakan dalam langkah ini mirip dengan peringkasan yang dikerjakan selama tahap ekstraksi. Beberapa perusahaan memilih untuk meringkas data dalam sebuah tempat penyimpanan tunggal. Fungsi fungsi *Agregate* yang sering digunakan antara lain : *summarizations, averages, minimum, maximum, dan count.*

- Pembersihan Data

Proses pembersihan data dilakukan untuk membuang record record yang keliru, menstandarkan field field, merasionalisasi struktur data, dan mengendalikan data yang hilang. Data yang tidak konsisten dan banyak kekeliruan membuat hasil data mining tidak akurat. Adalah sangat penting untuk membuat data konsisten dan seragam. Pembersihan data juga dapat membantu perusahaan untuk mengkonsolidasikan record. Ini sangat berguna ketika sebuah perusahaan mempunyai banyak record untuk seorang pelanggan. Setiap record atau file pelanggan mempunyai nomor pelanggan yang sama, tetapi informasi dalam tiap filenya berbeda.

- Bentuk Standar



Bentuk standar adalah bentuk data yang akan diakses oleh algoritma data mining. Bentuk standar ini biasanya dalam bentuk *spreadsheet like*. Bentuk *spreadsheet* bekerja dengan baik karena baris merepresentasikan kasus dan kolom merepresentasikan feature.

- Reduksi Data dan Feature

Setelah data berada dalam bentuk standar *spreadsheet* perlu dipertimbangkan untuk mereduksi jumlah feature. Ada beberapa alasan untuk mengurangi jumlah feature dalam *spreadsheet* kita. Sebuah bank mungkin mempunyai ratusan feature ketika hendak memprediksi resiko kredit. Hal ini berarti perusahaan mempunyai data dalam jumlah yang sangat besar. Bekerja dengan data sebanyak ini membuat algoritma prediksi menurun kinerjanya.

- Teknik Sampling Data

Salah satu masalah utama dalam penerapan Data Mining pada basis data yang besar adalah besarnya waktu komputasi yang diperlukan untuk mengevaluasi pola pola terbaik dalam data. Karena besarnya waktu komputasi ini disebabkan oleh pemindaian setiap record dalam basis data, maka solusinya adalah dengan jalan melakukan sampling. Ada banyak teknik sampling data yang bisa dilakukan, mungkin yang paling umum di pakai adalah *uniform random sampling*, dalam teknik ini setiap record mempunyai probabilitas yang sama untuk di pilih sebagai data *sampling*. Teknik lain yang bisa dipakai adalah *satisfied random sampling* dan *cluster sampling*, meskipun kedua teknik ini lebih umum digunakan pada proses akuisisi data dan jarang

digunakan pada tahap pembuatan model. Jika diterapkan dengan tepat, sampling dapat secara signifikan mengurangi biaya komputasi yang diperlukan untuk mengevaluasi pola, dan tidak berpengaruh pada kualitas pola-pola yang dihasilkan.

- Menjalankan Algoritma

Setelah semua proses diatas dikerjakan, algoritma data mining sudah siap untuk dijalankan.

2.1.4 Model Model Data Mining

Ada banyak model yang digunakan dalam data mining. Setiap model mempunyai kelebihan dan kelemahan masing masing. memilih model yang tepat untuk menyelesaikan persoalan utama dalam bisnis tidak selalu mudah. Beberapa model yang umum digunakan antara lain : klasifikasi, regresi, *time series*, klusterisasi, *assosiasi* dan pencarian *sequence*.

Klasifikasi adalah model yang paling sering digunakan dalam *prediktif data mining*. Ini digunakan untuk mengelompokan kasus dalam beberapa grup. Data digunakan menentukan pola pola yang terjadi dalam grup.

Contoh untuk model ini adalah prediksi terhadap resiko pemberian kredit. Data terdiri dari orang orang yang telah menerima kredit. Sebagian kreditur menjalankan kewajiban dengan baik, dan sebagian lagi tidak. Data mining harus mampu mendefinisikan atribut atribut apa yang paling berpengaruh dalam hal ini.

Regresi juga digunakan dalam prediksi. Regresi di referensikan juga sebagai fungsi aproksimasi. Perbedaan utama antara regresi dan klasifikasi adalah bahwa regresi digunakan untuk memprediksi kelas dalam format bilangan real dari pada format katagorikal atau boolean.

Model peramalan *time series*, digunakan ketika kita hendak melihat bagaimana sebuah variabel akan merubah nilainya tiap waktu. data terdiri dari satuan yang diambil dalam beberapa periode waktu.

Klusterisasi digunakan untuk membedakan kasus dalam segmen yang berbeda. Tujuannya adalah untuk menemukan bagaimana suatu kasus berhubungan dan mempunyai kemiripan dengan yang lain.

Asosiasi adalah model yang digunakan untuk menemukan relasi antar sesuatu, khususnya kejadian yang terjadi pada saat yang bersamaan. Contoh yang sering digunakan untuk menunjukkan model asosiasi adalah dalam toko retail. Mereka menggunakan asosiasi untuk menemukan jika seorang pelanggan membeli item A, maka ada 90% kemungkinan dia juga akan membeli Item B.

Pencarian *sequence*, bekerja dengan cara yang sama dengan model asosiasi, kecuali kejadian yang dicari adalah yang *sequence* bukan yang terjadi pada saat yang bersamaan.

2.2 ALGORITMA GENETIKA

Dalam bagian kedua Bab ini dibahas dasar teori dari algoritma genetika. Pembahasan tentang algoritma genetika pada Bab ini bersifat teoritis. Sedangkan

untuk penjelasan secara lebih rinci algoritma yang disusun untuk sistem ini akan dibahas pada Bab III. Bahan untuk pembahasan Algoritma genetika disarikan dari [Ber-89] dan [Wic-96].

2.2.1 Umum

Algoritma genetika sebagai cabang dari algoritma evolusi merupakan metode adaptive yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah masalah optimasi. Algoritma ini didasarkan pada proses genetic yang ada dalam makhluk hidup, yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti prinsip alam.

Algoritma ini menggunakan analogi secara langsung dari kebiasaan yang alami yaitu seleksi alam. Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu individu, yang masing masing individu merepresentasikan sebuah solusi yang mungkin bagi persoalan yang ada. Dalam kaitan ini, individu dilambangkan dengan sebuah nilai fitness yang akan digunakan untuk mencari solusi terbaik dari persoalan yang ada.

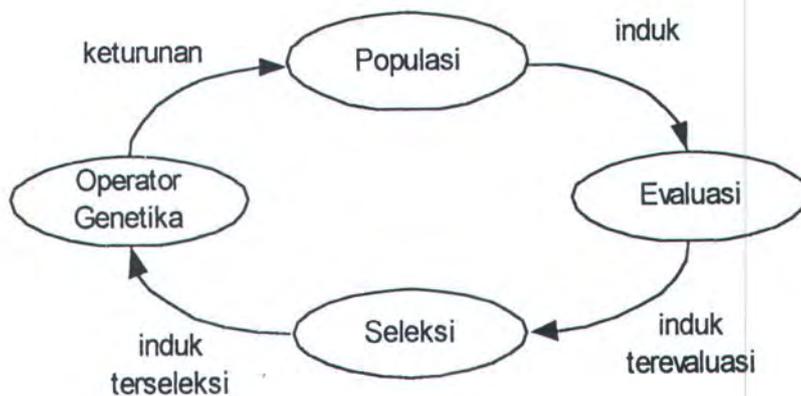
Pertahanan yang tinggi dari individu memberikan kesempatan untuk melakukan reproduksi melalui perkawinan silang dengan individu yang lain dalam populasi tersebut. Individu baru yang dihasilkan dalam hal ini dinamakan keturunan, yang membawa beberapa sifat dari induknya. Sedangkan individu dalam populasi yang tidak terseleksi dalam reproduksi akan mati dengan

sendirinya. Dengan jalan ini, beberapa generasi dengan karakteristik yang bagus akan bermunculan dari populasi tersebut. Dengan mengawinkan semakin banyak individu, maka semakin banyak kemungkinan terbaik yang bisa diperoleh.

Secara umum, dalam sebuah penerapan Algoritma Genetika akan melalui siklus sederhana yang terdiri 4 state, yaitu :

- Membangkitkan sebuah populasi
- Evaluasi masing masing individu
- Proses seleksi agar didapat individu yang lebih baik.
- Manipulais genetika untuk menciptakan populasi baru.

Untuk mendapatkan gambaran yang lebih jelas siklus ini, dapat dilihat pada gambar 2.1.



Gambar 2.1 Siklus generasional dalam GA

2.2.2 Sejarah Algoritma Genetika

Cikal bakal penggunaan Algoritma Genetika untuk pencarian dalam sistem diprakarsai oleh beberapa ahli biologi yang menggunakan komputer digital untuk mengerjakan simulasi dari sistem genetika. Diantaranya para ahli tersebut adalah [Cha-96] :

- Baricelli, N.A. pada tahun 1957 melakukan penemuan *Symbiogenetic Evolution Processes Realised by Artificial Method.*
- Baricelli, N.A. pada tahun 1962 melakukan penemuan *Numerical Testing Of Evolution Theories.*
- Fraser, A.S. pada tahun 1960 melakukan penemuan *Simulation Of Genetic Systems By Automatic Digital Computer : S-Linkage, Dominance, and Epistasis..*
- Fraser, A.S. pada tahun 1960 melakukan penemuan *Simulation Of Genetic Systems.*

Meskipun penelitian penelitian tersebut bertujuan untuk meneliti gejala alami, namun yang mereka kerjakan tidak begitu jauh dari pemikiran yang memunculkan ide tentang Algoritma Genetika. Fraser mensimulasikan evolusi dari 15-bit biner sebagai string generasi dan menghitung prosentase dari individu-individu yang terpilih oleh phenotype dengan generasi generasi yang berurutan. Pada saat itu Fraser tidak menyebutkan dalam laporannya bahwa algoritma pencarian dalam gejala alam akan berguna dalam sistem buatan, akan tetapi hasil dari penemuannya menyerupai fungsi optimasi.

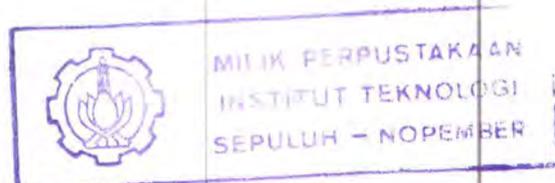
Hal itulah yang memberi inspirasi kepada Jhon Holland dan murid muridnya untuk mengaplikasikan proses genetika ini pada sistem buatan. Holland menancapkan pondasi terhadap aplikasi ini dengan karya tulisnya dalam teori sistem *Adaptive*, diantaranya :

- *Concerning Efficient Adaptive Systems (1962).*
- *Information Processing and Processing Systems (1962).*
- *Outline For A Logical Theory Of Adaptive Systems (1962).*

Pada tahun 1962-1965 Holland mengikuti kursus kursus dalam masalah adaptive di universitas Michigan. Salah satu kursusnya adalah *Theory Of Adaptive Systems*. Dalam seminar seminarnya Holland dan murid muridnya menyempurnakan detail dari Algoritma Genetika dan mengeksperimenkan dengan parameter parameternya, terutama menciptakan rumus standar dari Algoritma Genetika.

Selanjutnya teori mengenai Algoritma Genetika ini ditulis oleh Jhon Holland dalam bukunya *Adaptation In Natural And Artificial Systems* yang dipublikasikan pada tahun 1975.

Penemuan Holland, Algoritma Genetika ini adalah untuk membuktikan 2 hal. Yang pertama, bahwa teori Evolusi dapat digunakan sebagai sesuatu yang berguna untuk mencari fungsi fungsi optimasi pada sebuah komputer. Yang kedua, menciptakan lingkungan kerj abagi teori evolusi dengan sebuah permasalahan dunia nyata.



Dari sinilah Algoritma Genetika diterapkan dan dikembangkan oleh murid murid Holland. Salah satu muridnya menggunakan Algoritma Genetika pada aplikasi simulasi untuk mencari strategi yang paling mudah dalam permainan catur. Murid lainnya membuat disertasi yang menggunakan Algoritma Genetika untuk simulais fungsi fungsi dari organisme sel tunggal, sebagai simulasi biologi pertama yang menggunakan Algoritma Genetika. Ilmuwan ilmuwan yang selanjutnya mengembangkan Algoritma Genetika adalah D.E. Goldberg, K. De Jong, J.J. Grentenstette, L. Davis, Muhleinbein dan inspirator inspirator lainnya yang ikut menjadikan Algoritma Genetika berkembang.

2.2.3 Teori Dasar Algoritma Genetika

Sebelum kita menggunakan Algoritma Genetika, maka kita perlu memahami dulu beberapa teori dasar yang menunjang penerapan algoritma ini. Beberapa teori tersebut dibahas dalam beberapa sub bab berikut.

2.2.3.1 Beberapa Istilah dalam Algoritma Genetika

Seperti yang sudah dijelaskan sebelumnya, bahwa algoritma ini berlandaskan pada mekanisme genetika yang ada pada proses alami dan sistem buatan. Sehingga istilah istilah yang digunakan dalam Algoritma ini adalah gabungan dari 2 disiplin ilmu, yaitu Biologi dan Komputer.

Tabel 2.1 Beberapa istilah dalam Algoritma Genetika

Istilah	Keterangan
Kromosom	Individu, biasanya berupa segmen string yang sudah ditentukan
Gen	Bagian dari kromosom
Loci	Posisi gen
Alele	Nilai yang dimasukkan dalam gen
Phenotype	Solusi akhir
Genotype	Sejumlah individu yang berpotensi sebagai solusi

2.2.3.2 Parameter Parameter Algoritma Genetika

Dalam penerapan Algoritma Genetika, ada beberapa parameter yang dilibatkan, dimana parameter ini digunakan untuk melihat kompleksitas dari Algoritma Genetika. Jenis parameter yang digunakan tergantung pada permasalahan yang diselesaikan. Namun ada beberapa parameter yang menjadi standar, yaitu :

1 Ukuran Populasi (*Pop_Num*)

Ukuran populasi mempengaruhi kinerja dan keefektifan Algoritma Genetika. Jika populasinya kecil, biasanya kinerjanya buruk karena populasi tidak menyediakan cukup materi untuk mencakup ruang permasalahan. Populasi yang lebih besar dibutuhkan untuk merepresentasikan keseluruhan ruang persoalan.

Dan lagi dengan populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah lokal.

2 Probabilitas Pindah Silang (P_c)

Frekuensi operator pindah silang dikendalikan oleh sebuah nilai P_c . Dalam setiap populasi, sebanyak $P_c * Pop_num$ struktur melakukan pindah silang. Semakin tinggi nilai probabilitas pindah silang, semakin cepat struktur baru diperkenalkan dalam populasi. Jika probabilitas pindah silang terlalu tinggi, struktur dengan kinerja yang baik dapat hilang dengan lebih cepat pada proses seleksi, sehingga populasi tidak bisa meningkatkan unjuk kerja lagi. Sebaliknya, probabilitas yang rendah akan menghalangi proses pencarian.

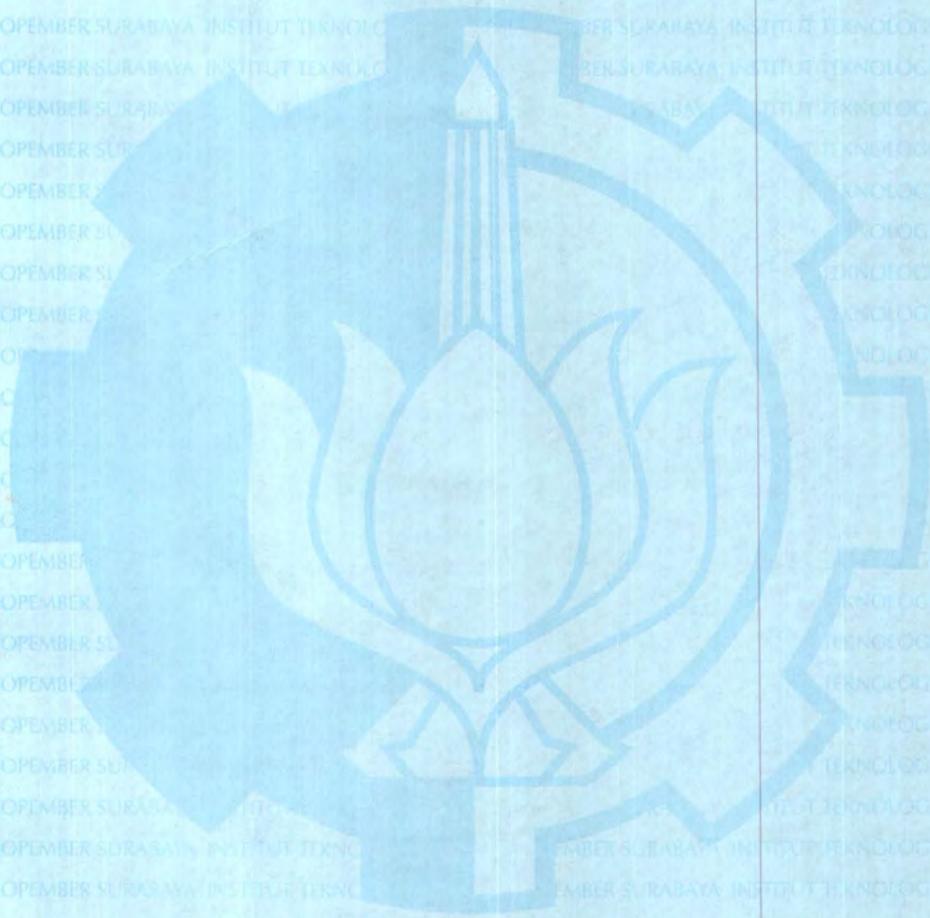
3 Probabilitas Mutasi (P_m)

Mutasi digunakan untuk meningkatkan variasi populasi. Mutasi ini dilakukan secara acak, tiap unit dasar (bit, posisi atau token) dalam struktur mempunyai kemungkinan tertentu untuk dipertukarkan. P_m yang rendah dapat mengakibatkan gen gen yang berpotensi tidak dicoba. Sebaliknya, tingkat mutasi yang tinggi dapat menyebabkan keturunan kehilangan kemiripan dengan induknya. Diperkirakan terjadinya mutasi sebanyak $P_m * Pop_Num * L$ pada tiap generasi, dimana L adalah panjang struktur dalam setiap individu.

2.2.3.2 Mekanisme Algoritma Genetika

Mekanisme yang ada dalam Algoritma Genetika sangat sederhana, yaitu hanya melibatkan operasi pindah silang, mutasi, evaluasi dan seleksi. Siklus

perkembangbiakan Algoritma Genetika diawali dengan pembuatan himpunan solusi secara random yang dinamakan populasi, dimana didalamnya terdapat individu individu yang dinamakan kromosom. Kromosom ini secara lambat laun mengalami iterasi pemilihan dalam sebuah generasi. Selama dalam sebuah generasi, kromosom kromosom ini dievaluasi dengan menggunakan rumus rumus yang ada dalam fungsi fitness. Untuk membuat generasi berikutnya dengan kromosom yang baru (dinamakan keturunan) dapat dilakukan sebelumnya dengan menggunakan operator pindah silang atau dengan memodifikasi sebuah kromosom dengan menggunakan operator mutasi. Sebuah generasi baru sebelum dievaluasi lagi, maka dia melalui proses seleksi berdasarkan fungsi fitnessnya. Dari seleksi ini, kromosom kromosom yang paling fit mempunyai kemungkinan besar untuk terseleksi.



BAB III
ALGORITMA GENETIKA
UNTUK PENCARIAN POLA IMPLIKASI
DALAM BASIS DATA

Dalam Bab ini dibahas suatu algoritma baru yang disusun untuk menerapkan Algoritma Genetika pada data mining. Seperti yang telah dijelaskan pada Bab II bahwa ada banyak model yang dapat diterapkan dalam data mining. Dan dalam Tugas Akhir ini, yang dijadikan perhatian utama adalah Pencarian Pola-pola Implikasi. Bentuk pola-pola implikasi yang dimaksud akan dibahas lebih lanjut dalam sub bab 3.1 yang materinya disarikan dari [FLO-96].

Algoritma yang disusun untuk menerapkan algoritma genetika pada Data Mining untuk pencarian pola-pola dalam basis data ini diberi nama GA-Miner. Untuk pembahasan selanjutnya, kata GA-Miner digunakan untuk menyebutkan algoritma ini. Seperti yang telah dijelaskan bahwa algoritma genetika telah banyak diterapkan dalam masalah optimasi dan pencarian. Namun, selama ini penerapannya masih terbatas pada optimasi permasalahan numerik, padahal teknik ini dapat digunakan untuk memanipulasi beragam struktur data, dan telah lama dieksploitasi untuk memanipulasi aturan (rule). Pembahasan tentang GA-Miner ini akan dimulai dengan pengkodean kemudian dilanjutkan ke algoritma algoritma dalam operator genetika, Algoritma-algoritma tersebut meliputi

operator pindah silang, mutasi dan lain lain. Sedangkan fungsi fungsi lain seperti fungsi evaluasi dan mekanisme seleksi dibahas dalam sub bab tersendiri.

3.1 Pola Implikasi

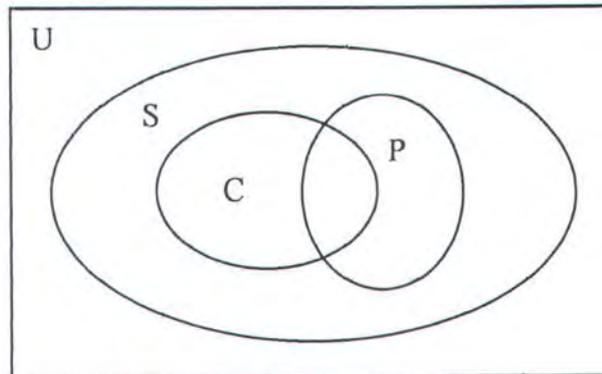
Tujuan Data Mining adalah untuk menemukan pola-pola yang menarik dan berharga dalam suatu basis data yang sulit atau bahkan tidak mungkin untuk ditemukan dengan menggunakan metode query standar dan teknik statistik klasik. Dalam hal ini, kami memandang bahwa bentuk yang paling baik untuk menampilkan Pola yang dihasilkan dari proses Data Mining adalah dalam bentuk yang kami sebut sebagai Pola eksplisit atau *Implication Rule*.

Aturan tersebut adalah sebuah predikat *If-Then* yang terdiri dari tiga bagian: *Specify (S)*, sejumlah klausa kondisional (C), dan sebuah klausa Prediktif (P). sehingga bentuknya akan seperti ini :

Untuk S, Jika C1 dan C2 dan : : : dan Cn Maka P1 dan : : : Pn

S mengindikasikan pada bagian mana dari basis data aturan tersebut diaplikasikan. C terdiri dari beberapa klausa tunggal yang digabungkan. Setiap klausa menunjuk pada satu atau dua field dalam basis data. P merupakan sebuah klausa tunggal. Ilustrasi penyeleksian record-record data dari basis data yang memenuhi batasan-batasan (*constraints*) yang telah dispesifikasikan dalam aturan dapat dilihat dalam gambar 3.1.

Dari subset yang terpilih ini, kemudian bisa di evaluasi aturan *If C Then P* (implikasi satu arah) atau *If only If C Then P* (implikasi dua arah) dengan menggunakan fungsi evaluasi yang di pilih.



Gambar 3.1 Ilustrasi penyeleksian record record basis data

Dari subset yang terseleksi kita dapat melakukan evaluasi apakah suatu aturan tersebut masuk dalam katagori bagus atau jelek. Ada beberapa ukuran yang dapat digunakan untuk melihat kualitas dari sebuah aturan, dalam tugas akhir ini yang digunakan antara lain :

Accuracy dari sebuah aturan didefinisikan sebagai jumlah record yang memenuhi C dan P di bagi dengan total jumlah record yang memenuhi C, sehingga :

$$Accuracy = \frac{|S \cap C \cap P|}{|S \cap C|} = \frac{|S \cap C \cap P|}{(|S \cap C \cap P| + |S \cap C \cap \bar{P}|)}$$

Coverage didefinisikan sebagai jumlah record yang memenuhi C dan P di bagi dengan total jumlah record yang memenuhi P.

$$Coverage = \frac{|S \cap C \cap P|}{|S \cap P|} = \frac{|S \cap C \cap P|}{(|S \cap C \cap P| + |S \cap \bar{C} \cap P|)}$$

Support didefinisikan sebagai jumlah record yang memenuhi C di bagi dengan total jumlah record S.

$$Support = \frac{|S \cap C|}{|S|}$$

Contoh kongkritnya akan seperti ini :

untuk Produk = {Produk1}, Jika bulan = {November, Desember} dan Curah hujan > 30 , Maka Total Penjualan > 100.

Accuracy = 82 %.

Coverage = 55 %.

Basis dari pola-pola aturan tersebut adalah sebuah tabel kontingensi 2X2 yang memuat jumlah record untuk masing masing subset seperti tabel berikut :

Tabel 3.1 Tabel Kontingensi 2 X 2

$S \cap$	\bar{P}	P
\bar{C}	$S \cap \bar{C} \cap \bar{P}$	$S \cap \bar{C} \cap P$
C	$S \cap C \cap P$	$S \cap C \cap \bar{P}$

Sebagai contoh, perhatikan pola ini : *Untuk S, C dan P adalah berelasi.*

Lebih tepatnya, istilah *berelasi* dapat diartikan *tidak independen secara statistik.*

Untuk melihat bagaimana hal ini dapat menginterpretasikan sebuah aturan, perhatikan dari contoh diatas, jika ternyata jumlah record yang memenuhi $S \cap C \cap P$ adalah lebih besar dari jumlah record yang memenuhi $S \cap C \cap \bar{P}$, maka hal ini berarti bahwa aturan : *jika C maka P* adalah benar untuk jumlah yang besar dari data record, meskipun mungkin saja *accuracy* nya tidak terlalu besar.

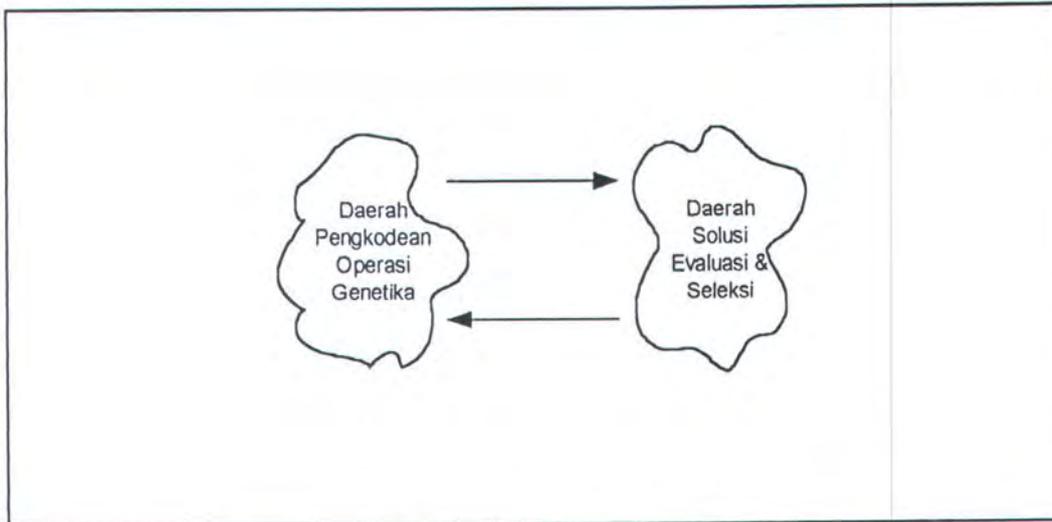
Hal penting yang harus diperhatikan dalam penyelesaian masalah optimasi dengan Algoritma Genetika adalah fungsi Evaluasi. Dalam kasus ini fungsi Evaluasi di gunakan untuk mengukur kualitas sebuah aturan. Kita mungkin berpikir bahwa *accuracy* seperti yang telah didefinisikan di atas mungkin akan

digunakan sebagai fungsi evaluasi. Namun, meskipun suatu aturan telah mempunyai *accuracy* 100% belum tentu informasi yang ada di dalamnya akan berguna.

Secara umum, *accuracy*, *coverage*, dan *support* sebuah aturan adalah bermanfaat untuk menggambarkan kondisi aturan, namun demikian, mereka belum cukup untuk digunakan sebagai suatu *fungsi* evaluasi. Ada beberapa Fungsi evaluasi yang akan digunakan dalam sistem ini. Untuk pembahasan lebih lanjut akan dibahas dalam sub bab tersendiri.

3.2 Pengkodean

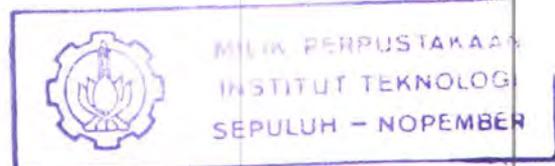
Pengkodean adalah salah satu bagian yang penting dalam algoritma genetika, karena dengan pengkodean yang tepat solusi yang dihasilkan akan lebih optimal dan dapat mengurangi biaya komputasi [Wic-96]. Dalam sistem ini masalah pengkodean menjadi lebih penting karena tidak seperti pada SGA dan optimasi permasalahan numerik, pengkodean dalam sistem ini bukan dengan sistem biner, namun sebuah struktur yang sangat kompleks. Dalam GA-Miner, Individu (kromosom) adalah sebuah aturan yang strukturnya didesain sesuai dengan pola yang hendak dicari (Pola Implikasi).

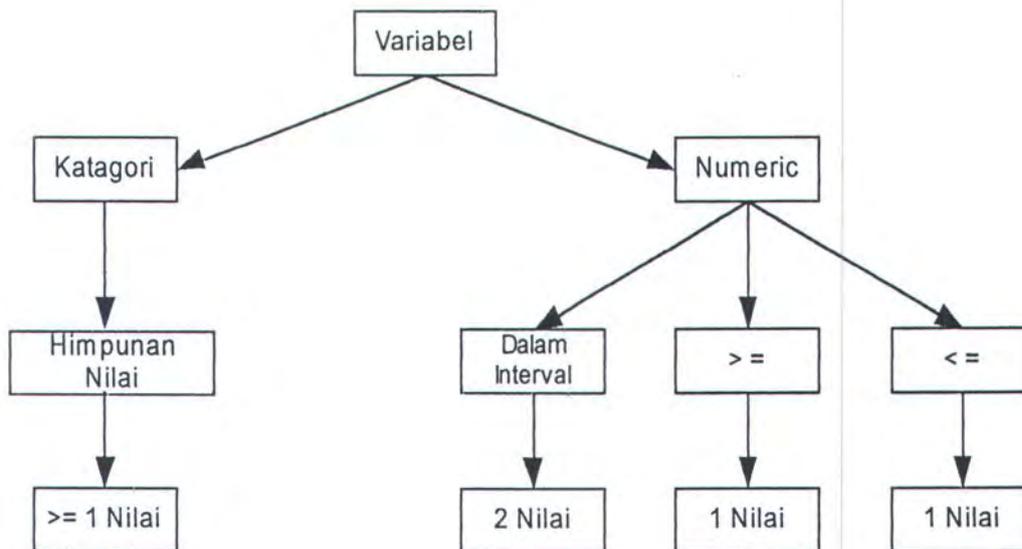


Gambar 3.2 Ilustrasi Pengkodean dalam GA

3.2.1 Pengkodean Aturan

Dalam sistem ini sebuah aturan adalah kromosom yang terdiri dari beberapa klausa yang merupakan materi genetik penyusun aturan. Sebuah klausa terdiri dari satu atau lebih klausa tunggal yang di sebut Term. Ada tiga bagian klausa dalam sebuah aturan sesuai dengan bentuk pola Implikasi (3.1), yaitu klausa specify S , klausa kondisional C, dan klausa prediktif P. Sedangkan sebuah Term terdiri dari tiga bagian yakni : variabel yang merferensi ke sebuah field dalam basis data, sebuah tanda dan satu atau beberapa nilai. Tanda dari sebuah term tergantung pada tipe data variabelnya. Sedangkan jumlah nilai tergantung pada tandanya (lihat gambar 3.3).





Gambar 3.3 Tipe variabel, Tanda dan nilai

Untuk lebih jelasnya akan diberikan contoh konkrit dalam patagrap berikut ini.

Misalkan sebuah basis data mempunyai field field berikut :

Tabel 3.2 Contoh Sebuah tabel dalam basis data

Produk	Bulan	Kota	Penjualan
Produk1	Januari	Surabaya	30
Produk1	Januari	Malang	8
Produk1	Pebruari	Surabaya	25
Produk1	Pebruari	Malang	10
Produk2	Januari	Surabaya	20
Produk2	Januari	Malang	5
Produk2	Pebruari	Surabaya	20
Produk2	Pebruari	Malang	5

Dari contoh tabel diatas, bisa dibangkitkan beberapa Term. Sebagai contoh :

Term 1 : produk =(Produk1)

Term 2 : bulan = {Januari}

Term 3 : Penjualan > = 30

dst.

Dari term term diatas bisa dibentuk sebuah aturan, misalkan untuk klausa sepecific nya adalah Term1, klausa kondisinya Term2 dan klausa Prediktifnya adalah Term3. Maka dapat dibentuk sebuah aturan :

Untuk Produk = {Produk1}, Jika bulan =(Januari), Maka Penjualan >= 30.

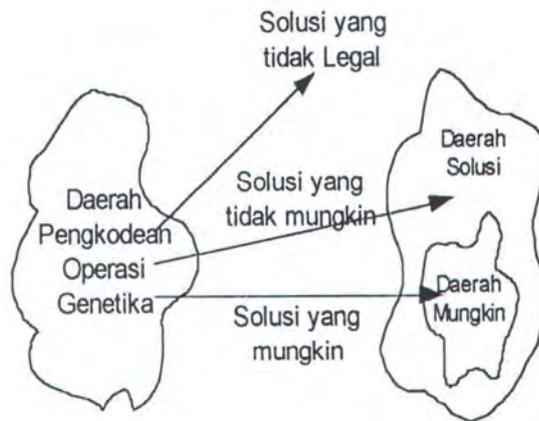
Masih ada beberapa aturan yang bisa dihasilkan dari contoh basis data diatas, tergantung field field mana yang ditentukan sebagai klausa specify, kondisional dan prediktif.

3.2.2 Mekanisme Kontrol

Dalam pengkodean sistem yang bukan biner, ada 2 hal yang harus diperhatikan :

- Feasibilitas dan legalitas dari sebuah individu.
- Pemetaan

Feasibilitas mempertanyakan apakah sebuah solusi yang direpresentasikan terletak dalam daerah yang fisibel dari domain permasalahan [Ber-89]. Legalistas merujuk pada kenyataan apakah sebuah individu benar benar merepresentasikan sebuah solusi dari domain permasalahan [Ber-89]. Sedangkan pemetaan yang dipakai dalam sistem ini adalah 1 ke 1, artinya sebuah individu (dalam hal ini sebuah aturan) merepresentasikan sebuah solusi.



Gambar 3.4 Feasibilitas dan legalitas sebuah individu.

Dalam usaha untuk menjaga feasibilitas dan legalitas individu maka dalam pengkodean sistem ini dibuat sebuah mekanisme *constraint handling*. Mekanisme tersebut menggunakan kumpulan tabel tabel yang dijadikan rujukan ketika algoritma pembangkitan populasi, aturan, klausa, term, variabel, tanda dan nilai. Dengan mekanisme ini sebuah individu dijaga agar tidak keluar dari domain solusi yang diperkenankan. Kumpulan tabel ini disebut sebagai Referensi Utama.

Kumpulan tabel tersebut adalah tabel fields, tabel kode dan tabel klausa. Tabel fields berisi data data tentang field field dalam basis data. Data tersebut adalah kode field, nama field, tipe datanya, nilai minimum dan maksimum untuk field yang bersangkutan. Tabel lainnya adalah tabel kode yang digunakan untuk menyimpan nilai nilai dalam suatu field dan kemudian dikodekan. Hal ini dimaksudkan agar dalam pengkodeannya dapat lebih hemat dan lebih mudah dalam operasi genetiknya, karena untuk menyimpan sebuah nama field yang panjang misalnya tidak diperlukan string yang panjang tapi cukup sebuah angka yang tidak lebih dari 2 digit. Sedangkan tabel klausa menyimpan field field yang telah ditentukan sebagai klausa specify, kondisi dan prediktif.

Referensi Utama sangat penting bukan hanya dalam pembangkitan variabel dan nilai, tapi juga tanda. Karena variabel tertentu dengan tipe data tertentu hanya diperbolehkan mempunyai tanda tertentu (keterangan lebih lanjut dalam 3.4.2).

3.3 Operator Genetika

Operator genetika mempunyai peranan yang besar dalam keberhasilan algoritma genetika, penyusunan operator genetika yang baik akan mengurangi waktu komputasi dan membuat solusi yang dihasilkan lebih optimal.

Operator operator genetika yang dibahas disini adalah antara lain : Operator Pembangkitan, pindah silang dan mutasi. Sedangkan algoritma untuk fungsi evaluasi dan metode seleksi dibahas dalam Sub Bab tersendiri. Tapi sebelumnya terlebih dulu akan dibahas Algoritma yang digunakan untuk pra pencarian (*pre search*).

3.3.1 Pra Pencarian.

Seperti yang telah dijelaskan, proses ini dimaksudkan untuk membentuk tabel tabel yang akan digunakan sebagai acuan dalam mekanisme *constraint handling*.

3.3.2 Algoritma Pembangkitan populasi

Algoritma pembangkitan populasi digunakan untuk membentuk populasi awal secara acak pada pencarian tidak terarah (*undirected*). Sedangkan pada

pencarian terarah (*directed*) parameter parameter yang diberikan user akan digunakan sebagai acuan dalam pembentukan populasi awal.

Proses ini tidak berdiri sendiri karena juga melibatkan algoritma pembangkitan aturan, sedangkan algoritma pembangkitan aturan nantinya akan memanggil algoritma pembangkitan klausa, demikian juga algoritma pembangkitan klausa memanggil algoritma pembangkitan term dan algoritma pembangkit term selanjutnya memanggil algoritma pembangkit variabel, sign, dan nilai. Semua algoritma pembangkitan yang telah disebutkan tadi juga akan dibahas dalam sub bab ini.

3.3.2.1 Algoritma Pembangkitan Term

Algoritma ini akan membuat sebuah term secara acak.

Langkah 1: Bangkitkan variabel secara acak.

Langkah 2: Bangkitkan sign secara acak sesuai variabelnya.

Langkah 3: bangkitkan nilai secara acak sesuai variabel dan signnya.

3.3.2.2 Algoritma Pembangkitan Klausa

Algoritma ini digunakan untuk membuat sebuah klausa baik itu kondisional maupun prediktif.

Langkah 1: Bangkitkan jumlah term n secara acak.

Langkah 2: Bangkitkan term sejumlah n dengan algoritma pembangkitan term.

3.3.2.3 Algoritma Pembangkitan Aturan

Algoritma ini digunakan untuk membuat sebuah aturan secara acak.

- Langkah 1: Bangkitkan sebuah klausa kondisional dengan menggunakan algoritma pembangkitan klausa.
- Langkah 2: Bangkitkan sebuah klausa prediktif dengan menggunakan algoritma pembangkitan klausa.
- Langkah 3: Periksa validitas aturan yang baru dibuat.

3.3.3 Pindah Silang

Operasi pindah silang pada sistem ini digunakan untuk mendapatkan sebuah aturan tunggal baru yang di hasilkan dari dua buah induk. Seperti juga pada algoritma pembangkitan populasi, operasi pindah silang pada sistem ini tidak berdiri sendiri, karena selain pada level aturan, pindah silang juga dilakukan pada level Klausa, bahkan pada level sign dan nilai. Algoritma untuk masing masing operasi akan dijelaskan lebih lanjut.

3.3.3.1 Pindah Silang Term

Dalam level ini term yang mempunyai variabel yang sama dari ke dua induk dipindahsilangkan untuk mendapatkan sebuah term anak tunggal. Algoritmanya adalah sebagai berikut :

- Langkah 1 : periksa apakah sign kedua Term sama, jika ya kerjakan langkah 2, jika tidak kerjakan langkah 6.
- Langkah 2 : periksa sign, jika sign=0 kerjakan langkah 3, jika sign=1 kerjakan langkah 3, jika sign = 2 atau sign = 3 kerjakan langkah 5.
- Langkah 3 : secara acak tentukan apakah nilai kedua term akan digabung atau di pilih salah satu. Jika di gabung kerjakan langkah 3a, jika sebaliknya kerjakan langkah 3b.
- Langkah 3a: gabungkan nilai dari kedua induk, jika ada yang sama pilih salah satu. Jika jumlah nilai lebih besar dari yang diijinkan nilai dari induk 1 di beri prioritas

dengan probabilitas sebesar $P_{crossTerm1}$, sedangkan probabilitas induk 2 sebesar $(1-P_{crossTerm1})$.

Langkah 3b: pilih daftar nilai dari salah satu induk.

Langkah 4 : secara acak pilih satu nilai dari keempat nilai sebagai batas bawah Range 1, kemudian secara acak pula pilih satu nilai dari ketiga nilai yang tersisa sebagai batas atas Range 2. Jika Rangel lebih besar dari Range 2 tukar nilainya.

Langkah 5 : pilih salah satu nilai dari kedua nilai induknya.

Langkah 6 : pilih salah satu sign dari kedua sign induknya beserta nilainya.

Untuk lebih jelasnya diberikan contoh berikut:

Term 1 : Bulan = { Januari }.

Term 2 : Bulan = { Pebruari }.

Setelah dipindahsilangkan Hasilnya dapat berupa :

Term 3 : Bulan = { Januari, Pebruari }.

3.3.3.2 Pindah Silang Klausa

Pindah silang klausa dalam sistem ini menggunakan metode *uniform crossover*, dalam metode ini hanya term yang mempunyai variabel yang sama pada kedua klausa induk saja yang dipasangkan untuk dipindah silang dengan menggunakan pindah silang term. Sedangkan term yang tidak mempunyai pasangan di turunkan ke anak dengan kemungkinan P_c untuk term dari induk pertama dan $(1-p_c)$ untuk term dari induk kedua. klausa baru yang dihasilkan selanjutnya di cek validitasnya dengan menggunakan fungsi validitas klausa.

Contoh :

Klausa I : produk = {Produk1, Produk2} dan bulan = {Januari}.

Klausa II : bulan = {Pebruari}.

Dari kedua klausa tersebut hanya term dengan variabel *bulan* saja yang sama, sehingga kedua term tersebut dipindahsilangkan dengan operasi pindah silang Term (lihat 3.3.3.3). Sedangkan term term yang lain akan dipertimbangkan untuk diturunkan ke anaknya.

Hasilnya dapat berupa :

Klausa hasil : produk = { Produk1, Produk2 } dan bulan = { Januari }.

3.3.3.3 Pindah Silang Aturan

Dalam level ini, setiap Klausa dalam induk pertama dipasangkan dengan Klausa induk kedua yang mempunyai tipe yang sama untuk dipindahsilangkan dengan Pindah Silang Klausa (3.3.3.2). Jadi klausa kondisi C pada induk pertama dipidahsilangkan dengan Klausa kondisi C pada induk kedua, hal yang sama juga untuk klausa prediktif P maupun Klausa Specify S. Hal ini dilakukan dengan menggunakan algoritma pindah silang Klausa untuk menghasilkan Klausa anak tunggal. Output dari algoritma ini adalah sebuah aturan baru.

Untuk lebih jelasnya diberikan contoh berikut :

Rule 1 :

Untuk Produk = {Produk2}, Jika bulan =(Januari), Maka Penjualan >= 30.

Rule 2 :

untuk produk = {produk1}, Jika kota = {Surabaya}, Maka Penjualan <=30.

Setelah dioperasikan hasilnya dapat berupa :

Rule 3 :

untuk produk = {produk1}, Jika bulan = {Januari}, Maka Penjualan >=30.

3.3.4 Mutasi

Operasi ini dimaksudkan untuk menghasilkan sebuah aturan baru dari sebuah induk tunggal. Operasi mutasi ini juga dilakukan pada beberapa level yang berbeda seperti pada operasi pindah silang. Disamping itu operasi mutasi juga di mungkinkan untuk melakukan pemanggilan terhadap fungsi fungsi pembangkitan term. Seperti pada pindah silang setiap klausa dalam aturan di mutasikan dengan memanggil fungsi mutasi klausa. Berikut ini akan di jelaskan operasi mutasi untuk masing masing level.

3.3.4.1 Mutasi Nilai Tunggal

Untuk nilai tunggal akan dimutasikan dengan terlebih dulu membangkitkan sebuah konstanta yang akan dijadikan interval. Selanjutnya nilai tersebut ditambah atau dikurangi dengan probabilitas P_{creep_inc} . Pengurangan atau penambahan nilai tersebut menggunakan interval yang telah dibangkitkan secara acak berdasarkan besarnya domain nilainya.

Contoh sebuah Term : `Penjualan >= 30`

Setelah dimutasikan Term tersebut dapat menjadi : `Penjualan >= 25.`

3.3.4.2 Mutasi Range

Seperti pada mutasi nilai tunggal, nilai dalam bentuk range akan dimutasikan dengan terlebih dulu membangkitkan sebuah konstanta yang akan dijadikan interval. Kedua nilai minimum dan maksimum ditambah atau dikurangi

dengan probabilitas P_{creep_inc} . Pengurangan atau penambahan nilai tersebut menggunakan interval yang telah dibangkitkan secara acak.

Contoh sebuah Term : Penjualan dalam interval 10 .. 20.

Setelah dimutasikan , dapat menjadi : Penjualan dalam interval 10 .. 20.

3.3.4.3 Mutasi Nilai Katagori

Operasi mutasi pada Nilai Katagori dilakukan dengan 4 pilihan. Yang pertama : mengganti himpunan nilai lama dengan himpunan nilai baru. Yang kedua : dengan menambahkan nilai baru pada himpunan nilai yang sudah ada. Cara yang ketiga : dengan menghapus nilai tertentu dalam himpunan nilai. Cara yang keempat : mengganti nilai tertentu dalam himpunan dengan nilai baru.

Contoh sebuah Term : Bulan = { Maret, September }.

Setelah dimutasikan, dapat menjadi : Bulan = { Maret, April }.

3.3.4.5 Mutasi Term

Sebuah term dimutasikan dengan probabilitas pmt . Term yang telah diseleksi untuk dimutasikan, variabelnya dimutasikan dengan probabilitas $pmtv$. Term yang bertanda tetap dicegah untuk tidak dimutasikan. Ketika sebuah variabel dipilih untuk dimutasikan, variabel tersebut akan digantikan dengan variabel baru yang dibangkitkan secara acak dengan fungsi pembangkitan variabel, kemudian seluruh bagian term akan dibangkitkan juga secara acak hingga terbentuk sebuah term yang sama sekali baru. Jika variabel tidak dipilih

untuk dimutasi, nilai tunggal atau dalam bentuk range dipertimbangkan untuk di mutasikan dengan probabilitas $pmtval$.

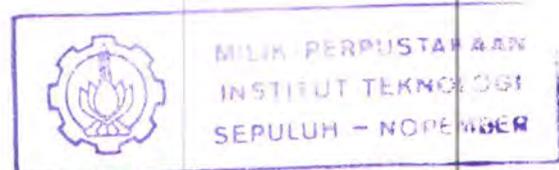
Mutasi pada nilai ini dibedakan berdasarkan tipe variabelnya. Untuk variabel yang bertipe numerik dilakukan dengan memanggil fungsi mutasi nilai yang menggunakan metode *creep mutation*. Penanganan mutasi nilai tunggal dibedakan dengan penanganan nilai range. Untuk nilai tunggal akan digantikan dengan nilai baru yang dibangkitkan dengan fungsi mutasi nilai tunggal (3.3.4.1). Sedangkan pada nilai dalam bentuk range akan dimutasikan dengan fungsi mutasi range (3.3.4.2). Sedangkan untuk variabel yang bertipe katagori dimutasikan dengan fungsi mutasi Nilai Katagori (3.3.4.3).

3.3.4.4 Mutasi Klausula

Setiap Term dalam klausula terlebih dulu dimutasikan dengan fungsi mutasi term. Kemudian term dalam klausula akan ditambah atau dikurangi dengan probabilitas $Pmutate$ (probabilitas penambahan dan pengurangan Term adalah sama yakni 0.5).

Term diseleksi untuk dihapus secara acak, dan jika kebetulan yang terpilih adalah term yang bertanda tetap, maka penghapusan dibatalkan dengan tidak menyeleksi term lain untuk dijadikan ganti. Penambahan Term dilakukan jika jumlah Term dalam satu klausula tidak melebihi jumlah yang diijinkan. Term baru yang ditambahkan tersebut dibangkitkan dengan menggunakan fungsi pembangkit Term (3.3.2).

Untuk lebih jelasnya ikuti contoh berikut :



Klausa I : Produk = { Produk1, Produk 2} dan Bulan = { Maret}.

Setelah dimutasi, kalusa tersebut dapat menjadi :

Klausa II : Produk = { Produk1 } dan Bulan = { Maret, April}.

3.4 Fungsi Evaluasi

Fungsi evaluasi yang digunakan dalam GA-Miner adalah Fungsi yang usulkan oleh piatetsky-saphiro [Fra-91] yang di beri nama *rule interest*.

$$Rule-Interest = \frac{|S \cap C \cap P| - (|S \cap C| \times |S \cap P|)}{|S|}$$

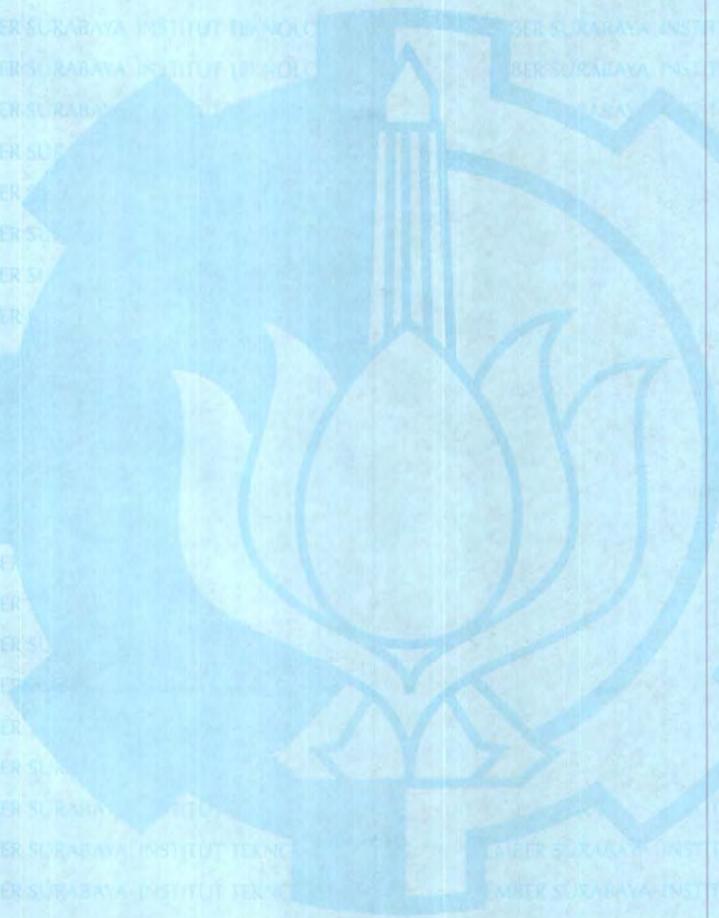
3.5 Mekanisme Seleksi dan Penggantian

Dalam GA-Miner, mekanisme seleksi dan penggantian digunakan untuk mendapatkan populasi baru yang diseleksi dari populasi induk dan keturunan yang didapat dari operasi pindah silang dan mutasi. Dalam mekanisme ini setiap individu keturunan *Ioff* terlebih dulu dievaluasi dengan fungsi evaluasi yang dipilih, kemudian akan dilakukan pencocokan ke semua anggota populasi lama untuk dicari individu individu yang mempunyai derajat kemiripan dalam tresshold tertentu hingga didapatkan himpunan individu *PSim*. Kemudian individu dalam *PSim* di urutkan dari yang paling jelek. Selanjutnya diperiksa, untuk semua individu dalam *PSim* dimulai dari individu yang paling jelek. jika fitness *Ioff* lebih besar, maka gantikan posisi individu tersebut. Jika tidak ditemukan individu yang mempunyai derajat kemiripan sesuai tresshold, maka dicari individu terjelek

dalam populasi lama, kemudian jika fitness I_{off} lebih besar, maka ditukar [Flo-96].

3.6 Siklus Generasional

Dalam algoritma genetika, individu individu terbaik akan dihasilkan setelah beberapa generasi. Pada mulanya sebuah populasi awal P_0 dioperasikan dengan operator pindah silang dan mutasi sehingga menghasilkan offspring (keturunan) P_0' . Selanjutnya dari P_0 dan P_0' ini dipilih individu individu terbaik sehingga menghasilkan populasi baru P_1 dengan menggunakan mekanisme seleksi. Proses selanjutnya P_1 ini kembali dioperasikan dengan operator genetika sehingga menghasilkan populasi baru P_2 , dan seterusnya sampai generasi ke- n . setelah iterasi ke- n , maka proses dihentikan dan dari populasi akhir yang terbentuk P_n diambil individu individu terbaik yang memenuhi treshold yang telah ditentukan [Flo-96].



BAB IV

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Dalam bab ini dibahas perancangan dan pembuatan perangkat lunak untuk mengimplementasikan algoritma yang telah dijelaskan dalam Bab iii. Sebelum pembahasan yang lebih mendalam tentang implementasinya dalam program, terlebih dulu akan dijelaskan deskripsi sistem yang dibangun. Kemudian dilanjutkan dengan pembahasan tentang perancangan dan pembuatan perangkat lunaknya sendiri. Dimulai dari perancangan data, perancangan proses dan perancangan antar muka. Kemudian dilanjutkan dengan pembahasan tentang implementasinya dalam program.

4.1 Deskripsi Sistem

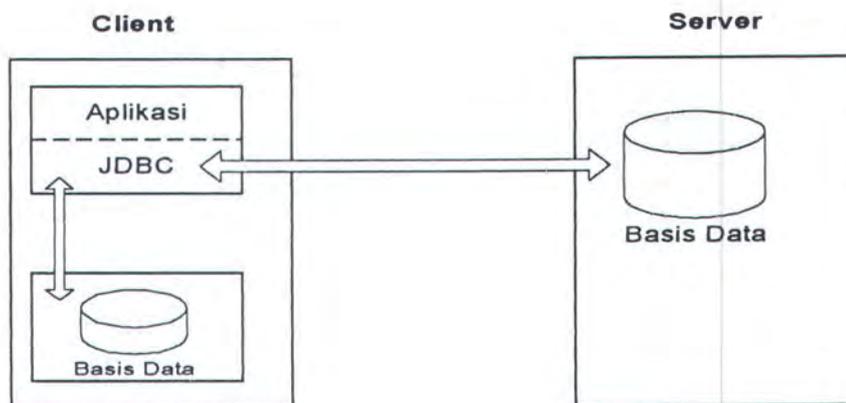
Sesuai dengan tujuan tugas akhir yang sudah diterangkan di Bab i, maka dibuat kriteria kriteria yang harus dimiliki oleh perangkat lunak yang akan dibangun. Kriteria kriteria tersebut antara lain :

1. Perangkat lunak mampu membuka suatu sumber data, yang berupa file file basis data relasional dalam format tertentu baik itu dalam mesin client maupun server.
2. Perangkat lunak mampu melakukan pencarian pola pola implikasi dalam basis data yang di maksud.

3. Perangkat lunak mampu memberikan interpretasi terhadap aturan aturan yang didapatkan dari proses pencarian dengan menampilkannya dalam bahasa yang mudah dimengerti.

Untuk memenuhi kriteria kriteria di atas, dibangunlah perangkat lunak yang akan dibahas dalam beberapa sub bab selanjutnya.

Perangkat lunak yang dibangun adalah sebuah sistem data mining yang bertujuan untuk menemukan pola pola implikasi (3.1) dari sebuah basis data dengan menggunakan algoritma genetika. Sistem ini didesain untuk mendukung teknologi client/server, dimana aplikasi dijalankan dari client dan data didapat dari akses ke server. Hal ini dilakukan karena biasanya data data dalam jumlah yang besar dikumpulkan dalam sebuah server. Namun demikian, juga dimungkinkan untuk mengakses data yang disimpan di mesin client. Ilustrasi sistem ini secara fisik dapat dilihat pada gambar 4.1



Gambar 4.1 Ilustrasi Sistem secara fisik

Sistem mengakses sumber data baik itu yang ada di server maupun di client untuk di proses hingga menghasilkan output berupa aturan aturan yang disajikan dalam bentuk yang mudah dimengerti.

4.2 Perancangan Perangkat Lunak

Dalam bagian ini dijelaskan mengenai perancangan sistem perangkat lunak. Perancangan tersebut meliputi perancangan data, perancangan proses, dan perancangan antar muka.

4.3.1 Perancangan Data

Data yang digunakan dalam sistem ini dibagi kedalam tiga katagori yakni : data masukan, data proses dan data keluaran.

4.3.1.1 Perancangan Data Masukan

Data yang digunakan sebagai masukan sistem ini berasal dari parameter parameter yang diberikan pengguna. Parameter parameter yang harus dimasukan oleh pengguna ini disajikan dalam bentuk tabel sebagai berikut :

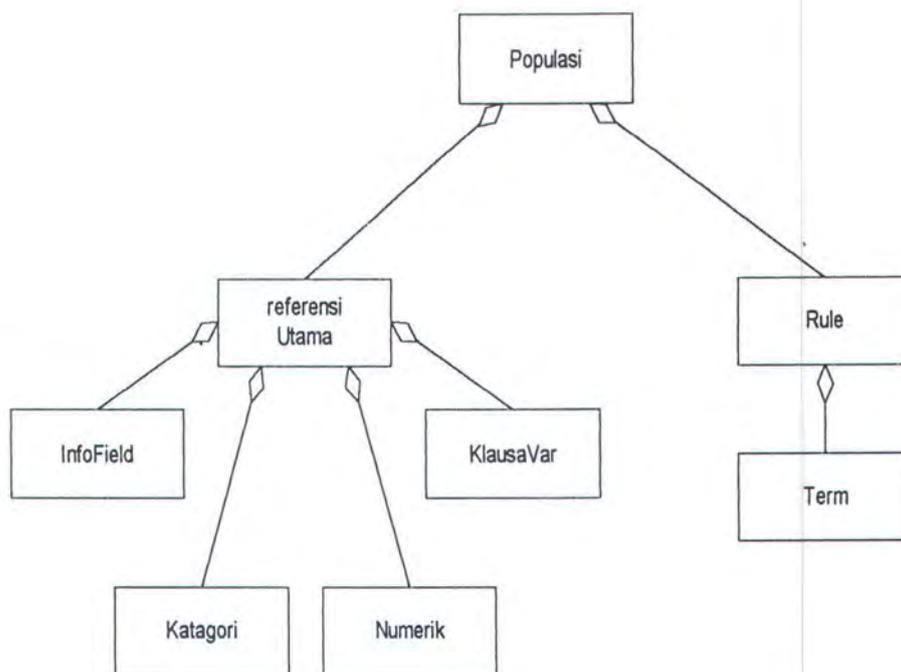
Tabel 4.1 Data Masukan

Nama Parameter	Tipe Data	Keterangan
Jumlah Generasi	Integer	menentukan berapa generasi sistem dijalankan.
Ukuran Populasi	Integer	menentukan berapa banyak aturan

		(individu) dalam sebuah populasi
CrossOverRate	Float	menentukan berapa kemungkinan terjadinya operasi pindah silang. Nilainya berkisar antara 0 s/d 1
MutateRate	Float	menentukan berapa kemungkinan terjadinya operasi mutasi. Nilainya berkisar antara 0 s/d 1.
Fungsi Evaluasi	Integer	menentukan tipe fungsi evaluasi yang digunakan.
Koneksi	Connection	merupakan koneksi ke mesin basis data server.
Nama Tabel	String	nama tabel dimana data yang akan diproses tersimpan.
Variabel Specify	List Of String	Nama variabel yang ditentukan sebagai Klausula Specify.
Variabel Kondisi	List Of String	Nama Variabel yang ditentukan sebagai Klausula Kondisi.
Variabel Prediktif	List Of String	Nama Variabel yang ditentukan sebagai klausula Prediktif

4.3.1.2 Perancangan Data Proses

Data yang digunakan selama proses adalah Referensi Utama (3.2.2) yang disimpan sebagai sebuah Data Agregasi yang terdiri dari beberapa tabel. Tabel tabel tersebut juga disimpan sebagai Data Agregasi. Dan data lainnya adalah data populasi yang terdiri dari aturan aturan dan atribut atribut populasi yang disimpan sebagai sebuah Data Agregasi. Sedangkan aturan sendiri juga disimpan sebagai Data Agregasi yang terdiri dari beberapa klausa dan sebuah klausa bisa terdiri dari beberapa Term. Untuk melihat hubungan antar data proses ini dapat dilihat pada gambar 4.2.



Gambar 4.2 Hubungan data data proses

- Data Agregasi InfoField

Dalam record ini disimpan informasi informasi mengenai tabel relasional yang sedang diproses. Berisi data data tentang field field dalam basis data. Data tersebut adalah kode field, nama field, tipe datanya dan nilai maksimum untuk field yang bersangkutan.

Tabel 4.2 Data Agregasi InfoField

Nama Field	Tipe Field	Keterangan
Kode	Integer	Kode dari field .
Nama	String	Nama Field yang dikodekan.
Tipe	Integer	Tipe field dalam (katagori atau numerik).
Maksimum	Integer	Jumlah macam nilai field.

- Data Agregasi KlausuVar

Tipe data ini digunakan untuk menyimpan variabel variabel yang telah ditentukan untuk masing masing klausa. Ketiga klausa (Specify, kondisi dan prediktif) disimpan dalam tabel ini.

Tabel 4.3 Data Agregasi KlausuVar

Nama Field	Tipe Field	Keterangan
Kode klausa	Integer	Kode pada klausa yang bersangkutan
Kode variabel	Integer	Kode variabel ybs (merujuk ke

		kode dalam Tabel InfoField)
Count	Integer	Jumlah Variabel yang ditentukan

- Data Agregasi Katagori

Digunakan untuk menyimpan kode nilai nilai dalam sebuah field untuk field yang bertipe katagori. Hal ini dimaksudkan agar dalam proses pembangkitan nilai, hasilnya tidak keluar dari domain yang ada. Disamping itu juga agar dalam penyimpanannya lebih hemat dan lebih mudah dalam operasi genetiknya.

Tabel 4.4 Data Agregasi Katagori

Nama Field	Tipe Field	Keterangan
Kode variabel	Integer	Kode dari field yang bersangkutan
Anggota	String	Daftar anggota Item dalam Field

- Data Agregasi Numerik

Digunakan untuk mengkodekan nilai nilai dalam sebuah field untuk field yang bertipe numerik. Hal ini dimaksudkan agar dalam proses pembangkitan nilai, hasilnya tidak keluar dari domain yang ada. Disamping itu juga agar dalam penyimpanannya lebih hemat dan lebih mudah dalam operasi genetiknya.

Tabel 4.5 Data Agregasi Numerik

Nama Field	Tipe Field	Keterangan
Kode variabel	Integer	Kode dari field yang bersangkutan
Anggota	Float	Daftar anggota Item dalam Field

- Data Agregasi Referensi Utama

Digunakan sebagai mekanisme *control handling* (Lihat 3.3.2). Untuk mencegah agar aturan aturan yang dibangkitkan tidak keluar dari solusi legal yang diijinkan. Data Agregasi ini terdiri dari beberapa tabel yang dijadikan acuan oleh algoritma pembangkitan. Tabel tabel tersebut antara lain tabel fields, katagori, Numeric dan tabel klausa.

Tabel 4.6 Data agregasi Referensi Utama

Nama Field	Tipe Field	Keterangan
Field	List Of InfoField	Menyimpan informasi field field dalam tabel yang diproses.
Catagorical	List Of Katagori	Menyimpan Item item dalam field field yang bertipe katagori.
Numerical	List Of Numerik	Menyimpan Item item dalam field field yang bertipe numerik.
Specify	List Of KlausaVar	Menyimpan field field yang ditentukan sebagai Specify.

Kondisi	List Of KlausaVar	Menyimpan field field yang ditentukan sebagai Kondisi.
Prediktif	List Of KlausaVar	Menyimpan field field yang ditentukan sebagai Prediktif.

Operasi operasi primitif yang dipunyai Data Agregasi ini antara lain :

Tabel 4.7 Operasi Operasi Primitif Referensi Utama

Nama Operasi	Parameter	Return Value	Keterangan
GetFieldCount	()	Integer	Mendapatkan jumlah field dalam basis data.
GetSpeCount	()	Integer	Mendapatkan jumlah field yang ditentukan sebagai Specify.
GetKonCount	()	Integer	Mendapatkan jumlah field yang ditentukan sebagai Kondisi.
GetPreCount	()	Integer	Mendapatkan jumlah field yang ditentukan sebagai Prediktif.

- Data Agregasi Term

Term merupakan bagian dari sebuah klausa yang merujuk ke sebuah field dalam basis data dan disimpan dalam sebuah record. Sebuah term terdiri dari variabel yang merujuk ke nama field, tanda dan nilainya.

Tabel 4.8 Data Agregasi Term

Nama Field	Tipe Field	Keterangan
Variabel	Integer	Kode variabel merujuk ke salah satu field dalam basis data
Tanda	Integer	Tanda variabel yang bersangkutan bergantung pada tipe variabelnya . (lihat 3.3.1)
Nilai	List Of Float	Nilai Term ybs.
Number	Integer	Jumlah Nilai

Operasi operasi primitif yang dipunyai Data Agregasi ini antara lain :

Tabel 4.9 Operasi Operasi Primitif Term

Nama Operasi	Parameter	Return Value	Keterangan
Extract	()	String	Melakukan Operasi Decoding pada Level Term.

- Data Agregasi Rule

Aturan disimpan sebagai sebuah record yang anggotanya adalah klausa klausa specify, kondisional dan prediktif. Masing masing klausa terdiri dari beberapa term. Disimpan juga nilai fitness, Accuracy dan coverage aturan tersebut.

Tabel 4.10 Data Agregasi Rule

Nama Field	Tipe Field	Keterangan
Specify	List Of Term	Term term specify
Kondisional	List Of Term	Term term kondisi
Prediktif	List Of Term	Term term Prediksi
Fitness	Float	Nilai fitness aturan
Accuracy	Float	Nilai accuracy aturan
Coverage	Float	Nilai coverage aturan
SpeCount	Integer	Jumlah Term Specify
KonCount	Integer	Jumlah Term Kondisional
PreCount	Integer	Jumlah Term Prediktif

Operasi operasi primitif yang dipunyai Data Agregasi ini antara lain :

Tabel 4.11 Operasi Operasi Primitif Rule

Nama Operasi	Parameter	Return Value	Keterangan
Extract	()	String	Melakukan Operasi Decoding.
Evaluate	()	Float	Melakukan evaluasi

			terhadap nilai fitness.
--	--	--	-------------------------

- Data Agregasi Populasi

Merupakan himpunan aturan aturan yang bertahan dalam suatu generasi tertentu dan aturan aturan yang dihasilkan oleh operasi mutasi dan operasi pindah silang. Data yang disimpan adalah aturan aturan dan atribut atribut populasi seperti `average_fitness`, `crossOverRate` dan `MutateRate`.

Tabel 4.12 Data Agregasi Populasi

Nama Field	Tipe Field	Keterangan
Genome	List Of Rule	Rule rule yang bertahan dalam generasi.
MutateResult	List Of Rule	Rule rule hasil proses Mutasi
CrossResult	List Of Rule	Rule rule hasil proses Pindah silang
Count	Integer	Jumlah genome
MutateCount	Integer	Jumlah MutateResult
CrossCount	Integer	Jumlah CrossResult
Average_fitnes	Float	Nilai rata rata fitness
CrossOverRate	Float	Nilai kemungkinan sebuah aturan dipindah silangkan
MutateRate	Float	Nilai kemungkinan sebuah aturan dimutasikan

Operasi operasi primitif yang dipunyai Data Agregasi ini antara lain :

Tabel 4.13 Operasi Operasi Primitif Populasi

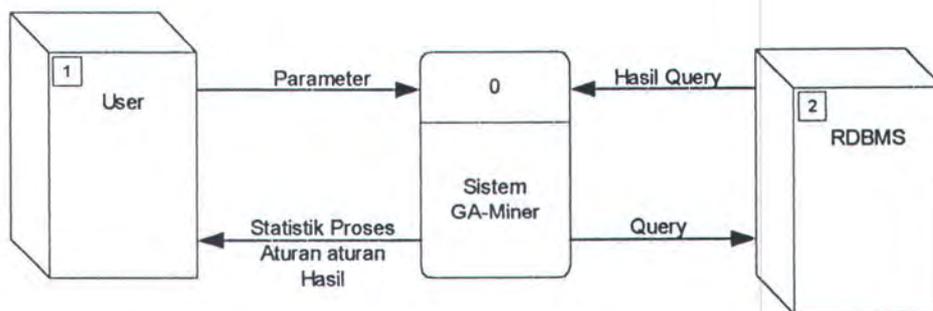
Nama Operasi	Parameter	Return Value	Keterangan
GetTheBest	()	String	Mendapatkan rule rule terbaik dalam populasi
Cross	()		Melakukan Pindah Silang
Mutate	()		Melakukan Operasi mutasi
Evaluate	()		Mengevaluasi semua rule dalam populasi.
Selection	()		Melakukan operasi seleksi

4.3.1.3 Perancangan Data Keluaran

Sebagai keluaran sistem ini adalah data data yang berupa aturan aturan hasil yang merupakan hasil dari proses data mining. Aturan aturan tersebut disertai juga dengan atribut atribut yang menunjukkan kualitas dari aturan yang bersangkutan. Atribut atribut tersebut diantaranya adalah : *accuracy*, *coverage*, *LeftSide* dan *RightSide*.

4.3.2 Perancangan Proses

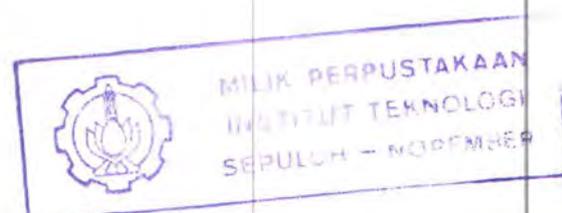
Untuk perancangan proses, digunakan sebagai diagram alir data dan hirarki proses sebagai alat untuk menyajikannya. Notasi DAD yang digunakan adalah notasi *Gane-sarson*.

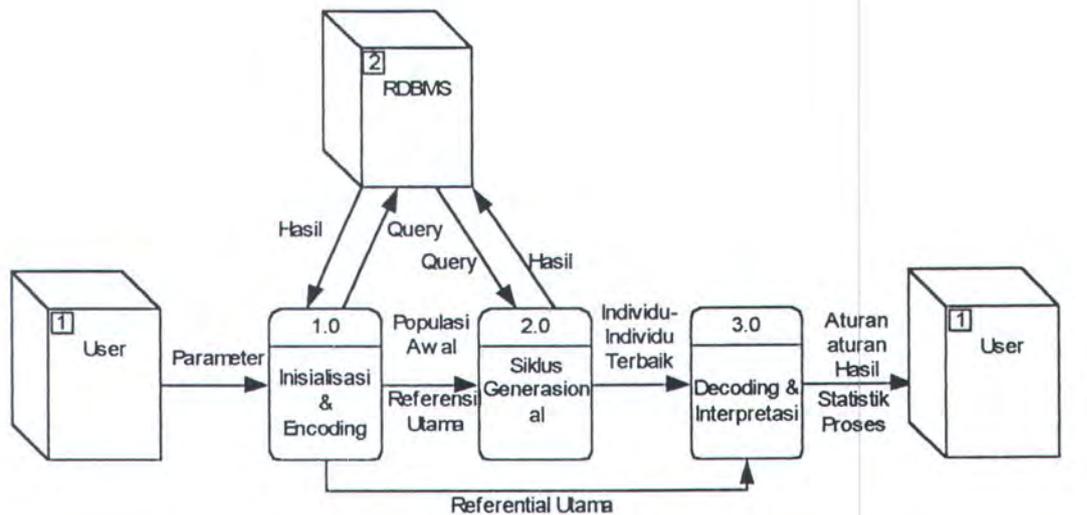


Gambar 4.3 DAD level 0, Sistem GA-Miner

DAD untuk proses utama dalam sistem GA-Miner ini dapat dilihat pada gambar 4.3. Proses dimulai dari pengguna dengan memasukkan parameter parameter genetik ke dalam sistem. Dengan parameter parameter yang diberikan pengguna sistem kemudian memulai pencarian aturan aturan dengan melakukan serangkaian query ke mesin basis data. Kemudian aturan aturan yang dihasilkan ditampilkan ke pengguna dalam bentuk yang mudah dimengerti.

Dalam DAD level 1 terdapat tiga proses utama (Gambar 4.4) yang merupakan penjabaran dari level 0, yaitu : Inisialisasi & Encoding, Siklus Generasional dan proses Decoding&Interpretasi. Proses inisialisasi melakukan validasi terhadap parameter, koneksi awal, pembuatan Referensi Utama dan pembangkitan populasi awal. Output dari proses ini adalah populasi awal dan Referensi Utama.

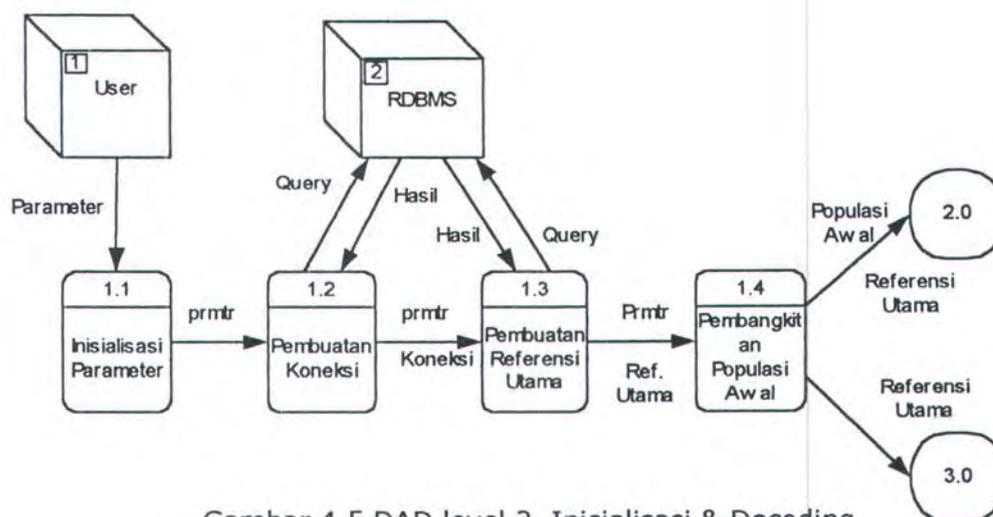




Gambar 4.4 DAD Level 1, GA-Miner

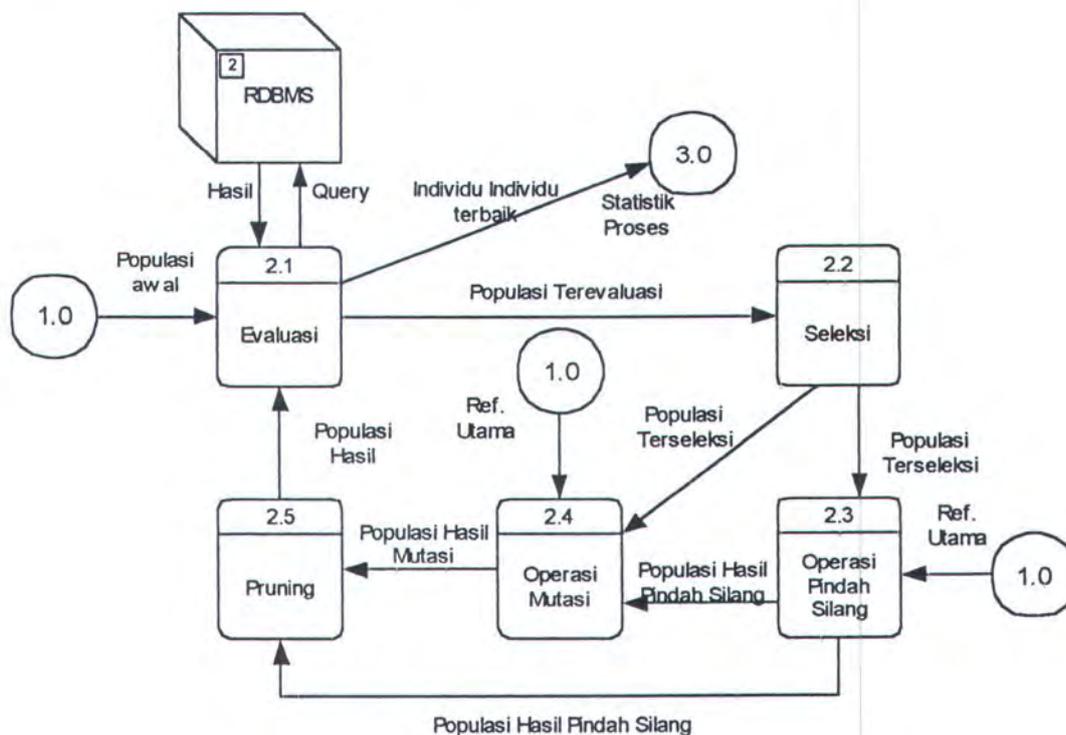
Dalam proses siklus generasional, populasi awal yang telah dibangkitkan akan diproses dengan serangkaian operasi genetika untuk mendapatkan populasi akhir yang kemudian di ambil individu individu terbaiknya sebagai output proses ini.

Aturan aturan terbaik yang didapatkan tadi kemudian menjadi input dari proses Decoding & Interpretasi. Dalam proses ini aturan aturan tersebut kemudian didecoding dan diinterpretasi untuk ditampilkan kepada pengguna.



Gambar 4.5 DAD level 2, Inisialisasi & Decoding

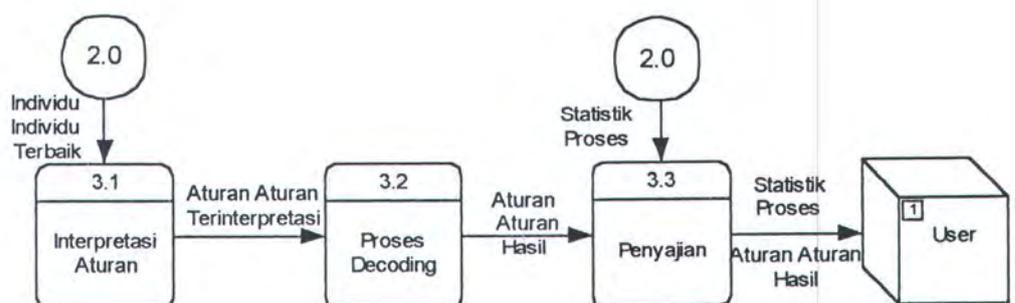
Penjelasan yang lebih rinci mengenai proses Inialisasi & Encoding dapat dilihat dalam gambar 4.5. Pengguna memasukan beberapa parameter kepada sistem yang kemudian divalidasi oleh sub proses 1.1 (Inialisasi Parameter). Output dari sub proses ini adalah parameter yang valid yang kemudian oleh sub proses 1.2 (Pembuatan Koneksi) akan digunakan untuk membuat Koneksi awal ke server RDBMS. Output dari sub proses 1.1 dan 1.2 ini oleh sub proses 1.3 (Pembuatan Referensi Utama) untuk membuat referensi utama. Output dari sub proses 1.1 dan 1.3 ini kemudian dijadikan parameter oleh sub proses 1.4 untuk membangkitkan populasi awal.



Gambar 4.6 DAD Level 2, Siklus generasional

Untuk gambaran yang lebih rinci mengenai proses Siklus generasional dapat dilihat pada gambar 4.6. Populasi awal yang dibangkitkan akan dievaluasi dengan fungsi evaluasi yang telah dipilih. Selanjutnya populasi yang telah

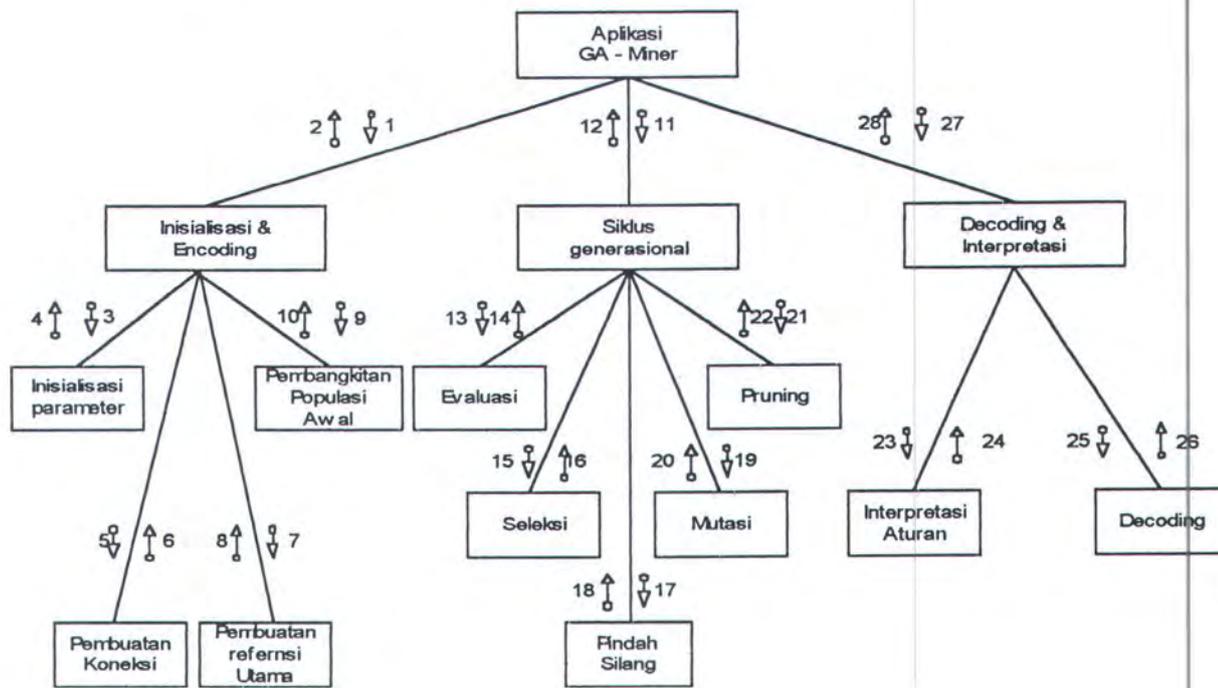
dievaluasi tadi diseleksi dengan mekanisme seleksi untuk menentukan aturan mana yang layak bertahan dan mempunyai kesempatan untuk mempunyai keturunan. Setelah itu populasi tersebut secara berturut turut dipindahsilangkan dan dimutasikan untuk mendapatkan offspring. Dan untuk selanjutnya offspring ini dievaluasi kembali. Kejadian ini akan berlangsung terus menerus sampai generasi ke-n, dimana n merupakan parameter yang diinputkan oleh pengguna.



Gambar 4.7 DAD Level 2, Decoding & interpretasi

Sedangkan gambaran yang rinci untuk proses Decoding & Interpretasi dapat dilihat pada gambar 4.7. Individu individu terbaik yang didapatkan dari proses Siklus generasional kemudian diinterpretasi dalam sub proses 3.1 (Interpretasi Aturan). Hasil sub proses ini selanjutnya didecoding dalam sub proses 3.2 (Proses Decoding). Output dari sub proses ini bersama sama dengan statistik proses yang dihasilkan oleh proses 2 (Siklus Generasional) ditampilkan ke pengguna.

Dari diagram alir data yang telah dijelaskan, selanjutnya dibuat hirarki proses seperti yang bisa dilihat dalam gambar 4.8.



Gambar 4.8 Hirarki Proses

Keterangan Gambar 4.8:

1. Parameter masukan.
2. Populasi awal.
3. Parameter.
4. Parameter tervalidasi.
5. Parameter koneksi.
6. Koneksi.
7. Koneksi.
8. Referensi Utama .
9. Referensi Utama .
10. Populasi awal.
11. Populasi awal.
12. Individu individu terbaik + Statistik Proses.

13. Populasi terevaluasi.
14. Populasi hasil.
15. Populasi hama + hasil.
16. Populasi terseleksi.
17. Populasi lama.
18. Populasi hasil pindah silang.
19. Populasi lama.
20. Populasi hasil mutasi.
21. Populasi hasil mutasi + populasi hasil pindah silang.
22. Populasi hasil.
23. Generasi terbaik.
24. Aturan aturan terinterpretasi.
25. Aturan aturan terinterpretasi.
26. Aturan aturan hasil.
27. individu individu terbaik.
28. aturan aturan hasil.

4.3.3 Perancangan Antar Muka

Perancangan antar muka dimaksudkan untuk memudahkan pengguna dalam menjalankan proses. Hirarki menu dapat dilihat secara lengkap pada gambar 4.9. Sedangkan penjelasan masing masing dialog yang dirancang adalah sebagai berikut :

- Menu Utama.

Merupakan sistem menu yang ditampilkan saat pertama kali program dipanggil. Dari sistem menu ini, nantinya dialog dialog yang lain diaktifkan.

- Dialog Koneksi Baru.

Dialog yang dirancang untuk melakukan koneksi ke suatu server basis data. Dalam dialog ini nantinya pengguna harus memasukan Driver yang digunakan, nama basis data (*connect string*), nama user dan passwordnya.

- Dialog Pemutusan Koneksi

Dialog yang dibuat untuk memutuskan koneksi yang sedang aktif.

- Dialog Informasi Koneksi.

Digunakan untuk melihat informasi koneksi yang sedang aktif.

- Dialog Pemilihan Tabel.

Digunakan untuk memilih tabel yang akan diproses oleh sistem.

- Dialog Pemilihan variabel specify, kondisi dan prediktif.

Sebelum proses dijalankan terlebih dulu harus ditentukan variabel variabel yang masuk dalam specify, kondisi dan prediktif.

- Dialog Penentuan *Direct Term*.

Jika pengguna bermaksud untuk melakukan *direct data mining*, maka terlebih dulu harus menentukan term term nya.

- Dialog Pelaksanaan Proses.

Merupakan dialog yang dirancang untuk memandu pengguna dalam melakukan proses mining.

- Dialog Penyajian Aturan hasil.

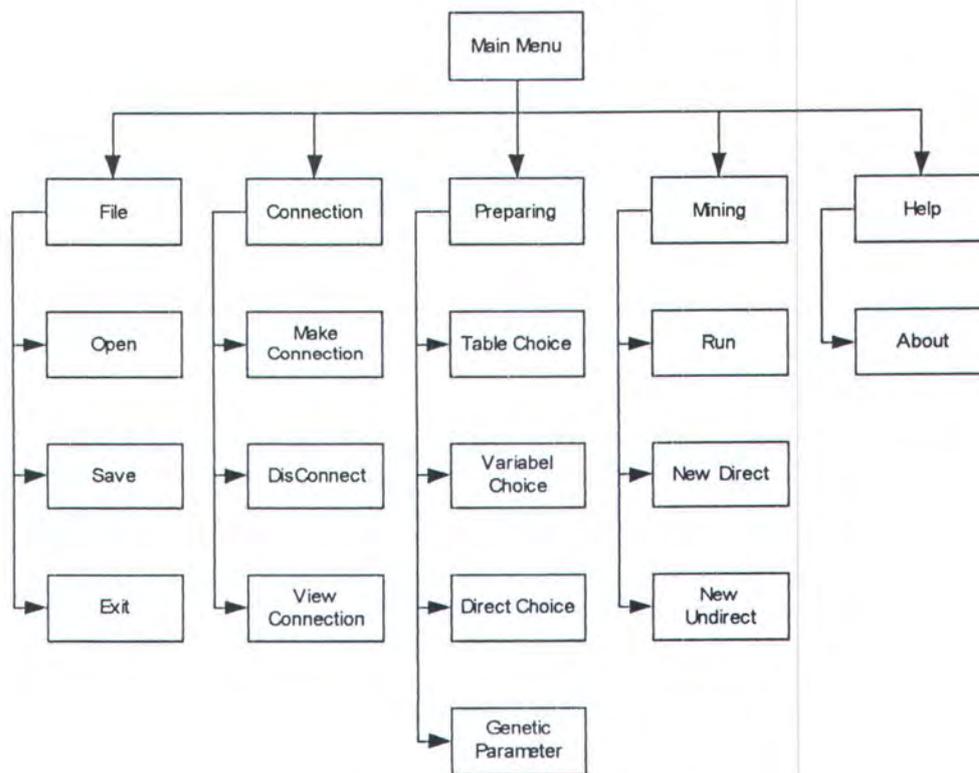
Merupakan dialog yang dirancang untuk menyajikan aturan aturan yang dihasilkan dari proses mining.

- Dialog Penyimpanan Aturan hasil.

Fasilitas yang diberikan kepada pengguna untuk menyimpan hasil proses kedalam berkas teks.

- Dialog Statistik Proses.

Dialog ini menyajikan data statistik dari proses. seperti waktu komputasi, nilai fitness tertinggi dan terendah serta kenaikan nilai rata rata fitness untuk tiap generasi.



Gambar 4.9 Hirarki Menu

4.3 Pembuatan Perangkat Lunak

Dalam sub bab ini dibahas pembuatan Perangkat Lunak yang merupakan penerapan dari perancangan yang telah dibuat dalam (4.2). pembahasan yang

dilakukan meliputi : Bahasa yang digunakan, Implementasi Struktur Data dan Implementasi proses.

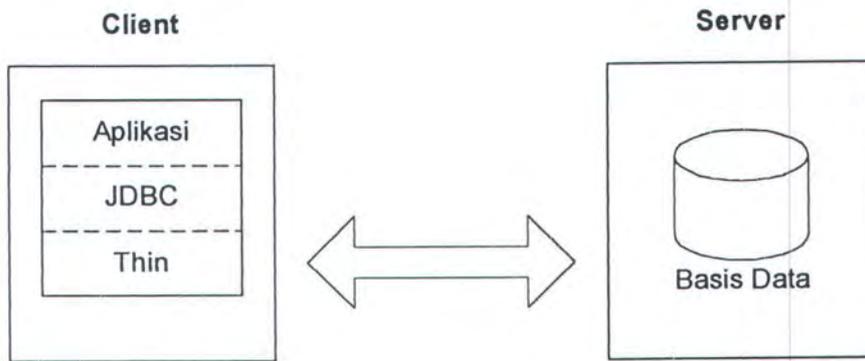
4.4.1 Lingkungan Implementasi

Sistem ini dibangun dengan menggunakan Oracle java Developer 2.0 dengan akses ke basis data melalui JDBC 2.0 dan data yang disimpan dalam server Oracle 8.01. keunggulan java adalah multi platformnya, sedangkan keunggulan JDBC adalah kemampuannya dalam mendukung *multitier connection*, dan tersedianya Driver yang disediakan oleh Oracle yakni : Oracle Thin Driver, Oracle OCI8 Driver dan Oracle OCI7 Driver.

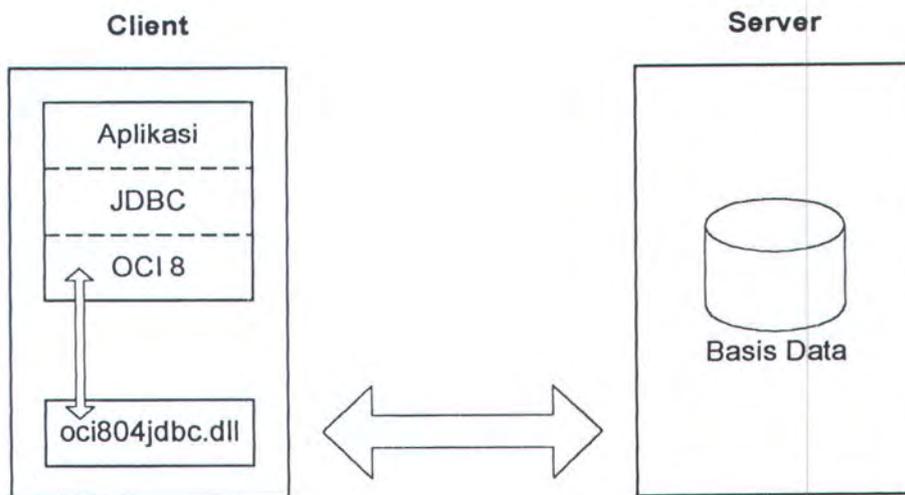
Secara umum Oracle Driver di bagi dalam dua bagian umum :

- Driver driver berbasis java (thin client drivers)
 - Diimplementasikan oleh semua produk java.
 - Memiliki portabilitas yang tinggi.
 - Dapat didownload dari server ke browser.
 - Dapat dikoneksikan dengan TCP/IP.
- Driver driver berbasis OCI
 - Diimplementasikan menggunakan library library method (OCI dlls).
 - Harus tersedia sebagai library di mesin client, dan tidak dapat di download ke browser.
 - Dapat di koneksikan dengan menggunakan banyak protokol SQL*net .
 - Unjuk kerja tinggi.

Ilustrasi, bagaimana sebuah aplikasi java mengakses suatu server basis data melalui JDBC dapat dilihat dalam gambar 4.10 dan 4.11



Gambar 4.10 Ilustrasi cara kerja Thin Driver



Gambar 4.11 Ilustrasi cara kerja OCI Driver

4.4.2 Implementasi Struktur Data

Bahasa yang digunakan untuk mengimplementasikan Sistem ini adalah Java yang merupakan bahasa yang murni OOP (*Object Oriented programming*). Karena itu, unit unit Data yang telah dirancang sebelumnya dalam (4.3.2) diimplementasikan sebagai kelas kelas. Dimana operasi operasi pada Data Agregasi diterapkan sebagai method method kelas.

4.4.2.1 Kelas RefUtama

Kelas ini merupakan implementasi dari Data Agregasi Referensi Utama (Tabel 4.6) dan (Tabel 4.7). Digunakan sebagai acuan oleh Algoritma pembangkitan untuk membangkitkan sebuah rule ataupun term. Operasi operasi yang dilakukan terhadap data data pada Data Agregasi ini merupakan operasi operasi untuk melakukan akses ke data yang bersangkutan. Hal ini terkait erat dengan pengkodean (3.3.1).

```
public class RefUtama {
// data member

private InfoField field[];
private Klausavar prediktif;
private Klausavar kondisi;
private Klausavar specify;
private Katagori catagori[] = new Katagori[60];
private Numerik numeric[] = new Numerik[60];

private byte count=0;
private byte countcat=0;
private byte countcont=0;
private Connection con;
private int totalRecord;
private String TableName;

//method method
public byte getCount()
public byte getCatagori_Count()
public byte getContinous_Count()
public byte getPre_count()
```

```

public byte getKon_count()
public byte getSpe_count()

public float getMax(byte var)
public float getMin(byte var)
public byte getVarKon(byte var)
public byte getVarPre(byte var)
public byte getVarSpe(byte var)
public String getName(byte var)
public String getCatagoricalName(byte var, float Katagori)
public float getContinous(byte var, float contin)
}

```

4.4.2.2 Kelas InfoField

Kelas ini merupakan implementasi dari Data Agregasi InfoField (Tabel 4.2). digunakan untuk menyimpan data data field dalam basis data. Semua data dalam kelas ini dibuat public supaya bisa diakses langsung dari luar kelas, karena operasi operasi yang mengakses data data dari kelas ini akan dilakukan langsung dari kelas RefUtama yang mengandung kelas ini.

```

public class InfoField {
//data member
    String name;
    float min;
    float max;
    byte type;
}

```

4.4.2.3 Kelas Katagori

Merupakan implementasi dari Data Agregasi katagori yang telah dirancang (Tabel 4.4). Digunakan untuk menyimpan nilai nilai sebuah field untuk field yang bertipe katagori. Seperti pada kelas InfoField, data data dalam kelas ini juga dibuat public supaya bisa diakses langsung dari luar kelas.

```

public class Katagori {
//data member
    String catagori[];
    int count;
    int var;
}

```

4.4.2.4 Kelas Numerik

Kelas ini merupakan implementasi dari Data Agregasi Numerik (Tabel 4.5). Digunakan untuk menyimpan nilai nilai sebuah field untuk field field yang bertipe numerik. Seperti pada kelas InfoField dan kelas Katagori, data data pada kelas ini juga dibuat public supaya bisa diakses langsung dari luar kelas.

```
public class Numerik {
    float continous[];
    int var;
    int count;
}
```

4.4.2.5 Kelas KlausuVar

Kelas ini merupakan implementasi dari Data agregasi KlausuVar (Tabel 4.3). Digunakan untuk menyimpan field field yang ditentukan untuk masing masing klausa. Seperti pada kelas FieldInfo, kelas Katagori dan kelas Numerik, data data dalam kelas ini juga dibuat public.

```
public class KlausuVar {
    int Kode[];
    int count;
}
```

4.4.2.6 Kelas Populasi

Merupakan implementasi dari Data Agregasi populasi (Tabel 4.12). Dalam kelas ini operasi operasi genetic seperti pembangkitan aturan aturan, operasi pindah silang, operasi mutasi, evaluasi dan seleksi dilakukan.

```
public class Populasi {
    // data member
    rule genome[];//rule tetap (terseleksi)
    rule result_cross[];//untuk menampung hasil crossover
    rule result_mutate[];//untuk menampung hasil mutasi
```

```

RefUtama mytab;
Random Myrand = new Random();
int rulecount;
int cross_count=0;
int mutata_count=0;
float average_fitness=0;
Term konTerms[];
Term preTerms[];
int kon_Tcount=0;
int pre_Tcount=0;
Connection con;

//method method
public Populasi(int NumRule,boolean direct, RefConstraint
mytabI, Term konTermsI[], Term preTermsI[], int kon_TcountI,
int pre_TcountI)

public void cross(double cross_rate,int num_rule)
public void mutata(double mutata_cross)
public void evaluate()
public void select()

private rule crossRule(rule rule1, rule rule2)
private Term CrossTerm(Term t1,Term t2,float p1,float p2)
private rule MutataRule(rule r1)
private Term MutataTerm(Term myterm,int clusaType, float
probNewMut, float probNewSign)
private rule GenRule(int direct)
private Term GenTerm(int clusaType)
private Term MutataIn(Term myterm)
private Term MutataRange(Term myterm,boolean percent)
private Term mutataSingle(Term myterm,boolean percent)

}

```

4.4.2.7 Kelas Rule

Kelas ini adalah implementasi dari Data Agregasi rule (Tabel 4.10). Data data yang disimpan dalam kelas ini adalah data data tentang rule yang bersangkutan seperti nilai fitness, CrossOverRate, MutataRate dsb. Sedangkan operasi operasi yang dilakukan adalah operasi operasi untuk mengeset data data tersebut.

```

public class rule {
//data member
Term specify[] = new Term[4];
Term conditional[] = new Term[11];

```

```

Term prediktif[]= new Term[6];
protected float crossRate;
protected float mutateRate;
protected float fitness;
protected float fitness_Chi;
protected byte Cc;
protected byte Cp;
protected byte Cs;
float Left;
float Right;
float accuracy;
float coverage;
boolean specified;

//method method
public rule()
public boolean isfull(int cla)
public boolean isempty(int cla)
public void addCon(Term temp)
public void delCon(int i)
public void addPre(Term temp)
public void delPre(int i)
public void addSpe(Term temp)
public void delSpe(int i)
public void Pruned()
public void Evaluate(RefConstraint mytab)
public String Extract(RefConstraint mytab)
public void setCross(float temp)
public void setMutate(float temp)
public void setFitness(float fit)
public byte getCc()
public byte getCp()
public byte getCs()
public float getCross()
public float getMutate()
public float getFitness()
}

```

4.4.2.8 Kelas Term

Kelas ini merupakan implementasi dari Data Agregasi Term (Tabel 4.8). Data data yang disimpan dalam kelas ini adalah data data tentang Term yang bersangkutan seperti nilai kode variabel, tanda nilai dsb. Sedangkan operasi operasi yang dilakukan adalah operasi operasi untuk mengeset data data tersebut.

```

public class Term {
// data member

byte var; //variabel
byte sign; //<, >, in dsb

```

```

float[] val = {0,0,0,0,0,0};
boolean mark;//tanda
byte Num; //jumlah value
int NumRecord;

//method method
public Term()
public void Count(RefConstraint mytab)
public String Extract (RefConstraint mytab)
}

```

4.4.3 Implementasi Proses

Proses proses yang dirancang dalam sistem ini (4.3.2) diimplementasikan sebagai method dari kelas kelas yang telah didefinisikan dalam (4.4.2). Tidak semua proses yang ada dalam sistem ini akan dijelaskan dalam bagian ini mengingat terbatasnya tempat yang tersedia, maka hanya proses proses utama saja yang akan di sampaikan disini. Proses proses tersebut antara lain : Pembangkitan Populasi, Pindah Silang, Mutasi, Evaluasi dan seleksi.

4.4.3.1 Pembangkitan Populasi Awal

Proses ini membangkitkan sekumpulan aturan secara acak sesuai parameter yang dimasukan oleh pengguna (lihat Gambar 4.4). Dalam implementasinya proses ini dibuat dalam beberapa method. Method method tersebut adalah : GenRule dan GenTerm. Kedua method tadi adalah anggota dari kelas populasi.

- GenRule

Seperti namanya, method ini menghasilkan sebuah aturan secara acak berdasarkan Referensi Utama .

```

public rule GenRule(int direct){
rule ret= new rule();
}

```

```

if (mytab.getSpe_count(>0) //artinya specified
{
byte Number =(byte) (GenInt(3));
if (Number==0) {Number=1;}
if (Number>3) {Number=1;}
Term tmp;
for (int i=1;i<=Number;i++)
{tmp = new Term();
tmp = GenTerm(0);
ret.addSpe(tmp);
}
ret.specified = true;
} else {ret.specified=false;}

if (mytab.getKon_count(>0)
{ byte Number =(byte) (GenInt(9)+1);
Term dirTerm = new Term();
if ((direct ==1)&&(kon_Tcount>0)){
int termKe = GenInt(kon_Tcount-1)+1;
dirTerm = konTerms[termKe];
dirTerm.mark=true;
ret.addCon(dirTerm);
Number--;
}
for (int i=1;i<=Number;i++)
{Term tmp = new Term();
tmp = GenTerm(1);
ret.addCon(tmp);
}}

byte Number =(byte) (GenInt(4)+1);
Term dirTerm = new Term();
if ((direct==1)&&(pre_Tcount>0)){
int termKe = GenInt(pre_Tcount-1)+1;
dirTerm = preTerms[termKe];
dirTerm.mark=true;
ret.addPre(dirTerm);
Number--;
}
for (int i=1;i<=Number;i++){
Term tmp = new Term();
tmp = GenTerm(2);
ret.addPre(tmp);
}
ret.Pruned();
return ret;
}

```

- GenTerm

Method ini menghasilkan sebuah Term secara acak berdasarkan Referensi

Utama .

```

public Term GenTerm(int clausaType){
    byte varmax;
    float min,max;
    Term temp = new Term();

    if (clausaType==0){ varmax = mytab.getSpe_count();}
    else if (clausaType==1){ varmax = mytab.getKon_count();}
    else { varmax = mytab.getPre_count();}
    byte var;
    var=(byte) (GenVar(varmax-1)+1);
    if (clausaType==0)
        { var = mytab.getVarSpe(var);} else if (clausaType==1)
        { var = mytab.getVarKon(var);}
        else { var = mytab.getVarPre(var);}
    temp.var=var;
    byte sign=0;
    byte tipe =0;
    max = mytab.getMax(var);
    min = mytab.getMin(var);
    tipe = mytab.getType(var);
    if (tipe==0) {sign=0;}
    else if (tipe==1) {sign = (byte) (GenInt(4)+1);}
    else {sign = (byte) (GenInt(4)+6);}
    if (sign==2) {sign=3;}
    if (sign==4) {sign=5;}
    temp.sign=sign;
    if (sign==0)//dalam catagorical {}
    {
        int NumVal = 2;
        int truNya=NumVal;
        boolean ok;
        for (int i=1;i<=NumVal;i++)
            {int val = GenInt( (int) (max-1) )+1;
            ok =true;
            for (int j=1;j<i;j++)
                { if (val==temp.val[j])
                {ok=false;truNya--;break;}
                }
            if (ok==true){temp.val[i]=val;}
            }
        temp.Num = (byte)truNya;
    }
    else if (sign==1)//percent dalam interval
    {int val1 = GenInt( (int) (max-3))+1;
    int val2 = val1+2;
    if (val2<val1)
        { int val3 = val2;
        val2=val1;
        val1=val3;
        }
    temp.val[1] = val1;
    temp.val[2] = val2;
    temp.Num=2;
    }
    else if ((sign>=2) & (sign <=5))
    {int val = GenInt( (int) (max-1))+1;

```



MILIK PERPUSTAKAAN
 INSTITUT TEKNOLOGI
 SEPULUH - NOPEMBER

```

temp.val[1] = val;
temp.Num = 1;
}
else if (sign ==6)//integer dalam interval
{ int val1 = GenInt( (int) (max-1))+1;
  int val2 = GenInt( (int) (max-1))+1;
  if (val2<val1)
    { int val3 = val2;
      val2 = val1;
      val1 = val3;
    }
  temp.val[1] = val1;
  temp.val[2] = val2;
  temp.Num = 2;
}
else //integer <=,>=,<,>
{ int val = GenInt( (int) (max-1))+1;
temp.val[1] = val;
temp.Num = 1;
}
return temp;
}

```

4.4.3.2 Pindah Silang

Proses ini melakukan operasi pindah silang untuk menghasilkan aturan aturan baru (lihat Gambar 4.5). Dalam implementasinya proses ini juga dibuat dalam beberapa method. Method method tersebut adalah : Cross, CrossRule dan CrossTerm. Ketiga method tersebut adalah anggota dari kelas populasi.

- Cross

Method ini melakukan operasi pindah silang dengan terlebih dulu memilih aturan aturan yang akan dioperasikan. Hasilnya adalah aturan aturan baru.

```

int local_cross=0;
int local[]=new int[num_rule+1];
for (int i=1;i<=rulecount;i++)
{double myDob = GenFloat(1);
if (myDob<=cross_rate)
  {local_cross++;
  local[local_cross]=i;
  }
}
if (local_cross % 2 != 0)
  {local_cross--;}

```

```

int Half = local_cross/2;
for (int i=1;i<=Half;i++)
    {rule rule_result =
    crossRule(genome[local[i]],genome[local[i+Half]]);
    cross_count++;
    result_cross[cross_count] = rule_result;
    rule rule_result2 =
    crossRule(genome[local[i]],genome[local[i+Half]]);
    cross_count++;
    result_cross[cross_count] = rule_result2;
    }
}

```

- CrossRule

Method ini melakukan operasi pindah silang untuk level aturan. Parameter masukannya adalah dua buah aturan dan keluarannya adalah sebuah aturan baru.

- CrossTerm

Method ini melakukan operasi pindah silang untuk level Term. Parameter masukannya adalah dua buah term dan keluarannya adalah sebuah Term baru.

4.4.3.3 Mutasi

Proses ini melakukan operasi mutasi untuk menghasilkan aturan aturan baru (lihat Gambar 4.5). Dalam implementasinya proses ini juga dibuat dalam beberapa method. Method method tersebut adalah : mutate, mutateRule, mutateTerm, mutateIn, mutateSingle dan mutateRange. Semua method tersebut adalah anggota dari kelas populasi.

- Mutate

Method ini melakukan operasi mutasi dengan terlebih dulu memilih aturan aturan yang akan dioperasikan. Hasilnya adalah aturan aturan baru.

```
public void mutate(double mutate_cross) {
    mutate_count=0;
    int local_mute=0;
    for (int i=1;i<rulecount;i++)
        {generate pobabilitas random
        double Mydob = GenFloat(1);
        if (Mydob<=mutate_cross)
            {rule rule_mutate = MutateRule(genome[i]);
            mutate_count++;
            result_mutate[mutate_count]=rule_mutate;
            }
        }
}
```

- MutateRule

Method ini melakukan operasi mutasi untuk level aturan. Parameter masukannya adalah sebuah aturan dan keluarannya adalah sebuah aturan baru.

- MutateTerm

Method ini melakukan operasi mutasi untuk level Term. Parameter masukannya adalah sebuah term dan keluarannya adalah sebuah Term baru.

- MutateIn

Method ini melakukan operasi mutasi terhadap term untuk term yang nilainya bertipe katagori. Parameter masukannya adalah sebuah term dan keluarannya adalah sebuah Term baru.

- MutateSingle

Method ini melakukan operasi mutasi terhadap term untuk term yang nilainya numerik tunggal. Parameter masukannya adalah sebuah term dan keluarannya adalah sebuah Term baru.

- MutateRange

Method ini melakukan operasi mutasi terhadap term untuk term yang nilainya numerik Range. Parameter masukannya adalah sebuah term dan keluarannya adalah sebuah Term baru.

4.4.3.4 Evaluasi

Proses ini digunakan untuk menentukan nilai fitness masing masing rule (lihat Gambar 4.5). Dalam implementasinya proses ini juga dibuat dalam sebuah method yang merupakan anggota dari kelas Populasi.

```
public void evaluate() {
    for (int i=1; i<=cross_count; i++)
        {result_cross[i].Evaluate(mytab, evalType); }
    for (int i=1; i<=mutate_count; i++)
        { result_mutate[i].Evaluate(mytab, evalType); }
}
```

4.4.3.5 Seleksi

Proses ini digunakan untuk menyeleksi aturan aturan terbaik yang layak bertahan dalam generasi selanjutnya dan mempunyai kemungkinan untuk mendapatkan keturunan. Dalam implementasinya proses ini juga dibuat dalam sebuah method yang merupakan anggota dari kelas populasi.

```
public void select() {
    boolean after_exchange=false;
    int cross=0,mutate=0,top;
    sort_worst();
    for (int i=1; i<=cross_count; i++)
        {top =0;
        after_exchange =false;
        for (int j=1; j<=rulecount; j++)
            {if (Is_similiar(result_cross[i], genome[j], (float)1)==true)
                {after_exchange=true; top++;
                if (result_cross[i].getFitness() > genome[j].getFitness())
                    {rule newRule = result_cross[i];
                    genome[j] = newRule;
                    sort_worst();
```

```

        break;
    }
}
}
if (after_exchange==false)
{for (int j=1;j<=rulecount;j++)
    {if (result_cross[i].getFitness() > genome[j].getFitness())
        {rule newRule = result_cross[i];
        genome[j] = newRule;
        sort_worst();
        break;
        }
    }
}
}
sort_worst();
after_exchange = false;
for (int i=1;i<=mutate_count;i++)
{top=0;
after_exchange =false;
for (int j=1;j<=rulecount;j++)
    {if (Is_similiar(result_mutate[i],genome[j],(float)1)==true)
        {after_exchange=true;
        top++;
        if (result_mutate[i].getFitness() >
        Genome[j].getFitness())
        {rule newRule = result_mutate[i];
        genome[j] = newRule;
        sort_worst();
        break;
        }
        }
    }
}
if (after_exchange==false)
{for (int j=1;j<=rulecount;j++)
    {if (result_mutate[i].getFitness() > genome[j].getFitness())
        {rule newRule = result_mutate[i];
        genome[j] = newRule;
        sort_worst();
        break;
        }
    }
}
}
}
sort_worst();
float total=0;
for (int i=1;i<=rulecount;i++)
    {total=total+genome[i].getFitness();}
average_fitness = total/rulecount;
}

```



BAB V

UJI COBA DAN EVALUASI PERANGKAT LUNAK

Dalam bab ini dibahas uji coba dan Evaluasi terhadap perangkat lunak yang telah dibuat. Karena dibuat dengan teknologi java maka perangkat lunak ini bersifat multi platform. Namun demikian untuk menjalankannya dibutuhkan Java Virtual Machine pada sistem operasi yang digunakan.

Untuk uji coba ini digunakan fasilitas di Lab Sistem Informasi , dimana Perangkat lunak dijalankan dari beberapa platform berbeda, mengakses sumber data yang disimpan dalam mesin server yang berbeda pula.

Meskipun tujuan utama pembuatan Perangkat Lunak ini adalah untuk Pencarian Pola Implikasi pada basis data (terutama basis data retail). Namun untuk menguji keakuratan dan keluwesan algoritma ini dalam banyak domain permasalahan, perangkat lunak ini akan diuji-cobakan pada beberapa basis data yang berbeda untuk pencarian aturan aturan klasifikasi. Kemudian untuk masing masing basis data tersebut akan dianalisa hasilnya satu persatu.

5.1 Data Untuk Uji Coba

Data yang digunakan dalam uji coba ini didapat dari beberapa sumber yang berbeda baik itu yang didapat dari internet maupun Data generator. Diantaranya adalah :

1. Basis data Market yang dibangkitkan dengan generator (Lampiran A). Digunakan untuk melihat pola-pola penjualan produk produk tertentu pada periode dan lokasi tertentu.
2. Basis data Dermatologi. Mengandung 34 atribut dan satu atribut kelas. 33 atribut bernilai katagorikal dan sebuah atribut bernilai kontinyu (numeric). Merupakan data yang dikumpulkan untuk mendiagnosa erythematous diseases (Lampiran B). Data didapat dari : <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/dermatology/dermatology.data>.
3. German Credit Data. Mengandung 20 atribut dan sebuah atribut kelas. 13 atribut bernilai katagorikal dan 7 buah atribut kontinyu (numeric). Merupakan data yang dikumpulkan untuk mengklasifikasikan usulan kredit kedalam katagori *good* atau *bad* (Lampiran C). Data didapat dari : <ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/german/german.data>.

5.2 Lingkungan Uji Coba

Uji coba terhadap sistem ini dilakukan dengan beberapa lingkungan perangkat keras dan perangkat lunak yang berbeda. Lingkungan lingkungan tersebut antara lain :

- Lingkungan I :

Perangkat Lunak dijalankan dari sisi klien dengan sistem operasi Windows NT Workstation 4.0, sedangkan data yang diolah disimpan dalam server Oracle 8 yang berjalan diatas sistem operasi HP-UX. Driver yang digunakan

adalah *Oracle Thin*. Adapun Spesifikasi Perangkat Kerasnya adalah sebagai berikut :

◆ Spesifikasi Klien

- Pentium II 450 Mhz
- RAM 64 Mb

◆ Spesifikasi Server

- Risc 150 Mhz
- RAM 512 Mb

• Lingkungan II:

Perangkat Lunak dijalankan dari mesin yang sama dengan tempat data di simpan. Mesin yang dimaksud berjalan diatas sistem operasi Windows NT Workstation 4.0, sedangkan data disimpan dalam format MS Access 2.0. Driver yang digunakan adalah *Sun.JDBC:ODBC Bridge*.

Spesifikasi Perangkat Kerasnya adalah :

- Pentium II 450 Mhz
- RAM 64 Mb

• Lingkungan III:

Perangkat Lunak dijalankan dari mesin yang sama dengan tempat data di simpan. Mesin yang dimaksud berjalan diatas sistem operasi HP-UX,

sedangkan data disimpan dalam format Oracle 8. Driver yang digunakan adalah *Oracle Thin*.

Spesifikasi Perangkat Kerasnya adalah :

- Risc 150 Mhz
- RAM 512 Mb

5.3 Evaluasi Terhadap Kinerja Perangkat Lunak

Dalam sub bab ini dibahas evaluasi terhadap kinerja sistem yang telah dibuat. Evaluasi dilakukan dengan memasukan beberapa parameter yang berbeda, hal ini dimaksudkan untuk melihat pengaruhnya terhadap kinerja Perangkat Lunak.

Uji coba terhadap kinerja perangkat lunak ini sendiri dilakukan dalam dua tahap. Tahap pertama dilakukan untuk mencari lingkungan yang terbaik dalam hal waktu komputasinya. Hal ini dilakukan dengan menggunakan tabel yang sama untuk ketiga lingkungan. Sedangkan untuk uji coba tahap kedua dilakukan pada Lingkungan dengan kinerja terbaik dengan menggunakan semua tabel yang dipersiapkan.

5.3.1 Parameter-Parameter Yang Diamati

Parameter-parameter yang diamati adalah parameter-parameter genetika seperti : jumlah populasi, jumlah generasi, probabilitas pindah silang, probabilitas mutasi dan fungsi evaluasi. Parameter lain yang dipertimbangkan adalah besar

data (jumlah field dan jumlah record) dan lingkungan uji coba. Sedangkan kinerja sistem dilihat dari waktu komputasi, nilai *average_fitness* awal dan nilai *average_fitness* akhir.

5.3.2 Uji Coba Pada Beberapa Lingkungan

Uji coba ini dilakukan pada beberapa lingkungan berbeda dengan menggunakan data yang sama. Data yang digunakan adalah Data Market yang terdiri dari 4 field dan 4800 record.

Tabel 5.1 Data uji coba pada beberapa lingkungan

Parameter Genetika				Waktu Komputasi		
Pop	Gen	Cross	Mutate	Ling I	Ling II	Ling III
10	10	0.5	0.5	15.16	30.534	21.224
10	10	0.75	0.75	22.16	40.414	31.024
10	10	1	1	32.12	56.866	44.968
20	20	0.5	0.5	60.6	120.754	84.84
20	20	0.75	0.75	96.76	189.272	135.46
20	20	1	1	125.03	254.966	175.04
40	20	0.5	0.5	124.629	240.345	174.48
40	20	0.75	0.75	186.639	530.073	261.29
40	20	1	1	250.36	533.937	325.46
40	40	0.5	0.5	247.546	473.681	321.80
40	40	0.75	0.75	388.789	788.553	505.42
40	40	1	1	522.822	1184.66	679.66
40	60	0.5	0.5	373.62	1087.92	485.70
40	60	0.75	0.75	555.518	1287.31	722.17
40	60	1	1	785.09	1487.72	1020.6
60	60	0.5	0.5	567.346	1314.31	737.54
60	60	0.75	0.75	862.149	2101.46	1120.7
60	60	1	1	1176.32	2728.44	1529.2
60	100	0.5	0.5	894.19	2214.89	1172.65
60	100	0.75	0.75	1381.285	3312.35	1718.97
60	100	1	1	1714.96	4319.46	2315.3

Keterangan Tabel 5.1:

- Gen : Jumlah Generasi.
- Pop : Jumlah Populasi.
- Cross : Probabilitas Pindah Silang.
- Mutate : Probabilitas Mutasi.
- Ling I : Waktu Komputasi pada Lingkungan I (dalam detik).
- Ling II : Waktu Komputasi pada Lingkungan I (dalam detik).
- Ling III : Waktu Komputasi pada Lingkungan I (dalam detik).

Dari tabel 5.1 terlihat bahwa kecepatan sistem (waktu komputasi) dipengaruhi oleh beberapa hal :

- Jumlah Populasi, semakin besar populasi semakin besar pula waktu komputasinya.
- Jumlah generasi, semakin banyak generasi semakin besar pula waktu komputasinya.
- Probabilitas CrossOver, semakin besar probabilitas pindah silang semakin besar pula waktu komputasinya.
- Probabilitas Mutasi, semakin besar probabilitas pindah silang semakin besar pula waktu komputasinya.
- Driver koneksi yang digunakan, driver Oracle Thin lebih bagus kinerjanya dibandingkan Driver JDBC:ODBC.
- Lingkungan yang mempunyai biaya komputasi terkecil adalah Lingkungan I. Hal ini disebabkan oleh dua hal. Yang pertama, Driver yang digunakan adalah Oracle Thin sehingga lebih cepat dari Lingkungan II yang menggunakan ODBC:JDBC meskipun diproses menggunakan mesin yang sama. Yang kedua, pemrosesan dilakukan pada Pentium II 450 MHz sehingga lebih cepat

dari lingkungan III yang diproses pada RISC 150 MHz meskipun menggunakan Driver yang sama.

5.3.3 Uji Coba Dengan Data Berbeda

Dari 5.3.2 didapatkan bahwa Lingkungan uji coba yang mempunyai Waktu Komputasi yang terbaik adalah Lingkungan I. Karena itu, dalam uji coba dengan data berbeda ini lingkungan yang digunakan adalah Lingkungan I. Hasil uji coba tersebut disajikan dalam tabel tabel berikut :

Tabel 5.2 Data Uji Coba pada Data Market
(jumlah field 4, jumlah Record 4800)

Parameter Genetika				Kinerja Sistem			
Pop	Gen	Cross	Mutate	Waktu	Fit awal	Fit akhir	Improve
10	10	0.5	0.5	15.16	-0.017	2.68	2.69
10	10	0.75	0.75	22.16	-0.017	14.0	14.017
10	10	1	1	32.12	-0.017	24.0	24.017
20	20	0.5	0.5	60.6	-0.083	24.26	24.35
20	20	0.75	0.75	96.76	-0.083	24.6	24.68
20	20	1	1	125.03	-0.083	32.825	32.9
40	20	0.5	0.5	124.629	-0.125	19.08	19.2
40	20	0.75	0.75	186.639	-0.125	24.2	24.33
40	20	1	1	250.36	-0.125	38.475	38.6
40	40	0.5	0.5	247.546	0.295	29.649	29.620
40	40	0.75	0.75	388.789	0.295	45.225	45.195
40	40	1	1	522.822	0.295	59.29	59.26
60	40	0.5	0.5	373.62	-0.149	36.88	37.03
60	40	0.75	0.75	555.518	-0.149	50.392	50.541
60	40	1	1	785.09	-0.149	62.875	63.0247
60	60	0.5	0.5	567.346	-0.149	59.267	59.416
60	60	0.75	0.75	862.149	-0.149	67.275	67.424
60	60	1	1	1176.32	-0.149	70.59	70.74



Tabel 5.3 Data Hasil Uji Coba dengan Data Dermatologi
(jumlah field 34, jumlah record 600)

Parameter Genetika				Kinerja Sistem			
Pop	Gen	Cross	Mutate	Waktu	Fit awal	Fit akhir	Improve
10	10	0.5	0.5	4.446	-0.842	10.69	11.53
10	10	0.75	0.75	5.608	-0.842	18.05	18.89
10	10	1	1	7.881	-0.842	19.74	20.58
20	20	0.5	0.5	15.262	0.094	26.811	26.716
20	20	0.75	0.75	21.892	0.094	25.387	25.29
20	20	1	1	28.401	0.094	28.458	28.36
40	20	0.5	0.5	27.921	0.22	21.978	21.76
40	20	0.75	0.75	40.428	0.22	23.999	23.777
40	20	1	1	53.648	0.22	26.896	26.67
40	40	0.5	0.5	53.147	0.225	29.17	28.94
40	40	0.75	0.75	78.062	0.225	31.122	30.89
40	40	1	1	103.99	0.225	29.94	29.71

Tabel 5.4 Data Hasil Uji Coba dengan Data *German Credit*
(jumlah field 20, jumlah record 1000)

Parameter Genetika				Kinerja Sistem			
Pop	Gen	Cross	Mutate	Waktu	Fit awal	Fit akhir	Improve
10	10	0.5	0.5	7.481	-0.448	6.31	6.763
10	10	0.75	0.75	10.225	-0.448	7.706	8.15
10	10	1	1	13.129	-0.448	9.663	10.11
20	20	0.5	0.5	23.203	-0.652	8.37	9.02
20	20	0.75	0.75	35.09	-0.652	9.52	10.17
20	20	1	1	46.447	-0.652	10.235	10.888
40	20	0.5	0.5	48.35	-0.313	7.548	7.77
40	20	0.75	0.75	70.532	-0.313	8.139	8.452
40	20	1	1	93.444	-0.313	8.29	8.6
40	40	0.5	0.5	90.062	-0.313	9.455	9.768
40	40	0.75	0.75	139.45	-0.313	10.16	10.47
40	40	1	1	183.35	-0.313	11.35	11.66

Keterangan Tabel 5.2, Tabel 5.3 dan Tabel 5.4 :

- Waktu : Waktu Komputasi.
- Fit_awal : Average Fitness Awal.
- Fit_akhir : Average Fitness Akhir.
- Improve : Peningkatan Average Fitness (Fit_awal – Fit_akhir).

Dari tabel 5.2, tabel 5.3 dan tabel 5.4 terlihat bahwa selain dipengaruhi oleh besar Populasi, jumlah generasi, probabilitas pindah silang dan probabilitas mutasi, Waktu komputasi juga dipengaruhi oleh besarnya data.

Sedangkan nilai Akhir *Average_fitness* dan *Improve* dipengaruhi oleh beberapa hal berikut :

- Jumlah Populasi, semakin besar populasi semakin besar pula peningkatan *Average_fitnessnya*, hal ini dimungkinkan karena semakin banyak materi yang disediakan oleh populasi sehingga ruang pencarian akan semakin luas.
- Jumlah generasi, semakin banyak generasi semakin besar pula peningkatan *Average_fitnessnya*, hal ini dikarenakan pencarian bisa lebih mendalam.
- Probabilitas *CrossOver*, semakin besar probabilitas pindah silang semakin besar pula peningkatan *Average_fitnessnya*, hal ini dikarenakan kemungkinan kromosom kromosom terbaik yang dihasilkan dari operasi pindah silang semakin besar.
- Probabilitas Mutasi, semakin besar probabilitas mutasi semakin besar pula peningkatan *average_fitnessnya*, hal ini dikarenakan kemungkinan kromosom kromosom terbaik untuk dicoba semakin besar.

5.4 Interpretasi Terhadap Aturan Yang Dihasilkan

Salah satu permasalahan dalam Tugas Akhir ini adalah Interpretasi terhadap aturan-aturan yang didapatkan dari proses pencarian.

Interpretasi tersebut dilakukan terhadap aturan berdasarkan atribut-atribut yang menunjukkan kualitas aturan tersebut.

5.4.1 Interpretasi Hasil pada Basis Data Market

Uji coba pada basis data ini dilakukan untuk pencarian aturan-aturan implikasi dengan Specify nama produk tertentu. Ada dua macam tabel untuk contoh basis data model ini yaitu : tabel Market1 dan tabel Market2. Perbedaan keduanya dapat dilihat pada Lampiran A. Beberapa contoh aturan yang didapat dalam proses mining terhadap basis data ini adalah :

Aturan 1:

```
UNTUK PRODUK = { Produk 14,Produk 13,Produk 3,Produk 6,Produk 9 } ,
JIKA BULAN = { Desember } MAKA PENJUALAN >= 31.0

Sisi Kiri :8.333334 % Dari Total Basis Data
Sisi Kanan :58.083332 % Dari Total Basis Data
Accuracy :100.0 % ( Expected Was 8.333334 % )
Coverage :14.347201 % ( Expected Was 58.083332 % )
```

Gambar 5.1 Aturan 1, contoh hasil dari Tabel Market2

Aturan 2:

```
UNTUK PRODUK = { Produk 20,Produk 9,Produk 19,Produk 4,Produk 15 } ,
JIKA BULAN = { Januari } MAKA PENJUALAN Dalam Interval 10.0 .. 15.0

Sisi Kiri :8.333334 % Dari Total Basis Data
Sisi Kanan :12.333334 % Dari Total Basis Data
Accuracy :100.0 % ( Expected Was 8.333334 % )
Coverage :67.567566 % ( Expected Was 12.333334 % )
```

Gambar 5.2 Aturan 2, contoh hasil dari Tabel Market2

Aturan 3:

```
UNTUK PRODUK = { Produk 10,Produk 11,Produk 17,Produk 5,Produk 16 },
JIKA BULAN = { Desember } MAKA PENJUALAN Dalam Interval 25.0 .. 28.0
```

```
Sisi Kiri :8.333334 % Dari Total Basis Data
```

```
Sisi Kanan :26.916668 % Dari Total Basis Data
```

```
Accuracy :88.0 % ( Expected Was 8.333334 % )
```

```
Coverage :27.244583 % ( Expected Was 26.916668 % )
```

Gambar 5.3 Aturan 3, contoh hasil dari Tabel Market2

Dari 2 buah contoh aturan yang dihasilkan dari proses mining terhadap tabel Market1 didapatkan beberapa informasi yang menarik tentang karakteristik dari tabel tersebut. Berdasarkan Aturan 1 (gambar 5.1), dapat disimpulkan bahwa untuk Produk 3, Produk 6, Produk 9, Produk 13 dan Produk 14 pada bulan Desember, maka penjualan akan lebih besar atau sama dengan 31. *Accuracy* aturan ini adalah 100 %, artinya : untuk kejadian S, 100 % kejadian C terjadi pada saat yang sama dengan kejadian P. sedangkan nilai *coverage* aturan ini adalah 14.35 %, artinya untuk kejadian S, hanya 14.35 % kejadian P yang disebabkan oleh C. Demikian pula dari aturan 2 dan aturan 3 bisa ditarik kesimpulan yang hampir sama.

Pada basis data dengan karakteristik seperti ini, jika yang diutamakan adalah prediksi penjualan untuk produk produk tertentu dalam periode dan lokasi tertentu pula, maka nilai *accuracy* lebih penting dari pada *coverage*. Karena yang lebih dibutuhkan adalah jawaban dari pertanyaan “apa yang terjadi jika kondisi adalah C” dari pada pertanyaan “apa yang menyebabkan terjadinya P”.

Aturan aturan di atas didapatkan dari sebuah proses pencarian dengan jumlah populasi 60 dan dijalankan dalam 40 generasi. Dengan probabilitas crossver dan mutasinya 1 dan fungsi evaluasi yang digunakan adalah *rule interest*. Sedangkan jumlah record dalam basis data ini adalah 4800 record.

5.4.2 Interpretasi Hasil pada Basis Data Dermatologi

Uji coba pada basis data ini dimaksudkan untuk pencarian aturan aturan klasifikasi untuk kepentingan diagnosa *erythemato-squamous diseases*. Sebelumnya perlu dijelaskan bahwa kesulitan utama dalam diagnosa *erythemato squamous diseases* ini adalah banyaknya gejala gejala yang sama pada stadium awal. Jadi sering terjadi, untuk sebuah kondisi C dapat menyebabkan beberapa kejadian P yang berlainan. Adapun beberapa contoh aturan yang didapat adalah :

Aturan 1:

```
JIKA kneeAndElbowInvolment = { 1,2 } DAN polygonalPapules =
{ 3,2,1 } MAKA Eryhemato_SquamousDiseases = { Lichen Planus }
```

```
Sisi Kiri Terdapat : 19 % Dari Total Basis Data
Sisi Kanan Terdapat : 20 % Dari Total Basis Data
Accuracy : 100 % ( Expected Was 19 % )
Coverage : 95 % ( Expected Was 20 % )
```

Gambar 5.4 Aturan 1, contoh hasil dari Tabel Dermatologi

Aturan 2:

JIKA follicularPapules = { 3,2 } MAKA Eryhemato_SquamousDiseases =
{ pityriasis rubra pilaris }

Sisi Kiri Terdapat : 6 % Dari Total Basis Data

Sisi Kanan Terdapat :5 % Dari Total Basis Data

Accuracy : 90 % (Expected Was 6 %)

Coverage : 81 % (Expected Was 5 %)

Gambar 5.5 Aturan 2, contoh hasil dari Tabel Dermatologi

Aturan 3:

JIKA elongationOfTheRete = { 2,0 } DAN melaninInco = { 0 }
MAKA Eryhemato_SquamousDiseases = { psoriasis }

Sisi Kiri Terdapat :81 % Dari Total Basis Data

Sisi Kanan Terdapat :31 % Dari Total Basis Data

Accuracy : 100 % (Expected Was 81 %)

Coverage : 38 % (Expected Was 38 %)

Gambar 5.6 Aturan 3, contoh hasil dari Tabel Dermatologi

Sama seperti yang telah dilakukan pada 5.4.1 dari contoh aturan aturan yang didapatkan dari proses pencarian, dapat disimpulkan beberapa informasi penting tentang *erythemato-squamous diseases*. Misalnya dari aturan 3 : bahwa jika *elongationOfTheRete* dalam stadium 0 atau 2 dan *melaninInco* dalam stadium 0, maka 100% kemungkinan disease berada dalam kategori *psiriosis*. Tetapi kondisi dimana disease dalam *psiriosis*, hanya disebabkan oleh 38 % kondisi dimana *elongationOfTheRete* dalam stadium 0 atau 2 dan *melaninInco* dalam stadium 0.

Aturan aturan diatas didapatkan dari sebuah proses pencarian dengan jumlah populasi 20 dan dijalankan dalam 40 generasi. Dengan probabilitas

crossver dan mutasinya 1. Sedangkan jumlah record dalam basis data ini adalah 690 record.

5.4.3 Interpretasi Hasil pada *German Credit Data*

Uji coba pada basis data ini dimaksudkan untuk pencarian aturan aturan klasifikasi untuk memprediksi apakah sebuah usulan kredit beresiko tinggi atau rendah. Beberapa contoh aturan yang didapatkan selama proses mining adalah sebagai berikut :

Aturan 1:

```
JIKA Credit_History = { A32,A33 } DAN Duration >= 60.0
DAN Personal_Status = { A94 } MAKA Credit_Risk = { Good }

Sisi Kiri :0.5 % Dari Total Basis Data
Sisi Kanan :69.9 % Dari Total Basis Data
Accuracy : 100.0 % ( Expected Was 0.5 % )
Coverage : 0.7153076 % ( Expected Was 69.9 % )
```

Gambar 5.7 Aturan 1, contoh hasil dari Tabel *German Credit Data*

Aturan 2:

```
JIKA Guarantors = { A102 } DAN existing_credits_at_this_bank <= 3.0
MAKA Credit_Risk = { Good }

Sisi Kiri Terdapat : 99 % Dari Total Basis Data
Sisi Kanan Terdapat : 69.9 % Dari Total Basis Data
Accuracy : 99 % ( Expected Was 99 % )
Coverage : 70 % ( Expected Was 69.9 % )
```

Gambar 5.8 Aturan 2, contoh hasil dari Tabel *German Credit Data*

Aturan 3:

```

JIKA Personal_Status = { A92 } DAN Property = { A121 }
DAN Installment_Rate <= 2.0 MAKA Credit_Risk = { Bad }

```

Sisi Kiri :0.1 % Dari Total Basis Data

Sisi Kanan :30.1 % Dari Total Basis Data

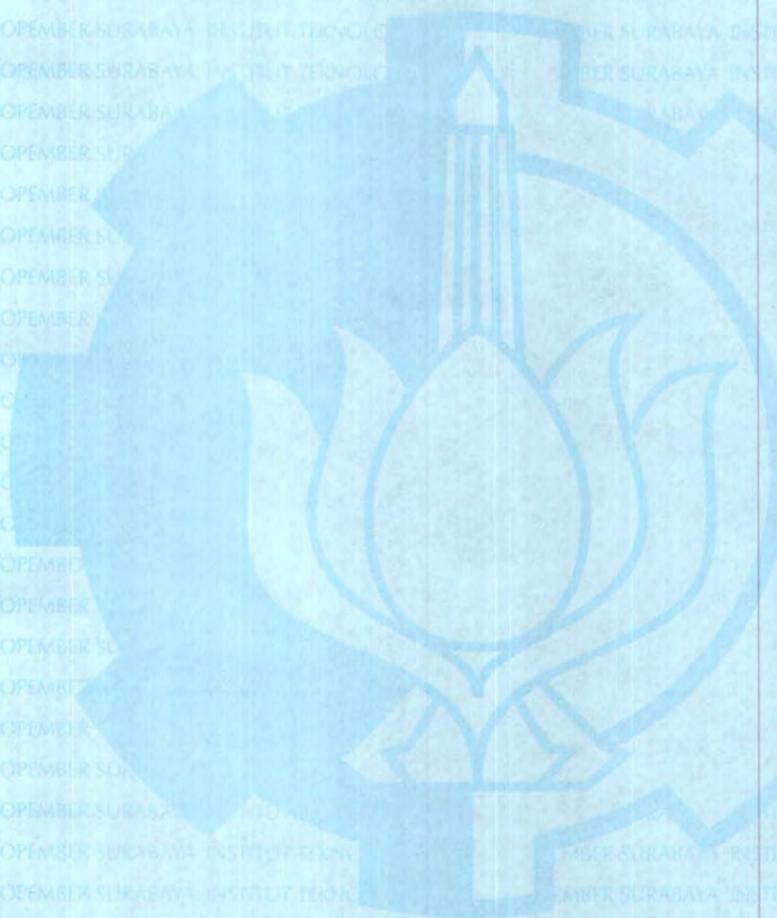
Accuracy : 100.0 % (Expected Was 0.1 %)

Coverage : 0.33222592 % (Expected Was 30.1 %)

Gambar 5.9 Aturan 3, contoh hasil dari Tabel *German Credit Data*

Dari aturan aturan yang dihasilkan dari proses mining terhadap *German Credit Data*, dapat diambil beberapa informasi yang berharga dari basis data tersebut. Sebagai contoh dari aturan 2 didapat : bahwa jika guarantors adalah *co applicant* (Lampiran C) dan jumlah kredit yang terdapat pada bank yang bersangkutan lebih kecil sama dengan 3 maka resiko kredit masuk dalam katagori baik (*good*). Kualitas aturan ini adalah *accuracy* = 99 %. Dan *coverage* = 70 %.

Aturan aturan di atas didapatkan dari sebuah proses pencarian dengan jumlah populasi 60 dan dijalankan dalam 20 generasi. Dengan probabilitas *crossver* dan mutasinya 1. Sedangkan jumlah record dalam basis data ini adalah 1000 record.



BAB VI

KESIMPULAN DAN SARAN

Pada bab ini diuraikan beberapa hal yang dapat disimpulkan dari penggunaan algoritma genetika pada data mining untuk pencarian pola-pola implikasi dalam suatu basis data. Kesimpulan tersebut didapatkan dari uji coba dan analisa terhadap perangkat lunak yang dibangun dalam Tugas Akhir ini. Disertai kemungkinan pengembangan Tugas Akhir ini dimasa datang.

6.1 Kesimpulan

Secara umum, dari hasil pengamatan selama uji coba, dapat diambil beberapa kesimpulan berikut :

1. Dengan pengkodean yang dilengkapi dengan penggunaan mekanisme pengendalian (*constraint handling*), penyusunan operator genetika yang optimal, mekanisme seleksi yang baik dan pemilihan fungsi evaluasi yang tepat, Algoritma Genetika telah berhasil digunakan pada Data Mining sebagai sebuah alat pencarian aturan-aturan Implikasi dalam suatu Basis Data. Hal ini ditunjukkan oleh hasil uji coba pada beberapa Basis Data dengan keberhasilannya mendapatkan aturan-aturan yang menarik dan berkualitas.
2. Secara umum Kinerja Perangkat Lunak dipengaruhi oleh dua hal utama. Yang pertama adalah Data yang diolah, meliputi : besar data yang diolah, tempat data disimpan (lokal atau remote) dan Driver Koneksi yang

digunakan. Yang kedua adalah Parameter genetika, meliputi : jumlah generasi, besar populasi, probabilitas pindah silang dan probabilitas mutasi. Hasil uji coba menunjukkan adanya kecenderungan - kecenderungan tersebut. Dengan semakin besar populasi, jumlah generasi, probabilitas pindah silang dan probabilitas mutasi akan semakin baik pula kinerjanya.

6.2 Kemungkinan Pengembangan

Beberapa hal yang dapat kami sarikan untuk kemungkinan pengembangan lebih lanjut adalah sebagai berikut :

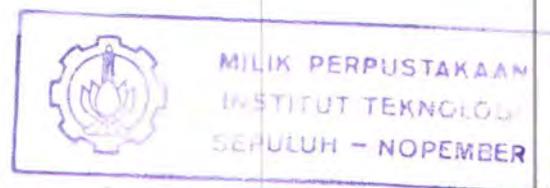
1. Untuk meningkatkan kinerja dan hasil yang lebih baik, kami menyarankan digunakannya *hybrid genetic algorithm* (algoritma genetika hibrida) dengan menggabungkan metode heuristik untuk pembentukan populasi awal. Karena dengan populasi awal yang baik dan menutup semua ruang lingkup pencarian, Operator Genetika akan lebih mudah menemukan kromosom kromosom yang terbaik. Sehingga solusi akhir yang dihasilkan akan lebih optimal.
2. Untuk meningkatkan kecepatan sistem dalam pencarian pada basis data yang sangat besar, perlu untuk dipikirkan implementasi algoritma genetika dengan menggunakan Algoritma Parallel.

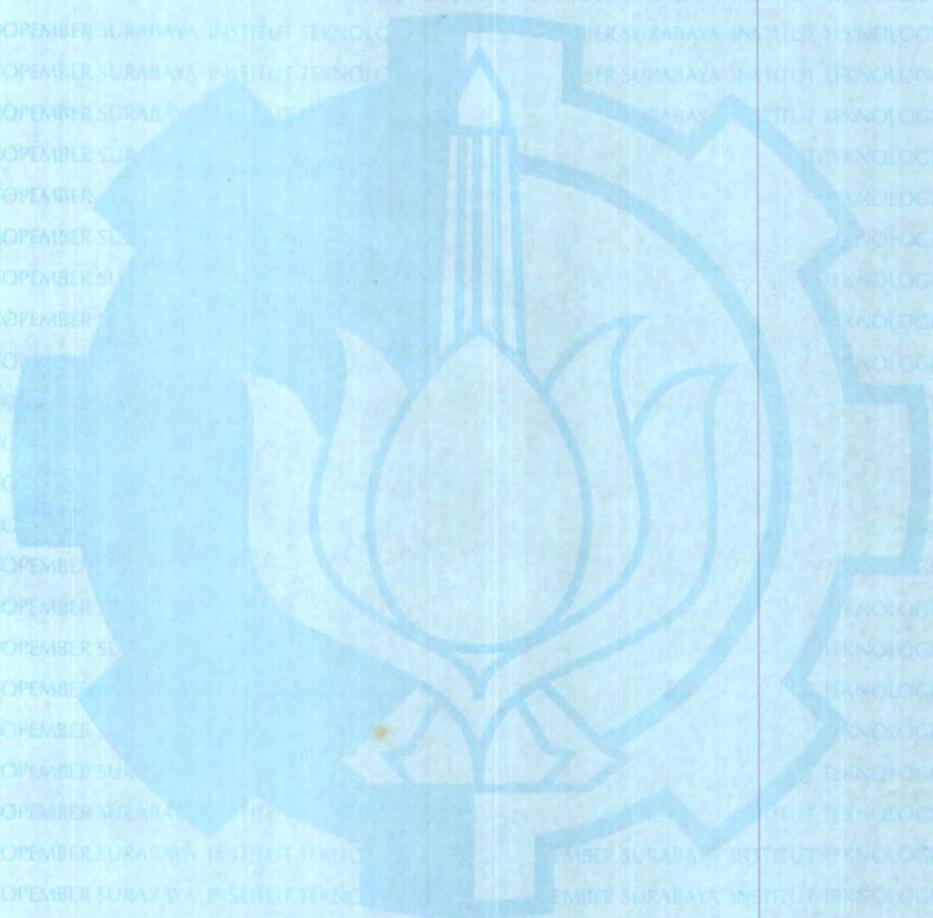


DAFTAR PUSTAKA

- [AUG-95] S. Augier, G. Venturini, and Y. Kodratoff (editors : Usama M. Fayyad and Ramasamy Uthurusamy), "Learning first order logic rules with a genetic algorithm", *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1995.
- [BER-89] Goldberg, David E, *Genetic Algorithm in Search, Optimization & Machine Learning*. Addison-Wesley, Reading, MA, New York, 1989.
- [CHA-96] Sophia Chan, "Genetic Algorithms, The Origin Of Genetic Algorithm", *White Paper*, 1996.
- [DEJ-93] Kenneth A. DeJong, William M Spears, and Diana F Gordon, "Using genetic algorithms for concept learning", *Journal Of Machine Learning (13:161—188)*, 1993.
- [FLO-96] Ian W. Flockhart and Nicholas J. Radcliffe, *A Genetic Algorithm Based Approach to Data Mining*, Department of Mathematics and Statistics, University Of Edhinburg, 1996.
- [FRA-91] William J. Frawley (editors : Gregory Piatetsky-Shapiro and William J. Frawley), "Using functions to encode domain and con-textual knowledge in statistical induction", *Journal Of Knowledge Discovery in Databases (261—275)*. MIT Press, 1991.

- [SKA-98] Jason Skaugc, "Data Mining", *White Paper*, Department of Computer Science, Winona State University, 1998.
- [WIC-96] Michelewicz, Zbignew, *Genetic Algorithms + Data Structure = Evaluation Programs*, Third Revised and Extended Edition, Springer – Verlag, Berlin, Heidelberg, New york, 1996.
- [ZYT-91] Jan M. Zytkow and John Baker (editors : Gregory Piatetsky-Shapiro and William J. Frawley), "Interactive mining of regularities in databases", *Journal Of Knowledge Discovery in Databases* (31–53). MIT Press, 1991.
- [ZYT-93] Jan M. Zytkow and Robert Zembowicz, "Database exploration in search of regularities", *Journal of Intelligent Information Systems* (2:39–81), 1993.





LAMPIRAN A

DESKRIPSI BASIS DATA MARKET

1. Nama : Basis Data Market

2. Sumber : Program Generator

3. Informasi :

Basis data ini dimaksudkan untuk mensimulasikan sebuah data warehouse yang multi dimensi. Dimensi yang dimaksud adalah : waktu, tempat dan produk. Ada 2 jenis tabel dalam basis data ini yang diberi nama Market1 dan Market2. Perbedaan kedua tabel ini terletak pada field penjualan. Untuk tabel Market1 field penjualannya bertipe katagori(qualitatif). Sedangkan pada tabel market2 field penjualannya bertipe numerik (kuantitatif).

4. Jumlah Record : 4800

5. Keterangan Atribut

Tabel Market1 :

- Produk (kuantitatif) item itemnya merupakan nama nama produk (dimensi produk).
- kota (kuantitatif) item itemnya merupakan nama nama kota (dimensi tempat).
- Bulan (kuantitatif) item itemnya merupakan nama nama bulan (dimensi waktu).
- Penjualan (kuantitatif) item itemnya merupakan keterangan penjualan untuk produk, bulan dan kota tertentu Yang sudah digolongkan kedalam katagori : Very Low, Low, Medium, Hight dan Very Hight.

Tabel Market2 :

- Produk (kuantitatif) item itemnya merupakan nama nama produk (dimensi produk).
- kota (kuantitatif) item itemnya merupakan nama nama kota (dimensi tempat).
- Bulan (kuantitatif) item itemnya merupakan nama nama bulan (dimensi waktu).
- Penjualan (kualitatif) item itemnya merupakan jumlah penjualan untuk produk, bulan dan kota tertentu yang masih berada dalam nilai kualitatif.

LAMPIRAN B

DESKRIPSI BASIS DATA DERMATOLOGI

1. Nama : Basis Data Dermatologi.

2. Sumber :

(a) Pemilik Asli:

1. Nilsel Ilter, M.D., Ph.D.,

Gazi University,
School of Medicine
06510 Ankara, Turkey
Phone: +90 (312) 214 1080

2. H. Altay Guvenir, PhD.,

Bilkent University,
Department of Computer Engineering and Information Science,
06533 Ankara, Turkey
Phone: +90 (312) 266 4133
Email: guvenir@cs.bilkent.edu.tr

(b) Donor: H. Altay Guvenir,

Bilkent University,
Department of Computer Engineering and Information Science,
06533 Ankara, Turkey
Phone: +90 (312) 266 4133
Email: guvenir@cs.bilkent.edu.tr

3. Informasi :

Basis data ini mengandung 34 atribut, 33 diantaranya bernilai katagori dan sebuah atribut numerik. Diagnosa erythemato-squamous diseases adalah masalah utama dalam dermatologi. semua desease dalam grup ini mempunyai feature klinik yang

sama dalam erythema dan scaling. Biasanya sebuah biopsy diperlukan untuk diagnosis tetapi malangnya, disease ini mempunyai banyak kesamaan feature hispatologi. Kesulitan lain dari diagnosa disease ini adalah karena banyaknya disease yang menunjukkan gejala yang sama pada fase awal. Feature family history mempunyai nilai 1 jika disease ini pernah terjadi dalam family, 0 jika sebaliknya. Feature umur secara sederhana merepresentasikan umur dari pasien. Semua feature yang lain (klinis dan hispatologi) telah diberikan derajad 0 sampai 3. Disini, 0 mengindikasikan bahwa feature tidak pernah ada. 3 menunjukkan penampakan terbesar. 1 dan 2 menunjukkan bahwa gejala nya sedang. Nama dan id pasien telah di buang dari basis data

4 jumlah record : 366

5. Informasi Atribut :

-- Dokumentasi Atribut:

Atribut Klinis: (bernilai 0, 1, 2, 3 , kecuali atribut 11 dan 34)

- 1: erythema
- 2: scaling
- 3: definite borders
- 4: itching
- 5: koebner phenomenon
- 6: polygonal papules
- 7: follicular papules
- 8: oral mucosal involvement
- 9: knee and elbow involvement
- 10: scalp involvement
- 11: family history, (0 atau 1)
- 34: Age (numerik)

Atribut atribut Histologi: (bernilai 0, 1, 2, 3)

12: melanin incontinence

13: eosinophils in the infiltrate

14: PNL infiltrate

15: fibrosis of the papillary dermis

16: exocytosis

17: acanthosis

18: hyperkeratosis

19: parakeratosis

20: clubbing of the rete ridges

21: elongation of the rete ridges

22: thinning of the suprapapillary epidermis

23: spongiform pustule

24: munro microabcess

25: focal hypergranulosis

26: disappearance of the granular layer

27: vacuolisation and damage of basal layer

28: spongiosis

29: saw-tooth appearance of retes

30: follicular horn plug

31: perifollicular parakeratosis

32: inflammatory mononuclear infiltrate

33: band-like infiltrate

LAMPIRAN C

DESKRIPSI BASIS DATA *GERMAN CREDIT*

1. Nama : German Credit data

2. Informasi sumber

Professor Dr. Hans Hofmann
Institut f"ur Statistik und "Okonometrie
Universit"at Hamburg
FB Wirtschaftswissenschaften
Von-Melle-Park 5
2000 Hamburg 13

3 jumlah record : 1000

4. Jumlah atribut : 20 (7 numerik, 13 katagori)

5. Deskripsi atribut :

Attribut 1: (kualitatif) Status of existing checking account

A11 : ... < 0 DM
A12 : 0 <= ... < 200 DM
A13 : ... >= 200 DM /
salary assignments for at least 1 year
A14 : no checking account

Attribut 2: (Numerik) Duration in month

Attribute 3: (Kualitatif) Credit history

A30 : no credits taken/all credits paid back duly
A31 : all credits at this bank paid back duly
A32 : existing credits paid back duly till now
A33 : delay in paying off in the past
A34 : critical account/other credits existing (not at this bank)

Attribute 4: (Kualitatif) Purpose

A40 : car (new)
A41 : car (used)
A42 : furniture/equipment
A43 : radio/television
A44 : domestic appliances
A45 : repairs
A46 : education
A47 : (vacation - does not exist?)
A48 : retraining
A49 : business
A410 : others

Attribute 5: (Numerik) Credit amount

Attribute 6: (Kualitatif) Savings account/bonds

A61 : ... < 100 DM
A62 : 100 <= ... < 500 DM
A63 : 500 <= ... < 1000 DM
A64 : .. >= 1000 DM
A65 : unknown/ no savings account

Attribute 7: (Kualitatif) Present employment since

A71 : unemployed
A72 : ... < 1 year
A73 : 1 <= ... < 4 years
A74 : 4 <= ... < 7 years
A75 : .. >= 7 years

Attribute 8: (Numerik) Installment rate in percentage of disposable income

Attribute 9: (Kualitatif) Personal status and sex

A91 : male : divorced/separated
A92 : female : divorced/separated/married
A93 : male : single
A94 : male : married/widowed
A95 : female : single

Attribute 10: (Kualitatif) Other debtors / guarantors

- A101 : none
- A102 : co-applicant
- A103 : guarantor

Attribute 11: (Numerik) Present residence since

Attribute 12: (Kualitatif) Property

- A121 : real estate
- A122 : if not A121 : building society savings agreement/
life insurance
- A123 : if not A121/A122 : car or other, not in attribute 6
- A124 : unknown / no property

Attribute 13: (Numerik) Age in years

Attribute 14: (Kualitatif) Other installment plans

- A141 : bank
- A142 : stores
- A143 : none

Attribute 15: (Kualitatif) Housing

- A151 : rent
- A152 : own
- A153 : for free

Attribute 16: (Numerik) Number of existing credits at this bank

Attribute 17: (Kualitatif) Job

- A171 : unemployed/ unskilled - non-resident
- A172 : unskilled - resident
- A173 : skilled employee / official
- A174 : management/ self-employed/
highly qualified employee/ officer

Attribute 18: (Numerik) Number of people being liable to provide maintenance
for

Attribute 19: (Kualitatif) Telephone

A191 : none

A192 : yes, registered under the customers name

Attribute 20: (Kualitatif) foreign worker

A201 : yes

A202 : no

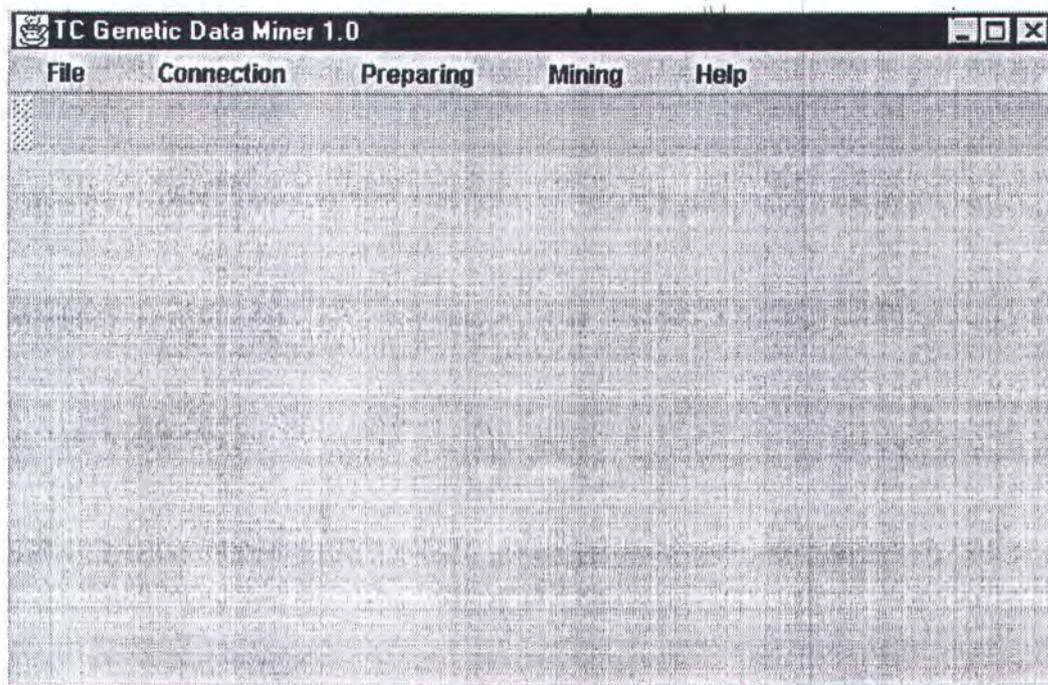
LAMPIRAN D

PETUNJUK PENGGUNAAN PERANGKAT LUNAK

Dalam lampiran ini dijelaskan petunjuk penggunaan perangkat lunak yang telah dibuat. Penjelasan yang diberikan dimulai dari tampilan utama perangkat lunak yang kemudian dilanjutkan dengan petunjuk penggunaan fasilitas fasilitas yang ada dalam sistem menu.

Tampilan Utama

Tampilan utama Perangkat Lunak ini dapat dilihat dalam gambar berikut ini :



Gambar D.1 Tampilan Utama

Fasilitas Menu

Beberapa menu yang disediakan dalam perangkat lunak ini antara lain :

- File

Dalam Menu ini terdapat dua buah item menu, yaitu :

- Item Open, digunakan untuk melihat hasil proses mining yang disimpan sebagai file Text.
- Item Exit, digunakan untuk keluar dari Perangkat Lunak sekaligus memutuskan semua koneksi ada.

- Connection

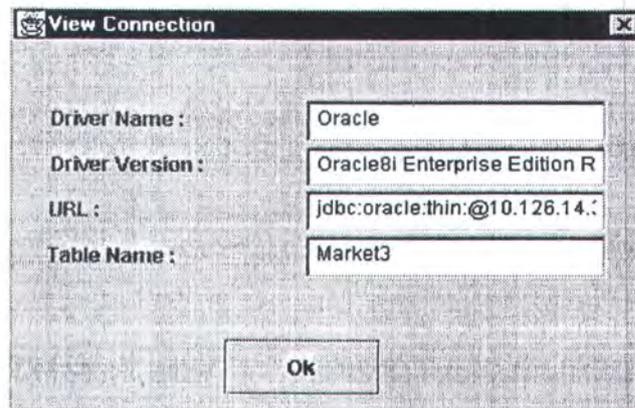
Menu ini digunakan untuk melakukan koneksi ke RDBMS, melihat koneksi dan memutuskan koneksi. Terdiri dari 3 buah item menu:

- Make Connection, digunakan untuk menampilkan dialog koneksi yang berfungsi untuk membuat koneksi ke RDBMS.



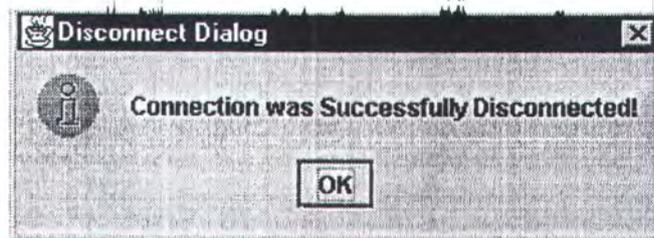
Gambar D.2 Dialog Koneksi

- View Connection, digunakan untuk menampilkan dialog *View Coonection* (melihat koneksi yang sedang aktif).



Gambar D.3 Dialog Lihat Koneksi

- Disconnect, digunakan untuk memutuskan koneksi yang sedang aktif.

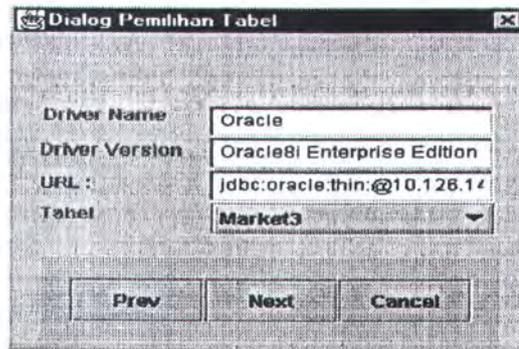


Gambar D.4 Dialog Pemutusan Koneksi

- Preparing

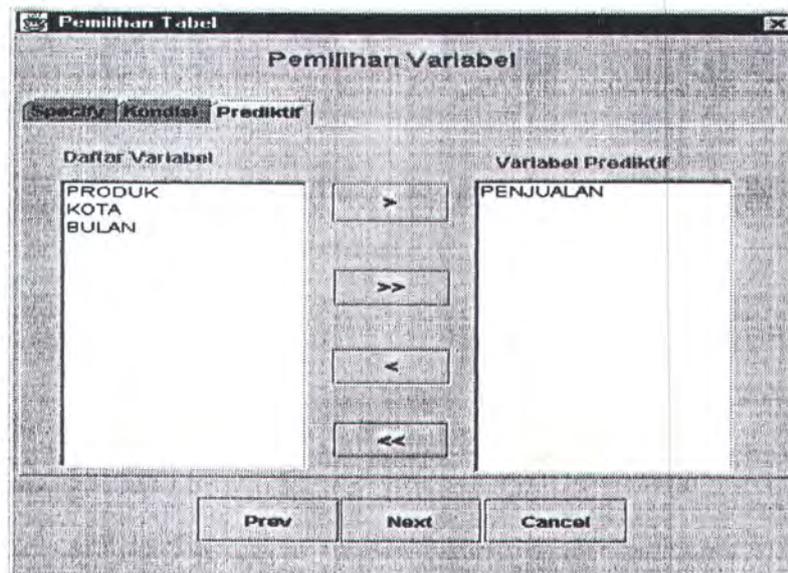
Digunakan untuk mempersiapkan parameter parameter yang akan digunakan dalam proses Mining. Terdiri dari 4 buah item menu :

- Table Choice, digunakan untuk menampilkan dialog pemilihan tabel yang akan diproses.



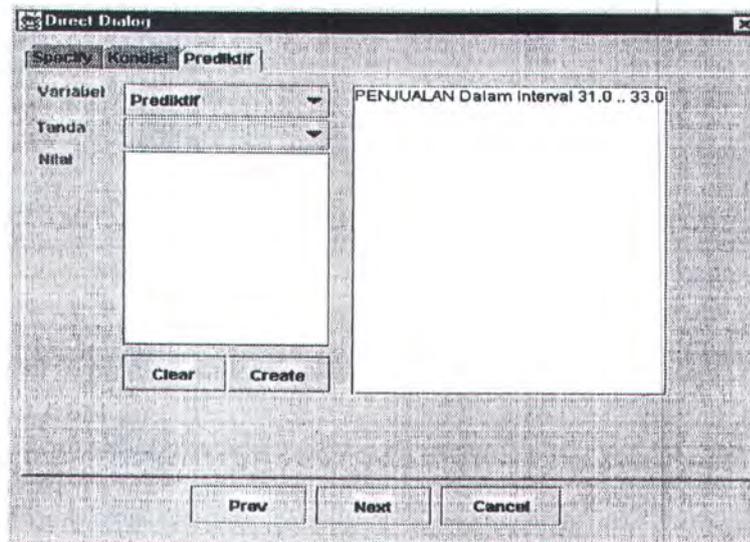
Gambar D.5 Dialog Pemilihan Tabel

- Variable Choice, digunakan untuk memilih variabel variabel yang ada di tabel yang ditentukan sebagai kalusa Specify, Kondisional ataupun Prediktif.



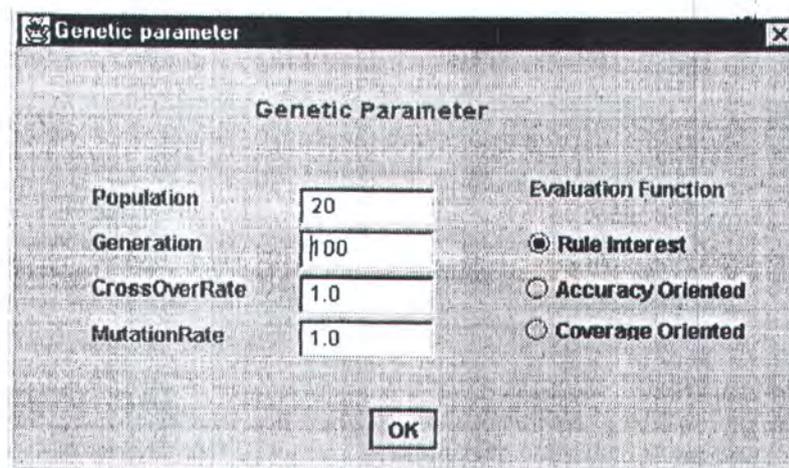
Gambar D.6 Dialog Pemilihan Variabel

- Direct, digunakan untuk menentukan Term Term direct.

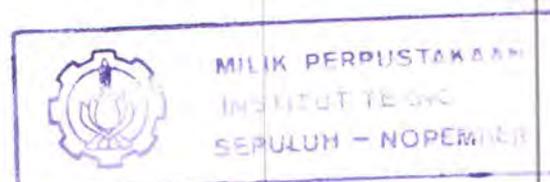


Gambar D.7 Dialog Directed

- Genetic Parameter, digunakan untuk menentukan parameter parameter genetika.

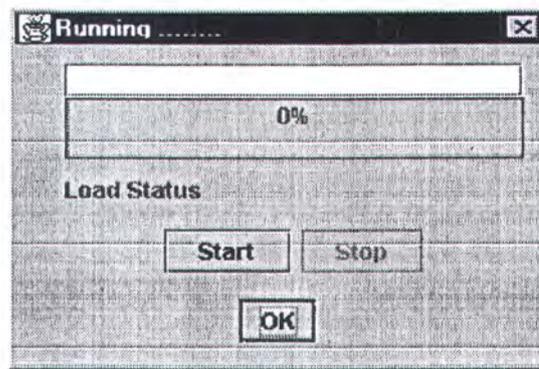


Gambar D.8 Dialog Parameter Genetika



- Mining

Digunakan untuk menjalankan beberapa macam proses yang berbeda.



Gambar D.9 Dialog Proses Mining

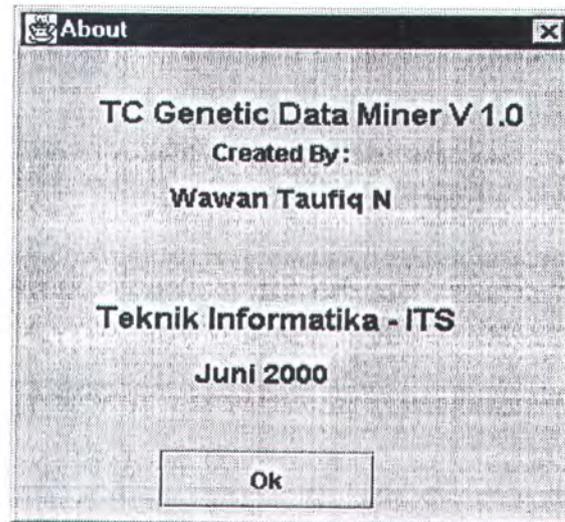
Dalam Menu ini terdapat beberapa Item Menu, diantaranya adalah:

- Run, digunakan untuk menjalankan proses Mining berdasarkan parameter parameter yang sudah diberikan.
- New Undirected, digunakan untuk menjalankan proses Mining secara Undirect dengan membuat koneksi baru dan semua parameter ditentukan ulang.
- New Directed, digunakan untuk menjalankan proses Mining secara direct dengan membuat koneksi baru dan semua parameter ditentukan ulang.
- New Hypothesis, digunakan untuk menjalankan proses Hypothesis dengan membuat koneksi baru dan semua parameter ditentukan ulang.

- Help

Dalam menu ini hanya terdaat satu item menu, yaitu :

- About, untuk melihat versi Perangkat Lunak dan sekilas keterangan tentang Perangkat Lunak yang dibuat.



Gambar D.10 Dialog About

LAMPIRAN E

CONTOH KELUARAN

E.1 Basis Data Market

▪ Script Parameter :

Data Source

Nama Tabel : Market3;
Tipe Mining : Directed;

Term Direct

Bulan = {Desember};

Parameter

Generasi : 40;
Populasi : 10;
CrossOverRate : 1;
MutateRate : 1;
Evaluation : 1;

▪ Aturan yang dihasilkan :

UNTUK PRODUK = { Produk 10,Produk 11,Produk 17,Produk 5,Produk 16 } ,

JIKA BULAN = { Desember } MAKA PENJUALAN Dalam Interval 25.0 .. 28.0

Sisi Kiri :8.333334 % Dari Total Basis Data

Sisi Kanan :26.916668 % Dari Total Basis Data

Accuracy :88.0 % (Expected Was 8.333334 %)

Coverage :27.244583 % (Expected Was 26.916668 %)

UNTUK PRODUK = { Produk 10,Produk 11,Produk 16,Produk 18,Produk 9 } ,

JIKA BULAN = { Desember } MAKA PENJUALAN Dalam Interval 26.0 .. 29.0

Sisi Kiri :8.333334 % Dari Total Basis Data

Sisi Kanan :29.249998 % Dari Total Basis Data

Accuracy :90.0 % (Expected Was 8.333334 %)

Coverage :25.641027 % (Expected Was 29.249998 %)

Gambar E.1 Contoh Keluaran pada Basis Data Market

E.2 Basis Data Dermatologi

- Script Parameter :

Data Source

Nama Tabel :Dermatologi;
Tipe Mining : Directed;

Term Direct

Eryhemato_SquamousDiseases = { cronic dermatitis }

Parameter

Generasi : 40;
Populasi : 10;
CrossOverRate : 1;
MutateRate : 1;
Evaluation : 1;

- Aturan yang dihasilkan :

JIKA familyHistory = { 0 } DAN fibrosis = { 3,2 }
MAKA Eryhemato_SquamousDiseases = { cronic dermatitis }
Sisi Kiri :12.568305 % Dari Total Basis Data
Sisi Kanan :14.207651 % Dari Total Basis Data
Accuracy :95.652176 % (Expected Was 12.568305 %)
Coverage :84.61539 % (Expected Was 14.207651 %)

JIKA koebnerPhenomenon = { 0 } DAN disappearOfTheGranular = { 0 }
MAKA Eryhemato_SquamousDiseases = { cronic dermatitis }
Sisi Kiri :47.267757 % Dari Total Basis Data
Sisi Kanan :14.207651 % Dari Total Basis Data
Accuracy :30.057802 % (Expected Was 47.267757 %)
Coverage :100.0 % (Expected Was 14.207651 %)

Gambar E.2 Contoh Keluaran pada Basis Data Dermatologi

E.3 Basis Data *German Credit*

▪ Script Parameter :

Data Source

Nama Tabel : German Credit;
Tipe Mining : Directed;

Term Direct

Credit_Risk = { Good }

Parameter

Generasi : 40;
Populasi : 10;
CrossOverRate : 1;
MutateRate : 1;
Evaluation : 1;

▪ Rule yang dihasilkan :

JIKA Credit_History = { A33,A31,A32 } MAKA Credit_Risk = { Good }

Sisi Kiri : 66.7 % Dari Total Basis Data

Sisi Kanan : 69.9 % Dari Total Basis Data

Accuracy : 71.66417 % (Expected Was 66.7 %)

Coverage : 68.38341 % (Expected Was 69.9 %)

JIKA Purpose = { A43 } MAKA Credit_Risk = { Good }

Sisi Kiri : 28.0 % Dari Total Basis Data

Sisi Kanan : 69.9 % Dari Total Basis Data

Accuracy : 76.78571 % (Expected Was 28.0 %)

Coverage : 30.758226 % (Expected Was 69.9 %)

Gambar E.3 Contoh Keluaran Pada Basis Data *German Credit*