

19.688/H/04

MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

PERANCANGAN DAN PEMBUATAN PERANGKAT
LUNAK PENGEKSTRAKSI KAIDAH ASOSIASI DAN
NAIVE BAYES BERBASIS BAHASA PREDICTIVE
MODEL MARKUP LANGUAGE DENGAN
MENGGUNAKAN KOMPONEN
ORACLE9i DATA MINING

TUGAS AKHIR

RSIF
005.1

Kri
P-1
2004



Oleh :

ADHITYA KRISNA

NRP. 5198 100 058

PERPUSTAKAAN ITS	
Tgl. Terima	24-2-2004
Terima Dari	H/
No. Agenda Prp.	219618

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2004

**PERANCANGAN DAN PEMBUATAN PERANGKAT
LUNAK PENGEKSTRAKSI KAIDAH ASOSIASI DAN
NAIVE BAYES BERBASIS BAHASA PREDICTIVE
MODEL MARKUP LANGUAGE DENGAN
MENGGUNAKAN KOMPONEN
ORACLE9i DATA MINING**

TUGAS AKHIR

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

Mengetahui / Menyetujui

Dosen Pembimbing

acc  5/2/04

**Rully Soelaiman, S.Kom., M.Kom.
NIP. 132 085 802**

**SURABAYA
FEBRUARI 2004**

ABSTRAK

Belakangan ini makin banyak teknologi baru data mining. Di antaranya yang menarik adalah keluarnya Oracle9i dan teknologi pemodelan dengan format Predictive Model Markup Language (PMML). Dengan Oracle9i fungsi-fungsi data mining sudah terintegrasi dalam basisdata(disebut Oracle9i Data Mining), sehingga pengguna tidak perlu memikirkan algoritma apa yang dipakai. Sedangkan dengan ditemukannya standar penulisan model PMML akan mempermudah pengguna dalam hal membandingkan model terutama yang lintas platform. Sebagai tambahan, Oracle9i Data Mining menyediakan kemampuan untuk melakukan proses eksport dan import model dalam format PMML.

Hubungan kedua teknologi, Oracle9i Data Mining dan PMML, menjadi sebuah topik yang dibahas dalam tugas akhir ini. Dalam tugas akhir ini dieksplorasi obyek-obyek yang disediakan oleh Oracle9i Data Mining untuk mengeksplorasi fitur-fitur yang tersedia terutama fitur untuk pembuatan model, eksport ke PMML, dan import dari PMML. Karena untuk sementara Oracle9i Data Mining hanya mendukung eksport-import model dari algoritmat Kaidah Asosiasi dan Naïve Bayes, maka dalam tugas akhir ini dibatasi hanya pada dua algoritma tersebut.

Uji coba dilakukan terhadap data dalam jumlah besar maupun kecil. Pengujian yang dilakukan menunjukkan kemudahan cara pembuatan model, juga validitas model yang diimport maupun model yang dieksport.

Kata Kunci : *Oracle9i Data Mining, Predictive Model Markup Language (PMML), Kaidah Asosiasi, Naïve Bayes*

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir berjudul :

**"PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK
PENGEKSTRAKSI KAI DAH ASOSIASI DAN NAIVE BAYES BERBASIS
BAHASA PREDICTIVE MODEL MARKUP LANGUAGE DENGAN
MENGGUNAKAN KOMPONEN ORACLE9i DATA MINING"**

Tugas akhir ini merupakan syarat akademis mahasiswa dalam rangka menyelesaikan studi kesarjanaan pada jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Besar harapan penulis Tugas Akhir ini dapat memberikan suatu kontribusi tersendiri bagi wahana ilmu pengetahuan dalam bidang Teknologi Informasi secara umum. Penulis juga menyadari bahwa penyusunan buku laporan ini tidak luput dari kesalahan-kesalahan dan kekurangan-kekurangan lainnya. Untuk itu penulis membuka diri dan akan sangat menghargai segala saran dan kritik yang sifatnya membangun.

Akhir kata, semoga laporan ini bermanfaat yang seluas-luasnya khususnya untuk penulis sendiri, umumnya buat masyarakat. Amin.

Surabaya, Februari 2004

Penulis

UCAPAN TERIMA KASIH

Segala puji bagi Allah SWT yang Maha Pengasih lagi Maha Penyayang. Pada kesempatan ini penulis menyampaikan terima kasih dan penghargaan yang sebesar-besarnya, atas bimbingan, dukungan, doa, serta bantuan yang diberikan kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini. Secara khusus penulis menyampaikan terima kasih sebesar-besarnya kepada :

1. Kedua orang tua penulis, ayahanda Agung Wibawanto dan ibunda Yayuk Suprih Utami. Terima kasih yang sebesar-besarnya atas kasih sayang, dukungan, didikan, dan segala bantuan yang telah ayahanda dan ibunda berikan. Hanya Allah SWT yang bisa membalasnya.
2. Bapak Rully Soelaiman, S.Kom, M.Kom selaku dosen pembimbing yang telah memberikan masukan, bimbingan, dan pengalaman yang begitu berarti terutama saat pengerjaan Tugas Akhir.
3. Bapak Yudhi Purwananto, S.Kom, M.Kom, selaku ketua Jurusan Teknik Informatika, juga Bapak Febriliyan Samopha, S.Kom,M.Kom selaku dosen wali penulis selama perkuliahan.
4. Mas Yudis, mbak Menuk, dan Aning, serta spesial buat si kecil Rangga.
Lee... pak lik lulus !!!
5. Keluarga di Ketintang Timur, eyang putri, paklik, bulik, dan adik-adik, yang sudah memberikan semangat buat menyelesaikan studi.
6. Yuni Lestari yang sudah memberikan dukungan, perhatian, dan semangat yang tiada henti. Juga buat Bapak dan Ibu Imam Rohmad di Blitar dan Deni di Bandung yang sudah memberikan doa buat kelancaran Tugas Akhir ini.

7. Teman baik penulis, Rossi Lazuardi, Suhariyono, Bhakti, Nges, Priyo. Sukses Mas !!
8. Teman-teman C0E. Buat yang suka berfutsal ria : Joko, Caca, Wong, Deni, Dian, Dimas, Deni, Didiet, Kendi, Yuda, Abi, mbah Nur. Yang ada di kantor IAO : Aryo, Intan, Ade, Ifa, Raras, Maman, Kamil, Budi, Yos, Ratna. Yang sidang bareng penulis : Luluk, Odi, Bomphi, Kakek, Andias, Danang. Yang lagi nyiapin sidang buat semester depan : Anang, Harry, Gusmul, Toshi, Decky, Trisno. Maaf buat yang belum disebut, yang pasti terima kasih banget.
9. Teman-teman angkatan lain, mas Khalid, Ade, Khalid Fathoni, Lintang, Primadoni, dan masih banyak lagi. Terima kasih buat kerja samanya selama ini.
10. Tim PDAM Kota Surabaya, Saenal, Wicak, Mas Budi, Mas Kuncoro, dan Pak Arif. Terima kasih atas keceriaan yang diberikan.
11. Pembesar-pembesar PT Infoglobal AutOptima, Bapak Achmad Hadi, Suhadi Lili, Khairul Huda, Fauzi, Bahtiar HS, dan Majid. Terima kasih sudah diberi kesempatan menimba ilmu di IAO.
12. Warung-warung makanan di Keputih dan sekitarnya yang sudah mendukung penulis dalam hal logistik.
13. Komputer yang setia menemani dan menghibur penulis dan juga sepeda motor Supra yang sudah rela mengantar penulis.
14. Untuk yang belum disebutkan dalam lembar terima kasih ini. Maaf bila nama Anda belum tercantum. Jasa Anda tidak akan pernah penulis lupakan.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN.....	ii
ABSTRAK.....	iii
KATA PENGANTAR	iv
UCAPAN TERIMA KASIH.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan.....	3
1.3 Tujuan dan Manfaat	4
1.4 Batasan Masalah.....	4
1.5 Metodologi Pembuatan Tugas Akhir	4
1.6 Sistematika Penulisan Buku Tugas Akhir.....	5
BAB II TEKNOLOGI ORACLE9i DATA MINING DAN PMML.....	7
2.1 Predictive Model Markup Language.....	7
2.1.1 Struktur Umum Sebuah Dokumen PMML	9
2.1.2 Tipe Data Dasar dan Entitas.....	13
2.2 Oracle9i Data Mining	14
2.2.1 Data Mining yang Melekat pada Basisdata Oracle9i	17
2.2.2 Algoritma Oracle9i Data Mining	19
2.3 Klasifikasi dan Prediksi dengan Naive Bayes.....	19
2.4 Menemukan Asosiasi dengan Kaidah Asosiasi.....	20
2.5 Obyek-obye dan Fungsi dalam Oracle9i Data Mining.....	22
BAB III PERANCANGAN PERANGKAT LUNAK.....	29
3.1 Desain Data	29
3.1.1 Data Masukan.....	30
3.1.2 Data Proses.....	34
3.1.3 Data Keluaran.....	34
3.2 Desain Proses Pemodelan, Eksport, dan Import	35
3.3 Desain Obyek	47
3.3.1 Membuat Koneksi ke DMS (Data Mining Server)	53
3.3.2 Mendeskripsikan Data yang Dibangun	53
3.3.2.1 Lokasi Akses Data untuk Membuat Data.....	53
3.3.2.2 Spesifikasi Fisik Data Transaksional	54
3.3.2.3 Spesifikasi Fisik untuk Membuat Data <i>Transactional</i>	54
3.3.3 Membuat Obyek MiningFunctionSetting.....	55
3.3.3.1 Menentukan Algoritma Untuk Klasifikasi dan Asosiasi.....	55
3.3.3.2 Menentukan Algoritma Naive Bayes	57
3.3.3.3 Melakukan Validasi terhadap Seting Fungsi Mining.....	57
3.3.4 Membuat Proses Sinkron dan Asinkron.....	58
3.3.4.1 Pembuatan Model Secara Sinkron	58

3.3.4.2 Pembuatan Model Secara Asinkron	59
3.3.5 Penentuan Lokasi Data Akses	59
3.3.6 Eksport Model ke Format PMML	60
3.3.7 Import Model dari Format PMML	61
3.4 Desain Antarmuka	61
BAB IV IMPLEMENTASI PERANGKAT LUNAK	65
4.1 Arsitektur Perangkat Lunak	66
4.2 Implementasi Data	67
4.2.1 Main	68
4.2.2 clsODMConnection	68
4.2.3 clsAssociationParameter	69
4.2.4 clsNBProsesParameter	71
4.2.5 clsAscToPMMLParameter	73
4.2.6 clsModelToPMML	74
4.3 Implementasi Proses	74
4.3.1 Pembuatan Model Kaidah Asosiasi (clsAssociationRule)	75
4.3.2 Pembuatan Model Naive Bayes (clsNBBuild)	76
4.3.3 Proses Eksport Model ke dalam Format PMML	77
4.3.4 Proses Import dari PMML ke Model Kaidah Asosiasi	78
4.4 Implementasi Antarmuka	79
4.4.1 Menu Login	80
4.4.1.a Login ODM	81
4.4.1.b Logout ODM	82
4.4.1.c Exit	82
4.4.2 Menu Association Rules	82
4.4.2.a Association Rules Model Build and Export	82
4.4.2.b Association Rules Import	88
4.4.2.c View Result	91
4.4.3 Menu Naive Bayes	92
4.4.3.a Naive Bayes Model Build and Eksport	93
4.4.3.a Naive Bayes Model Import	94
BAB V UJI COBA DAN EVALUASI	95
5.1 Lingkungan Uji Coba	95
5.2 Pembuatan Model dan Eksport	95
5.2.1 Kaidah Asosiasi	97
5.2.2 Naive Bayes	111
5.3 Proses Import	119
5.3.1 Kaidah Asosiasi	120
5.3.2 Naive Bayes	121
BAB VI KESIMPULAN DAN SARAN	123
6.1 Kesimpulan	123
6.2 Saran	124
DAFTAR PUSTAKA	125

DAFTAR GAMBAR

Gambar 2.01 Kegunaan PMML	8
Gambar 2.02 Komponen dan Fitur Utama Oracle9i Release 2	14
Gambar 2.03 Level Pada Business Intelegence	15
Gambar 3.01 <i>Use Case</i> Perangkat Lunak	35
Gambar 3.02 <i>Activity Diagram</i> – membuat koneksi ke skema ODM	36
Gambar 3.03 <i>Activity Diagram</i> – membuat model Kaidah Asosiasi	37
Gambar 3.04 <i>Activity Diagram</i> – eksport model ke PMML	37
Gambar 3.05 <i>Activity Diagram</i> – pembuatan model Naive Bayes	38
Gambar 3.06 <i>Activity Diagram</i> – mengimport model Kaidah Asosiasi	38
Gambar 3.07 <i>Activity Diagram</i> – mengimport model Naive Bayes	39
Gambar 3.08 <i>Activity Diagram</i> – menyimpan berkas hasil eksport	40
Gambar 3.09 <i>Activity Diagram</i> – mengeksekusi proses mining	40
Gambar 3.10 <i>Activity Diagram</i> – melihat kaidah model	41
Gambar 3.11 <i>Sequence Diagram</i> – membuat koneksi ke skema ODM	41
Gambar 3.12 <i>Sequnce Diagram</i> – membuat model Kaidah Asosiasi	42
Gambar 3.13 <i>Activity Diagram</i> – membuat model Naive Bayes	43
Gambar 3.14 <i>Sequence Diagram</i> – eksport model Naive Bayes	44
Gambar 3.15 <i>Sequence Diagram</i> – eksport Kaidah Asosiasi	44
Gambar 3.16 <i>Sequence Diagram</i> – mengimport model Kaidah Asosiasi	45
Gambar 3.17 <i>Sequence Diagram</i> – mengimport model Naive Bayes	45
Gambar 3.18 <i>Sequence Diagram</i> – menyimpan berkas	46
Gambar 3.19 <i>Sequence Diagram</i> – melakukan proses mining	46
Gambar 3.20 <i>Sequence Diagram</i> – melihat kaidah model	47
Gambar 3.21 <i>Class Diagram</i> - Pembuatan Model Kaidah Asosiasi	48
Gambar 3.22 <i>Class Diagram</i> - Pembuatan Model Naive Bayes	49
Gambar 3.23 <i>Class Diagram</i> - Eksport Model	50
Gambar 3.24 <i>Class Diagram</i> – import model	51
Gambar 3.25 Tahap Pembuatan Model	52
Gambar 3.26 Inisialisasi DMS	53
Gambar 3.27 Inisialisasi LocationAccessData	54
Gambar 3.28 Inisialisasi PhysicalDataSpecification - nonTransactional	54
Gambar 3.29 Inisialisasi PhysicalDataSpecification – Transactional	54
Gambar 3.30 Inisialisasi ClassificationFunctionSettings	56
Gambar 3.31 Inisialisasi AssociationRulesFunctionSettings	56
Gambar 3.32 Penentuan ClassificationFunctionSettings untuk Naive Bayes ..	57
Gambar 3.33 Pemakaian ClassificationFunctionSetting untuk Validasi ..	58
Gambar 3.34 Pemakaian AssociatioRulesFunctionSettings untuk Validasi ..	58
Gambar 3.35 Pembuatan Model Secara Sinkron	58
Gambar 3.36 Pembuatan Model Secara Asinkron	59
Gambar 3.38 Inisialisasi Model ExportTask	61
Gambar 3.39 Susunan Menu pada Perangkat Lunak	62

Gambar 4.01	Arsitektur Aplikasi	66
Gambar 4.02	Tampilan Utama Perangkat Lunak.....	80
Gambar 4.03	Sub Menu Login.....	81
Gambar 4.04	Antarmuka Login Perangkat Lunak	81
Gambar 4.05	Dialog Konfirmasi.....	81
Gambar 4.06	Dialog Konfirmasi Logout Berhasil	82
Gambar 4.07	Sub Menu Association Rule.....	82
Gambar 4.08	Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 1	83
Gambar 4.09	Dialog Konfirmasi Pembuatan Model.....	83
Gambar 4.10	Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 2	84
Gambar 4.11	Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 3	85
Gambar 4.12	Dialog Konfirmasi Proses Eksport Sukses.....	85
Gambar 4.13	Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 4	86
Gambar 4.14	Dialog Konfirmasi Pembuatan PMML	86
Gambar 4.15	Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 5	87
Gambar 4.16	Dialog Konfirmasi Penyimpanan PMML	88
Gambar 4.17	Dialog Import Model PMML	88
Gambar 4.18	Dialog Membuka Berkas PMML	89
Gambar 4.19	Dialog Import PMML Setelah Berkas Dibuka.....	89
Gambar 4.20	Dialog Konfirmasi Kesalahan Sintaks	90
Gambar 4.21	Konfirmasi Proses Sukses	90
Gambar 4.22	Dialog Laporan Proses Import	91
Gambar 4.23	Dialog View Result	92
Gambar 4.24	Sub Menu Naive Bayes	92
Gambar 4.25	Dialog Pembuatan dan Eksport Naive Bayes Step 1	93
Gambar 4.26	Dialog Pembuatan dan Eksport Naive Bayes Step 2	94
Gambar 5.01	Skenario Uji Coba Pemodelan dan Eksport	96
Gambar 5.01	Data Kaidah Asosiasi	100
Gambar 5.02	Hasil Uji Tabel Transaksional	100
Gambar 5.03	Hasil Uji Tabel non-Transaksional.....	101
Gambar 5.04	Kesalahan pada Pembuatan Model Kaidah Asosiasi	102
Gambar 5.05	Hasil Pemodelan untuk Tabel Bawaan Oracle	102
Gambar 5.06	Ouput Eksport ke PMML dari Tabel Transaksional	106
Gambar 5.07	Ouput Eksport ke PMML dari Tabel non-transaksional	110
Gambar 5.08	Ouput Pemodel Naive Bayes dengan ODMBrowser	112
Gambar 5.09	Ouput Eksport ke PMML dari Naive Bayes	119
Gambar 5.10	Skenario Uji Coba Import	120
Gambar 5.11	Tag PMML sebelum Perubahan.....	120
Gambar 5.13	Output Proses Import Model PMML	121
Gambar 5.14	Output Proses Import dengan ODMBrowser	122



DAFTAR TABEL

Tabel 2.01 Non-Transaksional, format data dengan satu record.....	18
Tabel 2.02 Format data transaksional	18
Tabel 2.03 Kemampuan Data Mining.....	23
Tabel 4.01 Class Main.....	68
Tabel 4.02 Class clsODMConnection.....	68
Tabel 4.03 clsAssociationParameter	69
Tabel 4.04 clsNBProsesParameter	71
Tabel 4.05 clsAscToPMMLParameter.....	73
Tabel 4.06 clsModelToPMML.....	74
Tabel 4.07 Class clsAssociationRule	75
Tabel 4.08 Class clsNBBuild	76
Tabel 4.09 clsAscToPMMLProses	77
Tabel 4.10 clsPMMLToAR.....	78
Tabel 5.01 Transaksi yang Diuji	98
Tabel 5.02 Data TABLE_UJI_TRANSAKSIONAL	98
Tabel 5.03 TABEL_UJI_NONTRANSAKSIONAL	99
Tabel 5.04 <i>Frequent Itemset (min_sup=50%)</i>	99
Tabel 5.05 Hasil Uji dengan Perangkat Lunak	101

BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun belakangan ini telah banyak model-model prediktif yang dibangun oleh komunitas data mining. Pada akhirnya muncul banyak ketertarikan untuk melakukan perbandingan dan evaluasi terhadap model-model yang berbeda. Predictive Model Markup Language (PMML) adalah sebuah solusi yang bisa diharapkan dalam menyelesaikan masalah pemindahan model prediktif terutama untuk membuat dan memadukan proses pembelajaran yang terdistribusi. Bahasa *markup* yang berbasis teks dan dibuat sebagai model prediktif ini membuat model-model prediktif menjadi lebih mudah untuk dianalisa dan dibandingkan serta juga dapat mengurangi kebutuhan terhadap kesesuaian antar platform tempat model tersebut dibentuk.

PMML juga mengijinkan penambahan model baru hanya dengan melakukan perubahan terhadap model yang ada di dalam dokumen yang biasa disebut sebagai Document Type Definition (DTD). Selain itu PMML menyediakan mekanisme yang sangat fleksibel dalam membuat definisi skema model prediktif dan juga mendukung seleksi dan analisa model saat terdapat banyak model yang diikutsertakan.

Seiring dengan dikeluarkannya Oracle versi 9i, Oracle juga mengeluarkan Oracle Data Mining Release 2 atau biasa disebut sebagai Oracle9i Data Mining

Release 2 (9.2) (ODM). Dalam rilis versi 2-nya terdapat beberapa kemampuan seperti berikut [OR1-02]:

1. *Decision Tree (Adaptive Bayes Network)*, mengembangkan kemampuan ODM dalam teknik pembelajaran *supervised* (teknik pembelajaran yang dapat memprediksi nilai target). Kemampuan ini diimplementasikan menggunakan teknologi *Adaptive Bayes Network*. Keuntungan utama dari penggunaan Pohon Keputusan (*Decision Tree*) adalah kemampuannya menghasilkan satu set aturan atau penjelasan yang dapat langsung dianalisa oleh para analis atau manajer. Kemudian pemakai dapat melakukan *query* terhadap basisdata untuk semua item yang memenuhi kriteria dari aturan yang telah dibuat.

2. *Clustering*, mengembangkan kemampuan ODM dalam teknik pembelajaran *unsupervised* (teknik pembelajaran yang tidak memiliki nilai target).

3. *Attribute Importance*, kemampuan untuk mengidentifikasi atribut-atribut yang memiliki efek terbesar. Hal ini dapat dipergunakan untuk membuat prediksi yang berguna dari setiap atribut yang tersedia dan mengurutkan atribut-atribut itu sesuai dengan kepentingan prediksinya.

4. *Model Seeker*, memberi kemampuan kepada ODM untuk secara otomatis membangun model majemuk data mining dengan input yang minimal, membandingkan model-model, serta memilih model terbaik yang berhasil dibentuk.

5. *Automated Binning*, melakukan otomatisasi tugas terhadap proses diskretisasi semua atribut ke dalam tempat yang terkategorisasi untuk keperluan perhitungan.

6. *Predictive Model Markup Language*, mendukung import dan eksport model PMML untuk *Naïve Bayes* dan model *Association Rules* (Kaidah Asosiasi). PMML mengijinkan aplikasi data mining untuk menghasilkan dan sekaligus membaca model yang digunakan oleh aplikasi Data Mining yang mengikuti standar PMML 2.0.

7. *Mining Task*, semua operasi data mining (pembuatan, pengujian, perhitungan, pengaplikasian, import, dan export) dibangun secara *asynchronous* dengan menggunakan *Mining Task*.

1.2 Permasalahan

Permasalahan yang diangkat dalam Tugas Akhir ini adalah :

- a. Membentuk aplikasi yang mampu menerima definisi-definisi model Kaidah Asosiasi atau Naive Bayes untuk kemudian hasil pemodelan ditampilkan ke dalam bahasa PMML.
- b. Membaca hasil pemodelan Kaidah Asosiasi atau Naive Bayes yang ditulis dalam bahasa PMML kemudian dilakukan proses terhadap data yang ada.
- c. Menampilkan hasil proses dari model yang telah didefinisikan.
- d. Memanfaatkan fitur dalam Oracle9i Data Mining untuk melakukan konversi dari model yang ditulis ke dalam PMML ke dalam model yang bisa diterima Oracle9i Data Mining dan juga sebaliknya merubah model Oracle9i Data Mining ke dalam PMML.

1.3 Tujuan dan Manfaat

Tujuan dari tugas akhir ini adalah membangun sebuah aplikasi yang bisa menerima model yang mengikuti Kaidah Asosiasi atau Naive Bayes untuk diekstraksi ke dalam bentuk bahasa PMML sekaligus bisa menerima model Kaidah Asosiasi dan Naive Bayes yang ditulis dalam bahasa PMML untuk kemudian dibaca dan dilakukan proses terhadap basisdata yang ada.

Apabila tujuan Tugas Akhir ini dapat tercapai maka dapat diperoleh manfaat dari perangkat lunak yang terbentuk yaitu sebuah perangkat lunak yang berdiri sendiri untuk membuat model Kaidah Asosiasi dan Naive Bayes sekaligus mampu melakukan eksport import dalam format PMML

1.4 Batasan Masalah

Untuk membuat penyusunan tugas akhir tetap sesuai dengan tujuan yang ingin dicapai maka perlu diberikan beberapa batasan-batasan dari masalah yang timbul di atas, yaitu:

- a. Model prediktif yang ditulis dalam PMML nantinya didukung standar versi 2.0.
- b. Model Data Mining yang dihasilkan hanya valid untuk Oracle9i Data Mining.
- c. Oracle9i Data Mining menghasilkan dan menerima dua macam tipe model yaitu model Kaidah Asosiasi dan Naive Bayes.

1.5 Metodologi Pembuatan Tugas Akhir

Metodologi penelitian yang dilakukan menggunakan langkah-langkah sebagai berikut :

- a. Studi Literatur

Pengumpulan literatur mengenai pemrograman PMML, Oracle 9i Data Mining, dan XML. Juga dipelajari fungsi-fungsi yang terdapat dalam Oracle 9i Data Mining sehingga bisa dibentuk model yang bisa diterima oleh Oracle9i Data Mining.

b. Perancangan Perangkat Lunak

Pada tahap ini dilakukan perancangan perangkat lunak yang meliputi diagram alur yang digunakan, struktur data yang diterima, desain antarmuka aplikasi, dan algoritma aplikasi.

c. Implementasi Perangkat Lunak

Tahap ini merupakan implementasi tahap sebelumnya. Pada tahap ini dilakukan pembuatan kode program perangkat lunak import dan eksport.

d. Uji Coba dan Evaluasi

Aplikasi yang telah dibuat diuji coba dan dievaluasi untuk mengetahui sejauh mana tercapainya tujuan dari aplikasi ini.

e. Penyusunan Tugas Akhir

Pada tahap akhir ini disusun sebuah dokumentasi dari awal hingga akhir dari pelaksanaan Tugas Akhir. Dokumentasi ini berguna dalam pengembangan aplikasi eksport dan import ini.

1.6 Sistematika Penulisan Buku Tugas Akhir

Penulisan buku tugas akhir ini, dibagi menjadi beberapa bab dengan isi dari masing-masing bab adalah seperti berikut :



Bab I berisi pendahuluan. Pada bab ini dibahas latar belakang pengembangan perangkat lunak, permasalahan, tujuan dan manfaat dikembangkannya perangkat lunak, dan batasan-batasan pada perangkat lunak yang dikembangkan.

Bab II membahas teknologi Oracle9i Data Mining dan Predictive Model Markup Language. Bab ini berisi bahan-bahan yang dijadikan acuan bagi pengembangan perangkat lunak, meliputi teori-teori tentang Predictive Model Markup Language, teori obyek-obyek Oracle9i Data Mining, Kaidah Asosiasi, dan Naive Bayes.

Bab selanjutnya yaitu Bab III membahas perancangan perangkat lunak. Pada bab ini dibahas bagaimana spesifikasi perangkat lunak yang dikembangkan, meliputi aturan-aturan dan kemampuan-kemampuan yang harus dimiliki. Dibahas juga kebutuhan sistem dan kebutuhan pengguna untuk perangkat lunak ini.

Bab IV merupakan pengimplementasian dari pembahasan bab sebelumnya .Bab ini membahas bagaimana perangkat lunak dibuat. Meliputi bagaimana arsitekturnya, struktur data yang digunakan, dan algoritma-algoritma penting yang ada didalamnya.

Bab V membahas uji coba dan evaluasi perangkat lunak. Perangkat lunak yang dikembangkan diuji dan dievaluasi agar dapat dipertanggungjawabkan kebenarannya.

Bab selanjutnya berisi kesimpulan dan saran dari pengembangan perangkat lunak pengekstraksi Kaidah Asosiasi dan Naive Bayes ini.

BAB II

TEKNOLOGI ORACLE 9i DATA MINING DAN PMML

BAB II

Teknologi Oracle 9i Data Mining dan PMML

Pada bab ini dibahas dasar teori yang mendukung pembuatan tugas akhir ini.

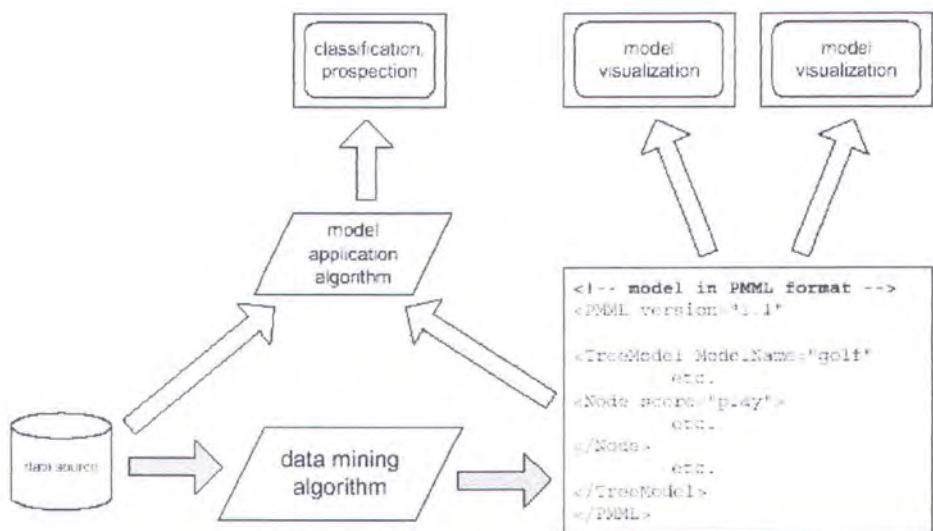
Akan dibahas dasar-dasar Predictive Model Markup Language, Oracle9i Data Mining, Kaidah asosiasi dan Naive Bayes.

2.1 Predictive Model Markup Language

Beberapa tahun belakangan ini bermacam model prediktif telah dibangun oleh komunitas data mining. Ada ketertarikan yang signifikan untuk membandingkan dan mengevaluasi model-model berbeda yang sudah ada. Predictive Model Markup Language (PMML) adalah sebuah solusi handal terhadap masalah pertukaran model-model prediktif dan untuk menghasilkan sebuah proses pembelajaran yang sesuai dan terdistribusi.

PMML adalah sebuah bahasa XML yang dipergunakan untuk menggambarkan dan menyimpan model data mining. PMML adalah sebuah bahasa berbasiskan XML yang menyediakan sebuah cara kepada sebuah aplikasi untuk mendefinisikan model statistik dan data mining dan untuk saling berbagi model di antara aplikasi yang mendukung PMML. Versi pertama (1.0) diperkenalkan pada Juli 1999 oleh Data Mining Group (DMG).

Model prediktif yang ditulis dalam bahasa markup berbasis teks ini mengijinkan kemudahan analisa dan perbandingan serta mengurangi kebutuhan terhadap kompatibilitas biner di antara platform-platform tempat model prediktif tersebut dibangun. PMML juga mengijinkan penambahan model baru hanya dengan mengakomodasikan Document Type Definition (DTD). Hal ini memberi manfaat dalam flexibilitas mekanisme untuk melakukan definisi skema untuk model prediktif dan mendukung seleksi model saat banyak model diikutkan. Sebagai tambahan, hal ini memfasilitasi perpindahan model antar aplikasi dan sistem.



Gambar 2.01 Kegunaan PMML

Gambar 2.1 [GRO-98] menunjukkan proses data mining yang disederhanakan. Model PMML menjadi sebuah output dari sebuah algoritma data mining. Karena PMML adalah standar terbuka, model ini dapat digunakan oleh aplikasi lain.

Alasan utama yang menyebabkan penggunaan PMML sebagai model prediktif adalah :

- universal : Sebuah bahasa markup sederhana harus dapat digunakan dan dimengerti dengan mudah oleh banyak orang. Selain itu bahasa tersebut haruslah bisa mengikuti teknologi yang ada.
- portabel : Menjadi sebuah kebutuhan bahasa markup tersebut portabel terhadap sebuah mesin data mining pada satu platform ke beberapa mesin pada platform perangkat lunak lain yang berbeda.
- mudah terbaca : Bahasa markup tersebut seharusnya mudah terbaca sehingga seseorang dapat dengan mudah melihat dan mengubah instan modelnya.
- bisa dikembangkan : Fitur ini menjadi sebuah perhatian utama akibat adanya fakta bahwa evolusi dari sebuah model baru di masa datang dapat diterima dan penambahannya ke dalam bahasa markup bisa semudah mungkin.

2.1.1 Struktur Umum Sebuah Dokumen PMML

PMML mempergunakan XML untuk merepresentasikan sebuah model. Struktur model dideskripsikan oleh sebuah DTD yang disebut PMML DTD. Satu atau lebih model mining dapat diikutkan dalam sebuah dokumen PMML. Sebuah dokumen PMML adalah sebuah dokumen XML dengan elemen utama adalah bertipe PMML [DMG-02]. Struktur umum dari sebuah dokumen PMML adalah tampak seperti berikut :

```
<?xml version="1.0"?>
```

```
<!DOCTYPE PMML
    PUBLIC "PMML 2.0"
    "http://www.dmg.org/v2-0/pmml_v2_0.dtd">
<PMML version="2.0">
...
</PMML>
```

Sebuah dokumen PMML tidak harus memiliki deklarasi DOCTYPE. Jika terdapat lebih dari satu sebuah dokumen PMML harus tidak tergantung kepada parameter eksternal, yang diasumsikan atribut asal standalone="yes" pada <?xml?>. Meskipun sebuah dokumen PMML harus valid sesuai dengan DTDnya, sebuah tidak membutuhkan validasi parser, yang mengambil entitas eksternal. Untuk mendapatkan sebuah format XML yang valid, dokumen PMML yang valid harus mematuhi sejumlah aturan yang dideskripsikan pada bermacam tempat dalam sebuah spesifikasi PMML.

Elemen utama dari sebuah dokumen harus memiliki tipe PMML.

```
<!ENTITY % A-PMML-MODEL '(TreeModel |
    NeuralNetwork |
    ClusteringModel |
    RegressionModel |
    GeneralRegressionModel |
    NaiveBayesModels |
    AssociationModel |
    SequenceModel )' >

<!ELEMENT PMML ( Header,
    MiningBuildTask?,
    DataDictionary,
    TransformationDictionary?,
    (%A-PMML-MODEL;)*,
    Extension* )>
<!ATTLIST PMML
    version   CDATA  #REQUIRED
    >
```

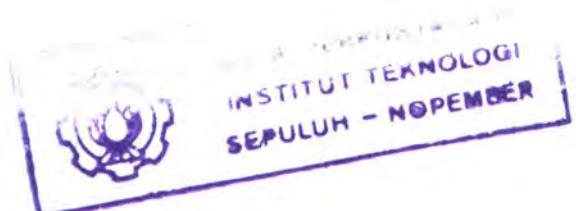
<!ELEMENT MiningBuildTask (Extension*) >

Sebuah dokumen PMML dapat berisi lebih dari sebuah model [DMG-02]. Jika sistem aplikasi menyediakan sebuah tujuan untuk menyeleksi model berdasarkan nama dan jika PMML menunjukkan sebuah model, maka model tersebut digunakan; dengan kata lain model yang pertama yang digunakan. Isi dari model mining dalam sebuah model PMML boleh kosong. Dokumen tersebut dapat digunakan untuk membawa metadata sebelum model sebenarnya dihitung. Sebuah dokumen PMML yang tidak berisi satu model tidak memiliki arti terhadap aplikasi pembaca PMML.

Untuk PMML versi 2.0 atribut versi harus memiliki angka 2.0.

Element **MiningBuildTask** dapat berisi beberapa nilai XML yang menggambarkan konfigurasi dari bagian pembelajaran yang menghasilkan instan model. Informasi ini tidak langsung dibutuhkan oleh pembaca PMML, namun dalam banyak kasus sangat menolong dalam melakukan pemeliharaan dan visualisasi model. Struktur isi khusus dari MiningBuildTask tidak didefinisikan oleh PMML 2.0. Akan tetapi, elemen ini menjadi isi asal untuk spesifikasi tugas yang didefinisikan oleh standar mining yang lain, misalnya dalam SQL atau Java.

Bagian dalam DataDictionary dan TransformationDictionary disebutkan dengan nama yang unik. Elemen lain dalam model dapat menunjukkan nama bagian ini. Model majemuk dalam sebuah dokumen PMML dapat berbagi bagian yang sama dalam TransformationDictionary. Beberapa tipe model seperti jaringan syaraf atau logistik regresi dapat digunakan untuk keperluan-keperluan yang



berbeda, yaitu beberapa instan yang mengimplementasikan prediksi dari nilai numerik, saat prediksi yang lain dapat digunakan untuk klasifikasi. Oleh karena itu, PMML mendefinisikan empat fungsi mining yang berbeda. Setiap model memiliki sebuah atribut “function name” yang menunjukkan fungsi mining yang diinginkan.

```
<ENTITY % MINING-FUNCTION '(associationRules |
    sequences |
    classification |
    regression |
    clustering )' >
```

Untuk semua model PMML struktur elemen model paling atas adalah sama dengan Xmodel berikut :

```
<!ELEMENT XModel (Extension*, MiningSchema,
    ModelStats?, ..., Extension* ) >
<!ATTLIST XModel
    modelName      CDATA          #IMPLIED
    functionName   %MINING-FUNCTION; #REQUIRED
    algorithmName  CDATA          #IMPLIED
>

<!ELEMENT MiningSchema (MiningField+) >
<!ELEMENT ModelStats (UnivariateStats+) >
```

Sebuah daftar dari bagian mining yang terisi mendefinisikan sebuah skema mining berikut ini :

- **Modelname :** nilai pada *modelname* menunjukkan model dengan nama yang unik dalam konteks file PMML. Atribut ini tidak harus ada. Pemakai PMML diberi kebebasan untuk mengatur nama model yang mereka buat sesuai dengan kekhususan yang diinginkan.

- **FunctionName** dan **algorithmName** menunjukkan macam model mining, seperti pengelompokan (*clustering*) atau klasifikasi (*classification*). Nama algoritma bisa menjadi deskripsi algoritma spesifik yang menghasilkan model. Atribut hanya bersifat informasi.

2.1.2 Tipe Data Dasar dan Entitas

Definisi berikut :

<!ENTITY % NUMBER "CDATA">

umumnya digunakan untuk membedakan nilai numerik dari data yang lain. Numerik boleh menjadi awalan, pecahan, atau eksponen. Dalam hubungan dengan tipe data *Number* ada definisi yang lebih spesifik :

- <!ENTITY % INT-NUMBER "CDATA">

Sebuah “CDATA” pada INT-NUMBER haruslah bilangan bulat, bukan pecahan atau eksponen.

- <!ENTITY % REAL-NUMBER "CDATA">

Sebuah “CDATA” pada REAL-NUMBER dapat diwakili oleh bahasa java sebagai *float*, *long*, atau *double*. Notasi eksponensial seperti 1,23 E3, diijinkan.

- <!ENTITY % PROB-NUMBER "CDATA">

Sebuah “CDATA” pada PROB-NUMBER adalah sebuah REAL-NUMBER diantara 0.0 dan 1.0 yang biasanya menunjukkan probabilitas.

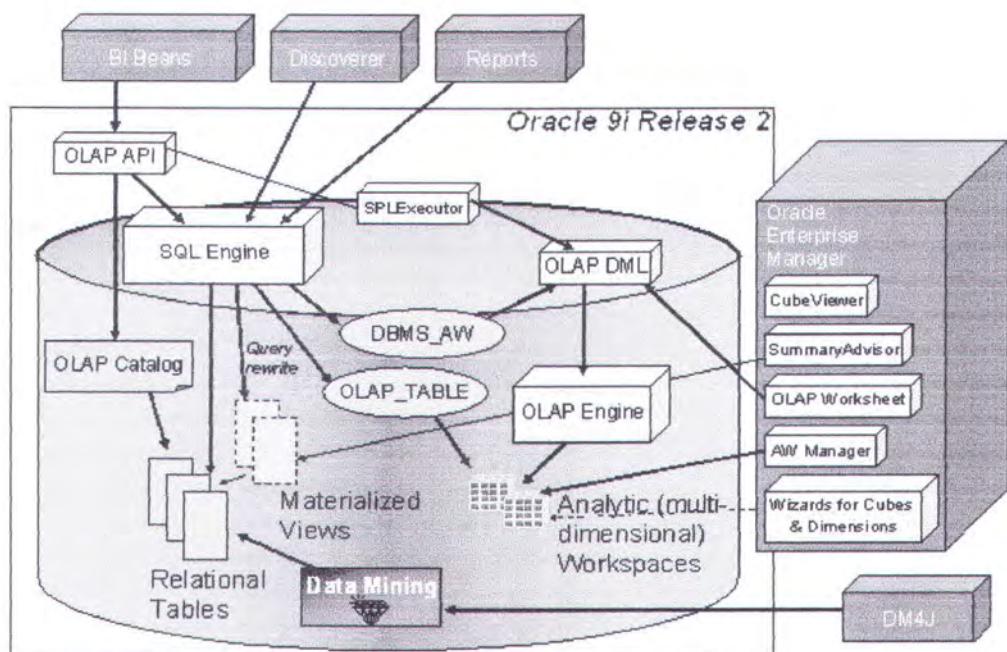
- <!ENTITY % PERCENTAGE-NUMBER "CDATA">

Sebuah “CDATA” pada PERCENTAGE-NUMBER adalah sebuah REAL-NUMBER diantara 0.0 dan 100.0.

Sebagai catatan, entitas-entitas tersebut tidak mengakibatkan parser XML untuk melakukan pengecekan tipe data, namun hanya mendefinisikan kebutuhan sebuah dokumen PMML yang valid.

2.2 Oracle9i Data Mining

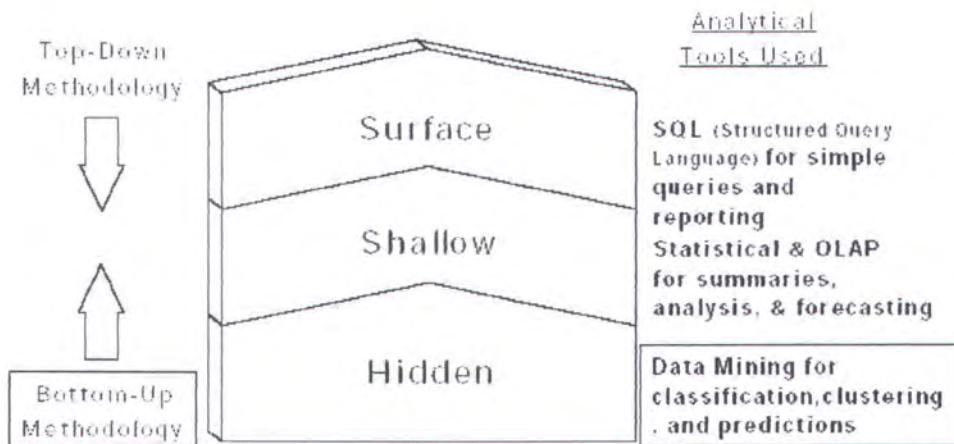
Basisdata Oracle9i , terutama rilis 2, menyediakan sebuah server yang terintegrasi untuk semua operasi Data Warehouse,OLAP, dan juga Data Mining. Operasi-operasi tersebut sebelumnya diproses dalam lingkungan yang berbeda misalnya dengan penggunaan basisdata multidimensional Oracle Express. Komponen dan fitur utama Oracle9i Release 2 untuk Data Warehouse, OLAP, dan Data Mining tampak berikut ini [OR3-2001] :



Gambar 2.02 Komponen dan Fitur Utama Oracle9i Release 2

Oracle9i Data Mining mengijinkan berbagai macam perusahaan untuk membangun aplikasi *Business Intelligence* yang melakukan eksplorasi data, menemukan wawasan baru, dan melakukan integrasi informasi yang ada ke dalam aplikasi bisnis. Oracle9i Data Mining merupakan pilihan dari Oracle9i Database Enterprise Edition (EE) yang memasukkan fungsionalitas datamining untuk melakukan klasifikasi, prediksi, dan asosiasi. Semua fungsi untuk membuat dan menghitung model dapat diakses melalui API berbasiskan Java.

Bila digambarkan pada sebuah level, posisi Oracle9i Data Mining pada sebuah *business intelligence* dapat digambarkan seperti berikut [SOL-02] :



Gambar 2.03 Level Pada Business Intelegence

Bagaimana perbedaan *data mining* dengan alat lain yang ada pada *business intelligence* ?

Query & Reporting dipergunakan untuk mendapatkan informasi dari basisdata [OR3-01]. Alat ini sesuai untuk menjawab pertanyaan : “Siapa yang paling sering melakukan pembelian pada 3 tahun terakhir ?”

OLAP mengijinkan pengguna untuk secara interaktif membuat ringkasan, melakukan perbandingan, membuat analisa, dan juga membuat ramalan. Alat ini sesuai untuk menjawab pertanyaan : “Berapa rata-rata pendapatan dari pelanggan berdasarkan tahun dan wilayah tempat tinggal ?”

Alat Statistik dipergunakan untuk membuat perbandingan dari contoh yang representatif yang didapatkan dari data dalam jumlah besar.

Query & Reporting, OLAP, dan Statistik berguna dalam hal mengijinkan pengguna melihat secara detil dan mengerti hal-hal yang terjadi di masa sebelumnya. Dengan alat-alat bantu tersebut, pengguna yang mengetahui sesuatu yang dicari, memiliki alat analitis yang handal, menjadi analis yang bagus, dan memiliki banyak waktu, akan menemukan jawabannya.

Data mining tidak memiliki keterbatasan tersebut. *Data mining* berada secara mendalam di dalam data. Dengan *data mining* akan didapatkan analisa detil dari semua peristiwa yang sudah terjadi, misal : “ apakah si A membeli barang B ? ”

Data mining membuat model dan menggunakan sebagai prediktor peristiwa yang akan terjadi, misal “ berapa besarnya kemungkinan si A membeli barang B ? ”

Dengan mempergunakan Oracle9i Data Mining, perusahaan dapat mengalirkan informasi tersembunyi yang ada dalam basisdata mereka untuk mendapatkan sebuah pandangan baru tentang pelanggan dan bisnis mereka. Pada setiap kondisi pada daur hidup pelanggan, Oracle9i Data Mining akan membawa nilai - yang akan bergerak menuju garis batas perusahaan.

Oracle9i Data Mining juga dapat menemukan pola yang tersembunyi dalam pengetahuan, pemerintahan, manufaktur, medis, dan tipe data yang lain. Aplikasi-aplikasi data mining akan berada dalam area berikut :

- melakukan prediksi terhadap sebuah bagian manufaktur,
- menemukan hubungan di antara pasien, obat dan pengeluaran,
- mengidentifikasi tabrakan jaringan yang mungkin terjadi,

2.2.1 Data Mining yang Melekat pada Basisdata Oracle9i

Oracle9i Data Mining mempermudah proses ekstraksi bisnis intelelegensi dari data yang besar. Dengan Oracle9i Data Mining, semua fungsi data mining melekat pada basisdata Oracle9i, akibatnya data, data preparat, aktivitas pembuatan model, dan perhitungan model berada dalam basisdata.

Karena Oracle9i Data Mining menyediakan semua fase dari data mining dalam basisdata, setiap fase data mining menghasilkan perkembangan yang signifikan dalam hal produktivitas, otomasi, dan integrasi. Perkembangan produktivitas yang signifikan dapat diraih dengan mengurangi ekstraksi data dari basisdata menuju alat khusus data mining dan melakukan import hasil proses kembali ke dalam basisdata. Pengembangan ini tercatat dalam data buatan yang sering kali menghabiskan 80% dari proses data mining. Dengan Oracle9i Data Mining, semua data buatan dapat diolah dengan menggunakan manipulasi SQL standar dan fungsi-fungsi yang ada dalam Oracle9i Data Mining.

Oracle9i Data Mining menerima tabel transaksional dan non-transaksional. Secara internal, Oracle9i Data Mining mempergunakan tabel transaksional dan secara otomatis menangani konversi yang dibutuhkan.

Nomor Kasus	Umur	Pendapatan	Jenis Kelamin
100	20	1000	Pria
200	30	2000	Wanita

Tabel 2.01 Non-Transaksional, format data dengan satu record

Nomor Kasus	Atribut	Nilai
100	Umur	20
100	Pendapatan	1000
100	Jenis Kelamin	Pria
Nomor Kasus	Atribut	Nilai
200	Umur	30
200	Pendapatan	2000
200	Jenis Kelamin	Wanita

Tabel 2.02 Format data transaksional

Oracle9i Data Mining mendukung *binning*, proses koleksi nilai-nilai ke dalam grup yang memiliki arti. Hal ini mengijinkan programmer untuk menempatkan data untuk tujuan pembuatan model untuk masalah yang lebih spesifik. Contohnya, pada permasalahan keamanan berkendara, nilai usia dapat digolongkan ke dalam rentang nilai 0-16, 17-21, dan diatas 21. Untuk permasalahan asuransi, usia digolongkan ke dalam 0 sampai 35, 35-55, dan lebih dari 55.

2.2.2 Algoritma Oracle9i Data Mining

Oracle9i Data Mining menyediakan akses program ke dalam algoritma data mining yang menempel pada Oracle9i Database melalui sebuah API berbasiskan Java. Algoritma data mining adalah teknik pembelajaran terhadap analisa data untuk masalah yang sudah digolongkan secara khusus. Algoritma-algoritma yang berbeda sesuai untuk analisa yang berbeda. Oracle9i Data Mining menyediakan dua algoritma, 1) Naive Bayes untuk klasifikasi dan prediksi, 2) Kaidah Asosiasi untuk menemukan pola dari kejadian yang berhubungan. Keduanya, menyelesaikan masalah yang sering terjadi dalam proses bisnis.

2.3 Klasifikasi dan Prediksi dengan Naive Bayes

Naive Bayes adalah teknik pembelajaran yang terarah untuk klasifikasi dan prediksi yang membangun model untuk melakukan prediksi probabilitas dari pengeluaran yang spesifik. Naive Bayes menggunakan data historis untuk menemukan pola dan hubungan serta kemudian membuat prediksi-prediksi. Oracle9i Data Mining menyediakan pengembang aplikasi untuk membangun model data mining untuk mengklasifikasi dan memprediksi berbagai macam keperluan, misalnya :

- a. mengidentifikasi pelanggan yang suka membeli suatu produk,
- b. mengidentifikasi pelanggan yang suka memilih,
- c. melakukan prediksi kemungkinan sebuah bagian menjadi tidak efektif lagi.

Algoritma Naive Bayes dapat melakukan prediksi keluaran dalam bentuk biner atau multikelas [OR3-01]. Pada permasalahan biner, setiap record akan atau tidak akan menunjukkan tujuan yang sudah dimodelkan. Misalnya, sebuah model

dapat dibangun untuk memprediksi apakah seorang pelanggan suka memilih atau lebih senang royal dalam melakukan pembelian. Untuk multikelas, contohnya, model dapat dibangun untuk melakukan prediksi kelas dari jasa yang dipilih dalam setiap kesempatan.

Contoh model biner :

Pertanyaan : Apakah pelanggan ini bisa menjadi pelanggan yang mampu memberikan keuntungan yang tinggi ?

Jawab : Ya dengan kemungkinan 85%.

Contoh model multikelas :

Pertanyaan : Yang mana dari lima lapis pelanggan berikut yang cocok untuk pelanggan ini – Meningkat, stabil, rusak, turun, atau tidak signifikan

Jawab : Stabil dengan kemungkinan 55%.

2.4 Menemukan Asosiasi dengan Kaidah Asosiasi

Kaidah Asosiasi menemukan “hubungan” yang tersembunyi dalam basisdata. Analisa asosiasi atau biasa disebut *unsupervised learning*, sering digunakan untuk menemukan tempat strategis (misal analisa berbasis pasar) dari produk-produk yang berhubungan dengan pelanggan, seperti susu dan cereal dihubungkan dengan pisang. Kaidah asosiasi dari Oracle9i Data Mining dapat digunakan untuk mengenali hubungan antar item atau kejadian di dalam bermacam masalah bisnis, seperti :

- a. atribut pasien dan obat yang mana yang berhubungan dengan pengeluaran,
- b. item atau produk mana yang seseorang paling sukai.

Algoritma kaidah asosiasi menghasilkan sebuah himpunan pasangan anteseden dan konsekuensi dalam bentuk A implikasi B dengan probabilitas $n\%$. Teknik pemodelan ini mengijinkan pengguna untuk menentukan hubungan antar item atau kejadian. Oracle9i Data Mining juga mengikutsertakan perlengkapan untuk menganalisa dan membuat laporan dari kaidah asosiasi. Asosiasi atau “aturan” yang ditemukan sangat berguna untuk membuat desain promosi spesial, kemasan produk, atau tataletak barang dagangan.

Oracle9i Data Mining dapat melakukan otomatisasi ekstraksi dan integrasi dari wawasan dan prediksi baru ke dalam bermacam aplikasi bisnis, termasuk pusat layanan, situs web, ATM, manajemen sumberdaya enterprise, dan aplikasi operasional dan perencanaan yang lain. Otomatisasi fungsi-fungsi data mining yang disediakan oleh Oracle9i Data Mining dengan menggunakan Java berbasis API. Pengembang aplikasi dapat melakukan kontrol terhadap semua aspek data mining -- pengembang dapat memasukkan konfigurasi yang kompleks untuk pengguna yang lebih tinggi tingkatannya atau benar-benar menyediakan otomatisasi untuk para pengguna bisnis.

Secara teoritis permasalahan dalam Kaidah Asosiasi dapat dibagi menjadi dua sub masalah [OR1-02]:

- a. Menemukan semua kombinasi dari item-item, disebut *frequent itemset*, yang nilai *support*-nya lebih dari minimal *support*.
- b. Menggunakan *frequent itemset* untuk menghasilkan aturan yang diinginkan. Misal ABCD dan AB adalah *frequent*, kemudian aturan AB mengimplikasikan CD didapat jika rasio dari *support* (ABCD) terhadap *support* (AB) adalah

lebih besar *minimum confidence*. Sebagai catatan aturan memiliki *minimum support*, sebab ABCD adalah *frequent*.

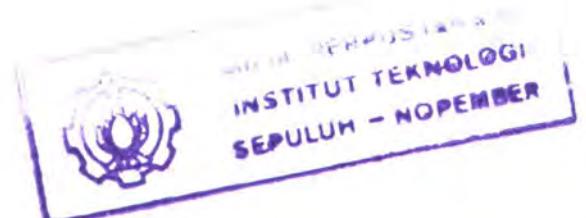
Algoritma Apriori untuk menemukan *frequent itemset* melewakan banyak data. Pada proses ke- k , algoritma ini menemukan semua *itemset* yang memiliki item sebanyak k , yang disebut k -*itemset*. Setiap proses terdiri dari dua fase. Misal F_k menunjukkan himpunan dari *frequent k-itemset*, dan C_k adalah himpunan kandidat k -*itemset*. Pertama, fase pembuatan kandidat dengan himpunan dari semua *frequent (k-1) itemset*, F_{k-1} ditemukan pada proses ke $(k-1)$ digunakan untuk membuat kandidat *itemset* C_k . Prosedur pembuatan kandidat memastikan bahwa C_k adalah superset dari semua *frequent k-itemset*. Kemudian data dibaca pada fase perhitungan *support*. Untuk setiap transaksi, kandidat di dalam C_k yang berisikan transaksi dibuat dengan menggunakan struktur data *hash-tree* dan nilai *support* dinaikkan. Pada akhir proses, C_k dibaca untuk menentukan kandidat yang sering muncul, sesuai dengan F_k . Algoritma berhenti saat F_k dan C_{k-1} kosong.

Pada Oracle9i Data Mining dipergunakan implementasi yang berbasiskan SQL. Pembuatan kandidat dan perhitungan nilai *support* diimplementasikan dengan mempergunakan SQL. Oracle9i Data Mining tidak mempergunakan struktur data memori secara khusus.

2.5 Obyek-obyek dan Fungsi dalam Oracle9i Data Mining

Kemampuan Data Mining dari tiap fungsi :

Function	Build	Tes	Compute Lift	Apply (Score)	Import PMML	Export PMML
Classification	X	X	X	X	NB	NB



Function	Build	Tes	Compute Lift	Apply (Score)	Import PMML	Export PMML
Clustering	X			X		
Association Rule	X				X	X
Attribute Importance	X					

Tabel 2.03 Kemampuan Data Mining

Objek-objek dan fungsionalitas yang ada dalam ODM [OR1-02]:

1. Physical Data Spesification

Sebuah objek *Physical Data Spesification* menentukan karakteristik data yang digunakan contohnya apakah data tersebut data *transactional* atau *nontransactional* dan peran dari macam-macam kolom data. Secara fisik data ODM harus berada dalam dua format berikut ini :

a. *transactional*

Pada format data transaksional, setiap kasus disimpan pada multiple record dalam sebuah tabel dengan kolom-kolom : *sequenceID*, *attribute_name*, *value* (nama-nama ini bisa didefinisikan sendiri oleh user).

- *sequenceID* adalah sebuah nilai integer yang berhubungan dengan *multiple record* pada sebuah data transaksional.
- *Attribute_name* adalah sebuah string yang berisi nama atribut.
- *Value* adalah sebuah integer yang merepresentasikan nilai dari atribut

b. *Nontransactional*

Pada format data *nonTransactional*, setiap kasus disimpan pada satu baris dalam sebuah tabel. Data non transaksional tidak membutuhkan sebuah kolom kunci secara unik untuk mengidentifikasi setiap *record*. Namun, sebuah atribut kunci direkomendasikan untuk menghubungkan kasus-kasus yang ada dengan hasil penilaian untuk *supervised learning*. Format ini biasa disebut *single-record case*.

2. Mining Function Setting

Sebuah *mining function setting* (MFS) berisi parameter tingkat tinggi untuk membangun sebuah model datamining. Setiap objek MFS berisi :

- parameter-parameter yang menunjukkan fungsi ODM,
- spesifikasi data lojik,
- spesifikasi data yang digunakan.

ODM terus-menerus mendukung MFS secara independen, dinamai entitas dalam DMS.

Function	Parameter	Default
Classification	CostMatrix	NULL
	Priors	NULL
Clustering	MaxNumberOfClusters	20
Association Rules	Minimum Support	0.1
	Minimum Confidence	0.1
Attribute Importance	Maximum Rule Length	2
	None	

Tabel Parameter pada ODM sesuai dengan fungsi

3. Mining Algorithm Setting

Sebuah objek *mining algorithm setting* berisi parameter-parameter yang berhubungan dengan algoritma tertentu untuk membangun sebuah model. Desain ODM, yang memisahkan *mining algorithm settings* dari *mining function setting*, mengijinkan *data miner* awam menggunakan ODM secara efektif, sedangkan *data miner* yang sudah ahli bisa mempergunakan sesuai dengan kebutuhan mereka.

4. Logical Data Specification

Sebuah objek *logical data specification* (LDS) adalah sebuah himpunan atribut instan yang menggambarkan sifat logik dari data yang digunakan sebagai input untuk membangun model. Setiap atribut *data mining* yang ada dalam LDS harus memiliki nama yang unik. Selain disimpan di DMS, setiap MFS memiliki salinan LDS, jika saat referensi dibagi pemakaiannya dalam proses API dari klien.

5. Mining Attributes

Mining atribut adalah sebuah konsep logik yang mendeskripsikan domain dari data yang digunakan sebagai input saat operasi ODM berupa numerik atau kategori. Sebuah *mining attribute* menunjukkan nama, tipe data, dan tipe atribut (numerik atau kategori).

6. Data Usage Specification

Sebuah objek *data usage specification* (DUS) menunjukkan bagaimana atribut-atribut yang ada dalam LDS digunakan untuk membangun sebuah model. Sebuah DUS berisi sekurangnya sebuah data yang digunakan untuk setiap atribut mining. Jika tidak ada data yang digunakan,

data *default* yang digunakan, mengimplikasikan bahwa atribut digunakan dalam pembentukan model.

DUS menunjukkan :

- apakah sebuah atribut adalah *active*(digunakan dalam proses pembuatan model), *inactive* (tidak dipergunakan), *supplementary* (sebuah atribut yang digunakan dengan hasil output pada saat *scoring* tetapi tidak dipergunakan pada saat pembuatan model).
- Apakah atribut tersebut adalah target *learning*

7. Mining Model

Sebuah objek *mining model* adalah hasil dari pembuatan sebuah model berdasarkan objek MFS. Representasi dari model bergantung pada algoritma yang digunakan atau dipilih oleh DMS. Beberapa model dapat dipergunakan untuk inspeksi langsung, misalnya melakukan pengujian terhadap aturan-aturan yang dihasilkan oleh Kaidah Asosiasi.

8. Mining Results

Objek *mining result* berisi produk akhir dari salah satu tugas mining berikut ini : pembuatan, pengujian, perhitungan, atau pengaplikasian. ODM mendukung penyimpanan model secara independen, sebagai entitas dalam DMS.

- sebuah hasil pembuatan berisi model secara detil. Hasil ini menyajikan nama fungsi dan algoritma dari model yang bersangkutan

- sebuah hasil pengaplikasian berisi tabel tujuan (nama skema dan nama tabel)
- sebuah hasil pengujian, untuk model klasifikasi berisi akurasi model dan mereferensi terhadap matrik *confusion*.

9. Confusion Matrix

Sebuah *confusion matrix* menyajikan sebuah pemahaman mudah terhadap akurasi model dan tipe kesalahan yang dibuat saat *scoring*. Matrik ini adalah hasil dari sebuah tugas pengujian untuk model klasifikasi.

10. Mining Apply Output

Instan *mining apply output* berisi beberapa item yang mengijinkan user untuk menyesuaikan hasil dari sebuah model. Output bisa berupa :

- data skalar yang dilewatkan ke output dari tabel input, contohnya atribut kunci
- nilai komputasi dari pengaplikasian itu sendiri, seperti skor dan probabilitas
- untuk input data transaksional, *sequence ID* diasosiasikan dengan kasus yang diberikan

Sebuah instan dari *miningapplyoutput* adalah sebuah spesifikasi dari data yang dimasukkan dalam pemakaian output (sebuah tabel atau output) yang dibuat sebagai hasil dari pemakaian operasi mining. Kolom atau atribut dalam pemakaian output dideskripsikan sebagai kombinasi dari banyak objek *ApplyContentItems*. Setiap item bisa berupa :

- Atribut asal

tabel output tempat pengaplikasian model bisa berisikan kolom hasil pengkopian langsung dari tabel input. Hal ini disebut atribut asal, dan setiap atribut asal direpresentasikan dengan sebuah instan dari *ApplySourceAttributeItem*. Atribut asal dapat digunakan untuk mengidentifikasi kasus individual pada pengaplikasian output, misal menghubungan sebuah kunci dengan setiap item output. Atribut asal pada tabel output tidak boleh lebih dari jumlahnya lebih dari 997.

- prediksi majemuk yang berbasiskan probabilitas

Sebuah instan *ApplyMultipleScoringItem* menghasilkan n prediksi bawah atau atas urut berdasarkan probabilitas dari prediksi, dengan n bernilai dari 1 sama dengan jumlah nilai tujuan. Suatu item menghasilkan dua kolom di dalam outputnya : prediksi dan probabilitas yang masing-masing diberi nama oleh penggunanya. Minimal terdapat sebuah *ApplyMultipleScoringItem* pada obyek *MiningApplyOutput*.

- Multipel prediksi yang berbasiskan pada nilai tujuan

ApplyTargetProbability menghasilkan prediksi untuk suatu nilai tujuan. Setiap nilai tujuan tersebut harus memiliki sekurangnya satu nilai tujuan asli yang dipergunakan untuk membuat model.

BAB III

PERANCANGAN PERANGKAT LUNAK

BAB III

PERANCANGAN PERANGKAT LUNAK

Oracle9i tidak menyediakan aplikasi yang berdiri sendiri yang memberikan kemudahan pada pengguna mengakses fungsi-fungsi yang tersedia di dalam Oracle9i Data Mining, termasuk juga untuk mengakses algoritma Kaidah Asosiasi dan Naive Bayes. Untuk itu perangkat lunak yang dibuat nantinya mempermudah akses terhadap fungsi Kaidah Asosiasi dan Naive Bayes dalam Oracle9i Data Mining.

Bab ini membahas perancangan perangkat lunak pengekstraksi model Kaidah Asosiasi dan Naive Bayes ke dalam bentuk PMML dan dari bentuk PMML. Akan ada dua bahasan besar dalam bab ini yaitu perancangan perangkat lunak yang melakukan proses pemodelan sampai dengan mengeluarkan output dalam bentuk PMML dan membaca model dalam bentuk PMML ke dalam bentuk model yang diterima oleh Oracle9i Data Mining.

3.1 Desain Data

Desain data ini bertujuan untuk menentukan kebutuhan data dari perangkat lunak yang dibuat termasuk format data yang harus dibuat. Perancangan data tersebut dikategorikan sebagai data masukan, data proses, dan data keluaran.

3.1.1 Data Masukan

Secara umum perangkat lunak yang dibentuk harus memiliki data masukan berupa koneksi ke skema Oracle9i Data Mining. Oracle9i membuat sebuah skema basisdata untuk menyimpan fungsi-fungsi data mining. Skema tersebut diberi nama odm.

Agar tidak selalu membuat koneksi baru saat melakukan proses maka diperlukan sebuah otentikasi yang disediakan oleh perangkat lunak. Otentikasi itu untuk masuk ke dalam skema odm. Untuk masuk ke dalam skema Oracle9i Data Mining diperlukan masukan sebagai berikut :

- a. nama pengguna, dalam hal ini yang memiliki hak akses ke dalam skema odm,
- b. kata sandi yang sesuai dengan kata sandi yang dimiliki atau digunakan oleh nama pengguna,
- c. nama *service oracle*,
- d. nama komputer tempat *service oracle* berjalan.

Semua masukan di atas memiliki tipe data *string*.

Untuk ekstraksi model, perangkat lunak nantinya terbagi dua, yaitu untuk ekstraksi model yang mendukung Kaidah Asosiasi dan yang mendukung Naive Bayes. Untuk perangkat lunak yang mendukung Kaidah Asosiasi memiliki tiga tahap.

1. Tahap Pembuatan Model Kaidah Asosiasi

Pada tahap ini perangkat lunak membutuhkan masukan antara lain :

- a. nama skema tempat data/tabel yang menjadi inputan algoritma, bertipe data *string*,

- b. nama tabel bertipe *string*,
- c. tipe tabel, bertipe data *string*,

Tipe tabel ini nantinya hanya ada dua pilihan sesuai dengan yang didukung oleh Oracle9i Data Mining yaitu : *transactional* dan *nonTransactional*.

- a. masukan *sequence ID*, bertipe data *string*,
- b. masukan *attribute name*, bertipe data *string*,
- c. masukan *value name*, bertipe data *string*,

Masukan *sequence ID*, *attribute name*, dan *value name* tersebut dibutuhkan jika tipe tabel yang dipilih adalah *transactional*.

- d. masukan minimum support, bertipe float,
- e. masukan minimum confidence, bertipe float,
- f. masukan max rule length, bertipe float,
- g. masukan status data yang diakses, bertipe *string*,

Status data yang diakses memiliki dua pilihan, yaitu : *discretized* dan *prepared*.

- h. masukan nama dari setting fungsi mining, bertipe *string*,
- i. masukan nama tugas mining, bertipe *string*,
- j. masukan nama model, bertipe *string*,
- k. masukan status *cleanUp* proses, bertipe *string*.

Masukan *cleanUp* memiliki tiga pilihan, yaitu : *cleanUpBeforeExecution*, *cleanUpOnly*, dan *noCleanUp*.

2. Tahap Pengisian Model PMML

Pada tahap ini dibutuhkan masukan konfigurasi untuk mengeksport model standar Oracle9i Data Mining ke dalam bentuk PMML. Untuk itu diperlukan masukan sebagai berikut :

- a. nama model kaidah asosiasi yang dieksport dalam bentuk PMML, tentunya sama dengan masukan pada tahap pertama, bertipe *string*,
- b. nama model hasil eksport, bertipe *string*,
- c. nama tabel tempat hasil eksport disimpan, bertipe *string*,
- d. nama skema tabel tempat hasil eksport disimpan, bertipe *string*,
- e. nama kolumn hasil eksport, bertipe *string*,
- f. masukan status *cleanUp* proses, bertipe *string*.

Masukan *cleanUp* memiliki tiga pilihan, yaitu :
cleanUpBeforeExecution, *cleanUpOnly*, dan *noCleanUp*.

3. Tahap penentuan letak file PMML disimpan

Pada tahap ini dibutuhkan masukan untuk menentukan tempat file PMML hasil eksport disimpan.

Untuk perangkat lunak yang mendukung Naive Bayes juga memiliki tiga tahap masukan, yaitu :

1. Tahap pembuatan model Naive Bayes

Pada tahap ini perangkat lunak membutuhkan masukan antara lain :

- a. nama skema tempat tabel yang menjadi masukan proses, bertipe *string*,
- b. nama tabel yang menjadi masukan proses, bertipe *string*,
- c. tipe transaksi, bertipe *string*,

Tipe transaksi pada Naive Bayes ada dua, yaitu : *transactional* dan *nonTransactional*,

- a. masukan *sequence ID*, bertipe data *string*,
- b. masukan *attribute name*, bertipe data *string*,
- c. masukan *value name*, bertipe data *string*,

Masukan *sequence ID*, *attribute name*, dan *value name* tersebut dibutuhkan jika tipe tabel yang dipilih adalah *transactional*.

- d. status data masukan, bertipe *string*,

Ada dua macam status data yang didukung oleh Naive Bayes, yaitu *unprepared* dan *discretized*.

- e. masukan nama setting fungsi, bertipe *string*,
- f. masukan atribut tambahan, bertipe *string*,
- g. masukan singleton threshold, bertipe float,
- h. masukan pairwise threshold, bertipe float,
- i. masukan nama atribut target, bertipe *string*,
- j. masukan nama *task*, bertipe *string*,
- k. masukan nama model keluaran, bertipe *string*,
- l. masukan status *cleanUp* proses, bertipe *string*.

Masukan *cleanUp* memiliki tiga pilihan, yaitu :

cleanUpBeforeExecution, *cleanUpOnly*, dan *noCleanUp*.

2. Tahap penentuan letak file PMML disimpan

Pada tahap ini dibutuhkan masukan untuk menentukan tempat file PMML hasil eksport disimpan.

Selain eksport dari model ke dalam format PMML, perangkat lunak nantinya juga bisa melakukan import model PMML ke dalam model yang diterima oleh Oracle9i Data Mining. Untuk import, perangkat lunak membutuhkan masukan berupa berkas dalam format PMML.

3.1.2 Data Proses

Data proses adalah data yang dibutuhkan pada proses aplikasi. Data proses ini digunakan untuk menyimpan informasi, yang digunakan untuk menjalankan proses ekstraksi data, yaitu proses pembuatan model, eksport ke bentuk PMML, dan import dari PMML. Detil data proses lebih dijelaskan pada saat desain sistem.

3.1.3 Data Keluaran

Sebagai perangkat lunak yang berbasiskan PMML, maka data keluaran untuk proses eksport adalah berkas dalam format PMML. Hal ini berlaku untuk kedua algoritma, yaitu Kaidah Asosiasi maupun Naive Bayes. Sedangkan untuk import, data keluaran adalah berupa model yang diterima oleh Oracle9i Data Mining, data keluaran ini disimpan dalam sebuah tabel tujuan.

Perangkat lunak yang dibentuk berbasiskan pada bahasa PMML, tetapi perangkat lunak juga mampu mengeluarkan data lain yang berhubungan dengan algoritma Kaidah Asosiasi dan Naive Bayes. Yang berhubungan dengan Kaidah Asosiasi, disediakan fitur untuk melihat besarnya *support* antara item-item yang dihitung. Sedangkan untuk Naive Bayes disediakan fitur untuk melihat matrik.

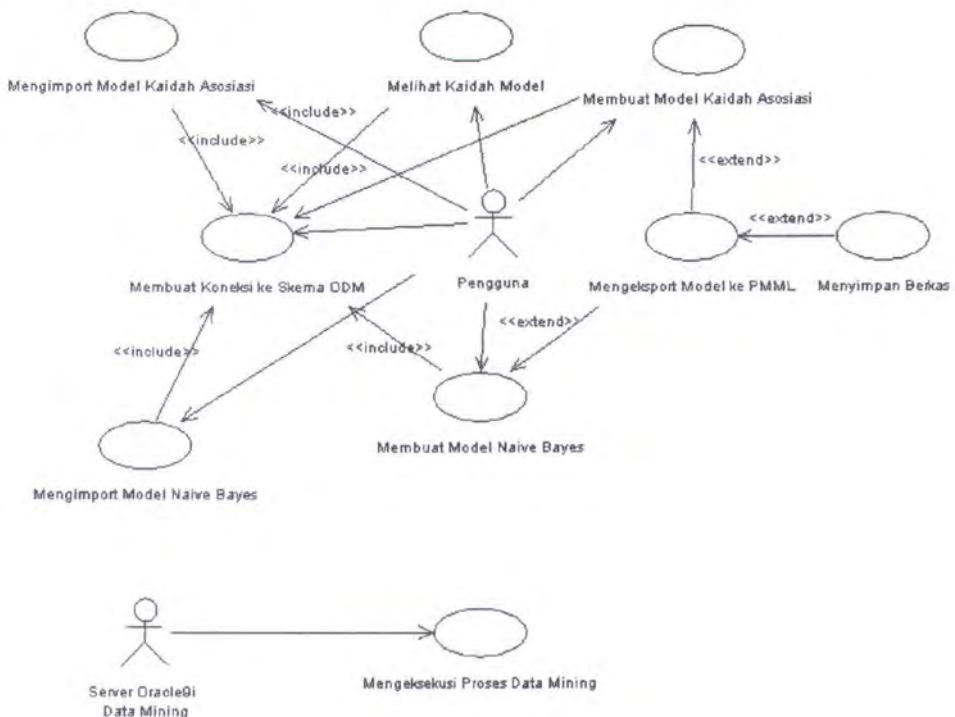
3.2 Desain Proses Pemodelan, Eksport, dan Import

Pada bagian desain proses ini dijelaskan desain proses yang dipergunakan oleh perangkat lunak. Ada tiga proses yang dibahas, yaitu proses pembuatan model, proses eksport model ke dalam bentuk PMML, dan proses import PMML ke dalam bentuk model Oracle9i Data Mining.

Perangkat lunak yang dibentuk didesain dengan menggunakan konsep *Unified Modelling Language* (UML).

1) Use Case

Berikut ini *use case* dari perangkat lunak yang dibuat :



Gambar 3.01 *Use Case* Perangkat Lunak

Pada gambar tampak ada dua aktor yang terlibat pada sistem ini, yaitu pengguna dan Server Oracle9i Data Mining. Aktor pengguna merupakan aktor

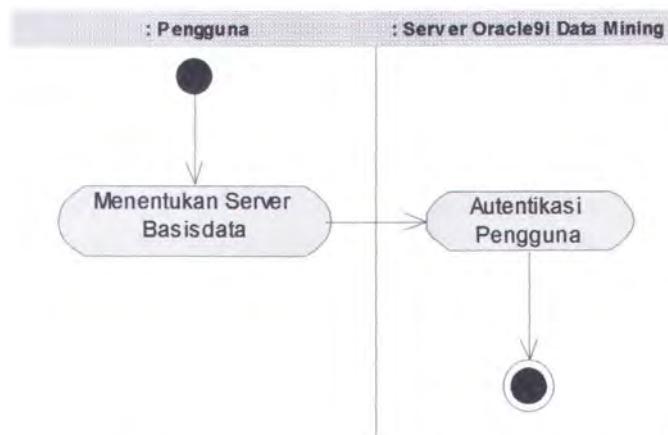
yang menggunakan perangkat lunak ekstraksi data. Aktor Server Oracle9i Data Mining bertugas untuk mengeksekusi tugas-tugas yang dikirim ke server.

2) *Activity Diagram*

Untuk menjelaskan maksud *use case* di atas diperlukan *activity diagram* dari masing-masing aktivitas yang dilakukan oleh aktor.

- membuat koneksi ke skema ODM

Aktivitas ini adalah aktivitas kunci pada perangkat lunak yang dibuat karena dibutuhkan oleh aktivitas yang lain. Pengguna memberikan konfigurasi basisdata tempat server Oracle9i Data Mining yang dituju. Sedangkan Oracle9i Data Mining melakukan otentifikasi.

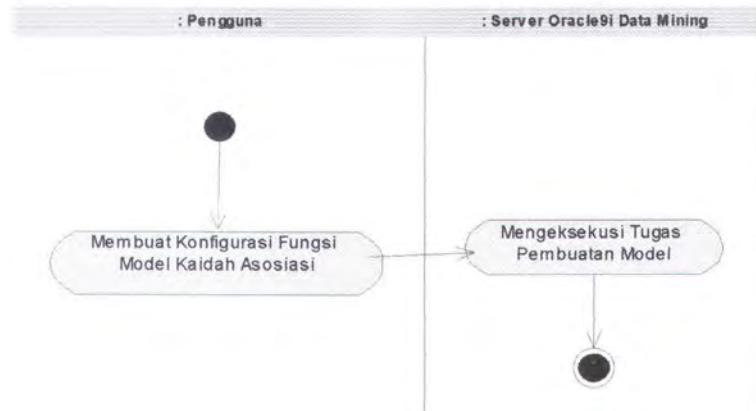


Gambar 3.02 *Activity Diagram – membuat koneksi ke skema ODM*

- membuat model Kaidah Asosiasi

Pengguna membuat konfigurasi fungsi model Kaidah Asosiasi. Konfigurasi di sini berupa konfigurasi yang dibutuhkan untuk membuat sebuah model Kaidah Asosiasi seperti minimum *support*, nama tabel, dan minimum *confidence*. Setelah konfigurasi diberikan, maka server Oracle9i

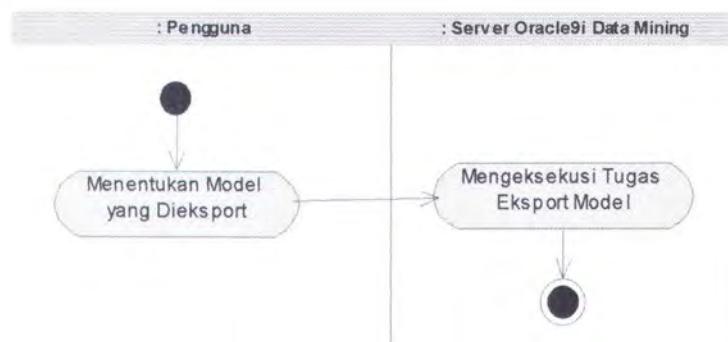
Data Mining mengeksekusi tugas pembuatan model berdasarkan konfigurasi yang diberikan.



Gambar 3.03 *Activity Diagram – membuat model Kaidah Asosiasi*

- mengeksport model ke PMML

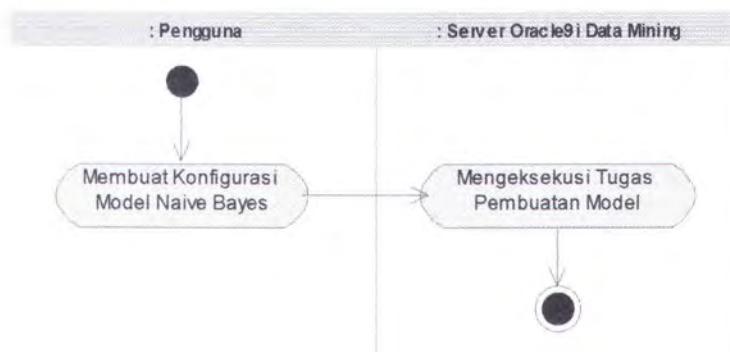
Untuk melakukan eksport model ke PMML pengguna harus menentukan nama model yang dieksport. Aktivitas ini sama untuk model Kaidah Asosiasi maupun model Naive Bayes. *Activity Diagramnya* terlihat seperti gambar berikut :



Gambar 3.04 *Activity Diagram – eksport model ke PMML*

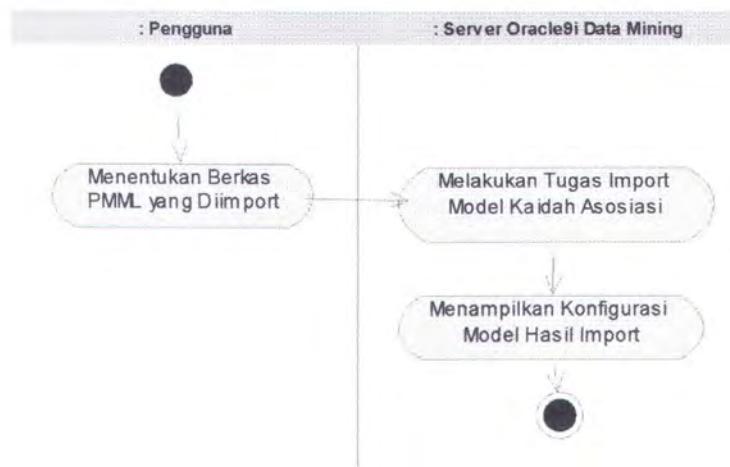
- membuat model Naive Bayes

Pengguna membuat konfigurasi fungsi model Naive Bayes. Konfigurasi di sini berupa konfigurasi yang dibutuhkan untuk membuat sebuah model Naive Bayes seperti *singleton threshold*, nama tabel, dan *pairwise threshold*. Setelah konfigurasi diberikan, maka server Oracle9i Data Mining mengeksekusi tugas pembuatan model berdasarkan konfigurasi yang diberikan.



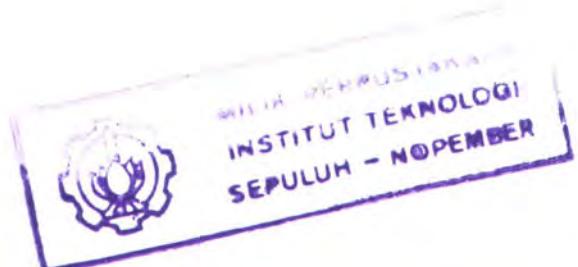
Gambar 3.05 *Activity Diagram* – pembuatan model Naive Bayes

- mengimport model Kaidah Asosiasi



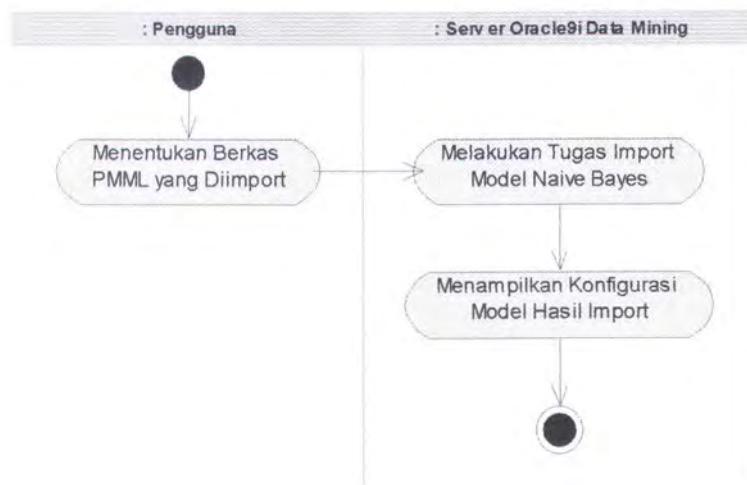
Gambar 3.06 *Activity Diagram* – mengimport model Kaidah Asosiasi

Untuk melakukan import model Kaidah Asosiasi, pengguna terlebih dahulu menentukan berkas PMML yang diimport. Server merespon berkas



yang diimport, jika berkas valid maka server memberikan respon konfigurasi model hasil import.

- **mengimport model Naive Bayes**

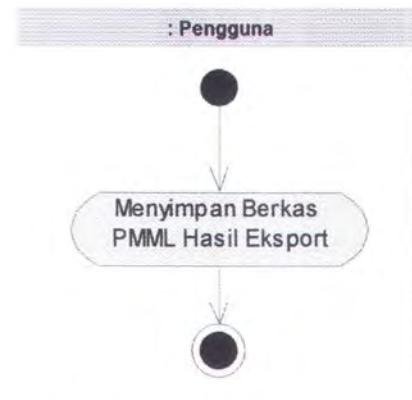


Gambar 3.07 Activity Diagram – mengimport model Naive Bayes

Untuk melakukan import model Naive Bayes, pengguna terlebih dahulu menentukan berkas PMML yang diimport. Server merespon berkas yang diimport, jika berkas valid maka server melakukan proses import dan memberikan respon konfigurasi model hasil import.

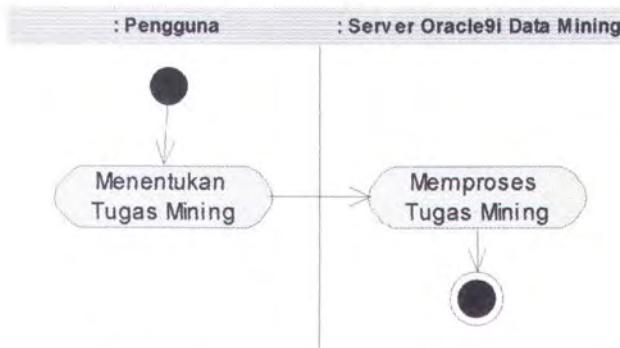
- **menyimpan berkas hasil eksport**

Setelah berhasil melakukan proses eksport, pengguna bisa menyimpan dokumen PMML ke media penyimpanan.



Gambar 3.08 *Activity Diagram – menyimpan berkas hasil eksport*

- mengeksekusi proses mining



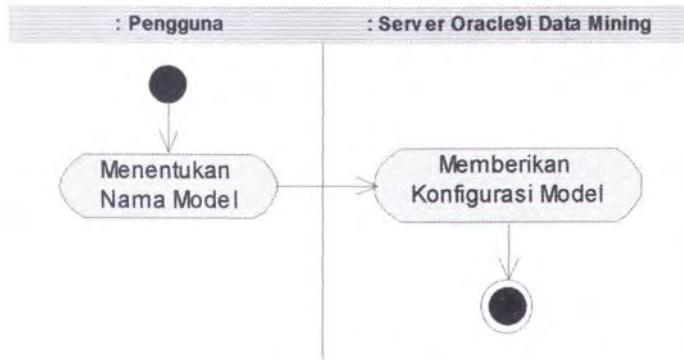
Gambar 3.09 *Activity Diagram – mengeksekusi proses mining*

Dalam aktivitas ini pengguna menentukan tugas mining yang harus dijalankan oleh server Oracle9i Data Mining. Tugas mining tersebut berupa membuat model, mengeksport model, mengimport model, dan melihat konfigurasi model.

- melihat kaidah model

Perangkat lunak menyediakan fitur untuk melihat kaidah dari sebuah model Kaidah Asosiasi. Pengguna memberikan nama model yang dilihat

kaidah yang terdapat di dalamnya, server Oracle9i Data Mining memberikan konfigurasi sesuai dengan model yang diinginkan.

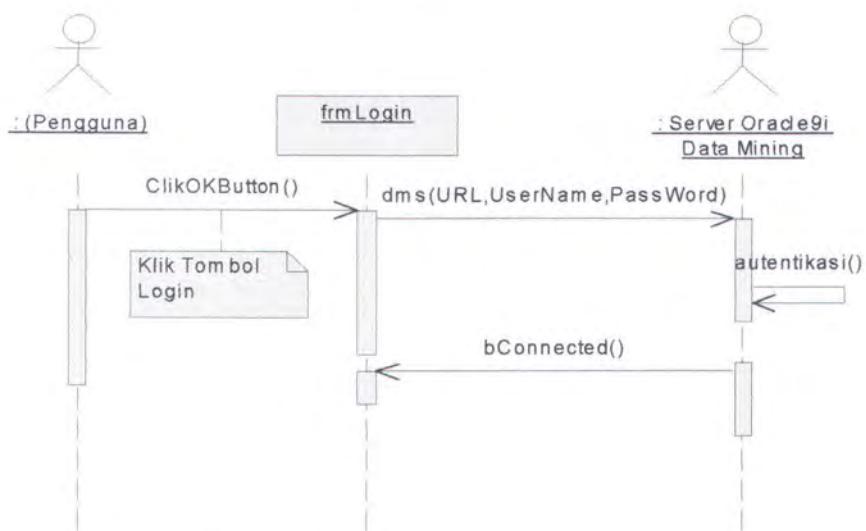


Gambar 3.10 *Activity Diagram – melihat kaidah model*

3) Sequence Diagram

Untuk menjelaskan *use case diagram* di atas diperlukan *sequence diagram* dari masing aktivitas. Berikut ini *sequence diagram* dari masing-masing aktivitas :

- membuat koneksi ke skema ODM



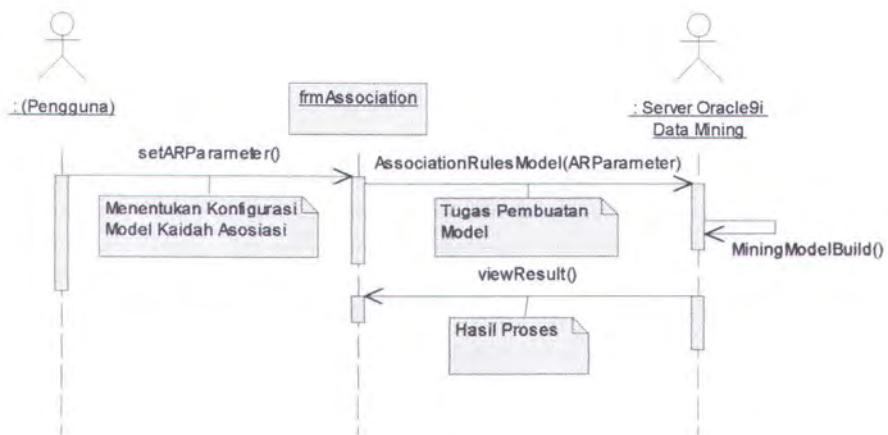
Gambar 3.11 *Sequence Diagram – membuat koneksi ke skema ODM*

Pengguna berhadapan dengan dialog login yang menentukan koneksi ke skema ODM. Dialog login mengirimkan parameter ke server Oracle9i Data Mining dan melakukan otentikasi.

- **membuat model Kaidah Asosiasi**

Untuk membuat model Kaidah Asosiasi pengguna berinteraksi dengan *frmAssociation*. Dialog ini menerima konfigurasi untuk membuat model Kaidah Asosiasi.

Sequence diagram dari aktivitas membuat model Kaidah Asosiasi tampak seperti berikut :

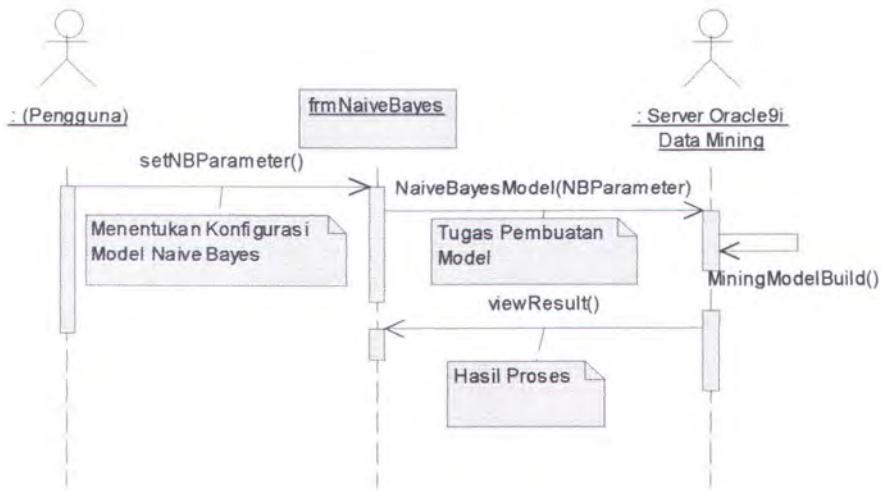


Gambar 3.12 *Sequence Diagram* – membuat model Kaidah Asosiasi

- **membuat model Naive Bayes**

Untuk membuat model Naive Bayes, pengguna berinteraksi dengan dialog *frmNaiveBayes*. Dialog ini menerima konfigurasi untuk membuat model Naive Bayes, kemudian mengirimkan ke server untuk dilakukan proses

Sequence diagram dari aktivitas ini tampak seperti berikut :

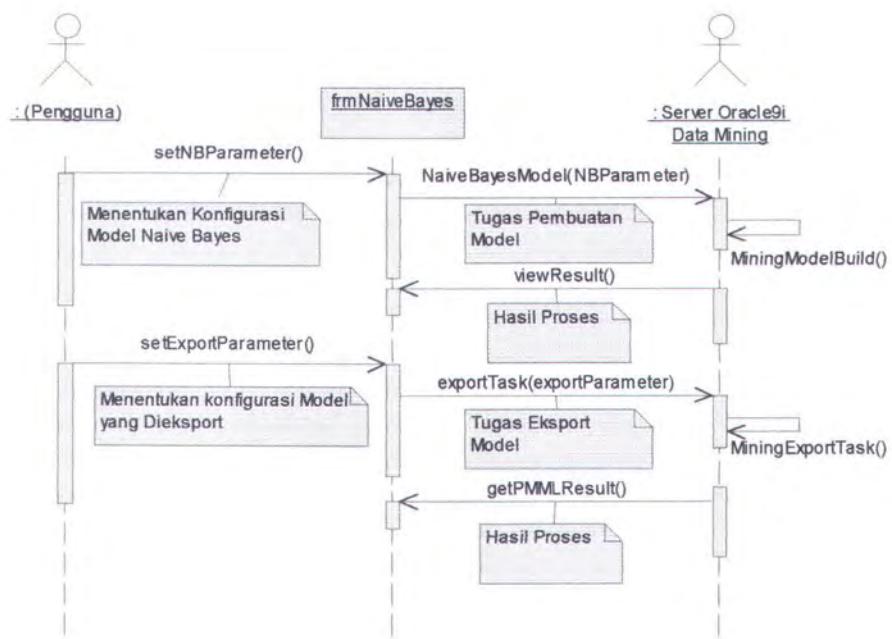


Gambar 3.13 *Activity Diagram – membuat model Naive Bayes*

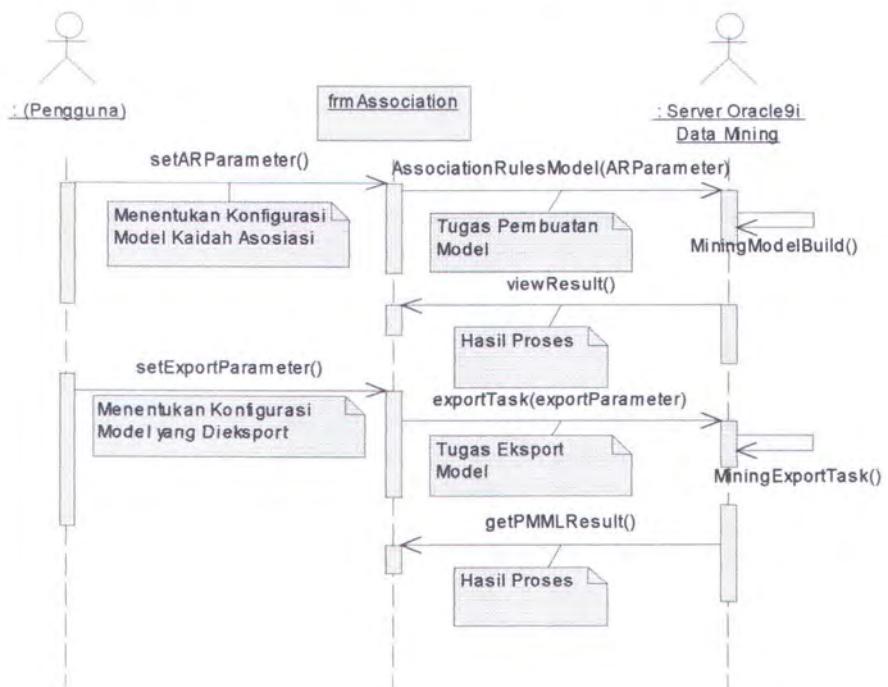
- **mengakses eksport model**

Untuk melakukan eksport model ini masing-masing model memiliki *sequence diagram* yang berbeda. Dari *sequence diagram* ini memperlihatkan bahwa untuk melakukan proses eksport harus didahului dengan proses pembuatan model. Untuk melakukan proses eksport model Kaidah Asosiasi harus didahului proses pembuatan model Kaidah Asosiasi dan juga proses eksport model Naive Bayes harus didahului proses pembuatan model Naive Bayes.

Sequence diagram dari masing-masing aktivitas tampak seperti berikut :



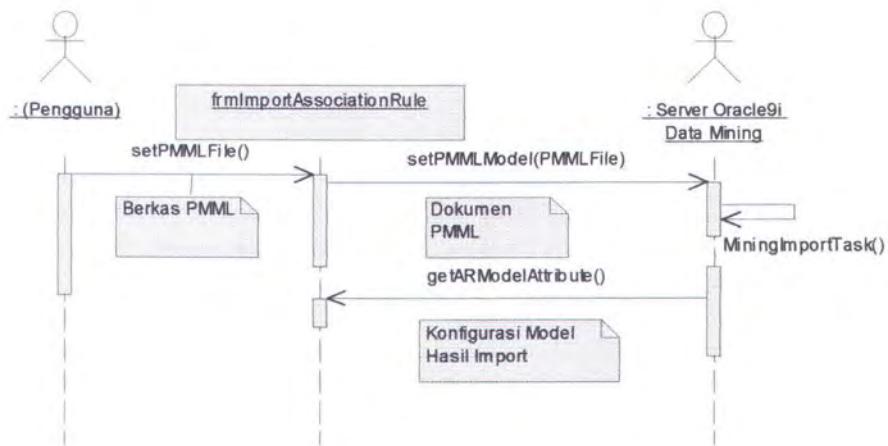
Gambar 3.14 Sequence Diagram – eksport model Naive Bayes



Gambar 3.15 Sequence Diagram – eksport Kaidah Asosiasi

- mengimport model Kaidah Asosiasi

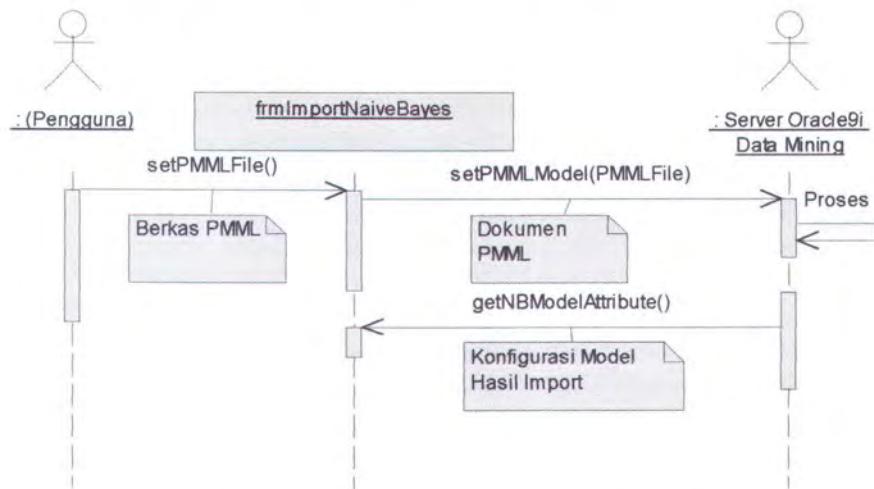
Sequence diagramnya tampak seperti berikut :



Gambar 3.16 Sequence Diagram – mengimport model Kaidah Asosiasi

- mengimport model Naive Bayes

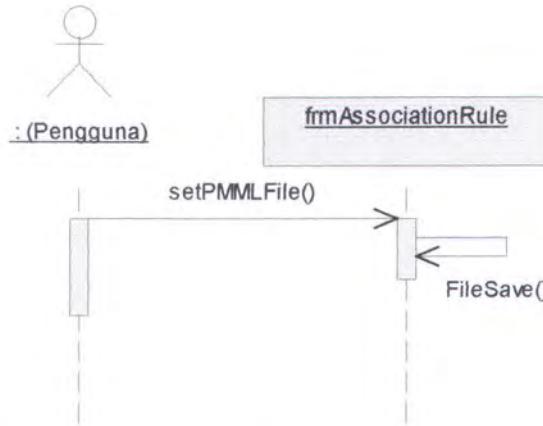
Sequence diagram-nya tampak seperti berikut :



Gambar 3.17 Sequence Diagram – mengimport model Naive Bayes

- menyimpan berkas hasil eksport

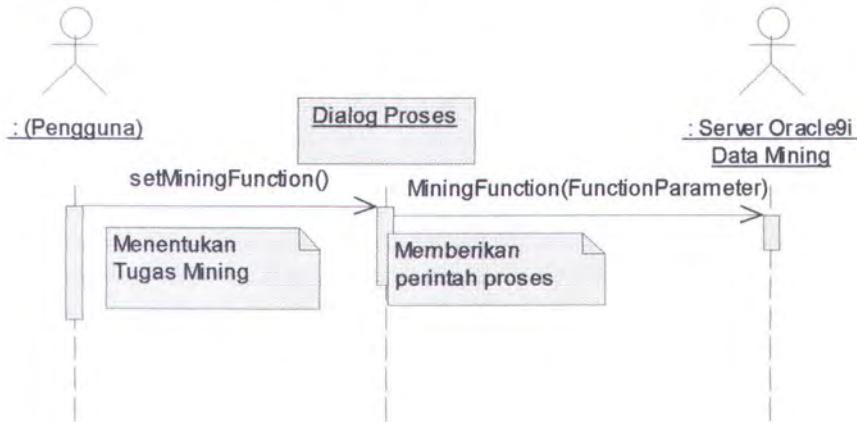
Sequence diagram berikut ini berlaku untuk semua model, baik Kaidah Asosiasi maupun Naive Bayes :



Gambar 3.18 *Sequence Diagram* - menyimpan berkas

- melakukan proses mining

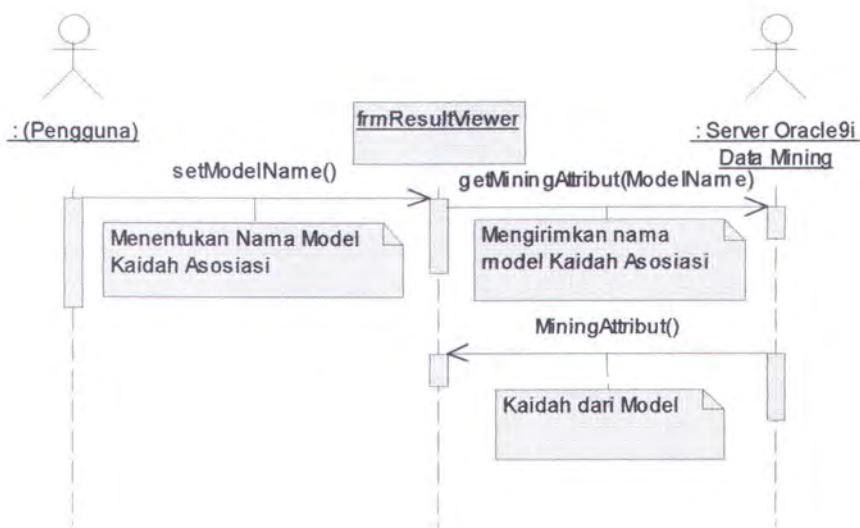
Sequence diagram dari aktivitas ini tampak seperti berikut :



Gambar 3.19 *Sequence Diagram* – melakukan proses mining

- melihat kaidah model

Sequence diagram dari aktivitas ini terlihat seperti berikut :



Gambar 3.20 *Sequence Diagram – melihat kaidah model*

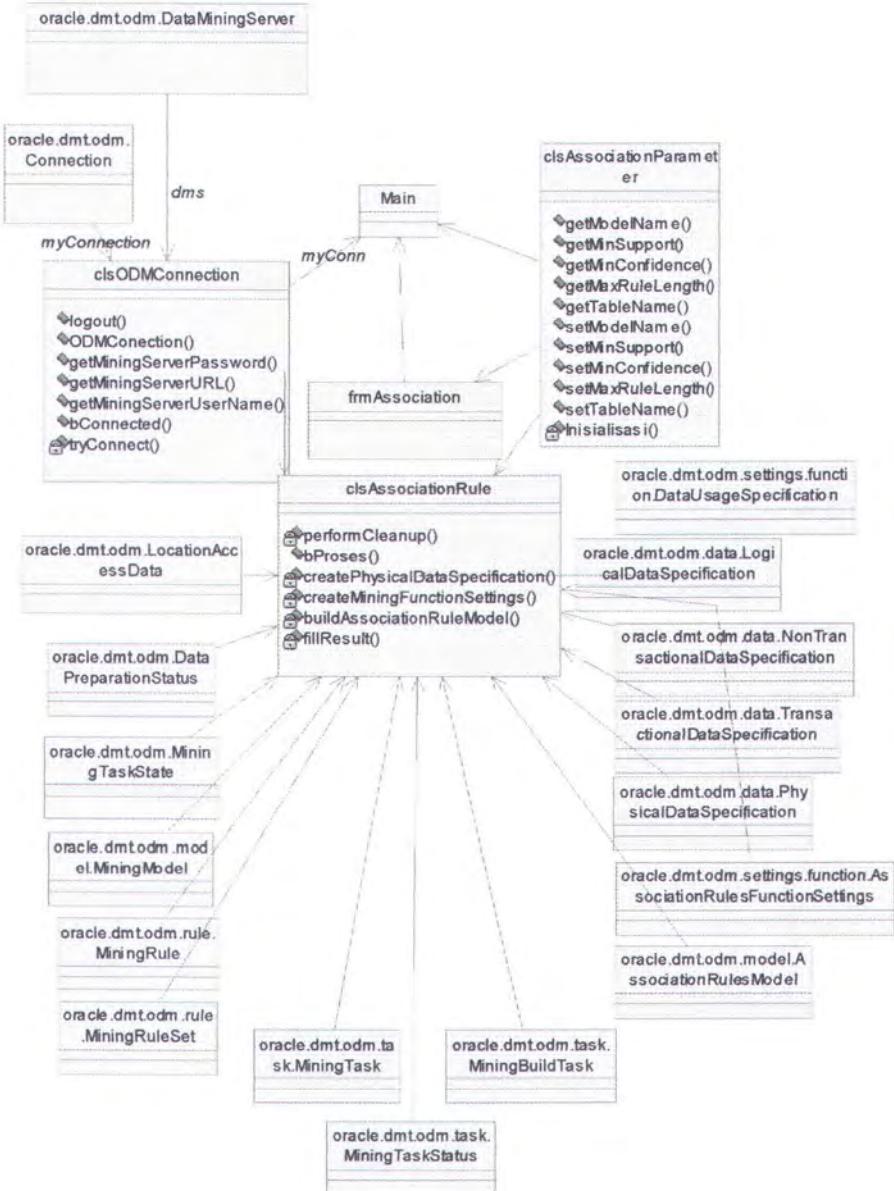
3.3 Desain Obyek

Pembuatan perangkat lunak ini berorientasikan obyek dengan bahasa Java sebagai implementasinya. Oleh karena itu diperlukan desain obyek untuk mengimplementasikan proses dan desain data yang telah diuraikan pada sub bab sebelumnya. Obyek-obyek yang dipakai direpresentasikan dalam bentuk *class-class* pada bahasa pemrograman Java.

Obyek-obyek nantinya terbagi menjadi beberapa bagian sesuai dengan aktivitas dari perangkat lunak, yaitu :

1) pembuatan model Kaidah Asosiasi

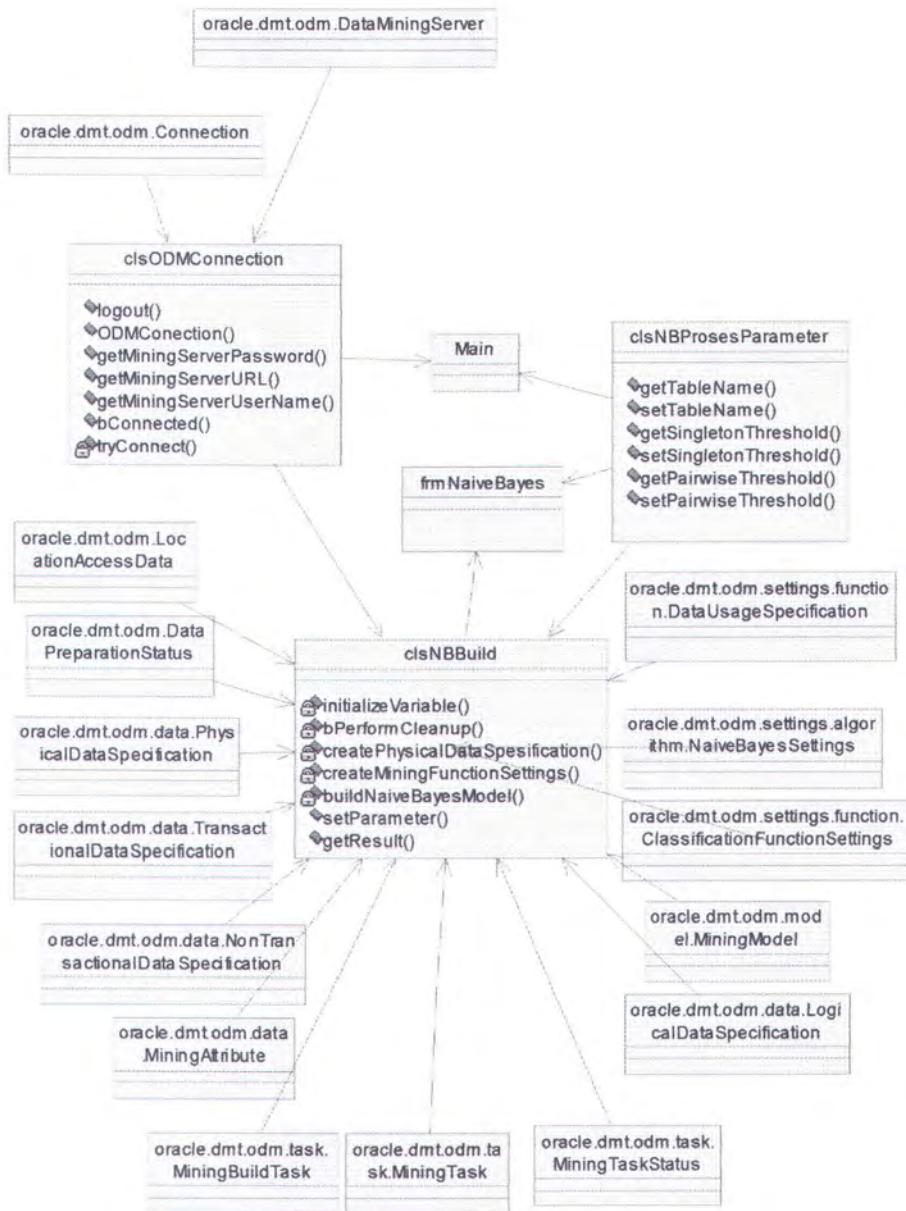
Untuk proses pembuatan model Kaidah Asosiasi memiliki desain obyek seperti berikut :



Gambar 3.21 *Class Diagram - Pembuatan Model Kaidah Asosiasi*

2) pembuatan model Naive Bayes

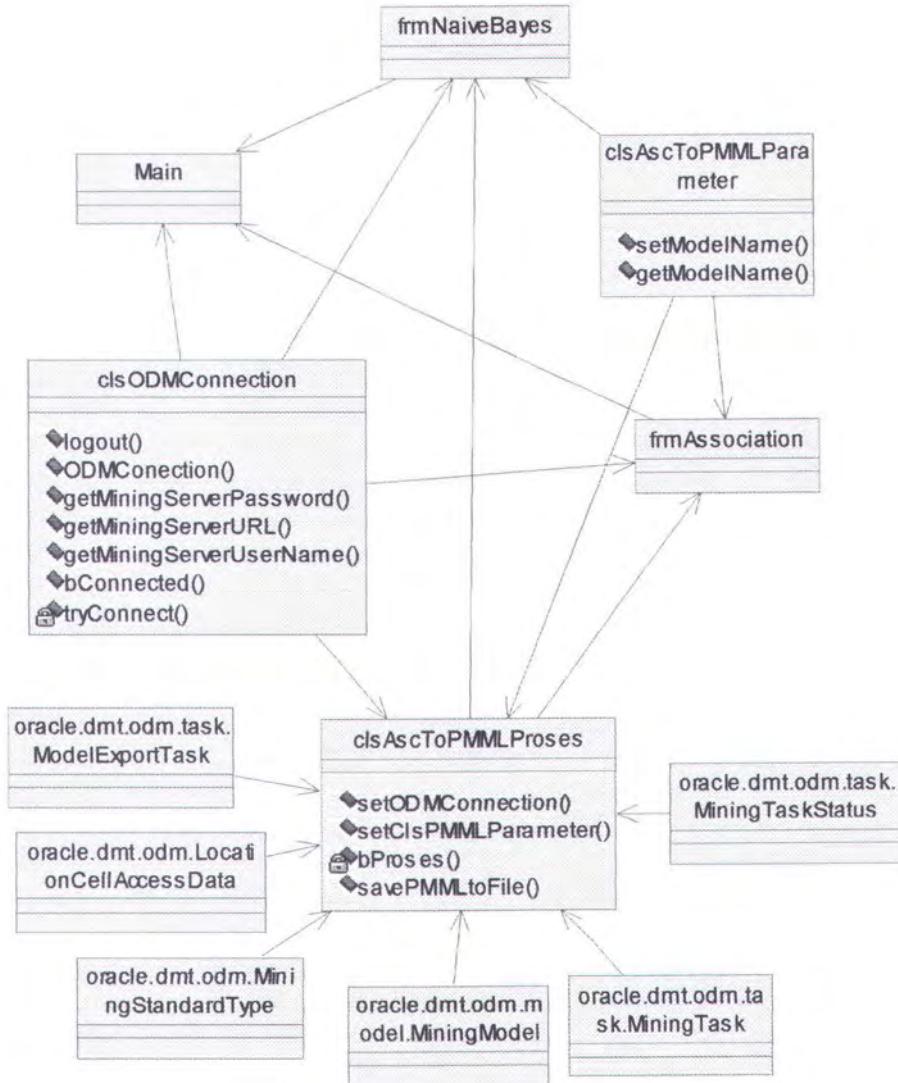
Untuk pembuatan model Naive Bayes melibatkan obyek-obyek sebagai berikut :



Gambar 3.22 *Class Diagram - Pembuatan Model Naive Bayes*

3) eksport model

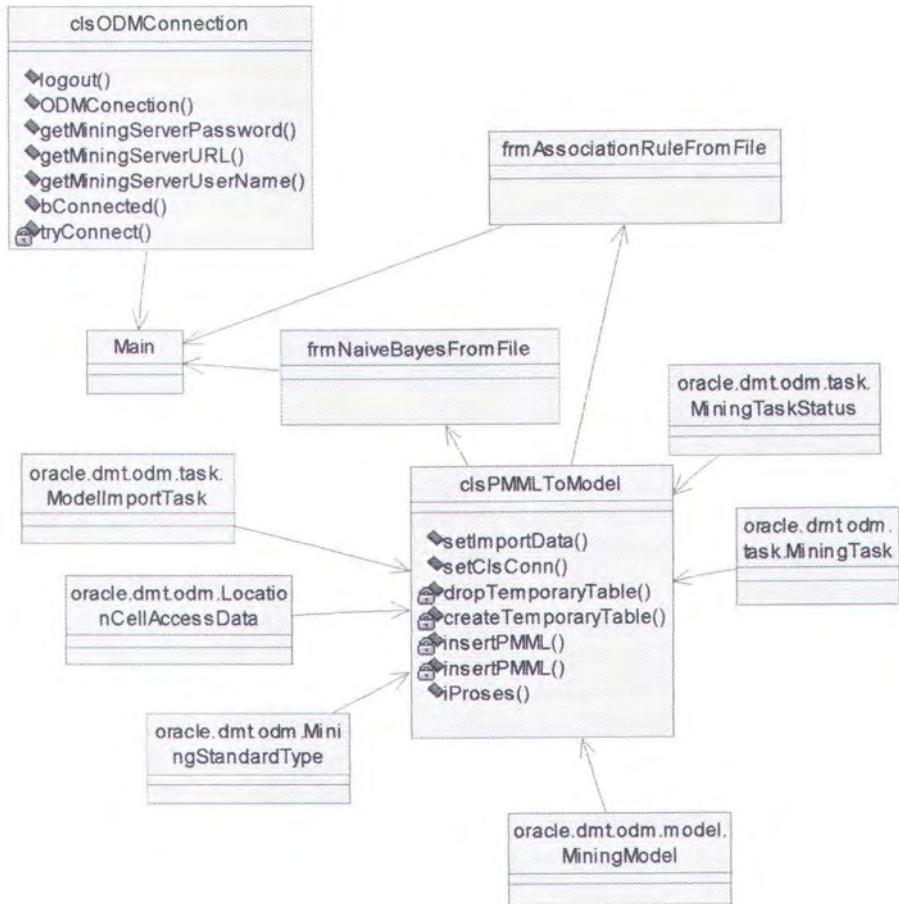
Untuk melakukan eksport model, baik model Kaidah Asosiasi maupun model Naive Bayes tampak seperti berikut :



Gambar 3.23 *Class Diagram - Eksport Model*

4) Import Model

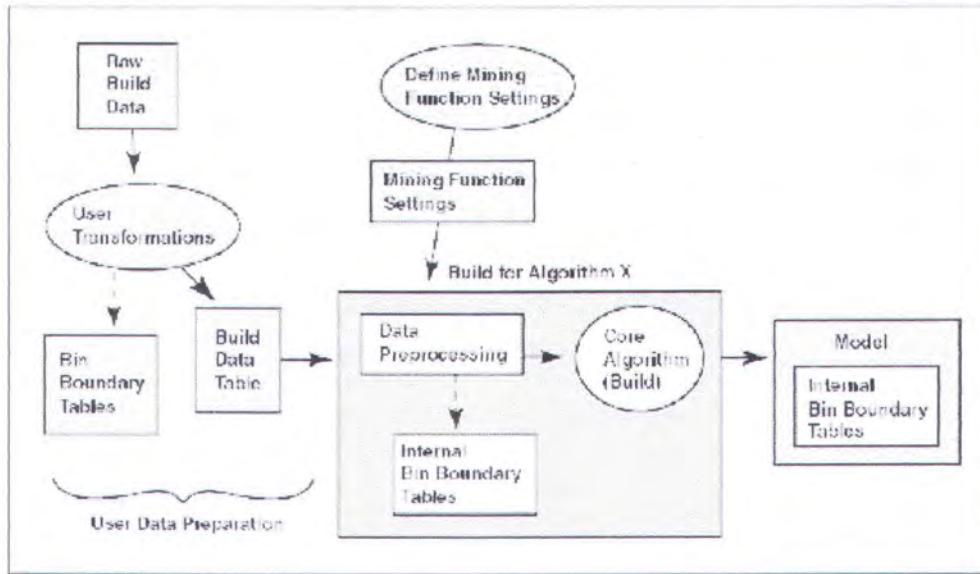
Untuk melakukan import model Kaidah Asosiasi maupun model Naive Bayes memakai desain *class diagram* seperti berikut :



Gambar 3.24 *Class Diagram – import model*

Class Main merupakan *class* utama. *Class* ini memanggil *class-class* yang lain sekaligus mengirimkan parameter koneksi yang dibutuhkan untuk melakukan proses.

Berikut ini gambaran umum tahap-tahap pada saat membuat model [OR1-02] :



Gambar 3.25 Tahap Pembuatan Model

Untuk membuat model harus dapat dipastikan hal-hal berikut :

- server yang dipilih sebagai tempat proses,
- letak data yang diproses dan cara proses terhadap data tersebut,
- tipe model yang dibentuk,
- fungsi yang dipilih,
- algoritma yang digunakan.

Berikut langkah umum yang dapat membantu memastikan kebenaran proses yang dibentuk :

- membuat koneksi ke DMS (data mining server),
- membuat sebuah obyek *PhysicalDataSpecification* untuk membuat data,
- membuat sebuah obyek *MiningFunctionSetting*,
- membuat model.

3.3.1 Membuat Koneksi ke DMS (Data Mining Server)

Sebelum membuat sebuah model, dibutuhkan sebuah instan dari *DataMiningServer*. Instan ini digunakan sebagai sebuah wakil untuk membuat koneksi ke Data Mining Server (DMS). Instan tersebut juga mengatur koneksi. DMS yang berada di sisi server, merupakan komponen basisdata yang menunjukkan operasi-operasi datamining sesungguhnya di dalam Oracle9i Data Mining. DMS juga menyediakan sebuah penyimpanan metadata yang berisi obyek masukan dan obyek hasil, bersama dengan nama tempat obyek-obyek tersebut disimpan dan diambil.

```
//Create an instance of the DMS server.
//The mining server DB_URL, user_name, and password for your
installation
//need to be specified
dms=new DataMiningServer("DB_URL", "user_name", "password");
//get the actual connection
dmsConnection = dms.login();
```

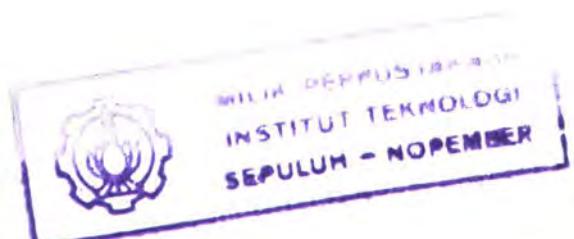
Gambar 3.26 Inisialisasi DMS

3.3.2 Mendeskripsikan Data yang Dibangun

Sebelum Oracle9i Data Mining (ODM) menggunakan data dari sebuah model, ODM harus mengetahui tempat data disimpan dan cara pengaturannya. Hal ini dapat dilakukan dengan membuat instan *PhysicalDataSpecification* yang menunjukkan data tersebut transaksional atau non-transaksional dan menunjukkan fungsi dari bermacam data kolom yang ada.

3.3.2.1 Lokasi Akses Data untuk Membuat Data

Sebelum membuat sebuah instan dari *PhysicalDataSpecification*, informasi dari lokasi data harus tersedia. Hal ini dapat dibuat dengan mempergunakan obyek *LocationAccessData*.



```
//Create a LocationAccessData using the table name
//(CENSUS_2D_BUILD_UNBINNED) and schema_name for your installation
LocationAccessData lad =
new LocationAccessData("CENSUS_2D_BUILD_UNBINNED", "schema_name");
```

Gambar 3.27 Inisialisasi LocationAccessData

Census_2d_Build_Unbinned pada gambar tersebut menunjukkan tabel lokasi data yang tersedia, *schema_name* menunjukkan nama skema dari tabel lokasi data.

Berikutnya, membuat instan *PhysicalDataSpecification*.

3.3.2.2 Spesifikasi Fisik Data Transaksional

Jika data dalam format non-transaksional, semua informasi yang dibutuhkan untuk membuat *PhysicalDataSpecification* berada di dalam obyek *LocationAccessData*.

```
//Membuat sebuah PhysicalDataSpecification untuk sebuah
//obyek NonTransactionalDataSpecification
//untuk kasus non-transaksional
PhysicalDataSpecification m_PhysicalDataSpecification =
new NonTransactionalDataSpecification(lad);
```

Gambar 3.28 Inisialisasi PhysicalDataSpecification - nonTransactional

3.3.2.3 Spesifikasi Fisik untuk Membuat Data Transactional

Jika data dalam format transaksional, harus ditentukan kegunaan dari berbagai kolom data.

```
// Membuat PhysicalDataSpecification untuk data kasus transaksional
PhysicalDataSpecification m_PhysicalDataSpecification =
new TransactionalDataSpecification(
"CASE_ID", //kolom menunjukkan id kasus
"ATTRIBUTES", //kolom menunjukkan nama atribut
"VALUES", //nama kolom yang memperlihatkan nilai
lad);
```

Gambar 3.29 Inisialisasi PhysicalDataSpecification – Transactional

3.3.3 Membuat Obyek MiningFunctionSetting

Obyek MiningFunctionSetting (MFS) memberi informasi kepada DMS tipe model yang dibentuk , fungsi model yang dipakai, dan algoritma yang digunakan.

ODM mendukung fungsi mining berikut ini :

- 1) Kaidah Asosiasi (*unsupervised learning*),
- 2) Clustering (*unsupervised learning*),
- 3) Classification (*supervised learning*),
- 4) Attribut Importance (*supervised learning*).

MFS mengijinkan seorang pengguna untuk menentukan tipe hasil tanpa harus menentukan sebuah algoritma tertentu. Jika sebuah algoritma tidak ditentukan, DMS yang digunakan memilih algoritma berdasarkan parameter-parameter yang dibuat oleh pengguna.

3.3.3.1 Menentukan Algoritma Untuk Klasifikasi dan Asosiasi

Untuk menentukan algoritma asal yang dipakai pada klasifikasi, terlebih dahulu dibuat sebuah obyek *ClassificationFunctionSettings* dengan sebuah obyek *MiningAlgorithmSettings* yang kosong untuk MFS. Cara mudah untuk membuat sebuah obyek *ClassificationFunctionSettings* dengan menggunakan metode *create* seperti digambarkan berikut. Pada kasus ini, dibutuhkan pemberian nama terhadap atribut tujuan, dan apakah data disiapkan oleh pengguna. Data yang belum dipersiapkan otomatis diberi tempat oleh ODM.

```
//menentukan "class" sebagai nama atribut tujuan, dengan tipe
//tipe atribut "categorical", dan menentukan DataPreparationStatus
//ke "unprepared".
//Pada kasus mempergunakan pengelompokan otomatis.
ClassificationFunctionSettings m_ClassificationFunctionSettings =
ClassificationFunctionSettings.create(
dmsConnection,
null,
m_PhysicalDataSpecification,
"class",
AttributeType.categorical,
DataPreparationStatus.getInstance("unprepared"));
}
}
```

Gambar 3.30 Inisialisasi ClassificationFunctionSettings

Sedangkan untuk kaidah asosiasi, menggunakan obyek *AssociationRulesFunctionSettings*. Pada inisialisasi obyek ini harus diikutsertakan dua karakter penting dari kaidah asosiasi, yaitu [OR2-02] :

- Support, *support* adalah rasio dari jumlah transaksi yang memiliki itemset ini terhadap jumlah transaksi yang ada.
- Confidence, *confidence* adalah nilai probabilitas adanya itemset A pada suatu transaksi, maka juga ada itemset B pada transaksi tersebut.

Pada ODM 9.2.0, hanya satu algoritma yang dipakai oleh Kaidah Asosiasi sehingga tidak memerlukan setting algoritma yang spesifik.

```
// init AssociationRulesFunctionSettings, dmsConn = Connection to ODM
//pds = physicaldataspecification,
//dps = datapreparationstatus,
arfs = AssociationRulesFunctionSettings.create
(
dmsConn,
pds,
dps,
fMinSupport,
fMinConfidence,
iMaxRuleLength);
```

Gambar 3.31 Inisialisasi AssociationRulesFunctionSettings

3.3.3.2 Menentukan Algoritma Naive Bayes

Jika terdapat suatu algoritma tertentu yang dipergunakan, informasi dari algoritma tersebut ditentukan lewat instan *MiningAlgorithmSettings*. Misalnya, jika diinginkan membuat sebuah model yang memakai algoritma Naive Bayes, langkah pertama harus dibuat terlebih dahulu sebuah instan *NaiveBayesSettings* untuk menentukan pemakaian algoritma Naive Bayes. Dua konfigurasi yang ada : *singleton threshold* dan *pairwise threshold*. Kemudian dibuat sebuah instan *ClassificationFunctionSettings* untuk operasi berikutnya.

```
//membuat algoritma Naïve Bayes dengan menentukan threshold
//bernilai 0.01.
NaiveBayesSettings algorithmSetting = new NaiveBayesSettings(0.01f
0.01f);
//membuat obyek ClassificationFunctionSettings dengan menggunakan
//algorithmSetting untuk MiningAlgorithmSettings. Menentukan "class"
//sebagai nama target atribut, "categorical" untuk tipe atribut
//, dan menentukan DataPreparationStatus bernilai "unprepared".
//Pada kasus ini mempergunakan pengelompokan otomatis
ClassificationFunctionSettings m_ClassificationFunctionSettings =
ClassificationFunctionSettings.create(
dmsConnection,
algorithmSetting,
m_PhysicalDataSpecification,
class,
Attribute Type.categorical,
DataPreparationStatus.getInstance(unprepared));
```

Gambar 3.32 Penentuan ClassificationFunctionSettings untuk Naive Bayes

3.3.3.3 Melakukan Validasi terhadap Seting Fungsi Mining

Karena obyek *MiningFunctionSettings* adalah obyek yang komplek, lebih baik dilakukan validasi terlebih dahulu untuk menentukan apakah seting yang sudah dimasukkan sudah benar sebelum memulai tugas pembuatan model yang sebenarnya. Jika obyek *MiningFunctionSettings* sudah valid, obyek ini dapat bertahan di dalam DMS untuk pemakaian berikutnya.

Berikut ini contoh cara pemakaian *ClassificationFunctionSetting*

```
//melakukan validasi dengan konfigurasi "NamaFungsi"
m_ClassificationFunctionSettings.validate();
m_ClassificationFunctionSettings.store(dmsConnection, "NamaFungsi");
```

Gambar 3.33 Pemakaian ClassificationFunctionSetting untuk Validasi

```
//Validate & store AssociationRulesFunctionSettings with
// SettingsName = sMiningSettingsName
arfs.validate();
arfs.store(dmsConn,sMiningSettingsName);
```

Gambar 3.34 Pemakaian AssociatioRulesFunctionSettings untuk Validasi

3.3.4 Membuat Proses Sinkron dan Asinkron

Jika semua kebutuhan informasi untuk membuat sebuah model sudah dimasukkan di dalam sebuah instan *PhysicalDataSpecification* dan *MiningFunctionSettings*, langkah selanjutnya adalah menentukan cara pembuatan model dalam proses dengan cara sinkron atau asinkron.

Jika ODM dipanggil dari sebuah aplikasi, desain dari aplikasi tersebut menentukan cara model dibangun, sinkron atau secara asinkron. Juga, jika data yang dipergunakan sangat besar, yang mengakibatkan pembuatan model membutuhkan waktu yang besar; pada kasus seperti ini lebih efektif mempergunakan pembuatan model secara asinkron.

3.3.4.1 Pembuatan Model Secara Sinkron

Untuk pembuatan yang dilakukan secara sinkron, maka mempergunakan metode *MiningModel.build*.

```
//membuat model dengan konfigurasi "NamaKonfigurasi" dan menyimpan
//model dengan nama "NamaModel".
MiningModel.build(
dmsConn,
lad,
m_PhysicalDataSpecification,
"NamaKonfigurasi",
"NamaModel");
```

Gambar 3.35 Pembuatan Model Secara Sinkron

3.3.4.2 Pembuatan Model Secara Asinkron

Untuk pembuatan model secara asinkron dapat dilakukan dengan membuat sebuah instan *MiningTask*. Sebuah *MiningTask* tetap berada di dalam DMS dengan mempergunakan metode *store*, dan bisa dieksekusi di waktu yang lain; namun hanya bisa dieksekusi satu kali. Saat proses dilakukan, permintaan terhadap informasi status suatu proses dapat dilakukan dengan mengeksekusi metode *getCurrentStatus*. Metode ini mengembalikan sebuah obyek *MiningTaskStatus*, yang menyediakan status yang lebih detil. Untuk status historis yang lebih detil dapat dilakukan dengan memanggil metode *getStatusHistory*.

```
//Membuat tugas pembuatan model Naïve Bayes dan mengesekusinya
//MiningFunctionsSettings dinamakan "SettingMining" dan
//nama model diberi nama "NamaModel"
MiningBuildTask task =
new MiningBuildTask(
m_PhysicalDataSpecification,
"SettingMining",
"NamaModel");

//menyimpan tugas mining dengan "NamaTugasMining"
task.store(dmsConnection, "NamaTugasMining");

//Mengeksekusi tugas
task.execute(dmsConnection);
```

Gambar 3.36 Pembuatan Model Secara Asinkron

Setelah bagian *MiningModel.build* atau *task.execute* berhasil dieksekusi, model disimpan dengan mempergunakan nama yang sudah ditentukan (pada contoh di atas, “Sample_NB_Model”) di dalam DMS.

3.3.5 Penentuan Lokasi Data Akses

Untuk melakukan eksport dan impor diperlukan penentuan terhadap lokasi data akses. Hal ini dapat dilakukan dengan membuat sebuah obyek

LocationCellAccessData. Obyek ini mengijinkan untuk menentukan sebuah kolom dan baris tertentu pada skema dan tabel yang sudah ditentukan.

```
Provide information about the location of the output data
LocationCellAccessData outputDataLocation = new LocationCellAccessData(
    m_ModelExportTableName,
    m_ModelExportSchemaName,
    m_ModelExportColumnName,
    rowID);
```

Gambar 3.37 Inisialisasi LocationCellAccessData

3.3.6 Eksport Model ke Format PMML

Untuk melakukan eksport model ke dalam format PMML dapat dilakukan dengan membuat sebuah obyek *ModelExportTask*. Instan dari *ModelExportTask* digunakan untuk melakukan eksport model ODM ke dalam format representasi model standar. Oracle9i Data Mining mendukung eksport model yang ada dalam DMS ke dalam standar industri PMML 2.0.

ODM mendukung eksport dua model, yaitu :

- Model Kaidah Asosiasi
- Model Naive Bayes.

Class ini mempunya kemampuan mengeksport model ke dalam sebuah tabel bertipe *XMLType*. Basisdata Oracle9i mendukung sebuah tipe data *SYS.XMLType* untuk menyimpan dokumen XML di dalam basisdata. *LocatioCellAccessData* menentukan lokasi dari tabel *XMLType* tempat model dieksport.

Akan terjadi kesalahan eksport jika model yang ditunjuk tidak ada dalam DMS, tabel atau sel yang diinginkan tidak valid, atau model tidak bisa diekspor.

```
Create the Model Export task
m_ModelExportTask = new ModelExportTask(
    m_ModelName,
    outputDataLocation,
    MiningStandardType.PMML20);
```

Gambar 3.38 Inisialisasi Model ExportTask

3.3.7 Import Model dari Format PMML

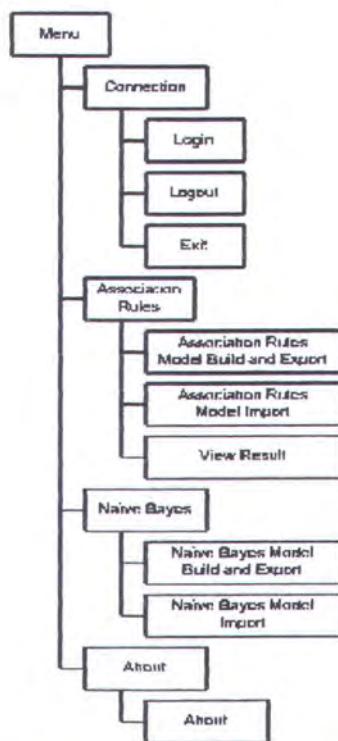
Untuk melakukan eksport model ke dalam format PMML dapat dilakukan dengan membuat sebuah obyek *ModelImportTask*. Sebuah instan dari *ModelImportTask* digunakan untuk mengimport format standar model ke dalam Oracle9i Data Mining. Obyek ini mendukung proses import sebuah model datamining dari tabel yang kolomnya bertipe XMLType. Basisdata Oracle9i mendukung tipe data SYS.XMLType untuk menyimpan dokumen XML di dalam basisdata. *ModelImportTask* membutuhkan obyek *LocationCellAccessData*. *LocationCellAccessData* menyediakan lokasi dari tabel bertipe XMLType yang menyimpan model yang diimport. Obyek lain yang dibutuhkan adalah *MiningStandartType*. Obyek ini menunjukkan format model yang diimport. Saat ini Oracle9i Data Mining hanya mendukung import ke dalam format PMML 2.0.

3.4 Desain Antarmuka

Perangkat lunak ekstraksi data ini memiliki desain antarmuka yang terdiri dari empat macam dialog, yaitu dialog login aplikasi sekaligus menentukan koneksi ke skema ODM, dialog pembuatan model Kaidah Asosiasi sekaligus eksport data ke dalam bentuk PMML, dialog pembuatan model Naive Bayes sekaligus eksport

model ke dalam bentuk PMML, dan dialog import model dari bentuk PMML ke dalam bentuk model yang diterima oleh Oracle9i Data Mining.

Perangkat lunak ini berbasiskan pada sistem operasi Windows 2000, oleh karena itu mengikuti umumnya perangkat lunak di bawah sistem operasi ini, yaitu berbasiskan GUI, yang mendukung Multiple Document Interface (MDI). Ada menu-menu yang tersedia untuk mengakses dialog.



Gambar 3.39 Susunan Menu pada Perangkat Lunak

Menu-menu tersebut adalah sebagai berikut :

a. Menu login

Menu ini terdiri dari sub menu :

- Login : digunakan untuk melakukan koneksi ke skema ODM.
- Logout : digunakan untuk keluar dari skema ODM.
- Exit : digunakan untuk keluar dari aplikasi.

b. Menu Association Rule

Menu ini terdiri dari sub menu :

- Association Rules Model Build and Export : sub menu ini digunakan untuk membuat model Kaidah Asosiasi, mengeksport ke dalam bentuk PMML, dan menyimpan hasil PMML ke dalam bentuk file.
- Association Rules Model Import : sub menu digunakan untuk membaca file PMML dan kemudian diimport menjadi model yang diterima oleh Oracle9i Data Mining.
- View Result : sub menu ini digunakan untuk melihat *rule* yang dimiliki oleh sebuah model hasil pembuatan maupun hasil import dari perangkat lunak ini.

c. Menu Naive Bayes

Menu ini terdiri dari sub menu :

- Naive Bayes Model Build and Export : sub menu ini digunakan untuk membuat model klasifikasi dengan memakai algoritma Naive Bayes.
- Naive Bayes Model Import: sub menu digunakan untuk membaca file PMML dan kemudian diimport menjadi model yang diterima oleh Oracle9i Data Mining.

d. Menu About

Menu ini hanya memiliki satu sub menu, yaitu sub menu About. Sub menu About ini berisi judul tugas akhir dan pembuatnya.

Pada tahap perancangan antar muka ini juga dibuat perancangan dialog dari masing-masing sub menu. Hanya saja bagian ini diikutsertakan pada bagian tahap implementasi antar muka.

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Bab ini menjelaskan implementasi perangkat lunak berdasarkan desain yang sudah dibuat pada bab sebelumnya. Setelah implementasi dari desain tersebut perangkat lunak nantinya mempunyai kemampuan :

- a. membuat model dengan algoritma Kaidah Asosiasi,
- b. membuat model dengan algoritma Naive Bayes
- c. mengeksport model Kaidah Asosiasi ke dalam format PMML 2.0,
- d. mengeksport model Naive Bayes ke dalam format PMML 2.0,
- e. mengimport model dari format PMML ke dalam model yang diterima Oracle9i Data Mining.

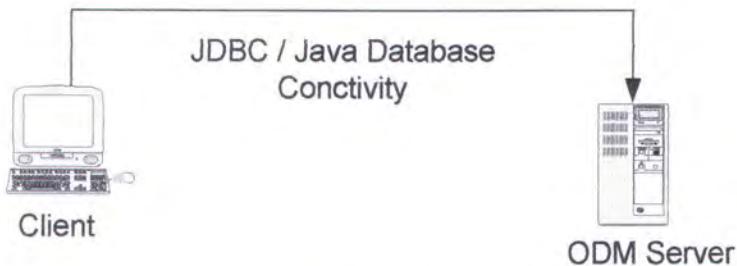
Untuk mendukung pembuatan perangkat lunak, digunakan perangkat lunak bantu dari Oracle, yaitu JDeveloper 9.0.3.1, yang berjalan di sistem operasi Windows 2000 Service Pack 3. Sedangkan basisdata yang digunakan adalah RDBMS Oracle9i versi 9.2.0.1.0 dengan Oracle9i Data Mining versi 9.2.0.3. Lingkungan pengembangan perangkat keras yang digunakan untuk membangun perangkat lunak ini adalah komputer dengan prosesor Intel Pentium 4 dengan kecepatan 1.8 GHz, serta memori fisik sebesar 512 Mb.

Tahap implementasi ini meliputi tiga aspek, yaitu implementasi data masukan, implementasi proses, dan implementasi antarmuka. Sesuai dengan yang

disebutkan pada bab sebelumnya, implementasi data terbagi menjadi tiga bagian, yaitu : implementasi data masukan, data proses, dan data keluaran.

4.1 Arsitektur Perangkat Lunak

Perangkat lunak ini menggunakan model arsitektur two-tier seperti pada gambar berikut :



Gambar 4.01 Arsitektur Aplikasi

Penggunaan model seperti ini mengakibatkan perangkat lunak yang berbasiskan Java berhubungan langsung dengan basisdata, terutama ODM Server. Oleh karena itu perangkat lunak ini membutuhkan sebuah JDBC driver yang bisa berkomunikasi dengan *Database Management System* (DBMS) yang diakses. Hal ini memungkinkan perangkat lunak untuk berhubungan dengan basisdata yang berada di mesin lain, yang berbeda dengan tempat perangkat lunak berjalan.

JDBC (Java Database Connectivity) didefinisikan sebagai sebuah antarmuka untuk berelasi dengan sumber data. JDBC memiliki kemampuan mengeksekusi perintah-perintah SQL untuk mengakses dan mengelola basisdata relasional. JDBC didesain berorientasi obyek, *application programming interface* (API) untuk mengakses basisdata yang berbasiskan Java, dan ditujukan sebagai standar oleh pengembang aplikasi Java dan perusahaan pembuat basisdata. Pemakaian

Java dan JDBC ini mengakibatkan pengembang aplikasi tidak perlu membuat aplikasi yang berbeda untuk sistem operasi yang berbeda.

Untuk terhubung dengan skema ODM, perangkat lunak ini menggunakan Oracle Thin JDBC. Driver ini merupakan *driver* yang menggunakan soket Java untuk melakukan koneksi secara langsung ke Oracle. Karena *driver* ini secara keseluruhan dibuat menggunakan Java, maka *driver* ini bersifat platform-independent.

4.2 Implementasi Data

Bagian implementasi data ini didasarkan pada apa yang sudah direncanakan pada bab III. Berdasarkan tahap perencanaan, ada tiga bagian data yang diimplementasikan, yaitu data masukan, data proses, dan data keluaran. Untuk mengimplementasikan hal tersebut sekaligus menunjukkan bahwa perangkat lunak yang dibuat ini berbasiskan obyek, maka perangkat lunak dibagi menjadi beberapa *class*. *Class-class* tersebut antara lain : clsAscToPMMLParameter, clsAscToPMMLProses, clsAssociationParameter, clsAssociationRule, clsFileFilter, clsModelToPMML, clsNBBuild, clsNBProsesParameter, clsODMConnection, frmAssociation, frmAssociationRuleFromFile, frmLogin, frmNaiveBayesProses, Main.

Class-class yang mempunyai nama dengan awalan frm menunjukkan bahwa kelas tersebut adalah *class dialog* yang berhubungan langsung dengan pemakai perangkat lunak, sedangkan *class-class* yang diawali dengan cls menunjukkan bahwa *class* tersebut tempat proses berlangsung.

Pada bagian berikutnya dijelaskan implementasi *class-class* tersebut terutama untuk menunjukkan implementasi data.

4.2.1 Main

Class ini merupakan *class* utama dari perangkat lunak, artinya pemakai banyak berinteraksi dengan *class* ini. *Class* ini berupa dialog yang mengimplementasikan paradigma *Multiple Document Interface* (MDI), sehingga berisi menu-menu, yang berfungsi untuk memanggil dialog-dialog yang lain.

Tabel 4.01 Class Main

Constructor Summary	
Main ()	membuat sebuah obyek Main baru
Method Summary	
void	EnableLogin(int iEnable) melakukan pengesetan terhadap aktif tidaknya menu sesuai dengan berhasil tidaknya login ke skema odm
void	InternalFrameClosed(InternalFrameEvent e) menangkap event yang terjadi pada <i>form-child</i>
void	main (String[] args) inisialisasi saat dialog utama dibuka pertama kali
JMenuBar	menuBar inisialisasi menu-menu pada Main, sekaligus event saat suatu menu diakses

4.2.2 clsODMConnection

Class ini terdiri dari metode-metode dan variabel-variabel yang dibutuhkan untuk melakukan koneksi dengan skema ODM.

Tabel 4.02 Class clsODMConnection

Constructor Summary	
clsODMConnection(String sComputerName, String sServiceName, String sUserName, String sPassword, String sPortNumber)	membuat sebuah objek clsODMConnection baru dengan nama komputer tempat skema ODM ditunjukkan oleh sComputerName, nama service Oracle9i ditunjukkan oleh sServiceName, nama <i>user</i> ODM ditunjukkan dengan sUserName, dan nomor port service ditunjukkan oleh sPortNumber

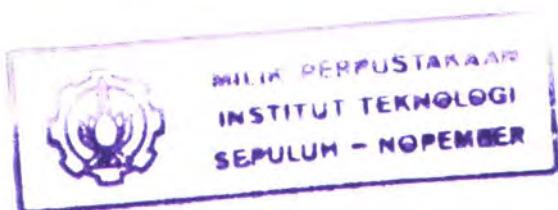
Method Summary	
boolean	iConnected() menunjukkan apakan <i>Class</i> sedang terhubung dengan skema ODM atau tidak
void	LogOut keluar dari skema ODM
Connection	ODMConnection mengembalikan parameter Connection, yaitu koneksi dengan skema ODM
String	sMiningServerPassword() mengembalikan string password ke skema ODM
String	sMiningServerURL mengembalikan URL dari koneksi ke skema ODM
String	sMiningServerUserName mengembalikan string <i>user name</i> ke skema ODM
int	tryConnect melakukan proses login ke skema ODM

4.2.3 clsAssociationParameter

Class ini berfungsi untuk menyimpan variabel-variabel input yang dibutuhkan dalam proses pembuatan model Kaidah Asosiasi.

Tabel 4.03 clsAssociationParameter

Constructor Summary	
clsAssociationParameter() membuat sebuah objek clsAssociationParameter baru, melakukan inisialisasi terhadap variabel lokal.	
Method Summary	
String	getAttributeName() mengembalikan String nama atribut
String	getBuildDataName() mengembalikan String nama data yang dibuatkan modelnya
String	getBuildDataType() mengembalikan tipe data dari data yang dibuatkan modelnya
String	getDataStatus() mengembalikan status data, <i>unprepared</i> atau <i>discretized</i>
String	getEnableCleanup() mengembalikan String enablecleanup



Method Summary	
String	getInputDataSchemaName() mengembalikan String skema tempat data yang dimodelkan
int	getMaxRuleLength() mengembalikan nilai integer yang menunjukkan maksimum panjang <i>rule</i>
float	getMinConfidence() mengembalikan nilai minimum <i>confidence</i>
float	getMinSupport() mengembalikan nilai minimum <i>support</i>
String	getMiningSettingsName() mengembalikan String nama setting yang digunakan
String	getModelName() mengembalikan String nama model yang dibangun
String	getSequenceIDName() mengembalikan String ID <i>sequence</i>
String	getTaskName() mengembalikan String nama <i>task</i>
String	getValueName() mengembalikan String nama <i>value</i>
void	setAttributeName(String sPass) menentukan nama attribut dengan nilai sPass
void	setBuildDataTableName(String sPass) menentukan nama tabel yang dimodelkan dengan nilai sPass
void	setBuildDataType(String sPass) menentukan tipe data dari tabel yang dimodelkan dengan nilai sPass
void	setDataStatus(String sPass) menentukan status data yang dimodelkan dengan nilai sPass
void	setEnabledCleanup(String sPass) menentukan nilai variabel <i>cleanup</i> sesuai dengan sPass
void	setInputDataSchemaName(String sPass) menentukan nama skema dari tabel yang dimodelkan sesuai dengan nilai sPass
void	setMaxRuleLength(int sPass) menentukan nilai maksimum <i>rule</i> sesuai dengan nilai sPass
void	setMinConfidence(float sPass) menentukan nilai minimum <i>confidence</i> sesuai dengan nilai sPass
void	setMinSupport(float sPass) menentukan nilai minimum <i>support</i> sesuai dengan nilai sPass
void	setMiningSettingsName(String sPass) menentukan nama setting sesuai dengan nilai sPass

Method Summary	
void	set modelName(String sPass) menentukan nama model sesuai dengan nilai sPass
void	set SequenceIDName(String sPass) menentukan nama kolom untuk pengidentifikasi <i>sequence</i> sesuai dengan sPass
void	set TaskName(String sPass) menentukan nama <i>task</i> sesuai dengan nilai sPass
void	set ValueName(String sPass) menentukan nama kolom untuk pengidentifikasi nilai traksaksi sesuai dengan sPass

4.2.4 clsNBProsesParameter

Class ini berfungsi untuk menyimpan variabel-variabel input yang dibutuhkan dalam proses pembuatan model Naive Bayes.

Tabel 4.04 clsNBProsesParameter

Constructor Summary	
	clsNBProsesParameter() menginisialisasi semua variabel lokal
Method Summary	
String	getAttributeName() mengembalikan String nama attribut
String	getEnableCleanup() mengembalikan nilai <i>cleanup</i>
String	getModelName() mengembalikan String nama model
float	getPairwiseThreshold() mengembalikan float nilai <i>pairwise threshold</i>
String	getPreparationStatus() mengembalikan status data yang dimodelkan
String	getSchemaName() mengembalikan String nama skema tempat tabel yang dimodelkan
String	getSequenceID() mengembalikan String ID <i>sequence</i>
String	getSettingName() mengembalikan String nama setting
float	getSingletonThreshold() mengembalikan float <i>Singleton Threshold</i>

Method Summary	
String	getSupplementalAttribute() mengembalikan nilai atribut supplemental
String	getTableName() mengembalikan String nama tabel yang dimodelkan
String	getTargetAttributeName() mengembalikan String nama atribut tujuan
String	getTaskName() mengembalikan String nama <i>task</i>
String	getType() mengembalikan tipe transaksi data yang dimodelkan
String	getValueName() mengembalikan String kolom yang menunjukkan <i>value</i>
void	setAttributeName(String sPass) menentukan nama atribut sesuai dengan nilai sPass
void	setEnableCleanup(String sPass) menentukan nilai variabel <i>cleanup</i>
void	setModelName(String sPass) menentukan nama model sesuai dengan nilai sPass
void	setPairwiseThreshold(float fPass) menentukan besarnya <i>pairwise threshold</i> sesuai dengan nilai sPass
void	setPreparationStatus(String sPass) menentukan nilai status data yang dimodelkan
void	setSchemaName(String sPass) menentukan nama skema dari tabel yang dimodelkan sesuai dengan nilai sPass
void	setSequenceID(String sPass) menentukan nama kolom untuk pengidentifikasi <i>sequence</i> sesuai dengan sPass
void	setSettingName(String sPass) menentukan nama setting sesuai dengan nilai sPass
void	setSingletonThreshold(float fPass) menentukan besarnya <i>pairwise threshold</i> sesuai dengan nilai sPass
void	setSupplementalAttribute(String sPass) menentukan nama atribut tambahan sesuai dengan nilai sPass
void	setTableName(String sPass) menentukan nama tabel data yang dimodelkan sesuai dengan nilai sPass
void	setTargetAttributeName(String sPass) menentukan nama atribut target sesuai dengan nilai sPass
void	setTaskName(String sPass) menentukan nama <i>task</i> sesuai dengan nilai sPass

Method Summary	
void	setType(String sPass) menentukan tipe traksasi yang digunakan oleh data yang dimodelkan sesuai dengan nilai sPass
void	setValueName(String sPass) menentukan nama kolom yang menunjukkan <i>value</i>

4.2.5 clsAscToPMMLParameter

Class ini berfungsi untuk menyimpan variabel-variabel input yang dibutuhkan dalam proses eksport model ke dalam format PMML. Proses dari eksport model Kaidah Asosiasi maupun Naive Bayes mempergunakan *class* ini.

Tabel 4.05 clsAscToPMMLParameter

Constructor Summary	
clsAscToPMMLParameter()	menginisialisasi variabel lokal
Method Summary	
String	getCleanUp() mengembalikan String variabel <i>cleanup</i>
String	getColumn Name() mengembalikan String nama kolom yang dieksport
String	getExportName() mengembalikan String nama proses eksport
String	getModelName() mengembalikan String nama model yang dieksport
String	getRowID() mengembalikan String RowID dari tabel yang dieksport
String	getSchemaName() mengembalikan String nama skema tempat model yang dieksport
String	getTableName() mengembalikan String nama tabel tujuan proses eksport
void	setCleanUp(String sPass) menentukan nilai variabel <i>cleanup</i> sesuai dengan nilai sPass
void	setColumnName(String sPass) menentukan nama kolom dari tabel tujuan eksport sesuai dengan nilai sPass
void	setExportName(String sPass) menentukan nama proses eksport sesuai dengan sPass

Method Summary	
void	setmodelName(String sPass) menentukan nama model yang dieksport sesuai dengan nilai sPass
void	setRowID(String sPass) menentukan RowID dari tabel tujuan eksport sesuai dengan nilai sPass
void	setSchemaName(String sPass) menentukan nama skema dari tabel tujuan ekport
void	setTableName(String sPass) menentukan tabel tujuan proses eksport

4.2.6 clsModelToPMML

Class ini berfungsi untuk membuka file model dalam format PMML 2.0 untuk diimport ke dalam model yang diterima oleh Oracle9i Data Mining.

Tabel 4.06 clsModelToPMML

Constructor Summary	
clsModelToPMML()	menginisialisasi semua variabel lokal
Method Summary	
Document	getDocumentResult() mengembalikan tipe data Document dari file yang dibuka
String	getFilePath() mengembalikan String tempat file asal
void	openFile(final File filePass) prosedur membuka file
boolean	setOpenFile() mengembalikan boolean proses membuka file PMML. Bernilai <i>true</i> jika file berhasil dibuka dan <i>false</i> jika sebaliknya

4.3 Implementasi Proses

Implementasi proses ini dibuat sesuai dengan perancangan proses yang disebutkan pada sub bab 3.3. Proses-proses yang dibahas pada sub bab ini adalah proses :

- a. pembuatan model Kaidah Asosiasi

- b. pembuatan model Naive Bayes,
- c. proses ekspor model Kaidah Asosiasi atau Naive Bayes ke dalam bentuk PMML
- d. proses import dari model PMML ke dalam model Kaidah Asosiasi atau Naive Bayes

4.3.1 Pembuatan Model Kaidah Asosiasi (clsAssociationRule)

Pada proses pembuatan model Kaidah Asosiasi ini menggunakan *Class* `clsAssociationRule` dan membutuhkan *Class* `clsAssociationParameter` untuk menampung variabel-variabel yang dibutuhkan. *Class* `clsAssociationRule` berisikan metode-metode dan variabel-variabel yang dibutuhkan untuk membuat model. Dalam prosesnya, *Class* ini menggunakan obyek-obyek seperti yang sudah disebutkan pada bab desain proses. Constructor dan metode yang ada tampak pada tabel berikut :

Tabel 4.07 Class clsAssociationRule

Constructor Summary	
	<code>clsAssociationRule (clsODMConnection classConn, clsAssociationParameter classParam)</code> membuat sebuah obyek <code>clsAssociationRule</code> baru dengan koneksi sesuai dengan <code>classConn</code> dan parameter-parameternya sesuai dengan <code>clsAssociationParameter</code>
Method Summary	
void	<code>initializeVariable()</code> melakukan inisialisasi semua variabel, mengkopi variabel-variabel dalam <code>clsParameter</code> ke dalam variabel lokal
int	<code>iProses()</code> mengembalikan nilai integer dari proses, jika berhasil bernilai 1 jika gagal bernilai 0
void	<code>CreatePhysicalDataSpecification (Connection dmsConn)</code> membuat obyek <code>PhysicalDataSpecification</code> sesuai dengan koneksi dalam <code>dmsConn</code>

Method Summary	
void	createMiningFunctionSettings (Connection dmsConn) membuat sebuah obyek <i>AssociationRulesFunctionSettings</i> , melakukan validasi, dan menyimpannya sesuai dengan koneksi yang ada pada dmsConn
String	buildAssociationRuleModel (Connection dmsConn) membuat sebuah obyek <i>MiningBuildTask</i> , mengeksekusi pembuatan model, dan mengembalikan string status pembuatan model
void	fillResult() mengisikan hasil <i>rule</i> pembuatan model
String	getProcessResult() mengembalikan string hasil pembuatan model

4.3.2 Pembuatan Model Naive Bayes (clsNBBuild)

Seperti pada pembuatan model Kaidah Asosiasi, untuk membuat model Naive Bayes diimplementasikan dengan menggunakan dua *class*, yaitu : clsNBParameter, yang berfungsi menyimpan atribut-atribut yang dibutuhkan serta clsNBBuild yang berfungsi membuat model. clsNBBuild mempergunakan obyek-obyek yang dibutuhkan untuk membuat model seperti yang telah disebutkan pada bab sebelumnya. *Constructor* dan metode yang ada tampak sebagai berikut :

Tabel 4.08 Class clsNBBuild

Constructor Summary	
clsAssociationRule ()	membuat sebuah obyek clsNBBuild baru
Method Summary	
void	setParameter(clsODMConnection clsNBProsesParameter clsNBPass) mengkopi koneksi dan parameter yang dibutuhkan dalam pembuatan model
void	initializeVariable() melakukan inisialisasi semua variabel lokal
boolean	bPerformCleanup() menghapus semua obyek yang ada sesuai dengan variabel <i>cleanup</i>

Method Summary	
boolean	bNBBuildProses() mengembalikan nilai boolean proses yang terjadi, <i>true</i> jika proses sukses, dan <i>false</i> jika sebaliknya
void	createPhysicalDataSpecification() membuat obyek <i>CreatePhysicalDataSpecification</i>
void	createMiningFunctionSettings(); membuat obyek <i>MiningFunctionSettings</i>
void	buildNaiveBayesModel() membuat model Naive Bayes

4.3.3 Proses Eksport Model ke dalam Format PMML

Untuk melakukan eksport model ke dalam format PMML 2.0 dipergunakan *class clsAscToPMMLProses*. *Class* ini digunakan untuk mengeksport model Kaidah Asosiasi maupun Naive Bayes. *Constructor* dan metode yang ada tampak sebagai berikut :

Tabel 4.09 *clsAscToPMMLProses*

Constructor Summary	
clsAscToPMMLProses()	membuat sebuah obyek <i>clsAscToPMML</i> baru dan melakukan inisialisasi variabel
Method Summary	
void	setClsPMMLParameter(<i>clsAscToPMMLParameter</i> <i>clsPass</i>) mengkopikan variabel dalam <i>clsAscToPMMLParameter</i> ke variabel lokal
void	setClsODMConnection(<i>clsODMConnection</i> <i>clsPass</i>) set koneksi ke ODM
int	iProses() melakukan proses dengan mengembalikan nilai 1 jika berhasil dan 0 jika gagal
void	performCleanup(); melakukan pengesetan ke dalam variabel <i>cleanUp</i> .
String	getPMMLResult mengembalikan hasil eksport dalam format PMML
void	saveToFile(Document <i>docPass</i> ,boolean <i>saveAs</i>) melakukan penyimpanan dokumen PMML yang dihasilkan dari proses eksport

4.3.4 Proses Import dari PMML ke Model Kaidah Asosiasi

Untuk melakukan import dari format PMML untuk masing-masing algoritma, Kaidah Asosiasi dan Naive Bayes, menggunakan *class* yang sama. *Class* tersebut adalah *clsPMMLToAR*.

Constructor dan metode yang ada tampak sebagai berikut :

Tabel 4.10 *clsPMMLToAR*

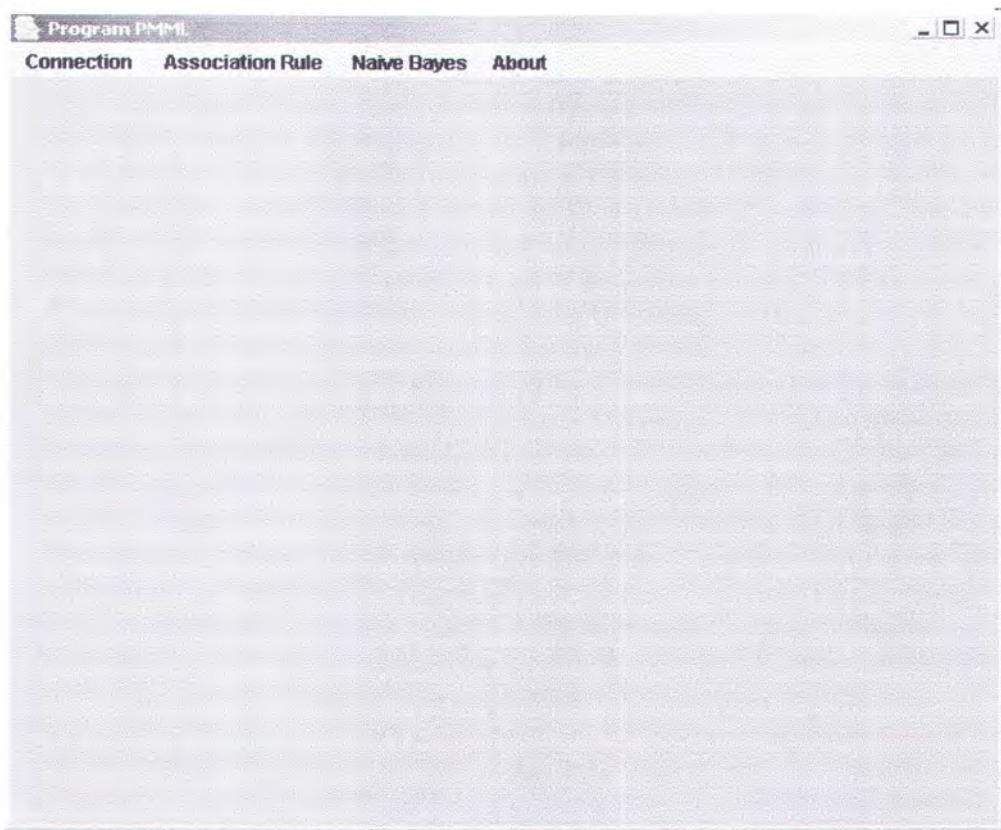
Constructor Summary	
clsPMMLToAR()	membuat sebuah obyek <i>clsPMMLToAR</i> baru dan melakukan inisialisasi variabel lokal
Method Summary	
void	setClsConn(<i>clsODMConnection</i> <i>clsPass</i>) mengkopikan variabel koneksi ke ODM
void	setImportData(String <i>sData</i>) set String yang diimport
int	PMMLProses() mengembalikan nilai int sesuai dengan hasil proses, 1 jika sukses dan 0 jika gagal
void	dropTemporaryTable() menghapus tabel untuk menampung dokumen PMML secara sementara
void	createTemporaryTable() membuat tabel sementara untuk menampung dokumen PMML
void	insertPMML(<i>sPMMLData</i> , <i>conn</i>) memasukkan data dalam format PMML ke dalam tabel sementara
void	importPMML() melakukan proses import dari data yang ada di tabel sementara ke bentuk model Oracle9i Data Mining
CLOB	getCLOB(String <i>xmlData</i> , Connection <i>conn</i>) memasukkan dokumen PMML ke dalam basisdata
String	getModelName() mengembalikan nama model hasil import
int	getNumberOfTransactions() mengembalikan jumlah transaksi dari model hasil import

Method Summary	
int	getMaxNumberOfItemsPerTransaction() mengembalikan banyaknya item tiap transaksi dari model hasil import
float	getMinimumSupport() mengembalikan nilai <i>Minimum Support</i> dari model hasil import
float	getMinimumConfidence() mengembalikan nilai <i>Minimum Confidence</i> dari model hasil import
int	getMaxRuleLength() mengembalikan panjang maksimal <i>rule</i> dari model hasil import

4.4 Implementasi Antarmuka

Implementasi desain antarmuka dibuat sesuai dengan perancangan antarmuka pada sub bab 3.3. untuk memudahkan pemakai menggunakan perangkat lunak ini. Dialog-dialog yang ada didesain untuk memudahkan pemakai mengoperasikannya. Misalnya untuk pembuatan Kaidah Asosiasi, pemakai cukup memasukkan parameter-parameter pada obyek-obyek yang tersedia.

Mengikuti kebiasaan pada perangkat lunak yang dibangun di bawah sistem operasi Windows 2000, maka digunakan bentuk MDI (Multiple Document Interface), dengan tampilan seperti berikut :



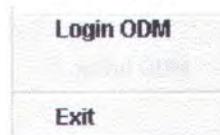
Gambar 4.02 Tampilan Utama Perangkat Lunak

Dari gambar tampilan perangkat lunak tampak keempat menu yang tersedia sesuai dengan desain yang sudah dibahas pada bab sebelumnya. Berikut ini pembahasan tampilan antarmuka perangkat lunak berdasarkan menu yang ada.

4.4.1 Menu Login

Menu ini ditujukan untuk membuat autentikasi pengguna sebelum memakai perangkat lunak. Sebelum memakai perangkat lunak pengguna diwajibkan mempunyai hak akses ke skema ODM. Dengan adanya menu ini berguna untuk membuat koneksi ke skema ODM, sehingga apabila ada proses yang membutuhkan koneksi ke ODM pengguna tidak harus login lagi.

Bila diklik tampak sub menu dari menu login sebagai berikut :



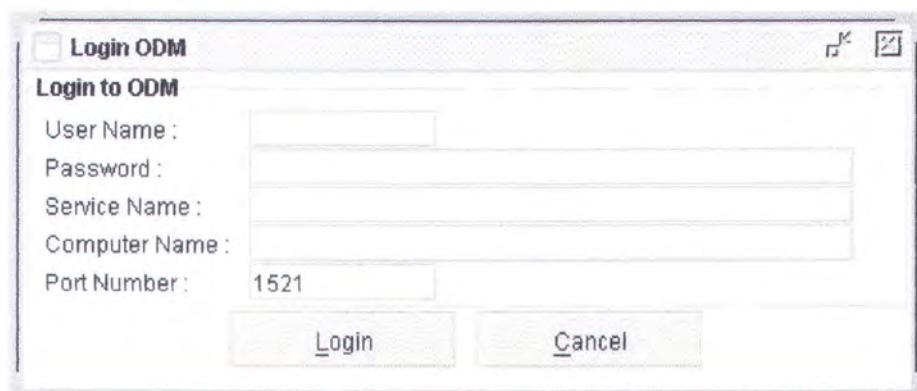
Gambar 4.03 Sub Menu Login

Saat pertama kali dijalankan tampak bahwa submenu Login ODM aktif dan sub menu Logout ODM tidak aktif.

Berikut ini efek yang dihasilkan saat sub menu diklik :

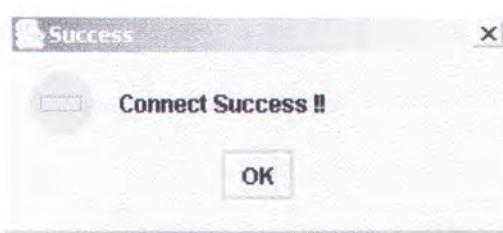
4.4.1.a Login ODM

Saat sub menu ini diakses menghasilkan tampilan dialog login perangkat lunak seperti berikut :



Gambar 4.04 Antarmuka Login Perangkat Lunak

Saat tombol *Login* ditekan, jika semua item pada dialog login ini sudah terisi dengan benar maka keluar dialog konfirmasi seperti berikut :



Gambar 4.05 Dialog Konfirmasi

4.4.1.b Logout ODM

Sub menu ini aktif jika pengguna berhasil login ke dalam perangkat lunak.

Saat sub menu ini diakses berarti pengguna memutuskan untuk keluar dari skema ODM, maka muncul dialog konfirmasi seperti berikut ini :



Gambar 4.06 Dialog Konfirmasi Logout Berhasil

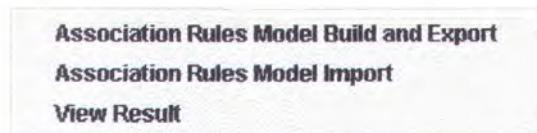
4.4.1.c Exit

Sub menu berfungsi untuk keluar atau menutup perangkat lunak. Saat menutup atau keluar tersebut juga dilakukan proses *Logout*.

4.4.2 Menu Association Rules

Pada menu ini terdapat sub menu yang berhubungan dengan Kaidah Asosiasi.

Bila diklik maka tampak sub menu seperti berikut :

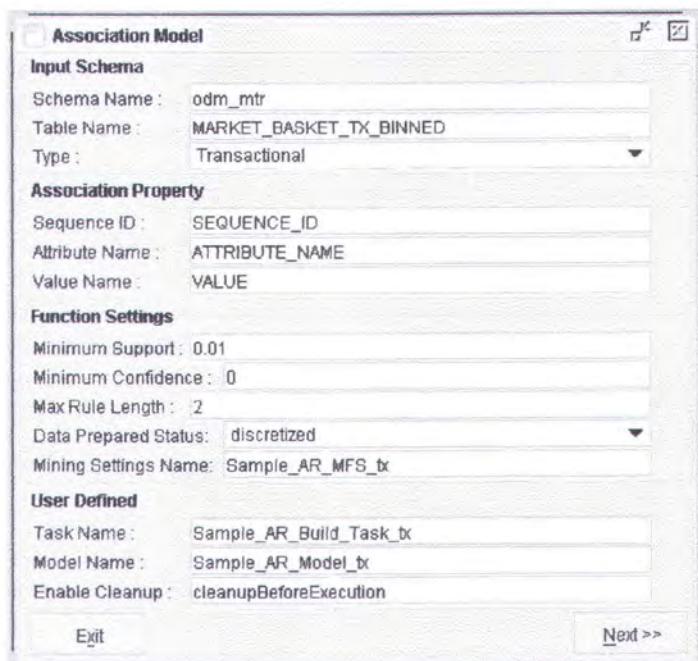


Gambar 4.07 Sub Menu Association Rule

Berikut ini *event* yang terjadi saat masing-masing sub menu tersebut diakses :

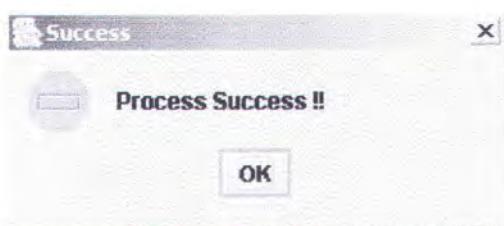
4.4.2.a Association Rules Model Build and Export

Menu ini berfungsi untuk memanggil dialog yang melakukan proses pembuatan model dan melakukan eksport ke dalam format PMML. Dialog tersebut tampak seperti berikut :



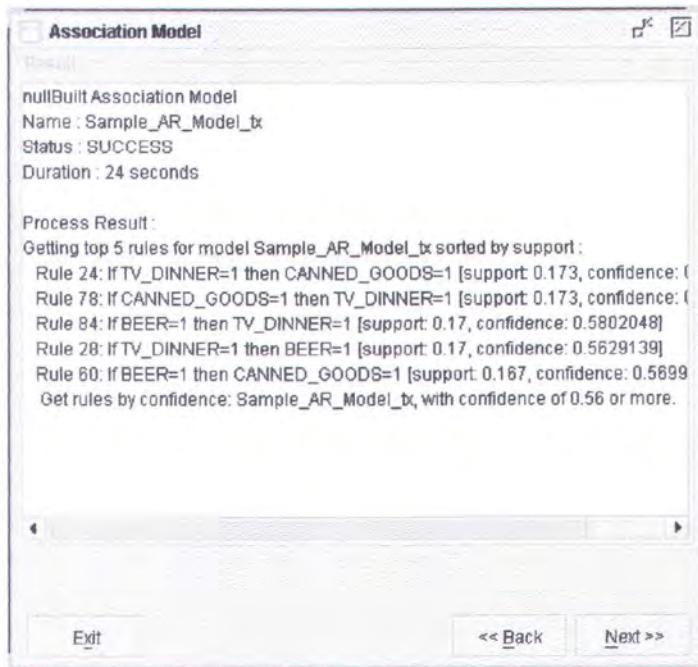
Gambar 4.08 Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 1

Pada gambar tersebut tampak dialog yang dibutuhkan untuk membuat model dan melakukan eksport. Terdapat beberapa langkah (*step*) pengisian dari pembuatan model sampai dengan menghasilkan file dalam bentuk PMML. Pada langkah pertama pengguna menginputkan parameter-parameter dalam pembuatan model Kaidah Asosiasi. Jika parameter-parameter tersebut sudah diisi dengan lengkap, pengguna dapat menekan tombol *Next>>* untuk mendapatkan proses pembuatan model dan menuju langkah selanjutnya. Jika proses berhasil dilakukan muncul dialog konfirmasi seperti berikut :



Gambar 4.09 Dialog Konfirmasi Pembuatan Model

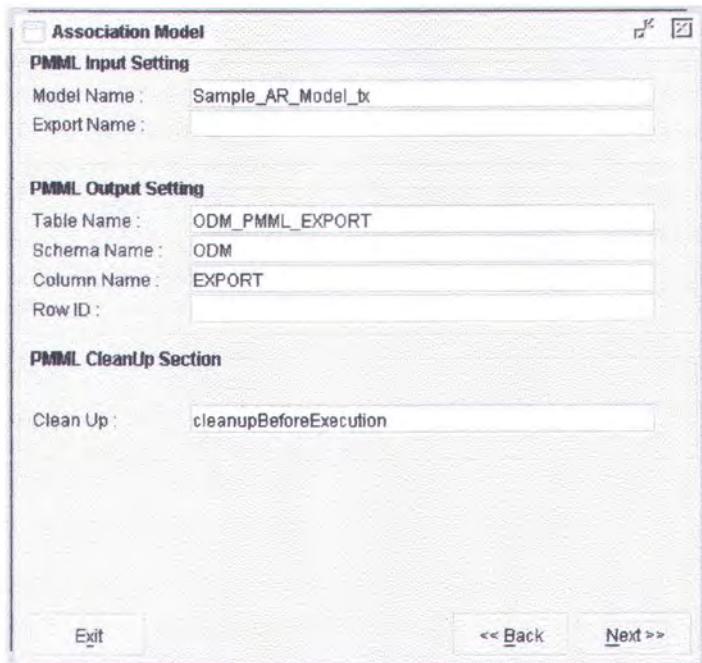
Setelah tombol OK ditekan maka pengguna mendapatkan laporan hasil pembuatan model yang tampak seperti berikut :



Gambar 4.10 Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 2

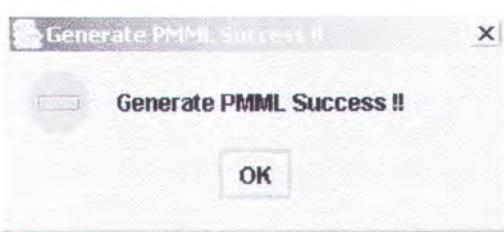
Pada tahap ini pengguna diberikan informasi status proses pembuatan model Kaidah Asosiasi, waktu yang dibutuhkan untuk membuat model, nama model yang dibentuk dan 5 *rule* teratas dari model yang berhasil dibangun berdasarkan *support* yang diberikan. Untuk merubah konfigurasi model yang sudah dibangun atau ingin membuat model dengan parameter yang berbeda, dapat dilakukan dengan menekan tombol *<<Back*. Sedangkan untuk meneruskan proses eksport model yang sudah dibentuk dapat dilakukan dengan menekan tombol *Next>>*.

Jika ditekan tombol *Next>>* maka pengguna mendapatkan tampilan berikut ini :



Gambar 4.11 Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 3

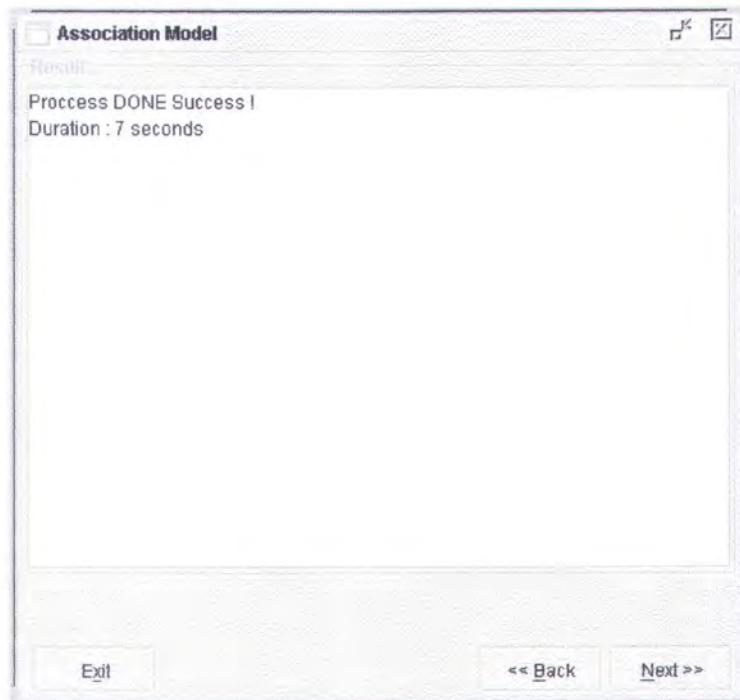
Pada tampilan ini pengguna dihadapkan pada pengisian parameter-parameter yang dibutuhkan untuk melakukan eksport model ke dalam format PMML. Setelah parameter diisi untuk melakukan proses eksport dapat dilakukan dengan menekan tombol Next >>. Jika proses eksport model berhasil maka muncul dialog konfirmasi seperti berikut :



Gambar 4.12 Dialog Konfirmasi Proses Eksport Sukses

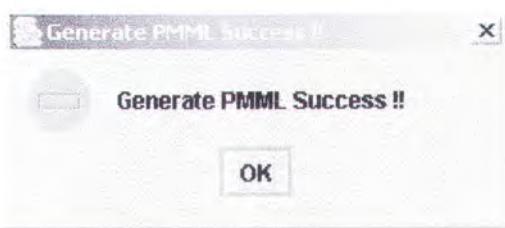
Seperti pada proses pembuatan model, pada tahap selanjutnya perangkat lunak mengeluarkan laporan proses eksport. Laporan tersebut berisi status proses

eksport dan waktu yang dibutuhkan untuk melakukan proses tersebut. Pada perangkat lunak tampak seperti berikut :



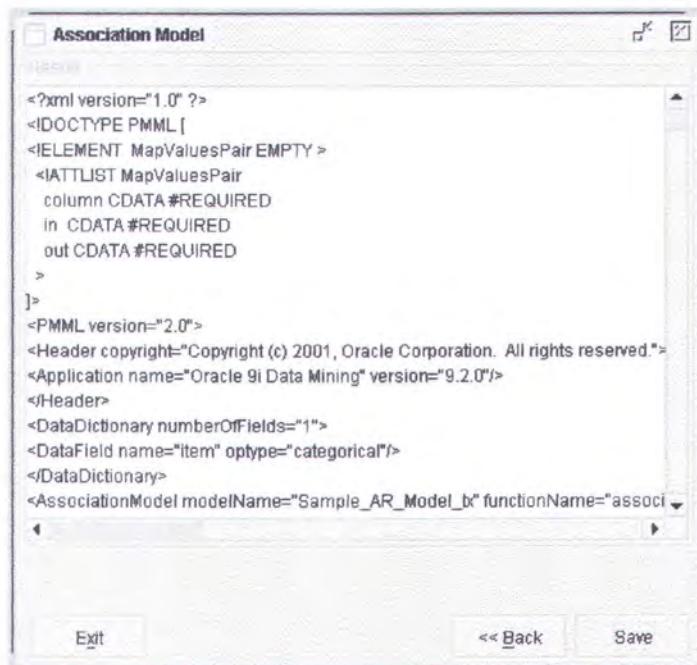
Gambar 4.13 Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 4

Untuk mengeksport model lain dapat dilakukan dengan menekan tombol <<Back. Hasil proses eksport model masih berada di dalam basisdata dalam format CLOB. Untuk mengambil dari basisdata dapat dilakukan dengan menekan tombol Next >>. Jika proses berjalan dengan sukses, tampak dialog konfirmasi sebagai berikut :



Gambar 4.14 Dialog Konfirmasi Pembuatan PMML

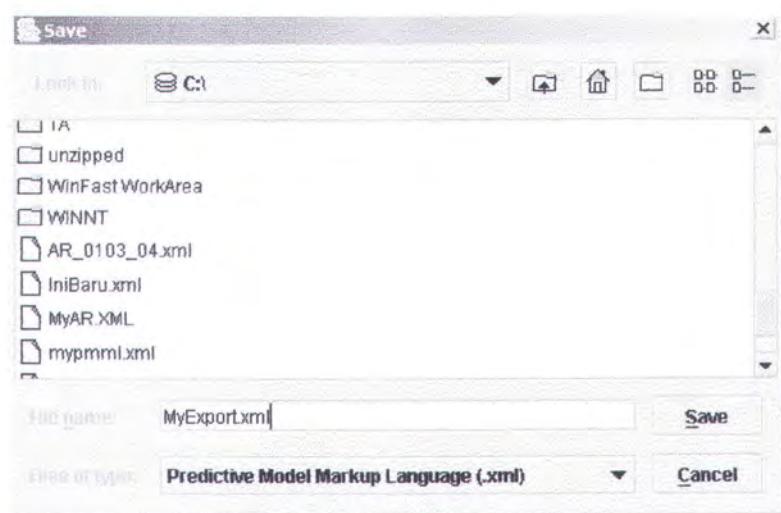
Setelah tombol OK dialog konfirmasi ditekan, maka dialog menuju tahap selanjutnya dan tampak seperti berikut :



Gambar 4.15 Dialog Pembuatan dan Eksport Kaidah Asosiasi Step 5

Pada tahap ini perangkat lunak telah mengkopikan dokumen yang berada dalam basisdata ke dalam dialog. Apabila diinginkan menyimpan model yang sekarang sudah dalam format PMML dapat dilakukan dengan menekan tombol *Save*. tampak dialog konfirmasi penyimpanan seperti berikut :





Gambar 4.16 Dialog Konfirmasi Penyimpanan PMML

Dengan menekan tombol *Save* maka model dalam format PMML tersimpan dalam media penyimpanan.

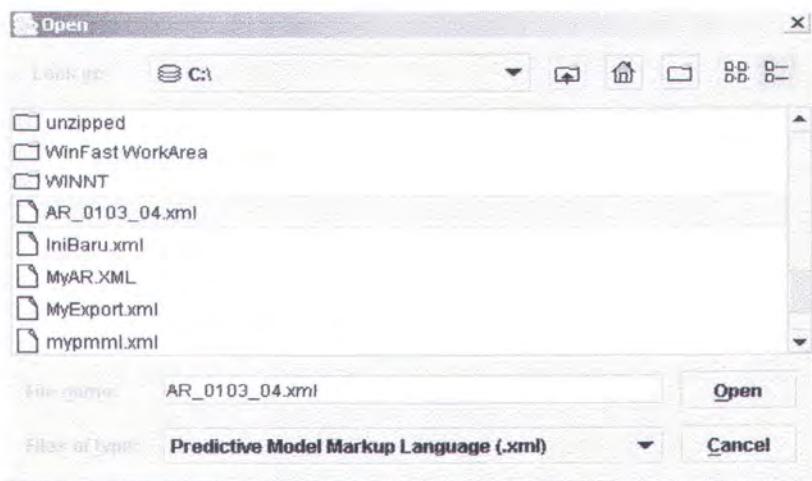
4.4.2.b Association Rules Import

Menu ini ditujukan untuk melakukan import model dalam format PMML yang berada dalam media penyimpanan (*file*) menuju model Oracle9i Data Mining. Dialog untuk import model tersebut tampak seperti berikut :



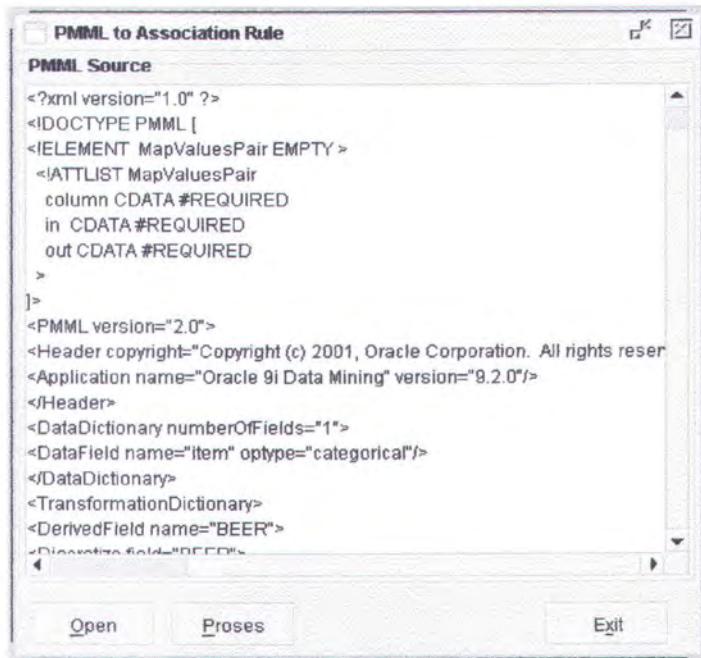
Gambar 4.17 Dialog Import Model PMML

Tombol Open pada dialog ini berfungsi untuk membuka berkas model dalam format PMML. Selanjutnya tampak dialog membuka berkas :



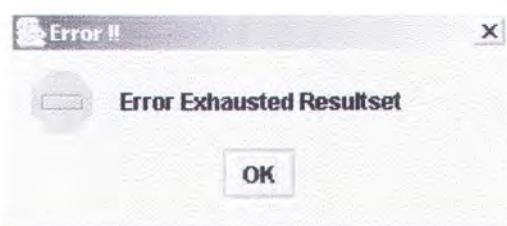
Gambar 4.18 Dialog Membuka Berkas PMML

Setelah memilih berkas dan menekan tombol Open, proses kembali ke dialog import model PMML. Pada kotak teks tampak isi berkas yang dibuka.



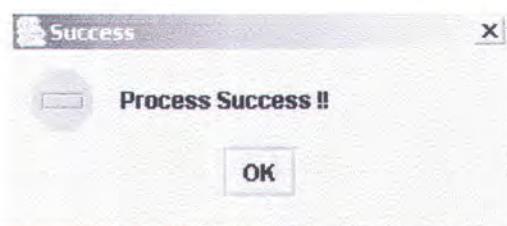
Gambar 4.19 Dialog Import PMML Setelah Berkas Dibuka

Pengguna diijinkan melakukan perubahan pada berkas yang diambil oleh dialog. Setelah berkas dibuka untuk memulai proses import model dilakukan dengan menekan tombol Proses. Jika terjadi kesalahan sintaks pada berkas yang dibuka keluar dialog konfirmasi, misalnya :



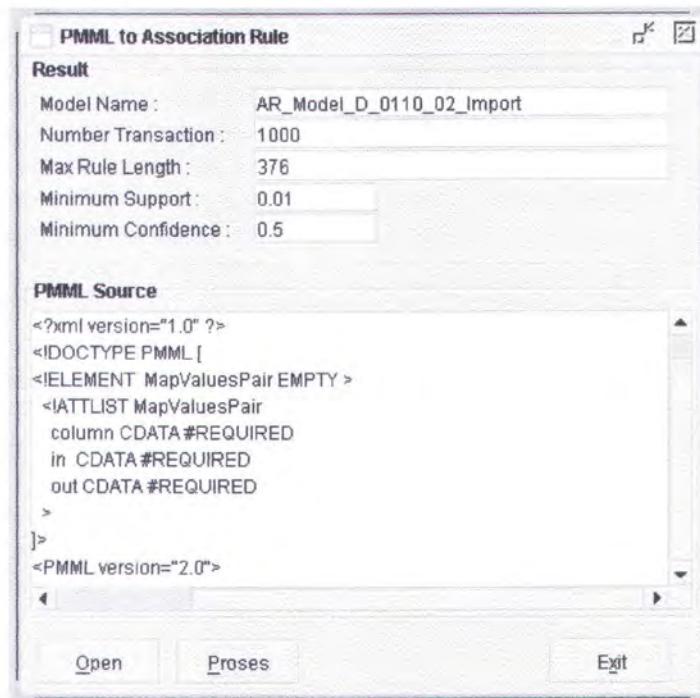
Gambar 4.20 Dialog Konfirmasi Kesalahan Sintaks

Perangkat lunak ini mempergunakan fitur dari Oracle9i untuk melakukan pengecekan terhadap validitas sintaks berkas yang dibuka dan juga validitas terhadap model yang bisa dibaca oleh Oracle9i Data Mining. Jika dokumen valid dan proses berhasil maka keluar dialog konfirmasi seperti berikut :



Gambar 4.21 Konfirmasi Proses Sukses

Jika proses berhasil dilakukan pada dialog import muncul laporan beberapa atribut pada model yaitu : nama model, minimum *support*, minimum *confidence*, jumlah transaksi, dan panjang maksimum dari *rule*.



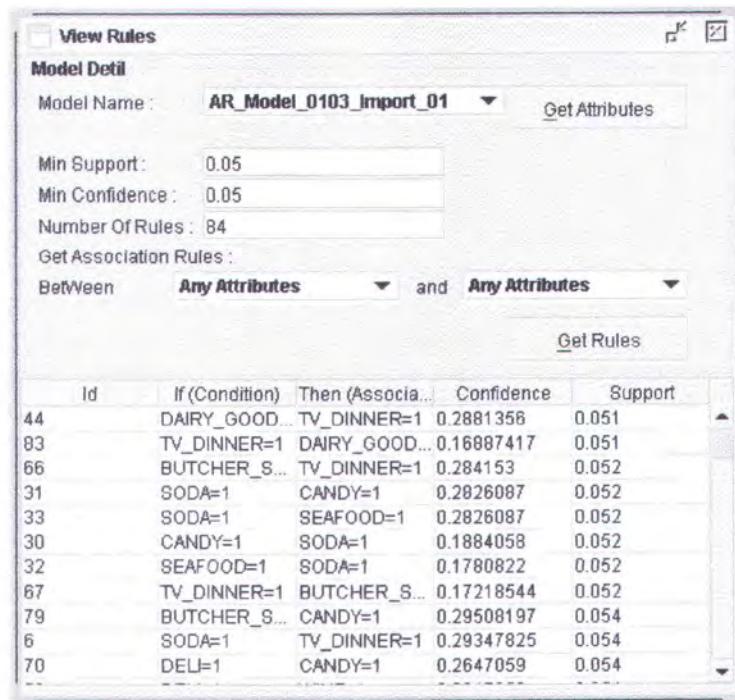
Gambar 4.22 Dialog Laporan Proses Import

4.4.2.c View Result

Menu berfungsi untuk memanggil dialog yang menunjukkan kaidah-kaidah yang dimiliki oleh suatu model yang dibuat maupun diimport oleh perangkat lunak ini. Dengan adanya fitur ini pengguna bisa mengetahui besarnya minimum *support*, minimum *confidence*, dan jumlah kaidah yang berhasil dibangkitkan. Selain itu juga bisa diketahui item-item dari transaksi yang dimodelkan.

Dengan adanya fitur ini dapat dilihat hubungan antar item, suatu item dengan semua item, maupun semua item dengan suatu item. Hubungan itu berupa besarnya *support* dan besarnya *confidence*. Hubungan antar item itu ditunjukkan oleh grid yang menempel pada dialog.

Dialog untuk melihat kaidah yang terbentuk itu tampak seperti berikut :



Gambar 4.23 Dialog View Result

Model name dapat dipilih dengan mengakses *combo box* yang berisikan model-model yang ada dalam skema ODM. Untuk melihat atribut yang dimiliki oleh suatu model dengan cara mengakses tombol *Get Attributes*. Item-item yang ada dalam model diisikan dalam dua *combo box*. Untuk melihat hubungan antar item dilakukan dengan mengakses tombol *Get Rules*. Hubungan antar item dapat terlihat di dalam grid.

4.4.3 Menu Naive Bayes

Pada menu ini terdapat sub menu yang berhubungan dengan Naive Bayes. Bila diklik maka tampak sub menu seperti berikut :

Naive Bayes Model Build and Export

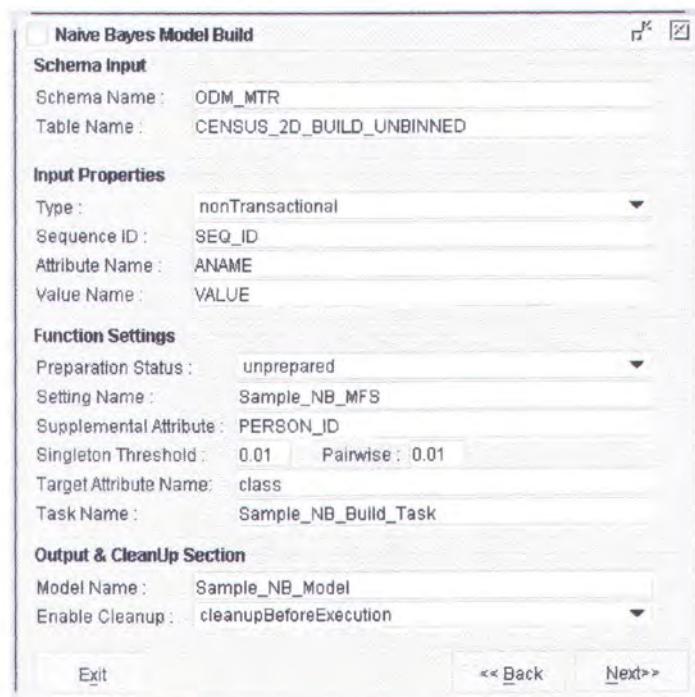
Naive Bayes Model Import

Gambar 4.24 Sub Menu Naive Bayes

Berikut ini *event* yang terjadi jika sub menu tersebut diakses oleh pengguna :

4.4.3.a Naive Bayes Model Build and Eksport

Menu ini berfungsi untuk memanggil dialog yang melakukan proses pembuatan model dan melakukan eksport ke dalam format PMML. Dialog tersebut tampak seperti berikut :



Gambar 4.25 Dialog Pembuatan dan Eksport Naive Bayes Step 1

Pada gambar tersebut tampak dialog yang dibutuhkan untuk membuat model dan melakukan eksport. Sama seperti pada Kaidah Asosiasi terdapat beberapa langkah (*step*) pengisian dari pembuatan model sampai dengan menghasilkan file dalam bentuk PMML.

Pada langkah pertama pengguna menginputkan parameter-parameter dalam pembuatan model Naive Bayes. Jika parameter-parameter tersebut sudah diisi

dengan lengkap, pengguna dapat menekan tombol Next>> untuk mendapatkan proses pembuatan model dan menuju langkah selanjutnya.

Selanjutnya, pengguna mendapatkan laporan dari pembuatan model yang telah dilakukan :



Gambar 4.26 Dialog Pembuatan dan Eksport Naive Bayes Step 2

Untuk langkah selanjutnya adalah langkah eksport model, melakukan pembuatan model format PMML, dan penyimpanan berkas PMML ke media penyimpanan. Langkah ini sama dengan langkah yang dimiliki proses ekport model Kaidah Asosiasi.

4.4.3.a Naive Bayes Model Import

Langkah-langkah pada proses ini sama dengan proses import model Kaidah Asosiasi. Perbedaan terdapat pada laporan model yang berhasil diimport.

BAB V

UJI COBA DAN EVALUASI

BAB V

UJI COBA DAN EVALUASI PERANGKAT LUNAK

Pada bab V Uji Coba dan Evaluasi Perangkat Lunak ini dibahas lingkungan uji coba yang digunakan untuk menjalankan perangkat lunak. Berikutnya diberikan pembahasan tabel yang dipergunakan untuk uji coba perangkat lunak nantinya, baik yang berupa tabel transaksional maupun tabel yang non-transaksional. Pada bab ini sebagian juga diberikan diuji kebenaran data keluarannya dengan mempergunakan perangkat lunak lainnya.

5.1 Lingkungan Uji Coba

Lingkungan sistem yang dipergunakan untuk uji coba perangkat lunak ini adalah sebagai berikut :

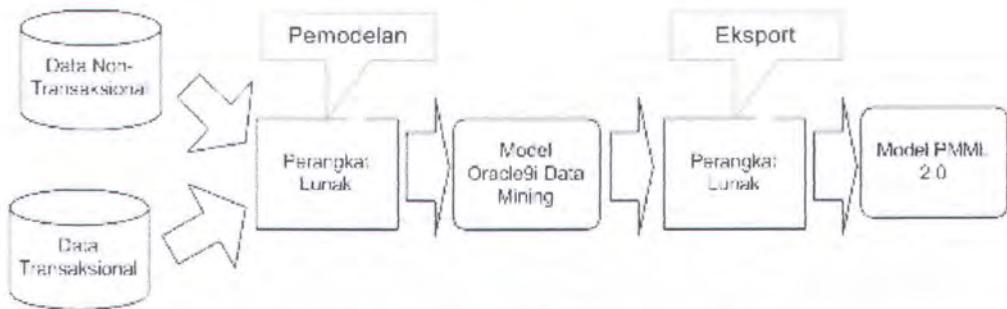
- Prosesor Intel Pentium IV 1.8 Ghz,
- Memori 500 Mb,
- Sistem Operasi Windows 2000 Service Pack 3,
- Server Database Oracle9i versi 9.2.0.1,
- Oracle9i Data Mining versi 9.2.0.3.

5.2 Pembuatan Model dan Eksport

Uji coba pertama adalah melakukan pemodelan dan meneruskan pemodelan tersebut dengan proses eksport. Pembuatan model dan eksport ini dilakukan terhadap kedua algoritma, Kaidah Asosiasi dan Naive Bayes. Data yang

adalah data yang tersimpan dalam tabel dalam format transaksional maupun non-transaksional dan mendefinisikan transaksi yang sama untuk kedua format tabel.

Untuk skenario uji coba terlihat seperti berikut :



Gambar 5.01 Skenario Uji Coba Pemodelan dan Eksport

Dari tabel yang diuji, transaksional maupun non-transaksional, dengan perangkat lunak dimodelkan sesuai dengan model yang didukung oleh Oracle9i Data Mining dengan konfigurasi tertentu sesuai dengan algoritma yang digunakan untuk pemodelan. Model ini disimpan dalam sebuah tabel dalam skema ODM dengan nama tabel ODM_MINING_MODEL dalam format BLOB (Binary Large Object) dan konfigurasi disimpan dalam tabel ODM_MINING_FUNCTION_SETTINGS [OR1-02].

Karena model dalam format BLOB, sehingga tidak dapat dilakukan analisa langsung terhadap model yang terbentuk. Untuk analisa model Kaidah Asosiasi dilakukan dengan menggunakan fitur yang tersedia oleh perangkat lunak, sedangkan untuk model Naive Bayes dianalisa dengan menggunakan ODMBrowser.

Setelah model terbentuk, tahap uji coba berikutnya adalah mencoba mengekspor dua model dari transaksional maupun non-transaksional ke dalam

format PMML. Proses ini bertujuan untuk melakukan analisa terhadap output yang berupa model dalam format PMML.

Jadi, tujuan dari uji coba yang pertama ini adalah :

1. menguji kebenaran hasil pemodelan dari perangkat lunak,
2. membandingkan hasil pemodelan dari data dengan tabel berformat transaksional maupun non-transaksional,
3. membandingkan hasil eksport data transaksional maupun non-transaksional

5.2.1 Kaidah Asosiasi

Untuk memodelkan Kaidah Asosiasi digunakan empat tabel, yaitu dua tabel transaksional dan dua tabel non-transaksional. Ada dua tabel transaksional dan non-transaksional yang merupakan tabel buatan sendiri yang berukuran kecil yang dipergunakan untuk mengecek kebenaran nilai model, sedangkan dua tabel yang lain merupakan tabel tansaksional dan non-transaksional yang sudah disediakan oleh Oracle9i Data Mining. Data yang disediakan oleh Oracle9i Data Mining itu adalah tabel MARKET_BASKET_2D_BINNED untuk tabel non-transaksional dan MARKET_BASKET_TX_BINNED untuk transaksional. Data-data pada tabel tersebut menunjukkan transaksi yang terjadi pada sebuah toko penjual bahan makanan. Pada tabel 2D (non-transaksional), setiap baris menunjukkan sebuah transaksi yang terjadi dan bernilai 1 untuk kolom produk yang dibeli.

Sedangkan tabel uji yang berukuran kecil berisi transaksi seperti berikut ini [TYA-02] :

Transaksi	Item yang dibeli
1	ACTW
2	CDW
3	ACTW
4	ACDW
5	ACDTW
6	CDT

Tabel 5.01 Transaksi yang Diuji

Pada tahapan berikut terlihat semua *frequent itemset*, yaitu yang memiliki nilai *support* sama atau lebih besar dari 3 (asumsi : $\text{min_sup} = 50\%$). Itemset CDW dan ACTW yang tercetak tebal adalah maksimal *frequent itemset*. Pada gambar terlihat semua Kaidah Asosiasi yang memiliki nilai *confidence* 100%.

Data transaksional dan data non-transaksional yang ada dalam basisdata tampak seperti berikut :

Tabel 5.02 Data TABLE_UJI_TRANSAKSIONAL

SEQUENCE_ID	ATTRIBUTE_NAME	VALUE
1	A	1
1	C	1
1	T	1
1	W	1
2	C	1
2	D	1
2	W	1
3	A	1
3	C	1
3	T	1
3	W	1
4	A	1
4	C	1
4	D	1

SEQUENCE_ID	ATTRIBUTE_NAME	VALUE
4	W	1
5	A	1
5	C	1
5	D	1
5	T	1
5	W	1
6	C	1
6	D	1
6	T	1

Tabel 5.03 TABEL_UJI_NONTRANSAKSIONAL

A	C	D	T	W
1	1	0	1	1
0	1	1	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1
0	1	1	1	0

Tabel 5.04 Frequent Itemset ($min_sup=50\%$)

Support	Itemset
100%	C
83%	W, CW
67%	A, D, T, AC, AW, CD, CT, ACW
50%	AT, DW, TW, ACT, ATW, CDW, CTW, ACTW

ASSOCIATION RULES (min_comf=100%)

A → C (4\4)	AC → W (4\4)	TW → C (3\3)
A → W (4\4)	AT → C (3\3)	AT → CW (3\3)
A → CW (4\4)	AT → W (3\3)	TW → AC (3\3)
D → C (4\4)	AW → C (4\4)	ACT → W (3\3)
T → C (4\4)	DW → C (3\3)	ATW → C (3\3)
W → C (5\5)	TW → A (3\3)	CTW → A (3\3)

Gambar 5.01 Data Kaidah Asosiasi

Pada uji coba yang pertama dipergunakan tabel transaksional dan non-transaksional yang berukuran kecil dengan nilai minimum *support* 50% dan minimum *confidence* sebesar 100%. Untuk melihat Kaidah Asosiasi yang terbentuk digunakan fasilitas *View Result* yang ada dalam perangkat lunak ini.

Dari dialog *View Result* didapatkan kaidah-kaidah yang sama seperti berikut ini :

View Rules

Model Detil

Model Name : UJI_Transaksional Get Attributes

Min Support: 0.5
Min Confidence: 1.0
Number Of Rules: 15

Get Association Rules:

Between Any Attributes and Any Attributes

Get Rules

Id	If (Condition)	Then (Associa...)	Confidence	Support
23	C=1 and T=1 ... A=1	A=1	1.0	0.5
22	A=1 and T=1 a... C=1	C=1	1.0	0.5
21	A=1 and C=1 ... W=1	W=1	1.0	0.5
10	T=1 and W=1	A=1	1.0	0.5
9	A=1 and D=1	C=1	1.0	0.5
18	T=1 and T=1	C=1	1.0	0.5
18	W=1 and W=1	C=1	1.0	0.5
17	A=1 and T=1	W=1	1.0	0.5
11	A=1 and W=1	C=1	1.0	0.6666667
12	A=1 and C=1	W=1	1.0	0.6666667
3	A=1	C=1	1.0	0.6666667

Gambar 5.02 Hasil Uji Tabel Transaksional

Lebih jelasnya dapat terlihat pada tabel berikut ini :

Tabel 5.05 Hasil Uji dengan Perangkat Lunak

if (condition)	then (association)	confidence	support
T=1	C=1	1.0	0.66667
A=1	C=1	1.0	0.66667
A=1	W=1	1.0	0.66667
D=1	C=1	1.0	0.66667
W=1	C=1	1.0	0.83333
A=1 And T=1	C=1	1.0	0.5
W=1 And T=1	A=1	1.0	0.5
A=1 And W=1	C=1	1.0	0.66667
A=1 And C=1	W=1	1.0	0.66667
D=1 And W=1	C=1	1.0	0.5
A=1 And T=1	W=1	1.0	0.5
W=1 And T=1	C=1	1.0	0.5
A=1 And C=1 And T=1	W=1	1.0	0.5
A=1 And W=1 And T=1	C=1	1.0	0.5
W=1 And T=1 And C=1	A=1	1.0	0.5

Sedangkan untuk tabel non transaksional tampak seperti berikut :

Id	If (Condition)	Then (Association)	Confidence	Support
23	C=1 and T=1 ... A=1		1.0	0.5
22	A=1 and T=1 a... C=1		1.0	0.5
21	A=1 and C=1 ... W=1		1.0	0.5
12	T=1 and W=1 ... A=1		1.0	0.5
11	A=1 and D=1 ... C=1		1.0	0.5
18	T=1 and T=1 ... C=1		1.0	0.5
13	W=1 and W=1 ... C=1		1.0	0.5
17	A=1 and T=1 ... W=1		1.0	0.5
9	A=1 and W=1 ... C=1		1.0	0.6666667
10	A=1 and C=1 ... W=1		1.0	0.6666667
3	A=1 ... C=1		1.0	0.6666667

Gambar 5.03 Hasil Uji Tabel non-Transaksional

Apabila konfigurasi fungsi yang sama dipergunakan untuk tabel uji yang telah disediakan Oracle9i Data Mining muncul dialog kesalahan. Dengan alat bantu dari ODMBrowser didapatkan status proses seperti berikut :

```
Message
ORA-20010: ODM_ASSOCIATION_RULE_MODEL.BUILD: Model building
failed with error (ORA-20012:
ODM_ASSOCIATION_RULE_MODEL.BUILD: Model has no rules.).
ORA-06512: at "ODM.ODM_ASSOCIATION_RULE_MODEL", line 145
ORA-06512: at "ODM.ODM_ASSOCIATION_RULE_MODEL", line 3424
ORA-20012: ODM_ASSOCIATION_RULE_MODEL.BUILD: Model has no
rules.
ORA-06512: at line 1
```

Gambar 5.04 Kesalahan pada Pembuatan Model Kaidah Asosiasi

Sedangkan untuk ujicoba dengan mempergunakan data tabel yang telah disediakan oleh Oracle9i Data Mining menghasilkan 93 aturan. Aturan ini didapatkan dengan mempergunakan minimum *support* sebesar 5% dan minimum *confidence* sebesar 5%.

Kaidah-kaidah yang didapatkan tampak seperti berikut :

View Rules																																																																													
Model Detil																																																																													
Model Name:		UJI_TABLE_ORACLE		Get Attributes																																																																									
Min Support:	0.05																																																																												
Min Confidence:	0.05																																																																												
Number Of Rules:	93																																																																												
Get Association Rules:																																																																													
Between	Any Attributes	and	Any Attributes																																																																										
Get Rules																																																																													
<table border="1"> <thead> <tr> <th colspan="2">Id</th><th>If (Condition)</th><th>Then (Associative Rule)</th><th>Confidence</th><th>Support</th></tr> </thead> <tbody> <tr> <td>101</td><td></td><td>BEER=1 and SEAFOOD=1</td><td>TV_DINNER=1</td><td>0.5882353</td><td>0.05</td></tr> <tr> <td>107</td><td></td><td>SEAFOOD=1 and TV_DL...</td><td>BEER=1</td><td>0.5555556</td><td>0.05</td></tr> <tr> <td>105</td><td></td><td>BEER=1 and TV_DINNE...</td><td>SEAFOOD=1</td><td>0.294117...</td><td>0.05</td></tr> <tr> <td>108</td><td></td><td>CANNED_GOODS=1 an...</td><td>TV_DINNER=...</td><td>0.5730337</td><td>0.051</td></tr> <tr> <td>106</td><td></td><td>SEAFOOD=1 and TV_DL...</td><td>CANNED_G...</td><td>0.566666...</td><td>0.051</td></tr> <tr> <td>102</td><td></td><td>CANNED_GOODS=1 an...</td><td>SEAFOOD=1</td><td>0.2947977</td><td>0.051</td></tr> <tr> <td>73</td><td></td><td>DAIRY_GOODS=1</td><td>TV_DINNER=...</td><td>0.2881356</td><td>0.051</td></tr> <tr> <td>17</td><td></td><td>TV_DINNER=1</td><td>DAIRY_GOO...</td><td>0.168874...</td><td>0.051</td></tr> <tr> <td>39</td><td></td><td>BUTCHER_SHOP=1</td><td>TV_DINNER=...</td><td>0.284153</td><td>0.052</td></tr> <tr> <td>60</td><td></td><td>SODA=1</td><td>CANDY=1</td><td>0.2826087</td><td>0.052</td></tr> <tr> <td>62</td><td></td><td>SODA=1</td><td>SEAFOOD=1</td><td>0.2826087</td><td>0.052</td></tr> </tbody> </table>						Id		If (Condition)	Then (Associative Rule)	Confidence	Support	101		BEER=1 and SEAFOOD=1	TV_DINNER=1	0.5882353	0.05	107		SEAFOOD=1 and TV_DL...	BEER=1	0.5555556	0.05	105		BEER=1 and TV_DINNE...	SEAFOOD=1	0.294117...	0.05	108		CANNED_GOODS=1 an...	TV_DINNER=...	0.5730337	0.051	106		SEAFOOD=1 and TV_DL...	CANNED_G...	0.566666...	0.051	102		CANNED_GOODS=1 an...	SEAFOOD=1	0.2947977	0.051	73		DAIRY_GOODS=1	TV_DINNER=...	0.2881356	0.051	17		TV_DINNER=1	DAIRY_GOO...	0.168874...	0.051	39		BUTCHER_SHOP=1	TV_DINNER=...	0.284153	0.052	60		SODA=1	CANDY=1	0.2826087	0.052	62		SODA=1	SEAFOOD=1	0.2826087	0.052
Id		If (Condition)	Then (Associative Rule)	Confidence	Support																																																																								
101		BEER=1 and SEAFOOD=1	TV_DINNER=1	0.5882353	0.05																																																																								
107		SEAFOOD=1 and TV_DL...	BEER=1	0.5555556	0.05																																																																								
105		BEER=1 and TV_DINNE...	SEAFOOD=1	0.294117...	0.05																																																																								
108		CANNED_GOODS=1 an...	TV_DINNER=...	0.5730337	0.051																																																																								
106		SEAFOOD=1 and TV_DL...	CANNED_G...	0.566666...	0.051																																																																								
102		CANNED_GOODS=1 an...	SEAFOOD=1	0.2947977	0.051																																																																								
73		DAIRY_GOODS=1	TV_DINNER=...	0.2881356	0.051																																																																								
17		TV_DINNER=1	DAIRY_GOO...	0.168874...	0.051																																																																								
39		BUTCHER_SHOP=1	TV_DINNER=...	0.284153	0.052																																																																								
60		SODA=1	CANDY=1	0.2826087	0.052																																																																								
62		SODA=1	SEAFOOD=1	0.2826087	0.052																																																																								

Gambar 5.05 Hasil Pemodelan untuk Tabel Bawaan Oracle



Uji coba selanjutnya adalah dengan eksport model ke dalam format PMML. Proses eksport dilakukan terhadap model yang sudah terbentuk terhadap data transaksi berukuran kecil. Untuk data transaksional menghasilkan dokumen PMML seperti berikut :

```

<?xml version="1.0" ?>
<!DOCTYPE PMML [
<!ELEMENT MapValuesPair EMPTY >
<!ATTLIST MapValuesPair
  column CDATA #REQUIRED
  in CDATA #REQUIRED
  out CDATA #REQUIRED
>
]>
<PMML version="2.0">
<Header copyright="Copyright (c) 2001, Oracle Corporation. All rights reserved.">
<Application name="Oracle 9i Data Mining" version="9.2.0"/>
</Header>
<DataDictionary numberOfFields="1">
<DataField name="item" optype="categorical"/>
</DataDictionary>
<AssociationModel          modelName="UJI_TRANSAKSIONAL"
functionName="associationRules"         numberOfTransactions="6"
maxNumberOfItemsPerTA="5"          avgNumberOfItemsPerTA="3.8333333"
minimumSupport="0.5"          minimumConfidence="1.0"          numberofItems="5"
numberOfItemsets="19"          numberofRules="15">
<MiningSchema>
<MiningField name="item" usageType="active"/>
</MiningSchema>
<Item id="1" value="A|I"/>
<Item id="2" value="C|I"/>
<Item id="3" value="D|I"/>
<Item id="4" value="T|I"/>
<Item id="5" value="W|I"/>
<Itemset id="1" support="0.666666666666666666666666666666666666667"
numberOfItems="1">
  <ItemRef itemRef="1"/>
</Itemset>
<Itemset id="2" support="1" numberOfItems="1">
  <ItemRef itemRef="2"/>
</Itemset>
<Itemset id="3" support="0.66666666666666666666666666666666666667"

```



```
<ItemRef itemRef="5"/>
</Itemset>
<Itemset id="12" support="0.5" numberOfItems="2">
    <ItemRef itemRef="3"/>
    <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="13" support="0.5" numberOfItems="2">
    <ItemRef itemRef="4"/>
    <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="14" support="0.5" numberOfItems="3">
    <ItemRef itemRef="1"/>
    <ItemRef itemRef="2"/>
    <ItemRef itemRef="4"/>
</Itemset>
<Itemset id="15" support="0.666666666666666666666666666666666666667" numberOfItems="3">
    <ItemRef itemRef="1"/>
    <ItemRef itemRef="2"/>
    <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="16" support="0.5" numberOfItems="3">
    <ItemRef itemRef="1"/>
    <ItemRef itemRef="4"/>
    <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="17" support="0.5" numberOfItems="3">
    <ItemRef itemRef="2"/>
    <ItemRef itemRef="3"/>
    <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="18" support="0.5" numberOfItems="3">
    <ItemRef itemRef="2"/>
    <ItemRef itemRef="4"/>
    <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="19" support="0.5" numberOfItems="4">
    <ItemRef itemRef="1"/>
    <ItemRef itemRef="2"/>
    <ItemRef itemRef="4"/>
    <ItemRef itemRef="5"/>
</Itemset>
<AssociationRule id="1" support="0.66666666666666666666666666666667">
```

```

confidence="1" antecedent="4" consequent="2"/>
<AssociationRule
support="0.666666666666666666666666666666666666667"
confidence="1" antecedent="1" consequent="2"/>
<AssociationRule
support="0.66666666666666666666666666666666666667"
confidence="1" antecedent="1" consequent="5"/>
<AssociationRule
support="0.66666666666666666666666666666666666667"
confidence="1" antecedent="3" consequent="2"/>
<AssociationRule
support="0.833333333333333333333333333333333333333"
confidence="1" antecedent="5" consequent="2"/>
<AssociationRule support="0.5" confidence="1" antecedent="7"
consequent="2"/>
<AssociationRule support="0.5" confidence="1" antecedent="13"
consequent="1"/>
<AssociationRule support="0.5" confidence="1" antecedent="7"
consequent="5"/>
<AssociationRule support="0.5" confidence="1" antecedent="13"
consequent="2"/>
<AssociationRule
support="0.66666666666666666666666666666666666667"
confidence="1" antecedent="8" consequent="2"/>
<AssociationRule
support="0.66666666666666666666666666666666666667"
confidence="1" antecedent="6" consequent="5"/>
<AssociationRule support="0.5" confidence="1" antecedent="12"
consequent="2"/>
<AssociationRule support="0.5" confidence="1" antecedent="14"
consequent="5"/>
<AssociationRule support="0.5" confidence="1" antecedent="16"
consequent="2"/>
<AssociationRule support="0.5" confidence="1" antecedent="18"
consequent="1"/>
</AssociationModel>
</PMML>
```

Gambar 5.06 Ouput Eksport ke PMML dari Tabel Transaksional

Setelah eksport model transaksional proses berikutnya adalah eksport terhadap data non-transaksional. Setelah proses eksport didapatkan model dalam format PMML seperti berikut :


```

</Itemset>
<Itemset id="13" support="0.5" numberOfItems="2">
  <ItemRef itemRef="4"/>
  <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="14" support="0.5" numberOfItems="3">
  <ItemRef itemRef="1"/>
  <ItemRef itemRef="2"/>
  <ItemRef itemRef="4"/>
</Itemset>
<Itemset id="15" support="0.666666666666666666666666666666666666666666666666667" numberOfItems="3">
  <ItemRef itemRef="1"/>
  <ItemRef itemRef="2"/>
  <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="16" support="0.5" numberOfItems="3">
  <ItemRef itemRef="1"/>
  <ItemRef itemRef="4"/>
  <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="17" support="0.5" numberOfItems="3">
  <ItemRef itemRef="2"/>
  <ItemRef itemRef="3"/>
  <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="18" support="0.5" numberOfItems="3">
  <ItemRef itemRef="2"/>
  <ItemRef itemRef="4"/>
  <ItemRef itemRef="5"/>
</Itemset>
<Itemset id="19" support="0.5" numberOfItems="4">
  <ItemRef itemRef="1"/>
  <ItemRef itemRef="2"/>
  <ItemRef itemRef="4"/>
  <ItemRef itemRef="5"/>
</Itemset>
<AssociationRule
  support="0.666666666666666666666666666666666666666666666666667" confidence="1"
  antecedent="4" consequent="2"/>
  <AssociationRule
    support="0.6666666666666666666666666666666666666666666666667" confidence="1"
    antecedent="1" consequent="2"/>
    <AssociationRule

```

```
support="0.6666666666666666666666666666666666666667" confidence="1"  
antecedent="1" consequent="5"/>  
    <AssociationRule  
support="0.6666666666666666666666666666666666666667" confidence="1"  
antecedent="3" consequent="2"/>  
    <AssociationRule  
support="0.833333333333333333333333333333333333333" confidence="1"  
antecedent="5" consequent="2"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="7"  
consequent="2"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="13"  
consequent="1"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="7"  
consequent="5"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="13"  
consequent="2"/>  
        <AssociationRule  
support="0.66666666666666666666666666666666666667" confidence="1"  
antecedent="8" consequent="2"/>  
        <AssociationRule  
support="0.66666666666666666666666666666666666667" confidence="1"  
antecedent="6" consequent="5"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="12"  
consequent="2"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="14"  
consequent="5"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="16"  
consequent="2"/>  
        <AssociationRule support="0.5" confidence="1" antecedent="18"  
consequent="1"/>  
    </AssociationModel>  
</PMML>
```

Gambar 5.07 Ouput Eksport ke PMML dari Tabel non-transaksional

Dari proses pembuatan model tampak bahwa untuk data yang kecil tidak didapatkan permasalahan proses. Hasil yang didapatkan sesuai dengan nilai jika dipergunakan perhitungan manual. Hal ini berlaku untuk tabel yang transaksional maupun tabel non-transaksional.

Sedangkan untuk proses eksport, hasil eksport ke dalam format PMML dari model yang dibangun dari tabel transaksional sama dengan tabel non-transaksional.

Untuk proses terhadap tabel yang telah disediakan oleh Oracle9i Data Mining, yaitu tabel dengan jumlah transaksi adalah 1000 transaksi, terjadi kesalahan jika dipergunakan setting yang sama, minimum *support* sebesar 0.5 dan minimum *confidence* sebesar 1, dengan tabel buatan sendiri yang memiliki 6 transaksi. Setelah dicoba dengan berbagai nilai, kesalahan ini tidak muncul setelah diisi nilai kurang dari sama dengan 0.1, untuk minimum *support* dan minimum *confidence*. Hasil model sama untuk model dari tabel transaksional maupun non-transaksional. Hal ini juga berlaku untuk proses eksport model ke dalam format PMML.

5.2.2 Naive Bayes

Proses pembuatan model Naive Bayes menggunakan tabel yang telah disediakan oleh Oracle9i Data Mining yaitu CENSUS_2D_BUILD_UNBINNED. Tabel ini berisi sensus terhadap penduduk di Amerika pada suatu waktu. Model ini mempergunakan setting *singleton threshold* sebesar 1% dan *pairwise threshold* sebesar 1%.

Setelah proses hasil dapat dilihat dengan mempergunakan alat bantu yang disediakan oleh Oracle, yaitu ODMBrowser. Hasil pembuatan model tampak seperti berikut :

Name	Data Type	Mining Type	Usage	Auto Binning
PERSON_ID	Int	Numerical	Suppl...	yes
AGE	Float	Numerical	Active	yes
WORKCLASS	String	Categorical	Active	yes
WEIGHT	Float	Numerical	Active	yes
EDUCATION	String	Categorical	Active	yes
EDUCATION_NUM	Float	Numerical	Active	yes
MARITAL_STATUS	String	Categorical	Active	yes
OCCUPATION	String	Categorical	Active	yes
RELATIONSHIP	String	Categorical	Active	yes
RACE	String	Categorical	Active	yes
SEX	String	Categorical	Active	yes
CAPITAL_GAIN	Float	Numerical	Active	yes
CAPITAL_LOSS	Float	Numerical	Active	yes
HOURS_PER_WEEK	Float	Numerical	Active	yes
NATIVE_COUNTRY	String	Categorical	Active	yes
CLASS	Int	Categorical	Target	no

Gambar 5.08 Output Pemodel Naive Bayes dengan ODMBrowser

Dari model ini dengan perangkat lunak dilanjutkan dengan proses eksport ke dalam format PMML. Hasil eksport dalam format PMML tampak seperti berikut :

```
<?xml version="1.0" ?>
<!DOCTYPE PMML [
<!ELEMENT MapValuesPair EMPTY >
<!ATTLIST MapValuesPair
  column CDATA #REQUIRED
  in CDATA #REQUIRED
  out CDATA #REQUIRED
>
]>
<PMML version="2.0">
<Header copyright="Copyright (c) 2001, Oracle Corporation. All rights reserved.">
<Application name="Oracle 9i Data Mining" version="9.2.0"/>
</Header>
<DataDictionary numberOfFields="16">
<DataField name="PERSON_ID" optype="continuous">
</DataField>
<DataField name="AGE" optype="continuous">
</DataField>
```

```

<DataField name="WORKCLASS" optype="categorical">
</DataField>
<DataField name="WEIGHT" optype="continuous">
</DataField>
<DataField name="EDUCATION" optype="categorical">
</DataField>
<DataField name="EDUCATION_NUM" optype="continuous">
</DataField>
<DataField name="MARITAL_STATUS" optype="categorical">
</DataField>
<DataField name="OCCUPATION" optype="categorical">
</DataField>
<DataField name="RELATIONSHIP" optype="categorical">
</DataField>
<DataField name="RACE" optype="categorical">
</DataField>
<DataField name="SEX" optype="categorical">
</DataField>
<DataField name="CAPITAL_GAIN" optype="continuous">
</DataField>
<DataField name="CAPITAL_LOSS" optype="continuous">
</DataField>
<DataField name="HOURS_PER_WEEK" optype="continuous">
</DataField>
<DataField name="NATIVE_COUNTRY" optype="categorical">
</DataField>
<DataField name="CLASS" optype="categorical">
</DataField>
</DataDictionary>
<TransformationDictionary>
<DerivedField name="AGE">
<Discretize field="AGE">
<DiscretizeBin binValue="17-31.6">
<Interval closure="closedOpen" leftMargin="17.0" rightMargin="31.6"/>
</DiscretizeBin>
<DiscretizeBin binValue="31.6-46.2">
<Interval closure="closedOpen" leftMargin="31.6" rightMargin="46.2"/>
</DiscretizeBin>
<DiscretizeBin binValue="46.2-60.8">
<Interval closure="closedOpen" leftMargin="46.2" rightMargin="60.8"/>
</DiscretizeBin>
<DiscretizeBin binValue="60.8-75.4">
<Interval closure="closedOpen" leftMargin="60.8" rightMargin="75.4"/>
</DiscretizeBin>
<DiscretizeBin binValue="75.4-90">
<Interval closure="closedOpen" leftMargin="75.4" rightMargin="90.0"/>
</DiscretizeBin>
</Discretize>
</DerivedField>
<DerivedField name="CAPITAL_GAIN">
<Discretize field="CAPITAL_GAIN">
<DiscretizeBin binValue="0-19999.8">
<Interval closure="closedOpen" leftMargin="0.0" rightMargin="19999.8"/>
</DiscretizeBin>

```

```

<DiscretizeBin binValue="1-20.6">
<Interval closure="closedOpen" leftMargin="1.0" rightMargin="20.6"/>
</DiscretizeBin>
<DiscretizeBin binValue="20.6-40.2">
<Interval closure="closedOpen" leftMargin="20.6" rightMargin="40.2"/>
</DiscretizeBin>
<DiscretizeBin binValue="40.2-59.8">
<Interval closure="closedOpen" leftMargin="40.2" rightMargin="59.8"/>
</DiscretizeBin>
<DiscretizeBin binValue="59.8-79.4">
<Interval closure="closedOpen" leftMargin="59.8" rightMargin="79.4"/>
</DiscretizeBin>
<DiscretizeBin binValue="79.4-99">
<Interval closure="closedOpen" leftMargin="79.4" rightMargin="99.0"/>
</DiscretizeBin>
</Discretize>
</DerivedField>
<DerivedField name="WEIGHT">
<Discretize field="WEIGHT">
<DiscretizeBin binValue="13769-230505.8">
<Interval closure="closedOpen" leftMargin="13769.0" rightMargin="230505.8"/>
</DiscretizeBin>
<DiscretizeBin binValue="230505.8-447242.6">
<Interval closure="closedOpen" leftMargin="230505.8" rightMargin="447242.6"/>
</DiscretizeBin>
<DiscretizeBin binValue="447242.6-663979.4">
<Interval closure="closedOpen" leftMargin="447242.6" rightMargin="663979.4"/>
</DiscretizeBin>
<DiscretizeBin binValue="663979.4-880716.2">
<Interval closure="closedOpen" leftMargin="663979.4" rightMargin="880716.2"/>
</DiscretizeBin>
<DiscretizeBin binValue="880716.2-1097453">
<Interval closure="closedOpen" leftMargin="880716.2" rightMargin="1097453.0"/>
</DiscretizeBin>
</Discretize>
</DerivedField>
<DerivedField name="EDUCATION">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">
<FieldColumnPair field="EDUCATION" column="CATEGORY"/>
<InlineTable>
<Extension>
<MapValuesPair column="CATEGORY" in="HS-grad" out="HS-grad"/>
<MapValuesPair column="CATEGORY" in="&lt; Bach." out="&lt; Bach."/>
<MapValuesPair column="CATEGORY" in="Bach." out="Bach."/>
<MapValuesPair column="CATEGORY" in="Masters" out="Masters"/>
<MapValuesPair column="CATEGORY" in="Assoc-V" out="Assoc-V"/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
<DerivedField name="MARITAL_STATUS">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">
<FieldColumnPair field="MARITAL_STATUS" column="CATEGORY"/>
<InlineTable>

```

```

<Extension>
<MapValuesPair column="CATEGORY" in="Married" out="Married"/>
<MapValuesPair column="CATEGORY" in="NeverM" out="NeverM"/>
<MapValuesPair column="CATEGORY" in="Divorc." out="Divorc."/>
<MapValuesPair column="CATEGORY" in="Widowed" out="Widowed"/>
<MapValuesPair column="CATEGORY" in="Separ." out="Separ."/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
<DerivedField name="NATIVE_COUNTRY">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">
<FieldColumnPair field="NATIVE_COUNTRY" column="CATEGORY"/>
<InlineTable>
<Extension>
<MapValuesPair column="CATEGORY" in="U.S." out="U.S."/>
<MapValuesPair column="CATEGORY" in="Mexico" out="Mexico"/>
<MapValuesPair column="CATEGORY" in="Phil." out="Phil."/>
<MapValuesPair column="CATEGORY" in="El-Sal." out="El-Sal."/>
<MapValuesPair column="CATEGORY" in="England" out="England"/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
<DerivedField name="OCCUPATION">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">
<FieldColumnPair field="OCCUPATION" column="CATEGORY"/>
<InlineTable>
<Extension>
<MapValuesPair column="CATEGORY" in="Prof." out="Prof."/>
<MapValuesPair column="CATEGORY" in="Crafts" out="Crafts"/>
<MapValuesPair column="CATEGORY" in="Exec." out="Exec."/>
<MapValuesPair column="CATEGORY" in="Sales" out="Sales"/>
<MapValuesPair column="CATEGORY" in="Cleric." out="Cleric."/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
<DerivedField name="RACE">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">
<FieldColumnPair field="RACE" column="CATEGORY"/>
<InlineTable>
<Extension>
<MapValuesPair column="CATEGORY" in="White" out="White"/>
<MapValuesPair column="CATEGORY" in="Black" out="Black"/>
<MapValuesPair column="CATEGORY" in="Asian" out="Asian"/>
<MapValuesPair column="CATEGORY" in="AIndian" out="AIndian"/>
<MapValuesPair column="CATEGORY" in="Other" out="Other"/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
<DerivedField name="RELATIONSHIP">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">

```

```

<FieldColumnPair field="RELATIONSHIP" column="CATEGORY"/>
<InlineTable>
<Extension>
<MapValuesPair column="CATEGORY" in="Husband" out="Husband"/>
<MapValuesPair column="CATEGORY" in="NotinFa" out="NotinFa"/>
<MapValuesPair column="CATEGORY" in="O-child" out="O-child"/>
<MapValuesPair column="CATEGORY" in="Unmarr." out="Unmarr."/>
<MapValuesPair column="CATEGORY" in="Wife" out="Wife"/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
<DerivedField name="SEX">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">
<FieldColumnPair field="SEX" column="CATEGORY"/>
<InlineTable>
<Extension>
<MapValuesPair column="CATEGORY" in="Male" out="Male"/>
<MapValuesPair column="CATEGORY" in="Female" out="Female"/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
<DerivedField name="WORKCLASS">
<MapValues outputColumn="DISPLAY_NAME" defaultValue="OTHER_CATEGORY">
<FieldColumnPair field="WORKCLASS" column="CATEGORY"/>
<InlineTable>
<Extension>
<MapValuesPair column="CATEGORY" in="Private" out="Private"/>
<MapValuesPair column="CATEGORY" in="SelfENI" out="SelfENI"/>
<MapValuesPair column="CATEGORY" in="Loc-gov" out="Loc-gov"/>
<MapValuesPair column="CATEGORY" in="SelfEI" out="SelfEI"/>
<MapValuesPair column="CATEGORY" in="Sta-gov" out="Sta-gov"/>
</Extension>
</InlineTable>
</MapValues>
</DerivedField>
</TransformationDictionary>
<NaiveBayesModel modelName="IMPORT_UJI_NAIVE_BAYES" threshold="0.000001">
<functionName="classification">
<MiningSchema>
<MiningField name = "AGE" usageType = "active"/>
<MiningField name = "WORKCLASS" usageType = "active"/>
<MiningField name = "WEIGHT" usageType = "active"/>
<MiningField name = "EDUCATION" usageType = "active"/>
<MiningField name = "EDUCATION_NUM" usageType = "active"/>
<MiningField name = "MARITAL_STATUS" usageType = "active"/>
<MiningField name = "OCCUPATION" usageType = "active"/>
<MiningField name = "RELATIONSHIP" usageType = "active"/>
<MiningField name = "RACE" usageType = "active"/>
<MiningField name = "SEX" usageType = "active"/>
<MiningField name = "CAPITAL_GAIN" usageType = "active"/>
<MiningField name = "CAPITAL_LOSS" usageType = "active"/>
<MiningField name = "HOURS_PER_WEEK" usageType = "active"/>

```

```

<MiningField name = "NATIVE_COUNTRY" usageType = "active"/>
<MiningField name = "CLASS" usageType = "predicted"/>
<MiningField name = "PERSON_ID" usageType = "supplementary"/>
</MiningSchema>
<BayesInputs> <BayesInput fieldName="CAPITAL_LOSS"> <PairCounts value="1560-2340">
    <TargetValueCounts> <TargetValueCount value="0" count="50"/>
    <TargetValueCount value="1" count="48"/> </TargetValueCounts> </PairCounts>
    <PairCounts value="0-780"> <TargetValueCounts> <TargetValueCount value="0" count="2212"/>
        <TargetValueCount value="1" count="604"/> </TargetValueCounts>
    </PairCounts> </BayesInput> <BayesInput fieldName="HOURS_PER_WEEK">
    <PairCounts value="1-20.6"> <TargetValueCounts> <TargetValueCount value="0" count="222"/>
        <TargetValueCounts> </PairCounts> <PairCounts value="20.6-40.2">
        <TargetValueCounts> <TargetValueCount value="0" count="1541"/>
        <TargetValueCount value="1" count="339"/> </TargetValueCounts> </PairCounts>
    </PairCounts> <PairCounts value="40.2-59.8"> <TargetValueCounts> <TargetValueCount value="0" count="383"/>
        <TargetValueCount value="1" count="239"/> </TargetValueCounts>
    </PairCounts> <PairCounts value="59.8-79.4"> <TargetValueCounts>
        <TargetValueCount value="0" count="99"/> <TargetValueCount value="1" count="69"/>
    </TargetValueCounts> </PairCounts> </BayesInput> <BayesInput
fieldName="MARITAL_STATUS"> <PairCounts value="Divorc."/>
    <TargetValueCounts> <TargetValueCount value="0" count="372"/>
    <TargetValueCount value="1" count="31"/> </TargetValueCounts> </PairCounts>
    <PairCounts value="Married"> <TargetValueCounts> <TargetValueCount value="0" count="769"/>
        <TargetValueCount value="1" count="588"/> </TargetValueCounts>
    </PairCounts> <PairCounts value="NeverM"> <TargetValueCounts>
        <TargetValueCount value="0" count="913"/> <TargetValueCount value="1" count="38"/>
    </TargetValueCounts> </PairCounts> <PairCounts value="Widowed">
        <TargetValueCounts> <TargetValueCount value="0" count="99"/>
    </TargetValueCounts> </PairCounts> <PairCounts value="Separ.">
        <TargetValueCounts> <TargetValueCount value="0" count="87"/>
    </TargetValueCounts> </PairCounts> <PairCounts value="OTHER_CATEGORY">
        <TargetValueCounts> <TargetValueCount value="0" count="32"/>
    </TargetValueCounts> </PairCounts> </BayesInput> <BayesInput
fieldName="NATIVE_COUNTRY"> <PairCounts value="OTHER_CATEGORY">
    <TargetValueCounts> <TargetValueCount value="0" count="110"/>
    </TargetValueCounts> </PairCounts> <PairCounts value="U.S."/>
    <TargetValueCounts> <TargetValueCount value="0" count="2049"/>
    <TargetValueCount value="1" count="613"/> </TargetValueCounts> </PairCounts>
    <PairCounts value="Mexico"> <TargetValueCounts> <TargetValueCount value="0" count="51"/>
        <TargetValueCounts> </PairCounts> </BayesInput> <BayesInput
fieldName="SEX"> <PairCounts value="Female"> <TargetValueCounts>
    <TargetValueCount value="0" count="898"/> <TargetValueCount value="1" count="85"/>
    </TargetValueCounts> </PairCounts> <PairCounts value="Male">
        <TargetValueCounts> <TargetValueCount value="0" count="1374"/>
    <TargetValueCount value="1" count="583"/> </TargetValueCounts> </PairCounts>
</BayesInput> <BayesInput fieldName="WORKCLASS"> <PairCounts value="Loc-gov">
    <TargetValueCounts> <TargetValueCount value="0" count="162"/>
    <TargetValueCount value="1" count="56"/> </TargetValueCounts> </PairCounts>
    <PairCounts value="Self-EI"> <TargetValueCounts> <TargetValueCount value="0" count="66"/>
        <TargetValueCount value="1" count="64"/> </TargetValueCounts>
    </PairCounts> <PairCounts value="Sta-gov"> <TargetValueCounts>
        <TargetValueCount value="0" count="82"/> </TargetValueCounts> </PairCounts>
    <PairCounts value="OTHER_CATEGORY"> <TargetValueCounts>
        <TargetValueCount value="0" count="57"/> <TargetValueCount value="1" count="32"/>
    </TargetValueCounts>

```

```

</TargetValueCounts>           </PairCounts>           <PairCounts value="Private">
<TargetValueCounts>           <TargetValueCount value="0" count="1571"/>
<TargetValueCount value="1" count="423"/>   </TargetValueCounts>   </PairCounts>
<PairCounts value="SelfENI">    <TargetValueCounts>   <TargetValueCount value="0"
count="182"/>      <TargetValueCount value="1" count="54"/>   </TargetValueCounts>
</PairCounts> </BayesInput> <BayesInput fieldName="AGE">   <PairCounts value="46.2-
60.8">      <TargetValueCounts>   <TargetValueCount value="0" count="393"/>
<TargetValueCount value="1" count="207"/>   </TargetValueCounts>   </PairCounts>
<PairCounts value="17-31.6">    <TargetValueCounts>   <TargetValueCount value="0"
count="932"/>      <TargetValueCount value="1" count="62"/>   </TargetValueCounts>
</PairCounts>      <PairCounts value="31.6-46.2">   <TargetValueCounts>
<TargetValueCount value="0" count="771"/>   <TargetValueCount value="1"
count="348"/>      </TargetValueCounts>   <PairCounts value="60.8-75.4">
<TargetValueCounts>           <TargetValueCount value="0" count="164"/>
<TargetValueCount value="1" count="49"/>   </TargetValueCounts>   </PairCounts>
</BayesInput> <BayesInput fieldName="CAPITAL_GAIN">   <PairCounts value="0-
19999.8">      <TargetValueCounts>   <TargetValueCount value="0" count="2272"/>
<TargetValueCount value="1" count="649"/>   </TargetValueCounts>   </PairCounts>
</BayesInput> <BayesInput fieldName="EDUCATION">   <PairCounts value="&lt;
Bach.">      <TargetValueCounts>   <TargetValueCount value="0" count="561"/>
<TargetValueCount value="1" count="120"/>   </TargetValueCounts>   </PairCounts>
<PairCounts value="Assoc-V">    <TargetValueCounts>   <TargetValueCount value="0"
count="98"/>      <TargetValueCount value="1" count="34"/>   </TargetValueCounts>
</PairCounts>      <PairCounts value="OTHER_CATEGORY">   <TargetValueCounts>
<TargetValueCount value="0" count="413"/>   <TargetValueCount value="1"
count="110"/>      </TargetValueCounts>   <PairCounts value="HS-grad">
<TargetValueCounts>           <TargetValueCount value="0" count="859"/>
<TargetValueCount value="1" count="138"/>   </TargetValueCounts>   </PairCounts>
<PairCounts value="Bach.">    <TargetValueCounts>   <TargetValueCount value="0"
count="264"/>      <TargetValueCount value="1" count="191"/>   </TargetValueCounts>
</PairCounts>      <PairCounts value="Masters">   <TargetValueCounts>
<TargetValueCount value="0" count="77"/>   <TargetValueCount value="1" count="75"/>
</TargetValueCounts>           </PairCounts>   </BayesInput>   <BayesInput
fieldName="EDUCATION_NUM">   <PairCounts value="4-7">   <TargetValueCounts>
<TargetValueCount value="0" count="160"/>   </TargetValueCounts>   </PairCounts>
<PairCounts value="7-10">    <TargetValueCounts>   <TargetValueCount value="0"
count="980"/>      <TargetValueCount value="1" count="153"/>   </TargetValueCounts>
</PairCounts>      <PairCounts value="10-13">   <TargetValueCounts>
<TargetValueCount value="0" count="724"/>   <TargetValueCount value="1"
count="177"/>      </TargetValueCounts>   <PairCounts value="13-16">
<TargetValueCounts>           <TargetValueCount value="0" count="370"/>
<TargetValueCount value="1" count="323"/>   </TargetValueCounts>   </PairCounts>
<PairCounts value="1-4">    <TargetValueCounts>   <TargetValueCount value="0"
count="38"/>      </TargetValueCounts>   <PairCounts value="Sales">   <TargetValueCounts>
<TargetValueCount value="0" count="249"/>   <TargetValueCount value="1"
count="88"/>      </TargetValueCounts>   <PairCounts value="Cleric.">
<TargetValueCounts>           <TargetValueCount value="0" count="296"/>
<TargetValueCount value="1" count="38"/>   </TargetValueCounts>   </PairCounts>
<PairCounts value="Prof.">   <TargetValueCounts>   <TargetValueCount value="0"
count="241"/>      <TargetValueCount value="1" count="153"/>   </TargetValueCounts>
</PairCounts>      <PairCounts value="Crafts">   <TargetValueCounts>
<TargetValueCount value="0" count="271"/>   <TargetValueCount value="1"
count="83"/>      </TargetValueCounts>   <PairCounts value="Exec.">

```

```

<PairCounts value="OTHER_CATEGORY"> <TargetValueCounts>
<TargetValueCount value="0" count="883"/> <TargetValueCount value="1"
count="121"/> </TargetValueCounts> </PairCounts> </BayesInput> <BayesInput
fieldName="RACE"> <PairCounts value="Black"> <TargetValueCounts>
<TargetValueCount value="0" count="294"/> </TargetValueCounts> </PairCounts>
<PairCounts value="Asian"> <TargetValueCounts> <TargetValueCount value="0"
count="63"/> </TargetValueCounts> </PairCounts> <PairCounts value="White">
<TargetValueCounts> <TargetValueCount value="0" count="1862"/>
<TargetValueCount value="1" count="615"/> </TargetValueCounts> </PairCounts>
</BayesInput> <BayesInput fieldName="RELATIONSHIP"> <PairCounts
value="Husband"> <TargetValueCounts> <TargetValueCount value="0"
count="662"/> <TargetValueCount value="1" count="534"/> </TargetValueCounts>
</PairCounts> <PairCounts value="NotinFa"> <TargetValueCounts>
<TargetValueCount value="0" count="681"/> <TargetValueCount value="1"
count="68"/> </TargetValueCounts> </PairCounts> <PairCounts value="Unmarr."
> <TargetValueCounts> <TargetValueCount value="0" count="319"/>
</TargetValueCounts> </PairCounts> <PairCounts value="OTHER_CATEGORY">
<TargetValueCounts> <TargetValueCount value="0" count="88"/>
</TargetValueCounts> </PairCounts> <PairCounts value="O-child">
<TargetValueCounts> <TargetValueCount value="0" count="435"/>
</TargetValueCounts> </PairCounts> <PairCounts value="Wife">
<TargetValueCounts> <TargetValueCount value="0" count="87"/>
<TargetValueCount value="1" count="52"/> </TargetValueCounts> </PairCounts>
</BayesInput> <BayesInput fieldName="WEIGHT"> <PairCounts value="447242.6-
663979.4"> <TargetValueCounts> <TargetValueCount value="0" count="50"/>
</TargetValueCounts> </PairCounts> <PairCounts value="13769-230505.8">
<TargetValueCounts> <TargetValueCount value="0" count="1638"/>
<TargetValueCount value="1" count="498"/> </TargetValueCounts> </PairCounts>
<PairCounts value="230505.8-447242.6"> <TargetValueCounts> <TargetValueCount
value="0" count="578"/> <TargetValueCount value="1" count="154"/>
</TargetValueCounts> </PairCounts> </BayesInput></BayesInputs><BayesOutput
fieldName="CLASS"> <TargetValueCounts> <TargetValueCount value="1"
count="668"/> <TargetValueCount value="0" count="2272"/>
</TargetValueCounts></BayesOutput></NaiveBayesModel>
</PMML>

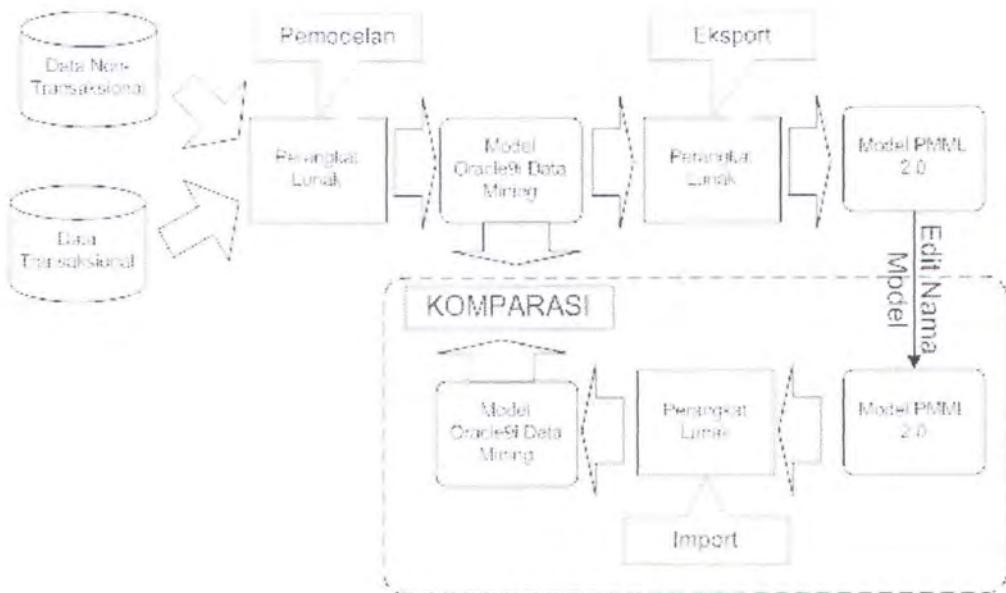
```

Gambar 5.09 Ouput Eksport ke PMML dari Naive Bayes

5.3 Proses Import

Uji coba ini dilakukan dengan melakukan perubahan pada berkas output yang sudah dalam bentuk PMML. Berkas PMML yang diimport adalah berkas hasil proses pemodelan dan eksport yang sudah dilakukan sebelumnya.

Apabila digambarkan skenario uji coba dari proses import model tampak sebagai berikut :



Gambar 5.10 Skenario Uji Coba Import

Perubahan pada berkas model PMML dilakukan terhadap nama model sedangkan properti yang lain dibiarkan seperti hasil eksport.

5.3.1 Kaidah Asosiasi

Perubahan itu dilakukan terhadap nilai pada tag *modelName*. Tag tampak seperti berikut :

```
<AssociationModel
  modelName="UJI_TRANSAKSIONAL"
  functionName="associationRules"
  maxNumberOfItemsPerTA="5"
  minimumSupport="0.5"
  numberOfWorksets="19" numberofRules="15">
  ...
</AssociationModel>
```

Gambar 5.11 Tag PMML sebelum Perubahan

Perubahan tampak seperti berikut :

```
<AssociationModel
  modelName="IMPORT_UJI_TRANSAKSIONAL"
  functionName="associationRules"
  maxNumberOfItemsPerTA="5"
  minimumSupport="0.5"
  numberOfWorksets="19" numberofRules="15">
  ...
</AssociationModel>
```

Gambar 5.12 Tag PMML setelah Perubahan



Perubahan hanya dilakukan terhadap nama model karena item-item yang lain pada berkas PMML merupakan setting dari model tersebut. Perubahan pada suatu setting berpengaruh pada setting yang lain.

Setelah dilakukan proses import maka didapatkan hasil seperti berikut :

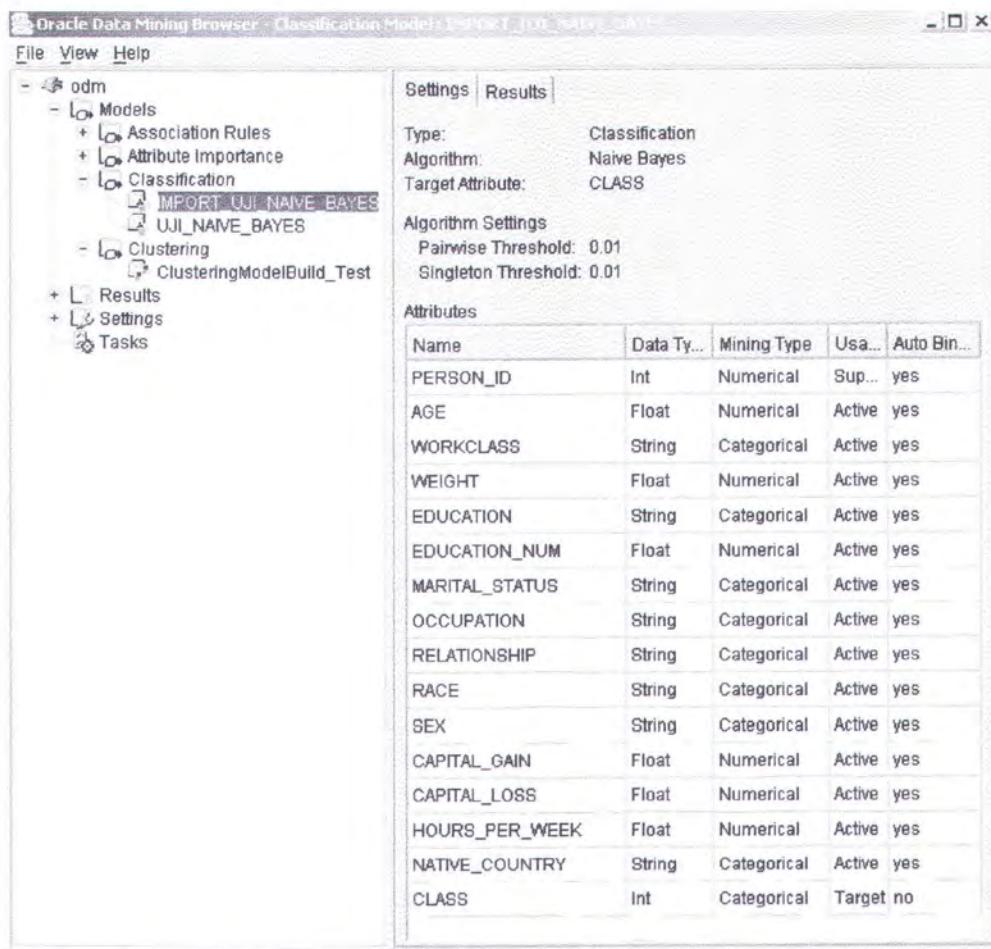
Id	If (Condition)	Then (Association)	Confidence		Support
			Get Rules		
15	C=1 and T=1 ... A=1		1.0	0.5	
14	A=1 and T=1 a... C=1		1.0	0.5	
13	A=1 and C=1 ... W=1		1.0	0.5	
7	T=1 and W=1 A=1		1.0	0.5	
6	A=1 and D=1 C=1		1.0	0.5	
9	T=1 and T=1 C=1		1.0	0.5	
12	W=1 and W=1 C=1		1.0	0.5	
8	A=1 and T=1 W=1		1.0	0.5	
10	A=1 and W=1 C=1		1.0	0.6666667	
11	A=1 and C=1 W=1		1.0	0.6666667	
2	A=1 C=1		1.0	0.6666667	

Gambar 5.13 Output Proses Import Model PMML

Dari proses import tampak bahwa proses import menghasilkan model yang sama dengan model yang dieksport.

5.3.2 Naive Bayes

Proses import ini dilakukan terhadap berkas PMML yang sudah disebutkan pada Gambar Output Eksport ke PMML dari Tabel CENSUS_2D_BUILD_UNBINNED. Perubahan dilakukan terhadap nama model. Dengan mempergunakan alat bantu ODMBrowser maka didapatkan hasil seperti berikut :



Gambar 5.14 Output Proses Import dengan ODMBrowser

Tampak proses import menghasilkan model yang sama dengan hasil pembuatan model.

BAB VI

KESIMPULAN DAN SARAN

BAB VI

KESIMPULAN DAN SARAN

Pada Bab ini dijelaskan mengenai kesimpulan yang dapat dari serangkaian ujicoba dan evaluasi terhadap perangkat lunak yang telah dibangun. Juga diberikan saran bagi pengembangan lebih lanjut dari perangkat lunak ini.

6.1 Kesimpulan

Dari hasil uji coba perangkat lunak yang telah dilakukan , maka dapat diambil kesimpulan seperti berikut :

- a. Berdasarkan uji coba proses pemodelan Kaidah Asosiasi dari Oracle9i Data Mining menghasilkan nilai yang sesuai bila dibandingkan dengan nilai dari perhitungan manual. Hal ini berlaku untuk model yang dibentuk dari tabel transaksional maupun model dari non-transaksional. Namun, untuk data yang besar, proses mengalami kesalahan bila digunakan konfigurasi minimum *support* dan minimum *confidence* bernilai lebih dari 0.1.
- b. Berdasarkan uji coba eksport model Kaidah Asosiasi dihasilkan bentuk model PMML yang sama untuk data transaksional maupun data non-transaksional.
- c. Berdasarkan uji coba, jika suatu model dieksport kemudian hasil eksportnya diimport, maka hasil import menghasilkan model yang sama dengan model asal yang dieksport. Hal ini berlaku untuk model Kaidah Asosiasi maupun model Naive Bayes.

- d. Berdasarkan uji coba, pembuatan model Naive Bayes menghasilkan model yang sama untuk tabel transaksional maupun non-transaksional.

6.2 Saran

Perangkat lunak ini dapat dikembangkan terutama untuk pemodelan Naive Bayes, tidak hanya sebatas pembuatan model atau bisa disebut *modeling*, tapi bisa sampai ke pengujian, perhitungan, atau pengaplikasian.

DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [DMG-02] Data Mining Group. *PMML Version 2.0*, Data Mining Group, 2002
- [HOR-00] Horton,Ivan. *Beginning Java 2*, Cananda,Wrox Press, Birmingham,2000
- [HUN-98] Hunter,David. *Beginning XML*, Wrox Press Inc,1998
- [GRO-98] Grossman,Robert.*D-Miner and PMML*,Data Mining Group,1998
- [OR1-02] Oracle Corporation.*Oracle9i Data Mining Concepts, Release 9.2.0.2*, Oracle Corporation, Redwood City, 2002
- [OR2-02] Oracle Corporation.*Data Mining API Documentation*, Oracle Corporation, Redwood City, 2002
- [OR3-01] Oracle Corporation.*Oracle9i Data Mining, An Oracle White Paper December 2001*, Oracle Corporation, Redwood City, 2001
- [SOL-02] Solari,Richard.*Oracle Data Mining Software*, Oracle Corporation,2002
- [TYA-02] Tyaspamadya,Daning.*Perancangan dan Pembuatan Perangkat Lunak Data Mining untuk Pengalian Kaidah Asosiasi dengan Menggunakan Metode Hybrid*, Tugas Akhir, Jurusan Teknik Informatika,FTIF,ITS,2002

