13582/H/01

ITS

DATA MINING PADA PENCARIAN POLA-POLA PERIODIK UNTUK DATA BERATRIBUT WAKTU

TUGAS AKHIR

RSIf
004
Kri
d-1
1999

Oleh :

ANTONIUS BOEDI KRISTRIYANTO

NRP. 2694.100.002

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

1999

Ky. 30.000.
PERPUSTAKAAN
ITS
TGL. TERIMA 04/01/2001
H
NO. INDUK 21 2717

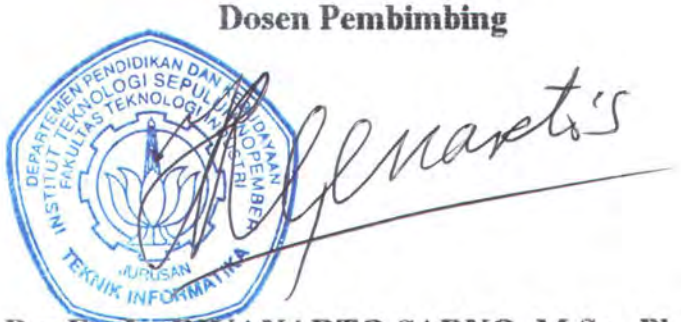
DATA MINING PADA PENCARIAN POLA-POLA PERIODIK UNTUK DATA BERATRIBUT WAKTU

TUGAS AKHIR

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

Mengetahui dan Menyetujui

Dosen Pembimbing



Drs.Ec. Ir. RIYANARTO SARNO, M.Sc., Ph.D.
NIP. 131 570 363

SURABAYA
Agustus, 1999

"Serahkanlah hidupmu kepada TUHAN dan
percayalah kepada-Nya, dan IA akan bertindak"
(Mazmur, 37:5)

Kata Pengantar

Puji syukur kepada Tuhan Yang Maha Pengasih karena hanya dengan rahmat dan perkenan-Nya penulis dapat menyelesaikan penulisan tugas akhir dengan judul

Data Mining pada Pencarian Pola-Pola Periodik
untuk Data Beratribut Waktu

dengan baik.

Tugas akhir ini merupakan salah satu wujud dari beberapa bidang yang penulis minati selama mengikuti perkuliahan di Jurusan Teknik Informatika ITS Surabaya, yaitu tentang masalah-masalah *Artificial Intelligence* dan *Knowledge Discovery in Database*. Masalah-masalah ini merupakan topik yang menarik dan sedang menjadi bahan penelitian dari berbagai peneliti di berbagai institusi akademis.

Harapan penulis, semoga tugas akhir ini menjadi sesuatu yang berguna dalam pengembangan-pengembangan selanjutnya dari topik yang telah diambil oleh penulis.

Pada kesempatan ini pula, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Kedua orang-tua ku yang tak henti-hentinya memberikan dorongan dan doa kepada penulis selama pengerjaan tugas akhir ini, juga kedua kakak-ku yang telah memberi motivasi, thanks berat
2. Bapak Drs.Ec.Ir. Riyanarto Sarno, M.Sc, Phd, selaku dosen pembimbing yang telah banyak memberikan masukan dan saran dalam pembuatan tugas ahir ini
3. Bapak Ir. Khakim Gozali, selaku dosen wali penulis selama penulis menjalani perkuliahan
4. Bapak Rully Soelaiman, S.Kom, atas masukannya pada seminar Tugas Akhir
5. Lusie, trims atas segala diskusi-diskusi kita dalam topik *Data Mining* dan menjadi moderator pada seminar Tugas Akhir

6. Monica dan Tinuk , trims atas bantuan salah satu sub prosedur dari program penulis
7. Mas Indra dari Abakus, trims atas tutorial Power Designer-nya
8. Kepada rekan-rekan cewek C0A terutama Myrna Ermawati, trims banyak, juga Rita, Intan, Lusie, Yetty, Maya, Dina, Bilqis serta Tita buat pinjeman printer-nya, dll, Trims atas segala bantuan, baceman, dan kebersamaan yang menjadikan hari-hari selalu terasa indah.
9. Kepada rekan-rekan dan sobat-sobat terbaik penulis di C0A yaitu: Firdaus, Waskitho, Anjeb, Deni, Firman, Yanuar Kusnadi, Yanuar Bagus, Tom, Heri, Pewee, Itonk, Arthur, Mudi, Roni, Jimmy, Putut, Agus SA, Agus Ulum, Agus Bagiarta, Fajar, Gusmang, Eddy, Sutama, Dewa, Arya, Herlambang, Tony, Hadi, Eko, Dudy, Salman, Yoyok, Anda, Paryono, Yudi, Iduy dan yang belum tersebut, trims banyak atas segala kebersamaan kita selama perkuliahan khusunya ketika quiz dan ujian
10. Kepada sobat terbaik sejak SMA, Wima Sangga, trims atas segala diskusi-diskusi kita
11. Kepada para karyawan di lingkungan Jurusan Teknik Informatika, terima kasih banyak atas segala bantuan selama penulis menjalani perkuliahan

Surabaya, 18 Agustus 1999

Penulis

DAFTAR ISI

Daftar Isi

Kata Pengantar ..... **i**

Daftar Isi ..... **ii**

Daftar Gambar ..... **iv**

Daftar Tabel ..... **vi**

Abstraksi ..... **vii**

BAB I PENDAHULUAN **1**

1.1 Latar Belakang 1

1.2 Permasalahan 3

1.3 Tujuan Penulisan 4

1.4 Batasan Permasalahan 5

1.5 Metodologi 5

1.6 Sistematika Penulisan 6

BAB II DATA MINING DAN DATA WAREHOUSING **7**

2.1 Data Warehousing 7

2.1.1 Data Multi-Dimensi (Multi-dimensional Data) 10

2.1.2 OnLine Analytical Processing 11

2.2 Data Mining 16

BAB III POLA-POLA PERIODIK **18**

3.1 Data Beratribut Waktu 18

3.2 Pola-pola Periodik 20

BAB IV METODA DAN SISTEM YANG DIGUNAKAN **26**

4.1 Partial Fixed Value-Based Search 28

4.2 Partial Fixed Trend-Based Search 35

4.3 Arbitrary Search 37

4.4 Analisa Cube 40

BAB V PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK 42

5.1 Perancangan Perangkat Lunak 43

5.1.1 Permodelan Data Konseptual 44

5.1.2 Diagram Alir Data (Data Flow Diagram) 48

5.2 Pembuatan Perangkat Lunak 55

5.2.1 Modul Masukan 55

5.2.2 Modul Proses 56

5.2.3 Modul Keluaran 57

BAB VI IMPLEMANTASI DAN EVALUASI PERANGKAT LUNAK 58

6.1 Pengujian Perangkat Lunak 58

BAB VII KESIMPULAN 72

DAFTAR GAMBAR

Daftar Gambar

Gambar 2.1Konsep Data Warehouse 8

Gambar 2.2 Contoh Sebuah Data Cube 12

Gambar 2.3 Contoh sebuah konsep hirarki pada dimensi waktu 14

Gambar 2.4 Algoritma K-Means Clustering 15

Gambar 2.5 Konsep Data Mining 17

Gambar 2.6 Proses Data Mining 17

Gambar 4.1 Contoh Working Cube 3-bulanan 29

Gambar 4.2 Algoritma untuk Fixed Value-Based 30

Gambar 4.3 Algoritma untuk Prosedur Find One Cycle Pattern 30

Gambar 4.4 Algoritma untuk Prosedur FormCandidatePattern 31

Gambar 4.5 Akgoritma untuk Prosedur CheckPatternExistence 31

Gambar 4.6 Slices untuk Working Cube pada Contoh 1 33

Gambar 4.7 Working Cube untuk Arbitrary Search 38

Gambar 4.8 Algoritma untuk Arbitrary Search 39

Gambar 4.9 Permodelan Konseptual Data untuk Sistem Retail Sederhana 46

Gambar 5.1 DFD Level 0 dari Sistem 48

Gambar 5.2 Modul Proses 57

Gambar 6.1 Data Cube hasil dari Pembentukan beberapa Database 59

Gambar 6.2 Time Series untuk Penjualan Women Accesories 60

Gambar 6.3 Hasil pada Fixed Value Based threshold 100 % 61

Gambar 6.4 Fixed Value Based dengan Threhold 90% 63

Gambar 6.5 Fungsi Autokorelasi Dari Data Asli 64

Gambar 6.6 Time Series setelah dilakukan proses Stasionerisasi 65

Gambar 6.7 Hasil Fixed Value-based pada Data Terstasionerisasi 66

Gambar 6.8 Fixed Value Based pada Contoh 2 dengan Jumlah Kategori = 5 68

Gambar 6.9 Fixed Value Based pada Contoh 2 dengan jumlah kategori 6 69

Gambar 6.10 Arbitrary Search pada Contoh 2 70

DAFTAR TABEL

Daftar Tabel

Table 2.1 Contoh Potongan Sebuah Tabel 10

Table 2.2 Contoh Penyajian Data Multi-Dimensi 11

Table 2.3 Contoh konsep kategori data 14

Table 4.1 Data Untuk Contoh 1 32

Table 4.2 Konsep Kategori untuk Contoh 1 32

Table 4.3 Daftar Trend Beserta Keterangannya 37

Table 5.1 Flow Mining Request 49

Table 5.2 Data Flow Data Item List 49

Table 5.3 Flow Periodic Patterns Informations 49

Table 5.4 Data Flow Data Item List 49

Table 5.6 Data StoreView 51

Table 5.7 Data Store Data Item List 51

Table 6.1 Konsep Kategori untuk Penjualan Women Accesories 62

Table 6.2 Konsep Kategori pada contoh 2 dengan jumlah Kategori = 5 67

Table 6.3 Konsep Kategori pada Contoh dengan Jumlah kategori - 6 68

ABSTRAKSI

Abstraksi

Dalam banyak aplikasi seperti perbankan, asuransi, retail, medis, dan juga masih banyak lagi, tiap hari terjadi transaksi dalam sistem mereka dan semuanya tercatat dalam sebuah sistem basis data. Data-data ini sangat berharga karena bila dilakukan analisa lebih lanjut, dapat menjadi tuntunan bagi perusahaan-perusahaan di atas untuk mengarahkan kebijakan dari perusahaan untuk masa yang akan datang. Pencarian informasi pada data-data seperti ini (*knowledge discovery of database*) lebih sering disebut sebagai *data mining*, yaitu seperangkat peralatan untuk menggali informasi berharga dari data yang mana sebelumnya belum diketahui untuk dijadikan sumber informasi bagi para pengambil keputusan untuk menentukan keputusan bisnisnya.

Dalam tugas akhir ini, akan dibahas salah satu topik dalam masalah *data mining*, yaitu pencarian pola-pola keperiodikan pada yang beratribut waktu. Pola-pola keperiodikan penting untuk diketahui karena dengan kita mengetahui sifat-sifat keperiodikan dari data, maka kita dapat memberikan perlakuan yang tepat bagi aplikasi yang menggunakan data tersebut termasuk juga sumber-daya yang terlibat di dalamnya Dengan metoda ini, pencarian informasi keperiodikan akan menjadi efektif dan efisien karena *user* diberi kesempatan untuk menentukan sendiri perioda-perioda keperiodikan yang diinginkan

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada banyak aplikasi saat ini, pengumpulan data dapat dilakukan dengan begitu mudah dan begitu cepatnya karena semakin tingginya teknologi. Data-data ini disimpan sebagai data perusahaan dan pada saat ini jumlahnya sudah sangat banyak, seperti yang dapat kita temui pada perusahaan-perusahaan retail, perbankan, asuransi, dan sebagainya.

Data-data ini sekarang disimpan dalam suatu mekanisme yang disebut *data warehouse*, yaitu suatu tempat penyimpanan data yang mana data-datanya dikumpulkan dari berbagai macam sumber yang terpisah dan mempunyai format yang berbeda-beda. Dengan *data warehouse* ini, diharapkan analisa terhadap data yang tersimpan didalamnya akan dapat dilakukan dengan mudah karena semua data telah dikumpulkan menjadi satu.

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

Salah satu yang dapat dilakukan dan juga menjadi topik dalam penyusunan tugas akhir ini adalah usaha untuk melakukan pencarian pola-pola periodik dari data-data yang telah terkumpul ini. Pencarian pola-pola periodik ini adalah usaha untuk mengetahui apakah suatu produk (misalnya) mempunyai suatu siklus (*cycle*) dalam penjualan tahunannya. *Cycle* di sini berarti terjadi perulangan pada suatu nilai yang relatif konstan pada tiap-tiap titik waktu dalam jangka waktu yang tetap.

Dengan diketahuinya pola-pola keperiodikan dari data, maka dapat didapatkan manfaat yaitu:

- Dengan diketahuinya pola-pola periodik dari suatu data, maka dapat dilakukan suatu alokasi dari suatu *resources* (sumber daya) baik manusia ataupun barang pada waktu dan jumlah yang tepat.
- Menjadi salah satu sumber pendamping bagi perangkat lainnya, misalnya menjadi referensi bagi metoda peramalan (*forecasting*) yang memang telah umum dilakukan

Alokasi sumber daya ini akan bermacam-macam tergantung dari aplikasi yang digunakan. Misalnya pada suatu perusahaan retail, bila diketahui bahwa suatu produk X mempunyai pola periodik dalam penjualan tahunannya, maka dapat diterapkan suatu kebijakan *stocking* yang tepat dalam hal jumlah dan waktu dari produk X tersebut, atau juga ditempatkan lebih banyak orang dengan kemampuan yang disesuaikan pada waktu-waktu dimana sebuh produk mengalami kenaikan atau membutuhkan perhatian khusus.

1.2 Permasalahan

Berikut adalah permasalahan yang timbul dari latar belakang yang telah diuraikan di atas sehingga timbul suatu usaha untuk mengetahui suatu pola-pola keperiodikan

1. Terdapat suatu kebutuhan untuk mengetahui perilaku suatu produk (bila berupa barang) atau perilaku umum *customer* (bila berupa usaha jasa) dalam lingkungan usaha tertentu. Perilaku ini bisa meliputi pada waktu kapan saja penjualan (misalnya) produk X mengalami nilai yang relatif sama, berapa nilai pada titik-titik dimana penjualan produk X ini mengalami nilai yang relatif sama, dan lain-lain. Perilaku ini berguna bagi para pengambil keputusan untuk menetapkan kebijakan perusahaan selanjutnya.
2. Beberapa metoda pencarian pola periodik telah dilakukan, tetapi metoda-metoda ini kurang bisa diterapkan karena beberapa alasan. *Fast Fourier Transformation* (FFT), misalnya, dapat mencari keperiodikan dari suatu data tetapi ia menganggap data adalah suatu nilai-nilai yang tidak dapat dipisahkan (*inseparable flow of values*) sehingga keperiodikan yang dihasilkan adalah *full cycle periodicity* dimana tiap titik dalam periode mempunyai kontribusi dalam *cycle* tersebut, padahal keperiodikan pada kasus-kasus yang ada umumnya adalah keperiodikan pada segmen-segmen tertentu, seperti misalnya: nilai perdagangan volume saham perusahaan M naik hampir tiap hari Rabu tetapi untuk hari-hari yang lain tidak dapat diperkirakan.
3. Beberapa perusahaan saat ini telah mulai mengumpulkan data-data histori mereka ke dalam sebuah *Data Warehouse*. *Data warehouse* adalah suatu

mekanisme penyimpanan data yang subyektif, terintegrasi, berbasis waktu, dan bersifat *read-only*. Pada *data warehouse* ini, data yang disimpan umumnya multi-dimensi (untuk lebih jelasnya dapat dilihat pada bab II) karena dikumpulkan dari berbagai sumber yang terpisah dan multi-format. Untuk itu dibutuhkan suatu mekanisme untuk menyajikan data-data yang terdapat dalam *Data warehouse* ini untuk dilakukan pencarian pola-pola keperiodikannya.

1.3 Tujuan Penulisan

Adapun tujuan dari penulisan tugas akhir ini adalah sebagai berikut :

- Mensimulasikan kerja sebuah *data warehouse* dimana semua data yang dipakai dalam semua metoda (*data mining*) yang disajikan dalam penyusunan tugas akhir ini akan diambilkan dalam *data warehouse* itu
- Mengimplementasikan metoda-metoda yang dapat mencari pola keperiodikan parsial dari data, parsial di sini berarti keperiodikan yang dihasilkan adalah berkaitan dengan segmen-segmen waktu yang tertentu (tiga-bulanan, empat bulanan, dan sebagainya.).
- Menyajikan pola-pola periodik yang (bila) ditemukan ke dalam suatu format bahasa yang mudah untuk dibaca
- Memanfaatkan cara dan teknik yang tepat untuk mencari pola-pola periodik dari suatu data mengingat jumlah data yang dijadikan sebagai sumber data bisa berjumlah besar.

1.4 Batasan Permasalahan

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

1. Data yang dijadikan sebagai input adalah data-data dari basis data relasional
2. Data yang akan digunakan dalam analisa keperiodikan akan dikumpulkan secara bulanan
3. Interpretasi yang dihasilkan dari pola-pola periodik yang telah didapatkan adalah bersifat subyektif, dalam arti diasumsikan tidak terdapat perubahan yang signifikan dari masalah yang sedang dianalisa. Seperti misalnya dalam menganalisa suatu produk dalam sistem retail, maka diasumsikan bahwa pola konsumsi masyarakat terhadap produk tersebut relatif tidak mengalami perubahan yang mencolok yang diakibatkan karena krisis ekonomi, perubahan kebijakan pemerintahan, dan lain-lain misalnya

1.5 Metodologi

Berikut adalah metodologi yang dipakai pada metoda-metoda yang terdapat dalam penyusunan tugas akhir ini:

1. Pembuatan *Data Cube*
2. Pembuatan *time series* berdasarkan kriteria tertentu dari *Data Cube* yang telah dibuat
3. Pembentukan *Working Cube* dari *time series*
4. Pencarian pola-pola periodik dengan metoda-metoda sebagai berikut :

- *Partial Fixed Value-Based Search*
- *Partial Fixec Trend-Based Search*
- *Arbitrary Search*

5. Pembentukan informasi (interpretasi) dari pola-pola periodik yang telah dihasilkan

1.6 Sistematika Penulisan

Penyusunan buku tugas akhir ini dibagi-bagi menjadi beberapa bab sebagai berikut :

- Bab I, berisi tentang latar belakang, tujuan penulisan, serta permasalahan yang diangkat dalam penyusunan tugas akhir ini
- Bab II dan Bab III berisi teori-teori penunjang yang mendasari penyusunan tugas akhir ini
- Bab IV berisi metoda dan sistem yang digunakan untuk mengimplementasikan tugas akhir ini, meliputi algoritma serta penjelasan-penjelasan rinci
- Bab V berisi perancangan dan pembuatan perangkat lunak yang merupakan salah satu bagian dari penyusunan tugas akhir ini
- Bab VI berisi implementasi dan evaluasi perangkat lunak
- Bab VII berisi kesimpulan dan penutup

BAB II

DATA MINING DAN DATA WAREHOUSING

BAB II

DATA MINING DAN DATA WAREHOUSING

Sebelum memasuki pada pembahasan utama dari topik penyusunan tugas akhir ini, berikut dalam bab ini akan diuraikan hal-hal dasar mengenai konsep *data warehouse* yang menjadi mekanisme penyimpanan data yang akan digunakan dalam seluruh metoda yang digunakan dalam topik tugas akhir ini.

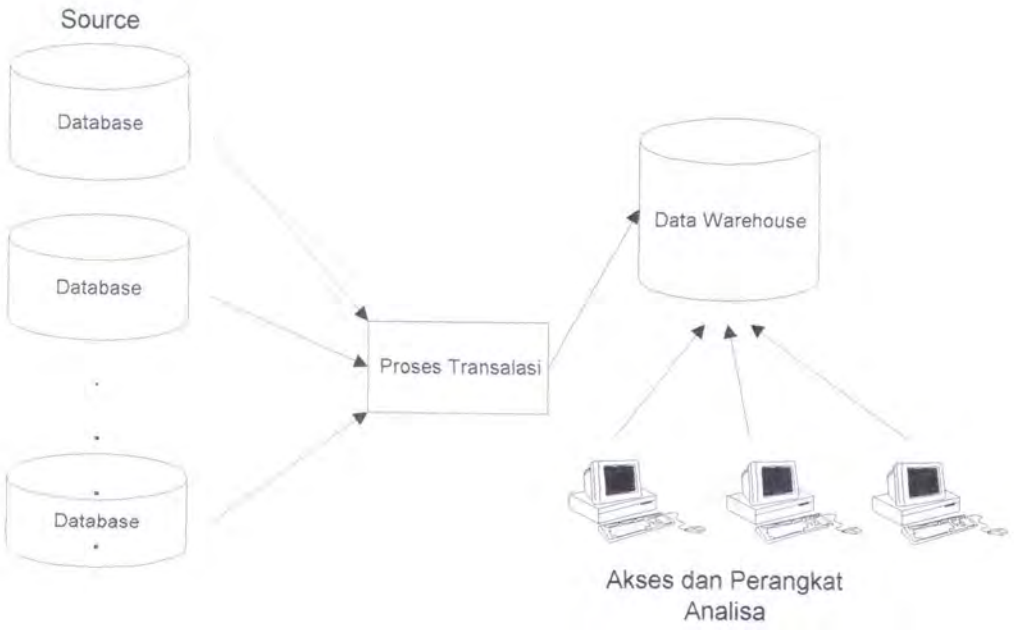
2.1 Data Warehousing

Data Warehouse[8] adalah sebuah proses pengintegrasian dari data-data perusahaan menjadi sebuah tempat penyimpanan tunggal yang mana pengguna (*end user*) dapat dengan mudah melakukan *queries*, membuat laporan, dan melakukan analisa. Ia juga adalah sebuah lingkungan pendukung keputusan yang meyimpan data dari berbagai sumber, mengorganisasikannya dan mengantarkannya kepada para pengambil keputusan. Pendek kata, *Data warehouse* adalah teknologi manajemen dan analisa data.

Ada beberapa hal yang membedakan antara sebuah *Data warehouse* dengan basis data biasa, yaitu:

- *Data warehouse* menyimpan data dengan operasi yang hanya dapat dibaca (*read only data*)

- *Data warehouse* adalah kumpulan dari berbagai sumber berbeda yang terpisah dan menggunakannya sebagai tempat dimana keterpisahan ini dapat disatukan
- Data disimpan dalam *Data warehouse* dengan maksud bahwa kita akan menjalankan berbagai aplikasi yang berbeda dengan menggunakan data yang tersimpan dalam *data warehouse* ini.



Gambar 2-1 Konsep Data Warehouse

Gambar di atas menggambarkan proses umum yang terdapat dalam sebuah *data warehouse*, yaitu *data warehouse* dibentuk dari berbagai macam *database* dimana dalam *database* satu dengan yang lain bisa jadi mempunyai format yang berbeda, misalkan saja format Microsoft Access, Dbase, Oracle, Microsoft SQL Server, dan lain-lain. Setelah diintegrasikan, barulah aplikasi-aplikasi dapat

menggunakannya untuk kepentingan-kepentingan analisa, laporan (*reporti*), dan lain-lain.

Berikut ini juga beberapa definisi dari sebuah *Data warehouse*

- Berorientasi Subyek (*Subject Oriented*), artinya bahwa *data warehouse* memfokuskan pada entitas-entitas dari data seperti pegawai, produk, dan sebagainya.
- Terintegrasi, artinya bahwa data disimpan dalam format yang konsisten atau tetap
- Beratribut Waktu (*Time Variant*), artinya bahwa data diasosiasikan dengan titik-titik waktu
- Tak Bisa Dihapus, artinya bahwa sekali data dimasukkan ke dalam *data warehouse*, maka data tak dapat diubah

Jadi, mengapa kita menggunakan *data warehouse* ? Jawaban utama dari pertanyaan ini adalah bahwa kita ingin melakukan analisa terhadap data dengan mudah dan tidak mengganggu proses operasional dari data itu sendiri, karena *data warehouse* merupakan tempat penyimpanan terpisah dari tempat operasional utama. Dengan mudah artinya bahwa kita tidak perlu direpotkan lagi dengan berbagai macam format dari data yang mungkin terjadi mengingat data yang terdapat dalam *data warehouse* dikumpulkan dari berbagai macam sumber yang terpisah dan berbeda sehingga kita bisa melakukan analisa dengan mudah dan cepat.

2.1.1 Data Multi-Dimensi (Multi-dimensional Data)

Seperti telah dijelaskan di atas bahwa *data warehouse* berisi data dari berbagai sumber yang terpisah, maka dalam *data warehouse* ini terdapat suatu data multi-dimensi. Artinya bahwa kita dapat melihat data-data itu sesuai dengan sumber dan kategorinya. Misalnya terdapat suatu potongan *record* dari tabel sebagai berikut :

Table 2-1 Contoh Potongan Sebuah Tabel

Kategori Produk	Nama Produk	Jumlah Penjualan
Pembersih Ruangan	Cairan X 120 ml	100
Pembersih Ruangan	Sapu Y ukuran medium	120
Minuman Ringan	Coca-cola	100
Minuman Ringan	Sprite	50

Maka dari potongan *record* diatas, dapat dilihat bahwa terdapat tiga dimensi dari tabel yang ada, yaitu dimensi kategori_produk, dimensi nama_produk, dan dimensi waktu yang diasosiasikan dengan jumlah penjualan (misalnya dalam satuan bulanan).

Bila *database* di atas tersebar dalam beberapa lokasi, bertambah lagi dimensinya yaitu dimensi lokasi. Dalam *data warehouse* , data-data multi dimensi ini dapat disajikan dengan format yang lebih "enak" untuk dilihat sehingga dapat digunakan untuk menganalisa data dengan cepat.

Berikut adalah salah satu contoh penyajian data multi dimensi dalam suatu mekanisme tabulasi dimana baris dan kolom-nya adalah merupakan dimensi-dimensi dari data. Dari penyajian ini, diharapkan dapat memperjelas arti dari data multi-dimensi tersebut.

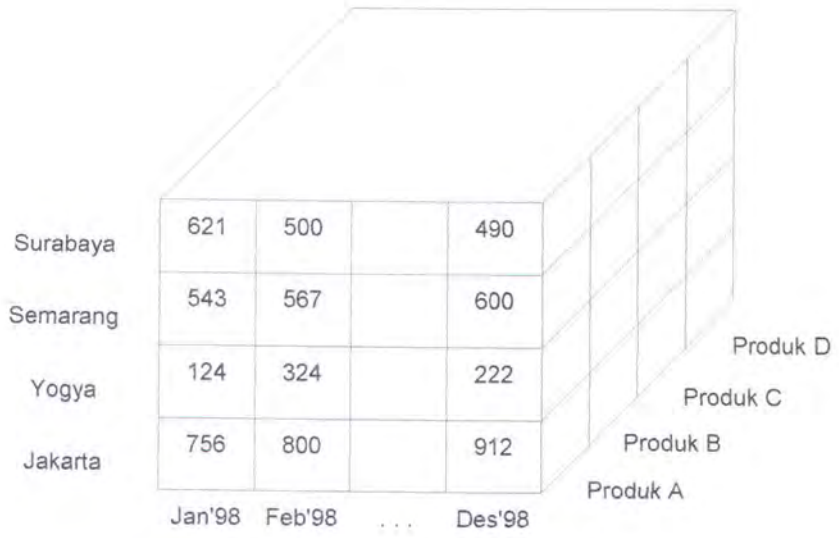
Table 2-2 Contoh Penyajian Data Multi-Dimensi

Lokasi	Kategori Pembersih Ruangan		Kategori Minuman Ringan	
	Cairan X	Sapu	Coca cola	Sprite
Surabaya	120	100	245	300
Semarang	76	86	200	300
Jakarta	245	200	120	250

2.1.2 OnLine Analytical Processing

Untuk menyajikan data-data multi-dimensi yang terdapat dalam *data warehouse* sehingga menjadi suatu informasi yang enak dan terintegrasi seperti di atas, maka dalam *data warehouse* terdapat suatu metoda yang disebut *On-Line Analytical Processing* (OLAP). Sesuai dengan namanya, OLAP ini berfungsi untuk menyajikan data multi-dimensi menjadi terintegrasi dengan baik sehingga memudahkan dalam analisa data. Data-data yang disajikan oleh OLAP ini biasanya adalah merupakan suatu fungsi agregasi seperti *summary, average*, dan lain-lain yang disajikan dalam sel-sel seperti pada tabel di atas.

Salah satu cara yang dipakai OLAP untuk menyajikan data dalam multi dimensi ini adalah dengan menggunakan *data cube*. *Data Cube* adalah metoda penyajian data dalam sebuah kubus dimana sisi-sisi dari kubus mewakili dimensi-dimensi dari data dan isi sel dari kubus adalah merupakan data itu sendiri. Jadi terdapat tiga dimensi yang dapat diamati tiap saat dari data bila disajikan dalam sebuah *data cube*.



Gambar 2-2 Contoh Sebuah Data Cube

Data Cube di atas adalah sebuah contoh penyajian data dari sebuah usaha retail, dimana akan dianalisa volume penjualan bulanan dari 4 buah produk (produk A, B, C, dan D) selama tahun 1998 dari 4 kota (Surabaya, Semarang, Yogya, dan Jakarta). Nilai-nilai tiap sel dari *cube* adalah merupakan hasil volume penjualan bulanan yang ada. Jadi setiap sumbu dari *cube* memegang sebuah dimensi dan sel-sel dari *cube* menyimpan hasil nilai dari *query* yang dilakukan. Pada contoh di atas, dimensi lokasi diletakkan pada sumbu Z, dimensi jenis produk diletakkan pada sumbu Y, dan dimensi waktu ada pada sumbu X.

Yang harus diperhatikan dalam penyusunan *data cube* adalah harus adanya atribut waktu dan data yang beratribut waktu. Pada contoh *data cube* di atas, atribut waktu ada pada sumbu X dan data beratribut waktu yang digunakan adalah "volume penjualan" yang diletakkan sebagai isi dari tiap-tiap sel dari *data cube*.

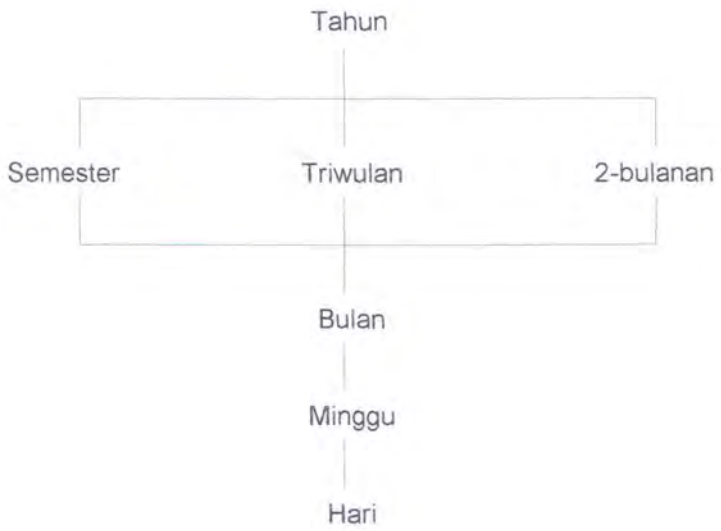
Data cube yang telah dibentuk seperti di atas sering juga disebut *reference cube* karena *data cube* yang telah dibentuk ini akan menjadi "referensi" untuk operasi-operasi lain yang diinginkan. Operasi-operasi ini antara lain:

- *Drill-down*, yaitu operasi menurunkan level agregasi dari data, misalnya *cube* telah terbentuk dalam data tahunan, maka *drill-down* dapat menjadikan *cube* dibentuk kembali dalam model data bulanan dengan jumlah data yang sama dengan periode tahun awal. Atau dapat juga terhadap dimensi lain yang mempunyai hirarki lebih kecil, seperti pada contoh multidimensi di atas, bila *cube* mula-mula dibentuk pada data dengan kategori minuman_ringan, maka *drill down* dapat dilakukan untuk mendapatkan data dengan nama_produk Coca-cola. Penurunan level ini tentu saja harus sesuai dengan konsep hirarki yang telah disepakati.
- *Roll-up*, kebalikan dari *drill-down* yaitu menaikkan level agragasi dari data
- *Slice*, adalah pemilihan data yang diinginkan, jadi misalnya kita telah memiliki *cube* seperti di atas, kita dapat melakukan *slice* untuk mendapatkan data untuk produk A saja.
- *Dice*, yaitu proyeksi dari *slice*, jadi setelah dilakukan *slicing*, bisa dilakukan *dice* untuk mendapatkan data produk di kota Semarang saja dari *slice* yang telah dibuat.

Konsep Kategori dan Konsep Hirarki

Pada beberapa bagian di atas, disinggung beberapa hal tentang konsep kategori dan konsep hirarki. Konsep hirarki dalam OLAP digunakan untuk

menentukan level agregasi bila dilakukan operasi-operasi seperti *drill-down* dan *roll-up*. Jadi bila suatu data dalam dimensi waktu akan dilakukan operasi *drill down* misalnya, maka ia harus menganut konsep hirarki waktu yang telah disepakati, misalnya sebagai berikut :



Gambar 2-3 Contoh sebuah konsep hirarki pada dimensi waktu

Jadi bila level agregasi data berada pada level tahunan, maka ia bisa di-*drill-down* ke level semesteran misalnya. Ini berlaku untuk semua dimensi data yang mempunyai hirarki.

Konsep kategori adalah suatu mekanisme pengelompokan data sehingga data bisa dipandang lebih abstrak. Misalnya saja kita mempunyai kategori data untuk suatu sistem penjualan sebagai berikut :

Table 2.3 Contoh konsep kategori data

Konsep Kategori	0 – 150	151-450	451-750	751-1000	1001- ~
Label	Sangat Jelek	Jelek	Stabil	Bagus	Bagus sekali

Dengan menggunakan konsep kategori, maka kita akan lebih mudah untuk mengetahui dan menyebutkan dimana data yang sedang kita amati berada.

Salah satu cara untuk mengelompokkan data dan yang juga dipakai dalam penulisan tugas akhir ini adalah dengan menggunakan algoritma *k-means clustering*. *Clustering* adalah proses untuk mengelompokkan data berdasarkan kemiripannya dan mengasosiasikannya ke dalam suatu *cluster*. Prinsip dari *clustering* adalah memaksimalkan perbedaan antar *cluster* dan meminimkan perbedaan antar data dalam satu *cluster*. Berikut adalah algoritmanya:

1. Pilih *n* titik awal dari data, baik secara random atau dipilih sendiri. Titik-titik awal ini adalah pusat (*centre*) awal dari *cluetsr* yang akan dibentuk
2. Pilih *n* titik awal dari data, baik secara random atau dipilih sendiri. Titik-titik awal ini adalah pusat (*centre*) awal dari *cluetsr* yang akan dibentuk.
3. Untuk tiap-tiap data, cari jarak antara data ini dengan titik pusat masing-masing dengan $cluster_i$. Isikan data ini pada $cluster_i$ dimana jarak data dengan titik pusat $cluster_i$ ini adalah minimum.
4. Cari nilai *mean* dari data-data yang ada pada tiap $cluster_i$. Nilai *mean* ini menjadi titik pusat baru untuk $cluster_i$
5. Ulangi langkah 2 dan 3 sampai nilai-nilai *mean* dari tiap $cluster_i$ konvergen.

Gambar 2-5 Algoritma K-Means Clustering

Beberapa keuntungan dengan dipakainya teknik *clustering* ini adalah sebagai berikut :

- Hasil *clustering* merupakan kumpulan alamiah data berdasarkan kemiripannya
- Jumlah *cluster* yang ingin dihasilkan dapat ditentukan sendiri

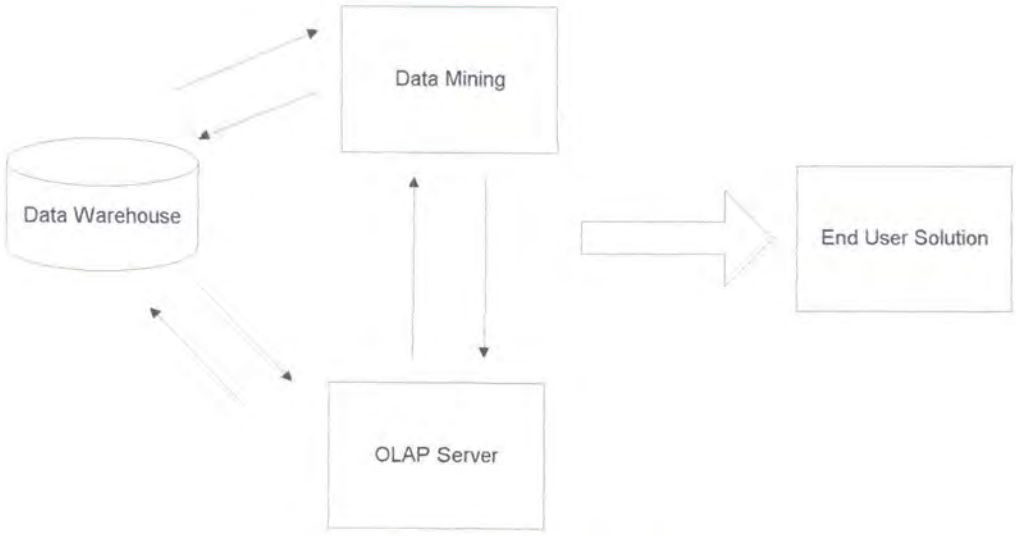
2.2 Data Mining

Data Mining adalah proses pengekstrasian informasi-informasi yang sebelumnya belum diketahui tetapi penting dari suatu sumber-sumber basis data dan menggunakannya untuk membuat sebuah keputusan bisnis yang penting. *Data Mining* adalah tentang sebuah penemuan hal-hal yang benar-benar nyata (*discovery of facts*) , sesuatu yang pada awalnya tampak tidak berhubungan, tetapi ketika dipandang dalam sudut yang lebih lebar dan digunakan dengan kepandaian manusia dan teknologi-teknologi yang mendukung, ia memegang arti-arti yang besar dan suatu informasi yang sangat berguna (*knowledge*).

Data Mining umumnya dilakukan pada suatu *data warehouse*, karena kemampuannya untuk menganalisa data dalam jumlah yang besar. Selain itu, tren pada saat ini adalah banyak perusahaan-perusahaan besar (*enterprise*) yang mulai menyimpan data-datanya dalam sebuah *warehouse* karena kemampuan *data warehouse* untuk mengintegrasikan berbagai macam basis-data dengan berbagai format dan struktur.

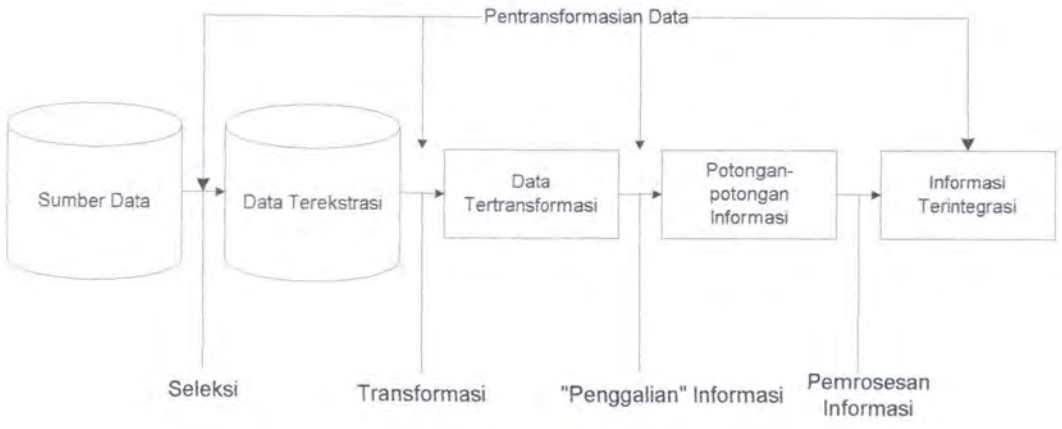
Karena *Data Mining* dilakukan pada sebuah *Data Warehouse*, maka metoda-metoda yang digunakan atau sedang dikembangkan diharapkan adalah metoda-metoda yang efisien dan efektif karena dengan jumlah data yang sangat besar, maka diharapkan *response time* dari metoda-metoda *Data Mining* adalah cepat.

Berikut adalah gambaran konsep dari *data mining*



Gambar 2.6 Konsep Data Mining

Konsep di atas menggambarkan bagaiamana *data mining* berada, jadi ia berada di atas sebuah *data warehouse* jadi dengan kata lain ia mengambil data dari *data warehouse* untuk mengintegrasikan sifat multi-dimensi yang dimiliki oleh data yang terdapat dalam *data warehouse*. Untuk proses *data mining*-nya sendiri, perhatikan gambar berikut:



Gambar 2.7 Proses Data Mining

Diagram di atas adalah sebuah gambaran proses dari *Data Mining*. Ia diawali dari sebuah sumber data yang biasanya adalah sebuah *Data Warehouse* lalu dilakukan seleksi-seleksi untuk memilih data-data apa yang akan dianalisa. Hasil seleksi ini kemudian diubah menjadi bentuk-bentuk dimana algoritma-algoritma *Data Mining* dapat dilakukan. Hasilnya adalah suatu informasi penting (*knowledge*) tentang karakterisitik dari data itu sendiri yang akan sangat membantu para pembuat keputusan untuk mengarahkan kebijaksanaan dari perusahaan mereka.

Dari beberapa uraian di atas dapat diambil kesimpulan bahwa *data mining* adalah bermacam-macam, tergantung dari informasi apa yang ingin dihasilkan. Beberapa teknik *data mining* yang sedang dikembangkan antara lain adalah sebagai berikut :

- Ektrasi *rule* secara otomatis, pada aplikasi *knowledge based system* tradisional, *rule-rule*nya dibuat secara manual, dengan beberapa algoritma yang mampu "menggali" informasi dari data, dapat dihasilkan *rule-rule* secara otomatis
- Pencarian pola-pola Asosiasi, pada beberapa sistem retail, akan sengat bermanfaat bila diketahui pola pembelian dari konsumen seperti "produk X biasanya dibeli dengan jumlah Y, kemudian diikuti dengan produk Z"
- Pencarian pola-pola periodik, yaitu yang menjadi topik penyusunan dalam tugas akhir ini, yaitu berusaha mencari keperiodikan data pada konsep kategori data dan periode tertentu

BAB III

POLA-POLA PERIODIK

BAB III

POLA-POLA PERIODIK

Dalam bab ini, akan dibahas segala sesuatu yang berkaitan dengan pola-pola periodik, baik itu berupa rumusan, notasi, serta asumsi-asumsi yang dipakai. Dengan memahami apa itu pola-pola periodik beserta notasi dan formatnya, maka metoda-metoda yang akan disajikan pada bab bab selamjutnya akan mudah dipahami karena metoda-metoda tersebut mengambil *inputan* data-data yang sudah diformat terlebih dahulu yang akan dibahas berikut.

3.1 Data Beratribut Waktu

Dalam sebuah sistem basis data, transaksi yang terjadi selalu dicatat berikut dengan kapan terjadinya transaksi tersebut. Jadi ada sesuatu yang berhubungan (dihubungkan) dengan waktu dari terjadinya transaksi tersebut. Data beratribut waktu adalah data dimana nilainya selalu berubah secara dinamis sesuai dengan waktu yang dihubungkan terhadapnya.

Misalnya dalam pada basis data sebuah retail, transaksi yang dicatat jika terjadi penjualan mungkin berupa atribut-atribut: "nama produk", "jumlah", "waktu transaksi", dan sebagainya. Pada transaksi ini, atribut "jumlah" adalah data beratribut waktu, karena bila data ditabulasikan menurut atribut "nama produk", maka nilai dari atrbut "jumlah" ini akan berubah secara dinamis

kemiringan (gradien, *slope*) dari kurva tiap-tiap segmen waktu yang telah ditentukan.

Jadi misalkan segmen waktu yang dipilih adalah semesteran, maka untuk mengetahui trend pada semester pertama, dilakukan penganbilan data pada 6 bulan pertama kemudian dilakukan teknik *regresi linear* untuk mengetahui nilai kemiringan dari kurva yang dihasilkan dari 6 data tersebut. Hasil dari analisa *trend-based* ini mungkin seperti "volume produksi produk A pada semester kedua naik x% dibanding semeseter pertama".

3.2 Pola-pola Periodik

Dalam sebuah *time series*, dikatakan terdapat pola-pola periodik pada *time series* tersebut jika dalam *time series* itu terdapat satu atau lebih siklus (*cycle*). Sebuah *cycle* dibentuk jika pada *time series* tersebut terdapat unit waktu T_i dan unit waktu Tj yang sama (*similar*), yaitu $T_i = T_j$ pada selang waktu $[t_i, t_{i+k}]$ dimana k adalah konstanta yang menandakan panjang *cycle* yang bisa ditemukan dalam *time series* tersebut.

Untuk lebih memperjelas, berikut adalah definisi-definisi formalnya dan juga definisi-definisi lain yang berhubungan dengan operasi *time series* tersebut. Sebelumnya berikut adalah beberapa simbol yang dipakai beserta penjelannya.

$\boldsymbol{P}$: Himpunan pola-pola periodik

$\boldsymbol{P^i}$: Anggota himpunan pola-pola periodik ke-i

P_j : Anggota himpunan pola-pola periodik sementara ke-j

P_j^i : Anggota himpunan pola-pola periodik sementara ke-i pada indeks-j

mengikuti besaran waktu yang dalam contoh ini dihubungkan dengan atribut "waktu transaksi".

Pengumpulan data sesuai dengan obyek tertentu (dalam contoh di atas menurut obyek "nama produk") dalam jangka waktu tertentu akan menghasilkan sebuah *time series*. Secara matematis, sebuah time series adalah sekumpulan nilai-nilai v_i dari sebuah variabel Y pada saat t_i. Variabel Y di sini adalah obyek yang telah dipilih. Karena nilai t_i adalah sebuah besaran waktu, maka dapat dikatakan bahwa Y adalah sebuah fungsi waktu.

Untuk menganalisa sebuah *time series*, sering *time series* ini digambarkan menjadi sebuah grafik Cartesian, dimana sumbu Y mewakili nilai-nilai v_i dan sumbu X mewakili nilai-nilai t_i dari *time series*. Dari grafik ini, akan terlihat trend dari *time series* ini.

Bila kita melihat grafik *time series* dan memberikan analisa dengan mengambil nilai-nilai v_i per titik dari t_i, maka dikatakan kita melakukan analisa secara berbasis nilai (*value-based*). Dalam analisa *value-based* ini, kita lebih menekankan pada terjadinya perubahan nilai-nilai v_i pada tiap-tiap waktu t_i. Jadi hasil dari analisa ini dapat berupa "volume produksi produk A pada bulan Januari meningkat x% dibandingkan dengan bulan Februari".

Tetapi terkadang, diinginkan juga untuk melihat suatu nilai dalam jangka waktu tertentu sesuai dengan segmen-segmen waktu yang lazim (triwulanan, semesteran, dan lain-lain). Analisa seperti ini disebut analisa berbasis trend (*trend-based*), karena kita lebih menekankan pada perubahan nilai dalam jangka waktu yang telah ditentukan. Besarnya perubahan ini dapat dilihat dari nilai

$\boldsymbol{CP}$: Himpunan kandidat pola-pola periodik

$\boldsymbol{CP^i}$: Anggota himpunan kandidat pola-pola periodik ke-*i*

C : Himpunan *cycle*

C_i : Anggota himpunan *cycle* ke-*i*

C_j^i : Anggota himpunan *cycle* ke-*i* indeks ke-*j*

Berikut adalah definisi formal dari pola-pola periodik

- Definisi 1, untuk semua *time series* yang mempunyai panjang *n*, jika $\exists l, o, \in Integer, 0 \leq l \leq n$ *dan* $0 \leq o \leq l$ *dimana* $\forall s \in Integer, 0 \leq s \leq n / l$, unit waktu ke $(l . s + o)$ semuanya sama dalam *time series* ini, maka terdapat sebuah *cycle* $C = (l, o, V)$ dimana *l* adalah panjang *cycle*, *o* adalah *offset* yang menandai pertama kali *cycle* muncul, dan *V* adalah nilai dari *time series* pada posisi tersebut. Jadi misalkan terdapat *time series* "132113412341", maka dapat ditemukan mulai dari t_1 (indeks mulai dari nol) setiap 4 bit dari *time series* selalu terdapat nilai yang sama, yaitu 3. Maka dalam *time series* ini, terdapat sebuah *cycle* $C = (4,1,3)$ atau bisa juga ditulis *3**. *Cycle* lain yang juga terdapat dalam *time series* ini adalah (4,3,1) dan (6,2,2).
- Definisi 2, untuk semua *time series* yang mempunyai panjang *n*, jika untuk beberapa $l, m, \in Integer, 0 \leq l \leq n$ *dan* $m \geq 0, \exists C_i$ *sebanyak m dengan panjang* $l, 0 \leq i \leq l$, maka *cycle-cycle* ini membentuk suatu pola periodik $\boldsymbol{P} = (l, m, C)$ dimana *l* adalah panjang *cycle*, *m* adalah jumlah *cycle*, dan *C* adalah *cycle* yang terbentuk. Sama dengan contoh *time series* di atas, maka *time series* di atas mempunyai pola-pola periodik $\boldsymbol{P} = (4, 2, \{(1,3),(3,1)\})$. Karena tidak semua unit waktu pada pola-pola ini mempunyai *cycle* maka dikatakan pola

periodik ini adalah pola periodik parsial. Sebuah pola periodik dikatakan mempunyai pola periodik yang lengkap bila untuk setiap unit waktunya terdapat *cycle*, jadi banyak pola *m* adalah sama dengan panjang *cycle*.

- Definisi 3, jika terdapat sebuah *cycle* C, maka $C' = kC$ adalah juga sebuah *cycle* dimana C' dikatakan sebagai perkalian dari C dan k adalah nilai pengalinya. $C' = kC = (k \,.\, l, o, V)$

Sebuah pola disebut *i-cycle* jika panjang pola tersebut adalah *i*.

3.3 Time Series Analysis

Untuk mengetahui perilaku dari sebuah *time series*, perlu dipahami beberapa model dari *time series* yang ada. Pemahaman ini akan berguna untuk mengetahui *time series* apa yang sedang 'dihadapi' dan informasi apa saja yang dapat diambil dari *time series* tersebut.

Sebuah *time series* disebut stasioner (*stationary*) bila ia berfluktuasi secara acak (*randomly*) pada sebuah nilai *mean* yang konstan. Sebaliknya ia disebut tak-stasioner (*non-stationary*) bila untuk keseluruhan *time series*, tidak ada nilai *mean* yang konstan. Pemahaman apakah sebuah *time series* itu stasioner dan tidak stasioner adalah penting mengingat semua analisis dari *time series* (baik rumus atau asumsi-asumsi) yang dipakai adalah untuk mengidentifikasi sebuah model untuk *time series* tersebut kebanyakan untuk menganalisa *time series* yang statsioner. Jika sebuah *time series* diketahui tak-stasioner, maka harus diusahakan untuk membuat *time series* tersebut stasioner karena diinginkan model yang dihasilkan adalah sebuah model *time series* yang stasioner.

Dalam sebuah model yang stasioner, akan lebih mudah untuk melihat komponen-komponen penting dari sebuah *time series*, yaitu:

- *Trend*

  Trend adalah suatu kecenderungan bentuk dari perilaku *time series*. Bentuk di sini mengacu pada suatu bentuk fisik dari *time series* ketika ia digambarkan dalam sebuah grafik. Sebagai contoh, sebuah *time series* dapat disebut mempunyai trend bila pada *time series* itu dapat ditarik garis lurus yang menginterpolasi semua titik-titik dari *time series* . Dengan memperhatikan bentuk dari garis yang telah dihasilkan itu, dapat diamati apakah trend dari *time series* itu naik, turun, atau stabil.

- *Musiman (Seasonality)*

  Suatu *time series* dikatakan mempunyai pola musiman bila nilai-nilai pada suatu titik-titik tertentu, berulang pada titik-titik yang lain dimana jarak antara titik-titik ini adalah tetap. Misalnya, penjualan terigu meningkat bila mendekati hari lebaran atau natal karena komsumsi akan terigu meningkat untuk periode-periode tersebut.

Untuk mengetahui stasioneritas, trend, ataupun pola musiman, berikut akan diberikan beberapa perangkat yang dapat digunakan untuk mendeteksinya. Perangkat-perangkat ini sebenarnya adalah perangkat untuk menganalisa *bi-variate* data, artinya terdapat dua himpunan data yang akan dianalisa untuk diketahui hubungan satu himpunan data dengan lainnya. Tetapi karena dalam

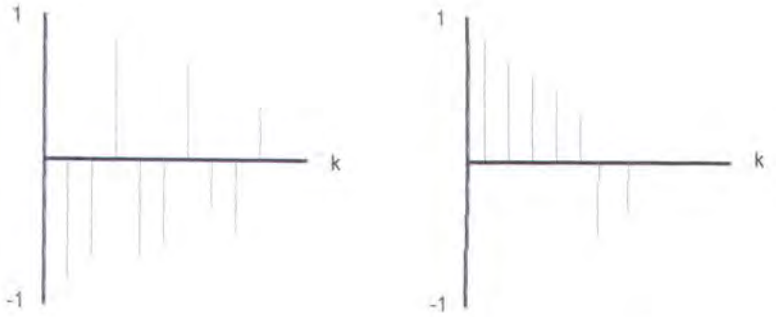
penyusunan tugas akhir ini hanya dibatasi pada satu himpunan data pada sebuah analisa, maka diambil asumsi bahwa data yang akan dianalisa adalah *Single Time Series Lagged on Itself*. Jadi nilai pengamatan pada x_t dihubungkan dengan nilai pengamatan pada x_{t-1} dari *time series* itu sendiri. Untuk itu ditambahkan istilah *auto* untuk menandakan bahwa analisa yang dilakukan adalah untuk mengamati hubungan antar data dalam *time series* itu sendiri.

Koefisien Auto-Korelasi

Koefisien auto-korelasi adalah nilai yang menandakan 'kedekatan' hubungan antara dua variabel. Karena data yang dipakai adalah dari satu *time series*, maka dua variabel di sini adalah dua data dari *time series* yang berselang k. Auto-korelasi pada selang k , yaitu r_k, dirumuskan sebagai berikut :

$$r_k = \frac{\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \qquad (3\text{-}1)$$

Adapun hubungan antara koefisien auto-korelasi dengan stasioneritas adalah bila nilai-nilai koefisien auto-korelasi ini digambarkan dalam sebuah grafik, dan nilai-nilai-nya cepat (dalam satu atau dua selang) berubah dari nilai positif ke negatif (atau sebaliknya, atau mendekati nilai nol) maka *time series* yang sedang diamati adalah stasioner. Sebaliknya bila nilai-nilai auto-korelasi dari suatu *time series* lambat menuju nilai nol (lebih dari lima selang) maka dapat dikatakan bahwa *time series* tersebut non-stasioner.



Gambar 3.1 Grafik Fungsi Auto-Korelasi

Grafik di atas adalah dua buah grafik fungsi auto-korelasi dari dua buah *time series* dimana grafik sebelah kiri menandakan sebuah *time series* yang stasioner dan grafik sebelah kanan menandakan *time series* yang non-stasioner. Untuk *time series* yang tidak stasioner, maka perlu dilakukan proses stasionerisasi dengan menggunakan metoda beda hingga orde pertama atau kedua yang akan dibahas berikut.

Beda Hingga

Suatu *time series* yang non-stasioner dapat dibuat menjadi stasioner dengan melakukan beda hingga orde-n pada *time series* tersebut. Beda hingga orde pertama dirumuskan sebagai berikut :

$$x_t' = x_t - x_{t-1} \qquad (3\text{-}2)$$

sedangkan untuk beda hingga order kedua dirumuskan sebagai berikut :

$$x_t'' = x_t' - x_{t-1}' \qquad (3\text{-}3)$$

dan seterusnya.

Suatu *time series* yang mempunyai pola musiman umumnya adalah stasioner, tetapi bila terdapat trend di dalamnya, maka kemungkinan *time series* tersebut menjadi non-stasioner.

BAB IV

METODA DAN SISTEM YANG DIGUNAKAN

BAB IV

METODA DAN SISTEM YANG DIGUNAKAN

Pencarian pola-pola periodik ini mengikuti algoritma *AprioriAll* yang dikembangkan oleh Rakesh Agrawal[2]. Algoritma ini pada awalnya diperuntukkan untuk mencari pola-pola sekuensial dari *customer* yang melakukan transaksi-transaksi yang tersimpan dalam sebuah basis data untuk kemudian diturunkan aturan-aturan asosiasi (*association rules*) antar transaksi itu.

Kemudian, algoritma *AprioriAll* ini dimodifikasi oleh Wan Gong[3] untuk kegunaan pencarian pola-pola keperiodikan pada suatu data dan merupakan topik utama dari penyusunan tugas akhir ini.

Pencarian pola-pola periodik dari suatu data beratribut waktu ini dibagi menjadi tiga bagian, yaitu:

- *Pencarian Pola Periodik Parsial Berbasis Nilai Pada Suatu Periode yang Tertentu (Partial Fixed Value-Based Search)*

  Parsial di sini berarti dari keseluruhan *time series*, maka hanya dicoba untuk dicari beberapa bagian dari *time series* yang membentuk suatu *cycle*. Keparsialan-nya akan ditentukan oleh sebuah nilai *threshold* γ, bila nilai *treshold*-nya 100% dapat juga diartikan kita berusaha mencari pola periodik yang lengkap (*complete patterns*) bukan lagi parsial. "Tertentu" (*Fixed*) di sini berhubungan dengan segmentasi waktu yang telah ditentukan dari awal dan

sifatnya adalah tetap, yaitu apakah akan mencari keperiodikan dalam periode triwulanan, semesteran, dan sebagainya. Berbasis nilai (*value-based*) berarti yang ingin ditekankan adalah perubahan nilai-nilai tiap-tiap titik dari *time series* yang sedang diamati.

- *Pencarian Pola Periodik Parsial Berbasis Trend Pada Suatu Periode yang Tertentu (Partial Fixed Trend-Based Search)*

  Hampir sama seperti *partial fixed value-based* , hanya di sini yang diamati adalah perubahan nilai-nilai dalam jangka waktu tertentu dari *time series*, atau pengamatan perubahan trend (*trend-based*)

- *Pencarian Pola Periodik Parsial yang Acak (Partial Arbitrary Search)*

  Acak (*Arbitrary*) di sini berarti bahwa tidak ada segmen waktu tertentu yang diasosiasikan terhadap *time series* seperti pada *partial fixed value-based* atau *partial fixed trend-based*. Jadi pencarian akan dilakukan terhadap pola-pola acak yang tidak mengikuti suatu periode waktu tertentu (semesteran, triwulanan, dan lain-lain)

Semua data yang akan dianalisa oleh ketiga metoda di atas diambilkan dari sebuah *data cube* . Berikut adalah alasan-alasan mengapa *data cube* dijadikan sumber masukan untuk ketiga metoda di atas:

- Data yang akan dianalisa adalah data-data yang disimpan dalam sebuah *data warehouse* dimana data-data ini mempunyai banyak dimensi seperti lokasi, kategori, produk, dan waktu. Dengan *data cube*, data-data dalam *data warehouse* ini dapat disajikan dengan baik menurut dimensi masing-masing yang diinginkan

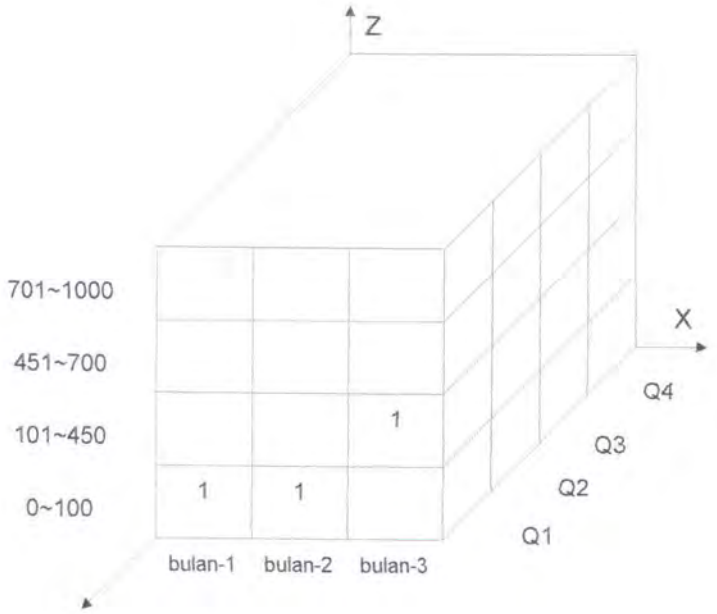
- Metoda-metoda di atas dalam operasinya "mengadopsi" teknik-teknik dari *data cube* dan OLAP, yaitu data diorganisasikan ke dalam kategori-kategori dan periode-periode yang diinginkan untuk kemudian dianalisa menurut periode-periodenya.

4.1 Partial Fixed Value-Based Search

Pada pencarian dengan metoda ini, kita memanfaatkan *data cube* untuk melakukan pencarian pola periodik dari data yang tersimpan dalam *data cube* tersebut. Berikut langkah-langkahnya:

1. Dari sumber basis data, buat sebuah *reference cube* yang berisi *query Q* dimana *query* ini menghasilkan paling sedikit satu dimensi waktu T, satu data beratribut waktu A_T, dan dimensi-dimensi lainnya diisi dengan data-data non-atribut waktu yang diinginkan
2. Buat suatu *time series* dari *reference cube* dimana *time series* ini merupakan nilai total dari data selama sebulan, atau dengan kata lain *time series* ini berisi data bulanan
3. Jika diinginkan, dapat dilihat terlebih dahulu apakah *time series* yang dihasilkan adalah *time series* yang stasioner atau tidak, jika tidak dapat dilakukan proses stasionerisasi dengan melakukan metoda beda hingga orde pertama
4. Dari *time series*, buat sebuah *working cube*. *Working Cube* adalah sebuah *cube* yang mana dua dari dimensinya adalah dimensi waktu, satu dimensi berisi

konsep kategori dari data, dan sel dari *cube* berisi pemetaan data. Untuk lebih jelasnya perhatikan gambar berikut



Gambar 4-1 Contoh Working Cube 3-bulanan

Cube di atas adalah sebuah *working cube* dimana data bulanan dalam setahun ingin disajikan dalam periode tiga-bulanan. Sumbu X merupakan indeks dari waktu (pada contoh ini adalah bulan, *time_index*) tiap tiga-bulanan, sumbu Y merupakan indeks dari periode (*periode_index)* tiga-bulanan dalam setahun (Q1 = tiga-bulanan pertama, Q2 = tiga-bulanan kedua, dst.), dan sumbu Z merupakan indeks kategorisasi dari data yang sedang diamati. Isi dari *cube* (sel) sendiri adalah pemetaan dari data dan disebut *value_id* untuk memudahkannya. Jadi misalnya data pada bulan Januari adalah 85, maka kita petakan ke koordinat (0,0,0) (bulan Januari adalah bulan pertama dari tiga-bulanan pertama, $Z = 0$ nilai 85 berada pada jangkauan 0~100 atau pada indeks 0)

dan kita isikan pada sel tersebut nilai satu yang artinya pada sel ini terdapat nilai seperti yang disajikan dalam sumbu-sumbu *cube*. Dengan cara yang sama, maka data 421 pada bulan Agustus akan dipetakan pada koordinat (1,2,2).

5. Mulai lakukan pencarian dengan algoritma sebagai berikut :

Algoritma Pencarian_Pola_Berbasis_Nilai_Periode_Tertentu(p)

Parameter : *p* adalah jumlah indeks periode

1. $\boldsymbol{P^1}$ = Cari_Pola_1_Cycle()
2. Untuk *i* = 2 sampai *p* lakukan langkah 3 sampai 5
3. $\boldsymbol{CP^i}$ = Bentuk _Pola_Kandidat(*i*)
4. $\boldsymbol{P^i}$ = Periksa_Keberadaan_Pola($\boldsymbol{CP^i}$)
5. Jika $\boldsymbol{P^i}$ tidak kosong maka hapus tiap-tiap pola-pola (*i*-1)-*cycle* yang membentuk P_j dalam $\boldsymbol{P^{i-1}}$, jika kosong maka **selesai**

Hasil : Pola-pola Periodik $\boldsymbol{P}$

Selesai

Gambar 4-2 Algoritma untuk Fixed Value-Based

Berikut adalah detil dari tiap-tiap prosedur yang ada di algoritma di atas

```
Procedure Cari_Pola_1_Cycle()
Begin
  NumCycle:= 0
  For time_index:= 0 to Number of Time Index do
    If number of value_id on time_index that fall
    into V_value_id is no less than minimal_pattern Then
      C_NumCycle:= (p, time_index, V_value_id) is a cycle
      Inc(NumCycle)
    End
  End
  Return 1-cycle periodic pattern P^1
End.
```

Gambar 4-3 Algoritma untuk Prosedur Find One Cycle Pattern

```
Procedure Bentuk_Pola_Kandidat(i)
Paramater i: Penanda i-cycles keberapa
Begin
  NumCandidate:= 0
  For every pair of patterns P_1^{i-1} = (p,i-1,C_1^{i-1}) and
  P_2^{i-1} = (p,i-1,C_2^{i-1}) in P^I Do
    If C_1^{i-1}[h] = C_2^{i-1}[h] ∀1 ≤ h ≤ i-2 And C_1^{i-1}[i-1] has a
    time index before C_2^{i-1}[i-1] Then
      CP^i_NumCandidate := (p,i,C^i) where C^i[h] = C_1^{i-1}[h] ∀1 ≤ h ≤ i-1
                           And C^i[i] = C_2^{i-1}[i-1]
    Insert CP^i into CP^i
    Inc(NumCandidate)
  End

  For every candidate i-cycle in CP^i do
    If ∃ some pattern P^{i-1} whose cycle set C_{i-1} is a
    subset of C_i of CP^i such that C_{i-1} is not in any
    pattern of P^{i-1} Then
      Delete CP^i from CP^i
    End
  End
  Return i-cycle periodic pattern set CP^i
End.
```

Gambar 4-4 Algoritma untuk Prosedur FormCandidatePattern

```
Procedure Periksa_Keberadaan_Pola(CP^i)
Parameter CP^i : Kandidat pola periodik i-cycles
Begin
  For every candidate i-cycle CP^i in CP^i Do
    Count:= 0
    For every periode_index on periode index dimension Do
      If all cube cells (periode_index, o_j, V_j) are non-empty
      ∀ 1 ≤ j ≤ i, (o_j, V_j) ∈ C_i in CP^i Then
        Inc(Count)
      End
    End
    If (Count / Number of Periode) ≥ γ Then
      Insert CP^i into P^i
    End
  End
  Return i-cyle periodic pattern set P^i
End.
```

Gambar 4-5 Algoritma untuk Prosedur CheckPatternExistence

Untuk lebih memperjelas algoritma serta detil-detil prosedurnya, maka berikut akan diberikan sebuah contoh sederhana. Katakanlah kita telah membentuk sebuah *reference cube* yang berisi data penjualan bulanan pada th. 1998 dari sebuah produk X dan ditunjukkan dalam tabel berikut:

Table 4-1 Data Untuk Contoh 1

Jan	Feb	Mar	Apr	Mei	Juni	Juli	Agust	Sept	Okt	Nov	Des
638	588	1055	221	795	1042	647	824	1043	637	804	1047

Table 4-2 Konsep Kategori untuk Contoh 1

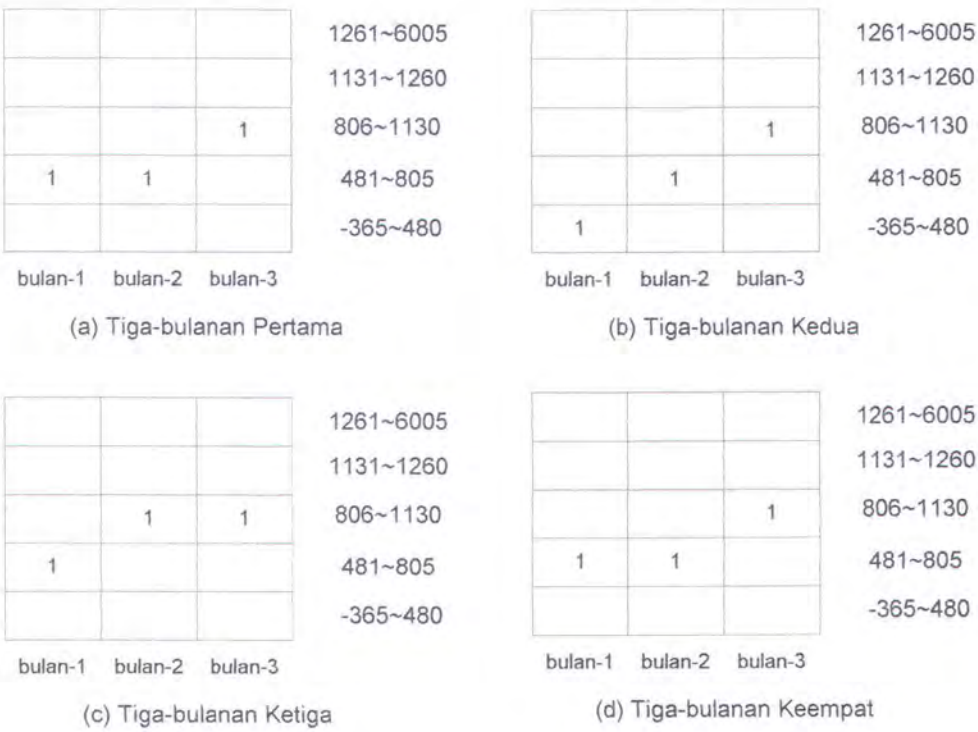
Kategori Indeks	Nilai
0	-365 ~ 480
1	481 ~ 805
2	806 ~ 1130
3	1131 ~ 1260
4	1261 ~ 6005

Kemudian ingin diperiksa, apakah ada pola periodik dalam periode tiga-bulanan dari data penjualan di atas. Diberikan juga nilai *threshold* $\gamma = 75\%$, maka dengan data ditentukan dalam periode tiga-bulanan, maka *time_index* adalah sama dengan 3 dan jumlah periode yang dapat dibentuk adalah 4, jadi pola minimal yang harus ada supaya ia disebut periodik minimal adalah γ x *number of periode* = 75%.4 = 3. Jadi secara matematis suatu data adalah disebut kandidat pola periodik bila kemunculannya dalam *working_cube* untuk kategori ke-i adalah sama dengan

$$Minimal_pattern = \gamma * \text{jumlah_indeks_periode_dalam_cube}$$

Langkah pertama, buat sebuah working cube dengan spesifikasi di atas. Berikut adalah *working cube-nya* dengan disajikan dalam *slice-slice* menurut periodenya.

Cara pembuatan *working cube* adalah dengan memetakan tiap data terhadap nilai indeks kategorinya, serta terhadap "tempat" dimana data itu dihasilkan baik dalam satuan bulan-nya dan satuan tempat bulan itu berada dalam periodenya. Contoh: data bulan Januari adalah 638 termasuk dalam kategori kedua (indeks = 1) jadi nilai pada sumbu Z = 1. Bulan Januari sendiri adalah bulan pertama dalam periode 3 bulanan, jadi nilai pada sumbu X = 1. Karena datanya adalah data selama satu tahun, maka bulan Januari pada di atas adalah terletak pada periode tiga-bulanan pertama atau indeks-nya menurut indeks perioda adalah 0. Berikut adalah *working cube* dengan dipotong-potong menurut indeks periodanya.



Gambar 4-6 Slices untuk Working Cube pada Contoh 1

Langkah kedua, yaitu mencari pola-pola 1-cycle. Pola *1-cycle* adalah pola dengan panjang 1 dan mempunyai jumlah *value_id* ≥ nilai minimal pola yang diijinkan , yaitu 3. Jadi kita mencari mulai dari *time_index* = 0 (yaitu *bulan-1*) sampai *time_index* = 2 (bulan-3) untuk setiap kategori. Pada *bulan-1*, jumlah *value_id* yang bernilai 1 pada kategori 1(481 ~ 805) adalah 3, jadi terdapat *cycle* di *bulan-1*, yaitu $C_j = (l, o_j, v_j)$. = $C_0 = (3,0,1)$, *l* adalah jumlah kemunculan nilai di o_j dan v_j, o_j adalah *time_index*, dan v_j adalah indeks kategori. Dengan cara yang sama kita dapatkan untuk *bulan-2* dan *bulan-3*, yaitu $C_1 = (3,1,1)$ dan $C_2 = (3,2,2)$ yang membentuk pola periodik $\boldsymbol{P^1} = \{ P_0^1 = (3,1,\{(0,1)\}),\ P_1^1 = (3,1,\{(1,1)\}),\ P_2^1 = (3,1,\{(2,2)\}) \}$.

Langkah ketiga, pencarian kandidat-kandidat pola. Di sini, kita berusaha mencari pola-pola *i-cycle* yang lain yang mana nilai *i* adalah sampai *number of time index*. Alasannya adalah kita berusaha mencari pola-pola selengkap mungkin dengan mencoba menggabungkan pola–pola *(i-1)cycle* yang telah terbentuk. Maka untuk *i* = 2, kita akan menggabungkan tiap pola di pola *1-cycle* yang membentuk $CP_0^2 = (3,2,\{(0,1),(1,1)\})$, $CP_1^2 = (3,2,\{(0,1),(2,2)\})$, dan $CP_2^2 = (3,2,\{(1,1),(2,2)\})$. Selanjutnya adalah memeriksa apakah dari tiap-tiap kandidat pola ini, terdapat C_j yang merupakan subset dari CP^i tetapi bukan merupakan subset dari $\boldsymbol{P^{i-1}}$. Jadi misalkan kita mempunyai pola $\boldsymbol{P^2} = \{(3,2,(\{0,1),(2,2)\}), (3,2,\{(1,1),(2,2)\})\}$ maka hasil penggabungannya adalah $CP^3 = (3,3,\{(0,1),(1,1),(2,2)\})$, tetapi kandidat ini tidak bisa disertakan karena salah satu sub-pola dari CP^3 ini yaitu $CP_0^3 = (3,2,(\{0,1),(1,1)\})$ adalah bukan subset dari $\boldsymbol{P^{i-1}}$.

Langkah 4, memeriksa keberadaan pola. Langkah ini adalah memeriksa apakah masing-masing pola yang ada di kandidat $\boldsymbol{CP^i}$ memenuhi jumlah minimal pola yang diijinkan, yaitu dengan menghitung jumlah *kemunculan* (*occurence*) dari kandidat-kandidat ini pada tiap-tiap *slice* dari *working cube*. Sebagai contoh, kita ambil salah satu sub-pola dari $\boldsymbol{CP^2}$, yaitu CP_0^2 = (3,2,{(0,1),(1,1)}). Jika kita periksa, pola ini mewakili *value_id* untuk bulan pertama,bulan kedua dan indeks kategori 1. Ternyata pola ini hanya muncul pada *slice* pertama dan *slice* keempat atau pada tiga-bulanan pertama dan tiga-bulanan keempat. Jadi tidak memenuhi jumlah pola minimal yang diijinkan, yaitu 3. Jadi pola kandidat ini harus dihapus dari $\boldsymbol{CP^2}$.

Langkah 5, lakukan langkah 3 dan 4 sampai tidak ada kandidat lagi yang bisa dibentuk.

Pada contoh ini, pola akhir yang bisa diperoleh, yaitu berjumlah dua pola $\boldsymbol{P}$ = {(3,2,{(0,1),(2,2)}), (3,2,{(1,1),(2,2)}) }. Pola ini bisa diinterpretasikan bahwa pada volume penjualan produk X selama th. 1998, terdapat pola periodik tiga-bulanan, yaitu volume penjualan pada bulan pertama dan ketiga tiap tiga bulan sekali berada pada nilai 406 ~ 805.

4.2 Partial Fixed Trend-Based Search

Pencarian pola periodik dengan metoda ini pada dasarnya sama seperti dengan *Partial Fixed Value-based Search*, hanya yang dicari keperiodikannya adalah nilai-nilai dalam jangkauan waktu tertentu. Jadi bila dilakukan pencarian pola periodik pada data bulanan dalam periode tiga bulanan, maka akan dicari

nilai trend dari nilai data tiap tiga bulan. Nilai trend ini adalah nilai kemiringan (*slope, gradien*) dari garis lurus yang akan diaproksimasi dari ketiga titik (nilai data). Teknik yang akan digunakan untuk mencari nilai kemiringan ini adalah dengan menggunakan teknik regresi linear.

Persamaan garis lurus dari sehimpunan n buah titik dengan koordinat (x_i, y_i) adalah $y = ax + b$, dimana a nilai kemiringan dari garis dan b adalah sebuah konstanta. Nilai a dan b dapat diperoleh dengan menyelesaikan persamaan berikut

$$a = \frac{n(\sum_{i=1}^{n} x_i y_i) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2} \tag{4.1}$$

$$b = \frac{(\sum_{i=1}^{n} x_i^2)(\sum_{i=1}^{n} y_i) - (\sum_{i=1}^{n} x_i y_i)(\sum_{i=1}^{n} x_i)}{n(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2} \tag{4.2}$$

Dalam hal ini, x_i adalah indeks dari data sedangkan y_i adalah nilai dari data pada indeks ke x_i . Jadi data-data pada Table 4.1 dapat disajikan seperti titik-titik koordinat sebagai berikut : {(1,638), (2,588), (3,1055), ...}.

Setelah nilai-nilai kemiringan dari tiap-tiap periode didapatkan, maka nilai-nilai ini yang akan dimasukkan ke dalam *working cube*. Tentu saja, dengan perubahan ini, akan terjadi juga perubahan satuan indeks tiap dimensi *cube*. Jika pada *partial fixed value-based*, dimensi waktu-nya adalah bulan, maka pada *trend-based* ini dimensi waktunya menjadi periode. Sedangkan dimensi periodenya dapat digeneralisasi menjadi periode yang lebih tinggi sesuai konsep hirarkinya, jadi bisa dibawa ke arah semesteran atau tahunan.

Selain itu, karena trend melibatkan beberapa unit waktu, maka dapat dibuat suatu keterangan yang menunjukkan apa yang terjadi dalam trend tersebut, naik, stabil, atau turun. Ini bisa dilakukukan dengan mengisikan keterangan-keterangan yang diasosiasikan dengan konsep kategori yang dikehendaki. Sebagai contoh, berikut adalah konsep kategori dari trend penjualan produk seperti pada contoh untuk kasus *partial fixed value-based* di atas.

Table 4-3 Daftar Trend Beserta Keterangannya

Indeks Kategori	Nilai	Keterangan
0	-100 ~ 0	Turun Tajam
1	1 ~ 200	Turun
2	201 ~ 500	Stabil
3	501 ~ 800	Naik

Sekali lagi, nilai pada tabel di atas, adalah nilai kemiringan (*slope*) dari *m* data dengan *m* adalah nilai dari suatu segemen periode yang telah ditentukan, jadi bila dipilih periode tiga-bulanan, maka nilai *m* adalah 3 atau jumlah data yang dibutuhkan untuk menghitung nilai kemiringan adalah 3.

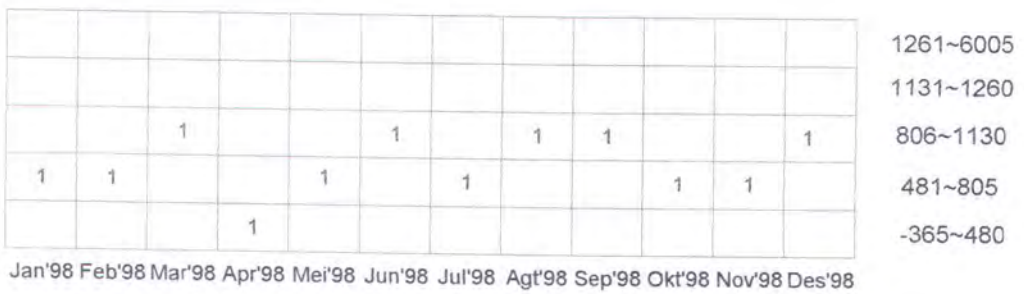
Untuk lebih jelasnya, berikut adalah *working cube* dengan mengambil contoh data seperti pada contoh pada bagian *partial fixed value-based*.

4.3 Arbitrary Search

Pada pencarian pola periodik dengan pola acak (*arbitrary*) ini, tidak ada suatu segmen periode tetap yang diberikan atau diasosiasikan sebagai asumsi pencarian awal, karena sering dijumpai bahwa data tidak berpola periodik menurut suatu segmen periode tertentu.

Sama seperti metoda-metoda pencarian sebelumnya, di sini juga memanfaatkan suatu *working cube* untuk menyajikan data ke dalam suatu bentuk yang cocok untuk metoda pencarian ini. Karena tidak ada suatu segmen periode tertentu yang diasosiasikan terhadap data, maka bentuk *cube*-nya sendiri sedikit berbeda dengan *cube-cube* yang telah dibentuk pada contoh-contoh sebelumnya.

Dengan mengambil contoh data yang masih sama seperti contoh-contoh sebelumnya, maka *working cube* di atas adalah hasil pembentukan untuk pencarian pola model acak ini. Jadi data bulanan dipetakan pada *cube* sesuai dengan nilainya dan karena tidak ada segmen periode, maka dimensi indeks periode dari *cube* (sumbu Y) tidak ada.



Gambar 4.7 Working Cube untuk Arbitrary Search

Dengan melihat *cube* di atas, maka maka data dapat dikodekan (ditransformasikan) menjadi sebuah *time series* menurut indeks kategori dari data (dalam contoh ini berarti kategori dari volume penjualan produk X) sebagai berikut : S =1 1 2 0 1 2 1 2 2 1 1 2. Panjang dari *time series* ($|S|$)adalah sama dengan jumlah data bulanan dan nilai v_i dari *time series* adalah nilai indeks dari data terhadap konsep kategorinya.

Pencarian pola-pola periodik acak yang akan dibahas ini mengacu erat pada prinsip bahwa hasil perkalian dari sebuah pola-pola periodik adalah juga

menghasilkan pola-pola periodik. Jadi bila (l, o, V) adalah sebuah *cycle* dalam sebuah pola yang mempunyai panjang l, maka untuk sebuah pola dengan panjang $l \times s$ (tiap kelipatan dari l), tiap-tiap l unit waktu dimulai dari t_i pasti mempunyai sebuah *cycle* dengan nilai V.

Satu hal lagi, bila *threshold* γ pada dua metoda terdahulu dikaitkan dengan jumlah segmen periode waktu yang dapat dihasilkan, maka γ pada *arbitrary search* ini dikaitkan dengan panjang *time series* atau jumlah data dan merupakan panjang pola maksimal yang akan coba dicari. Jadi $p_max = \gamma x|S|$.

Berikut adalah algoritmanya

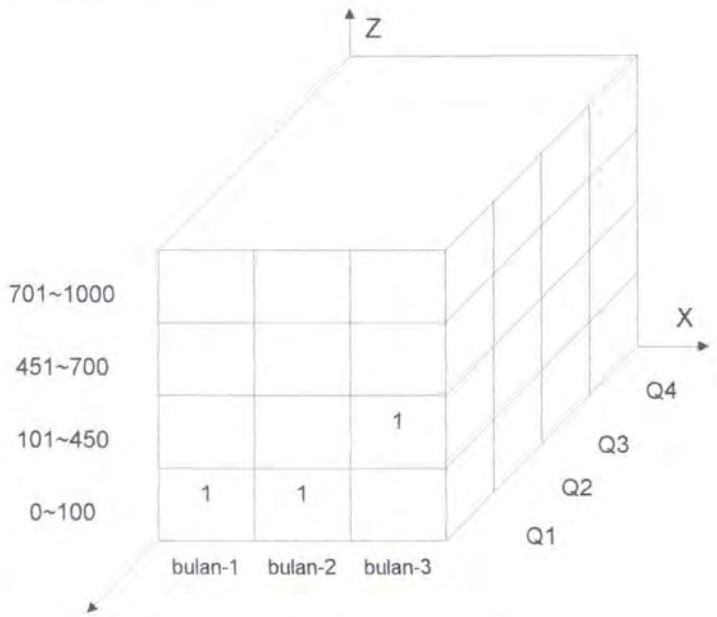
```
Procedure ArbitrarySearch(S)
Parameter S : time series
Begin
  For j:= 1 to p_max Do
    Pattern_length_{j-1}:= j
    Num_cycle_{j-1}:= 0
    P_{j-1}:= (Pattern_length_{j-1} , Num_cycle_{j-1} , C = empty)
  End
  For k:= 0 to p_max - 1 DO
    Is_complete_pattern:= true
    New_cycle_exist:= false
    For j:= 0 to pattern_length_k - 1 Do
      If (j,S[j]) not ∈ C_k Then
        If S[j] = S[j+k.t] ∀t > 0 and j+k.t < |S| Then
          New_cycle_exist:= true
          Num_cycle_k:= num_cycle_k + 1
          Insert (j, S[j]) into C_k of P_k
        End Else is_complete_pattern:= false
      End
    End
    If is_complete_pattern Then
      For every P_s where pattern_length_s is multiple of
      pattern_length_k Do Delete P_s
      Insert P_k into P^j
    End Else
     If new_cycle_exist Then
     End Else Delete the current pattern P_k
  End
End.
```

Gambar 4.8 Algoritma untuk Arbitrary Search

Dalam melakukan pencarian dengan metoda *arbitrary* ini, pemilihan nilai *threshold* harus dilakukan dengan hati-hati, karena *threshold* di sini adalah untuk menentukan panjang pola maksimal dari *time series* yang ada. Jadi bila diinginkan (diduga) terdapat pola 7 bulanan, maka harus dicari nilai *threshold* dimana jika nilai ini dikalikan dengan panjang *time series* adalah menghasilkan nilai yang mendekati tujuh.

4.4 Analisa Cube

Dari beberapa uraian di atas, ada dua macam analisa *cube* yang digunakan pada penyusunan tugas akhir ini



Gambar 4.9 Contoh Working Cube 3-bulanan

- Analisa keperiodikan, analisa ini dilakukan pada *cube* dengan melakukan operasi pemeriksaan tiap sel-sel dari *working cube* mulai dari depan ke belakang, atau sepanjang sumbu Y (pada Gambar 4.9 di atas) yang merupakan

kumpulan dari *quarterly data*. Jadi misalkan terdapat data dua tahun dan *working cube* dibentuk pada periode tiga-bulanan, maka terdapat 15 periode tiga-bulanan, maka analisa keperiodikan dilakukan pada tiap sel-sel mulai dari periode tiga-bulanan pertama sampai periode tiga-bulanan ke limabelas

- Analisa Trend, analisa ini dilakukan pada *cube* dengan melakukan pemeriksaan terhadap tiap-tiap sel dari *working cube* dari kiri ke kanan. Seperti pada Gambar 4.9 berarti sepanjang sumbu X yang merupakan sel-sel dari pemetaan data bulanan Jadi misalkan terdapat data dua tahun dengan *working cube* dibentuk pada periode tiga-bulanan, maka untuk tiap lima belas periode yang terjadi, tiap-tiap bulannya dilakukan pemeriksaan (kiri ke kanan) untuk didapatkan trend-nya

BAB V

PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

BAB V

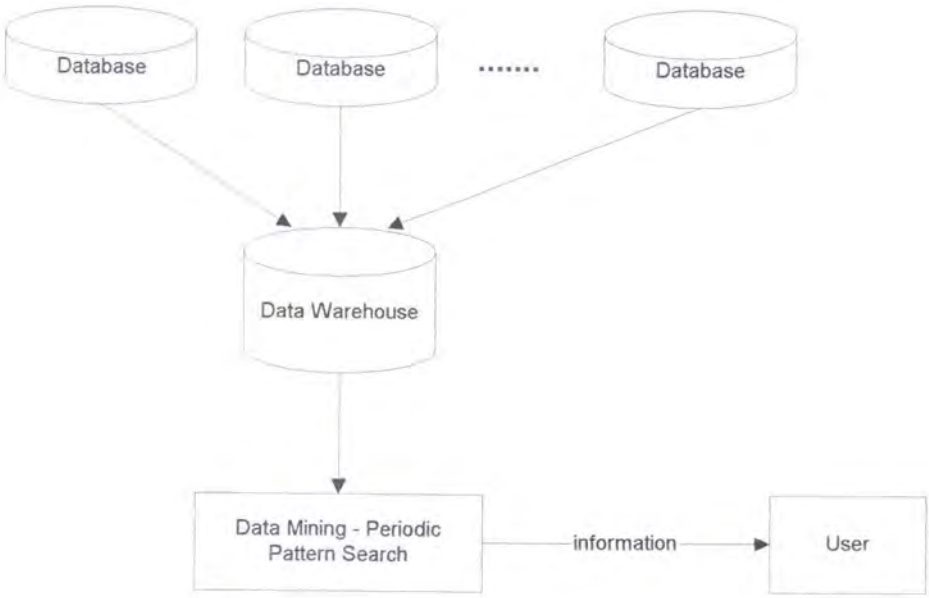
PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK

Sesuai dengan tujuan dari penulisan tugas akhir ini yang terdapat pada bab I, maka dapat disusun beberapa kriteria yang nantinya akan menjadi spesifikasi dari perangkat lunak yang dihasilkan. Kriteria-kriteria ini adalah sebagai berikut :

1. Perangkat lunak mampu membuka suatu sumber data, yaitu *file-file* basis data relasional dengan format tertentu
2. Perangkat lunak mampu melakukan operasi-operasi pra-proses untuk data sebelum memasuki proses utama, pra-proses ini meliputi analisa awal dari data, yaitu apakah ia stasioner, mempunyai pola musiman, atau mempunyai trend, atau gabungan dari padanya
3. Perangkat lunak mampu melakukan pencarian pola-pola periodik dari sumber data yang telah dimasukkan. Pencarian ini meliputi *fixed value-based search, fixed trend-based search,* dan *arbitrary search*
4. Perangkat lunak mampu memberikan interpretasi dari pola-pola periodik yang (jika) ditemukan dengan bahasa yang mudah untuk dibaca

Untuk memenuhi kriteria-kriteria seperti di atas, maka dibuat suatu rancangan dalam pembuatan perangkat lunak ini yang akan dijelaskan pada bab

ini. Perangkat lunak yang akan dibuat ini akan mengambil contoh sebuah sistem retail sederhana. Dalam sistem retail ini, terdapat beberapa macam format basis-data penjualan dari retail ini diantaranya adalah format Microsoft Access (*.mdb) dan dBASE (*.dbf). Jadi dengan perangkat lunak ini diharapkan dapat memberi ilustrasi dari sebuah *data warehouse* sederhana. Berikut adalah diagram dari proses perangkat lunak.



Gambar 5-1Diagram Proses System

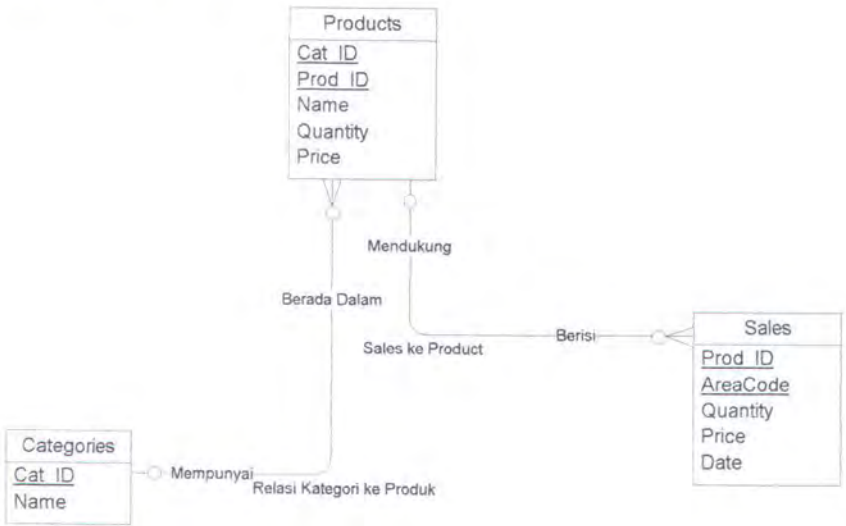
5.1 Perancangan Perangkat Lunak

Perancangan perangkat lunak di sini meliputi data apa saja yang akan dipakai sebagai masukan dari perangkat lunak, serta proses apa saja yang terjadi dalam perangkat lunak nantinya. Proses-proses ini akan menjadi tuntunan bagi

user dalam menjalankan perangkat lunak ini nantinya (*guidelines*) dan juga menjadi dokumentasi proses dari perangkat lunak.

5.1.1 Permodelan Data Konseptual

Permodelan data konseptual di sini adalah memodelkan data masukan untuk perangkat lunak yang akan dibuat dan telah ditetapkan untuk mengambil contoh sebuah sistem *warehouse* sederhana dari sebuh sistem retail. Seperti telah digambarkan pada Gambar 5.1, sumber data dari *warehouse* adalah banyak *database*. *Database-database* ini diasumsikan tersebar pada cabang-cabang di beberapa kota dengan berbagai format. Jadi bisa saja *database1* ada di kota A dengan memakai *enginee* Microsoft Acess, kemudian *database2* ada di kota B dan memakai *enginee* Dbase.



Gambar 5.2 Pemodelan Konseptual Data untuk Tiap-tiap Database dari sistem

Gambar di atas adalah diagram pemodelan konseptual data dari masing-masing *database* yang menjadi sumber dari *data warehouse*. Dari diagram di atas,

dapat dilihat bahwa terdapat tiga tabel pada masing-masing *database*, yaitu : *Categories*, *Products*, dan *Sales*. Data-data dari masing-masing tabel di atas adalah sebagai berikut :

Table 5.1 Tipe Data dari Tabel Categories

Nama Field	Tipe	Ukuran
Cat_ID	Number	2
Name	Char	25

Table 5.2 Tipe Data dari Tabel Products

Nama Field	Tipe	Ukuran
Cat_ID	Number	2
Prod_ID	Number	2
Name	Char	25
Quantity	Float	5
Price	Currency	5

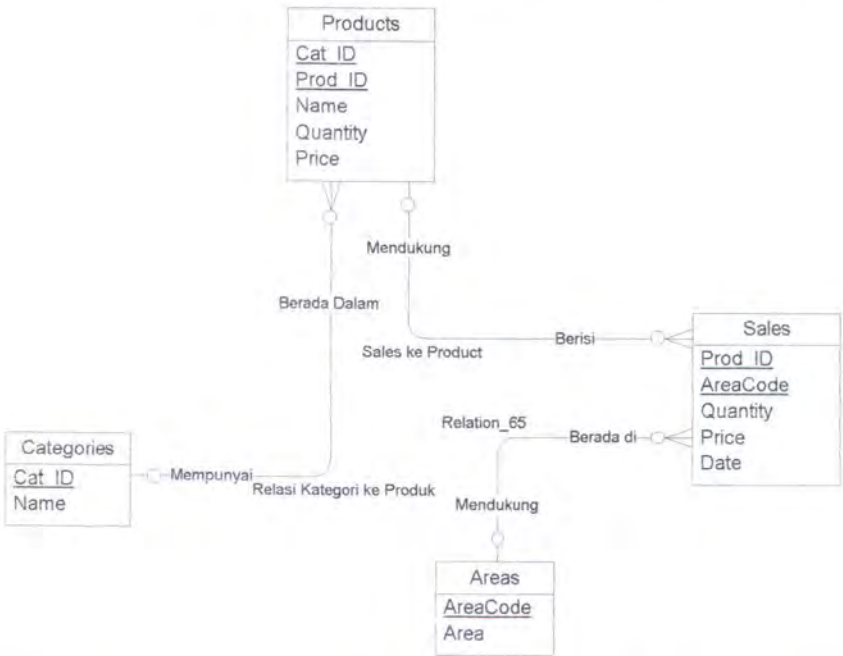
Table 5.3 Tipe Data dari Tabel Sales

Nama Field	Tipe	Ukuran
Prod_ID	Number	2
AreaCode	Number	2
Quantity	Float	5
Price	Currency	5
Date	Date	-

Struktur di atas terdapat dalam semua *database* yang terlibat. Hal ini untuk memudahkan dalam melakukan pengorganisasian data dalam *data warehouse* nantinya. Di samping itu, desain yang standar adalah dianjurkan meskipun dalam kondisi yang ekstrem, tiap-tiap *database* dalam *data warehouse* bisa saja mempunyai struktur yang berbeda.

Langkah selanjutnya (masih seperti desain pada Gambar 5.1) adalah mengumpulkan *database-database* ini ke dalam suatu *data warehouse* yang

merupakan tempat penyimpanan terahir dari data yang berasal dari berbagai macam sumber (dalam kasus ini adalah dari berbagai macam lokasi). Berikut adalah pemodelan konseptual data dari *data warehouse* yang terbentuk.



Gambar 5.3 Permodelan Konseptual Data untuk Sistem Retail Sederhana

Pada dasarnya, entitas-entitas yang terlibat dalam konseptual daa dari *data warehouse* adalah sama dengan konseptual data dari *database* sumber, hanya terdapat satu tambahan yaitu entitas *areas* yang mencatat lokasi dari masing-masing *database*.

Terdapat empat buah entitas di sini, yang masing-masing mempunyai kegunaan sebagai berikut :

- *Entitas Area*

  Entitas ini berisi nama-nama lokasi dimana cabang-cabang retail ini berada

- *Entitas Categories*

Entitas ini berisi kode-kode kategori dari produk-produk yang ada dalam dalam sistem retail ini. Jadi kategori dan produk adalah suatu konsep hirarki (lihat penjelasan bab II.1-1), contohnya adalah produk sapu, pembersih debu elektrik, dan lain-lain adalah masuk dalam kategori peralatan rumah tangga

- *Entitas Products*

  Entitas ini berisi properti-properti dari produk yang dijual dalam sistem retail, yaitu meliputi harga, kuantiti, dan lain-lain.

- *Entitas Sales*

  Entitas ini adalah entitas yang menrupakan suatu transaksi harian dari retail tersebut. Pembelian pada waktu (tanggal), jumlah pembelian, dan harga per buah dari produk dicatat di sini setiap terjadi transaksi pembelian oleh *customer*. Juga dicatat tempat terjadinya transaksi ini (pada atribut *areaCode*).

Dari ketiga entitas ini, terdapat empat buah dimensi yang mana dengan adanya multi dimensi ini, adalah cocok untuk masalah *data mining*. Dimensi-dimensi tersebut adalah

1. Lokasi
2. Kategori
3. Produk
4. Waktu

Dimensi-dimensi ini pula yang nantinya akan ada dalam *data cube* yang akan melayani semua permintaan untuk melakukan proses penggalian informasi tentang keperiodikan dari data penjualan ini.

5.1.1 Diagram Alir Data (Data Flow Diagram)

Hal selanjutnya adalah memodelkan proses dan data dari perangkat lunak ini yang akan digambarkan dengan menggunakan *Data Flow Diagram* (DFD). Dalam penjelasan DFD ini, disertakan juga *Data Dictionary* dari tiap-tiap data dan juga aliran dari tiap data. Selain itu, tiap proses akan dijelaskan dengan rinci.

5.1.1.1 Level 0

Gambar 5.4 DFD Level 0 dari Sistem

Proses utama dari sistem digambarkan pada DFD level-0 di atas, dimana *user* memasukkan *mining request*, yaitu suatu input tentang basis-data mana saja yang akan dipilih dan properti-properti dari metoda pencarian pola seperti kategori, metoda pencarian pola apa, berapa nilai thresholdnya, dan lain-lain. Basis data di sini adalah basis data *sales* dari berbagai macam lokasi dan format. Setelah itu *user* memasukkan *task*, yaitu pola-pola keperiodikan apa saja yang ingin diketahui dari basis-data yang dipilihnya. Seperti diketahui ada tiga macam metoda untuk mencari pola-pola keperiodikan, yaitu *fixed partial value-based* ,

fixed partial trend-based dan *arbitrary*. Hasil akhirnya adalah informasi-informasi pola-pola keperiodikan yang diterima oleh *user*.

Berikut adalah *Data Dictionary* dari *Mining Request* dan *Periodic Patterns Informations*

Table 5.4 Flow Mining Request

Name	Mining Request
Code	FLOW_MINING_REQUEST
Label	
Type	Logical
Source	DBExplorer Mining System (Process)
Destination	User (External Entity)

Table 5.5 Data Flow Data Item List

Name	Code	Data Type
Area	AREA	A10
Category	CATEGORY	A10
Product	PRODUCT	A10
OrderDate	ORDERDATE	D
MiningMethod	MININGMETHOD	A10
MiningPeriode	MININGPERIODE	A10
MiningThreshold	MININGTHRESHOLD	F3

Table 5.6 Flow Periodic Patterns Informations

Name	Periodic Pattern Informations
Code	FLOW_RESULT
Label	
Type	Logical
Source	User (External Entity)
Destination	DBExplorer Mining System (Process)

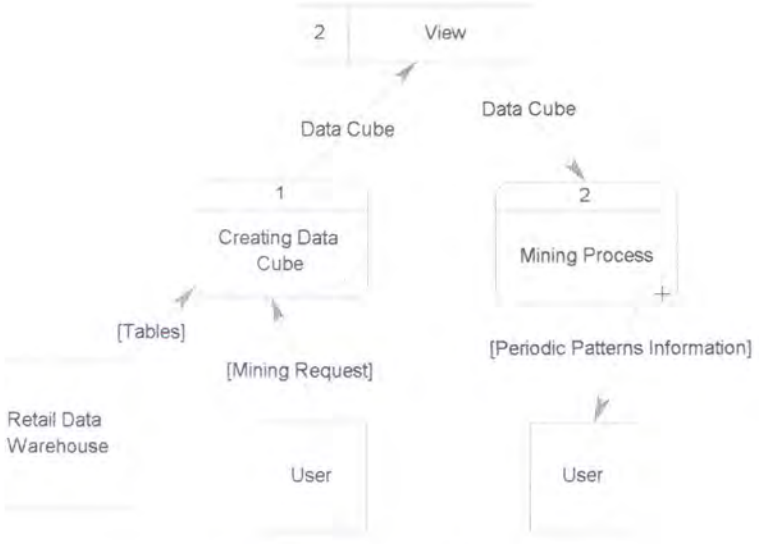
Table 5.7 Data Flow Data Item List

Name	Code	Data Type
Periodic_Pattern	PERIODIC_PATTERNS	A100

Retail Data Warehouse adalah sebuah entitas luar (*external entity*) dimana data-data pada entitas ini yang masuk dalam *DBExplorer System* . Entitas ini berisi tabel-tabel yang mana desain konspetual-nya telah dijelaskan pada bab IV.1-1. Jika entitas ini diganti dengan entitas lainnya, perbankan misalnya, maka *DBExplorer System* diartikan akan menggali informasi keperiodikan dari data yang ada pada sistem perbankan tersebut. Jadi, *DBExplorer System* adalah perangkat lunak yang bersifat umum.

5.1.1.2 Level 1

Selanjutnya dilanjutkan dengan DFD level-1, yaitu untuk lebih memperjelas proses apa saja yang berada dalam *DB Explorer System.*



Gambar 5.5 DFD Level 1 dari Sistem

Proses selanjutnya adalah pembentukan *Data Cube*. *Data Cube* adalah proses penyajian data yang ada dalam sistem retail ke dalam bentuk multidimensi,

yaitu lokasi, kategori, produk, dan waktu transaksi. Proses pembentukan *data cube* ini dilakukan dengan perintah SQL sebagai berikut :

```
Select Areas.Area, Categories.CategoryName,
 Products.ProductName, Sales.OrderDate,
 Sum(Sales.Price*Sales.Quantity) as Total_Sales
From Areas, Categories, Products, Sales
Group By Areas.Area, Categories.CategoryName,
 Products.ProductName, Sales.OrderDate
```

Hasil dari eksekusi perintah SQL di atas kemudian disimpan untuk sementara dalam sebuah *view*. *View* inilah yang nantinya menjadi *server data* untuk semua proses di dalam *mining process* ini.

Table 5.8 Data StoreView

Name	View
Code	STOR_VIEW
Label	
Number	2
Is Entity	Yes

Table 5.9 Data Store View Item List

Name	Code	Data Type
Area	AREA	A10
Category	CATEGORY	A10
Product	PRODUCT	A10
OrderDate	ORDERDATE	D

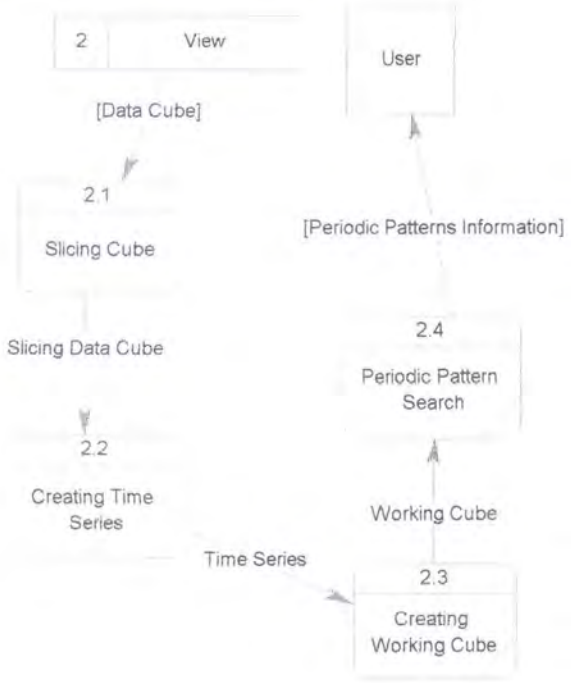
5.1.1.3 Level 2

Selanjutnya berikut adalah DFD level 2 yang akan memperjelas proses dari penggalian informasi ini (*mining process*).

Proses *mining* dimulai dengan melakukan *slicing cube*, yaitu operasi pemilihan data yang diinginkan untuk dianalisa. Jadi misalkan diinginkan untuk menganalisa apakah terdapat keperiodikan pola pada data penjualan produk A di

Semarang selama tahun 1993 - 1996, maka kita melakukan *slicing* dari *data cube* sehingga menghasilkan data-data penjualan produk A di Semarang selama tahun 1993-1996. Perintah SQL untuk melakukannya adalah sebagai berikut :

```
select * from data_cube
where (data_cube.Area = "Semarang") and
      (Year(data_cube.OrderDate) between 1993 and 1996)
```



Gambar 5.6 DFD Level 2 dari Sistem

Hasil dari perintah SQL ini adalah *slicing data cube* yang mana tipe dari data adalah sama dengan dengan yang ada pada *data cube*, tetapi telah mengandung data yang berbeda sesuai dengan kriteria yang dimasukkan pada *slicing* seperti pada di atas.

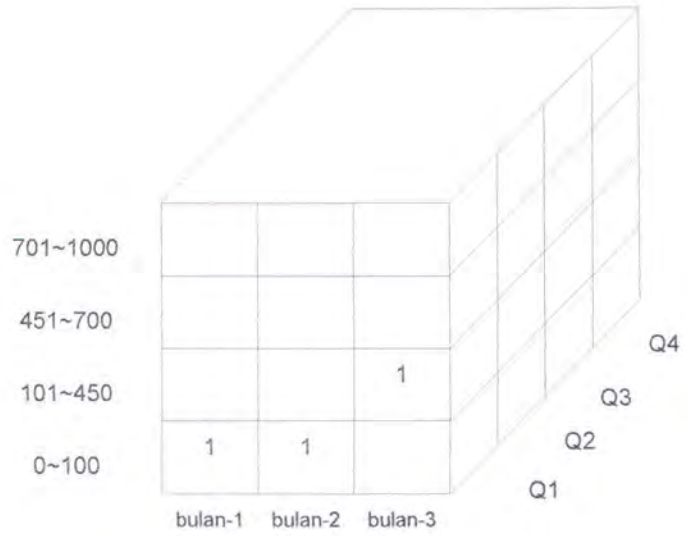
Selanjutnya dilakukan trasnformasi data hasil *slicing* ini menjadi sebuah *time series* dimana data pada *time series* ini adalah menjadi data bulanan. Dalam contoh sistem retail ini berarti data Total_Sales yang akan dikumpulkan secara

bulanan. Sekaligus di sini juga diperiksa apakah *time series* ini stasioner atau tidak, jika belum maka lakukan proses stasionerisasi. Proses stasonerisasi pada data dilakukan dengan melakukan beda hingga orde pertama atau order kedua tergantung hasil dari fungsi auto-korelasinya. Tipe data dari *time series* ini adalah *array [0..n-1] of float*, dimana *n* adalah jumlah bulan.

Time series hasil di atas, adalah merupakan masukan bagi proses pembuatan *working cube*. Proses pembuatan *working cube* ini memerlukan informasi dari *user* (dari *task*) tentang bagamanan pembentukan *working cube* ini, apakah dalam periode tiga-bulanan, empat-bulanan, enam-tahunan, atau periode tahunan. Cara pembentukan *working cube* ini adalah dengan memetakan tiap-tiap data yang ada pada *time series* sesuai dengan kategori, urutan bulan, dan urutan periodenya. Misalkan akan dibuat *working cube* dengan periode 3-bulanan, dan terdapat empat konsep kategori, maka hasil dari *working cube* ini adalah seperti pada Gambar 5.7.

Data pada *time series* adalah urut menurut bulannya, jadi data ke-0 adalah data pada bulan Januari tahun pertama, data ke-0 adalah data pada bulan Februari tahun pertama, dan seterusnya. Jadi, untuk memetakan data ke dalam *wirking cube* ini, yang menentukan adalah indeks dari *time series*, indeks kategori dimana data ini berada, dan indeks periodenya. Contoh, misalkan data pada bulan Januari tahun pertama adalah 75, maka ia akan dipetakan ke sel (0,0,0) (semua indeks mulai dari nol) yang mana sumbu X adalah mewakili indeks bulan, sumbu Y mewakili indeks periode, dan sumbu Z mewakili indeks kategori. X = 0 artinya bahwa data adalah data pada bulan Januari, Y = 0 artinya adalah data terletak pada

periode tiga-bulanan pertama, dan Z = 0 artinya adalah data berada pada indeks kategori pertama yaitu antara 0 ~ 100.



Gambar 5.7 Working Cube

Proses pencarian pola periodik dilakukan pada *working cube* yang telah dibuat. Di sini, *user* memilih, dengan metoda apa pola periodik ingin dicari, selain itu dimasukkan pula nilai *thershold* untuk metoda yang akan dilakukan. Metoda pencariannya sendiri telah dijelaskan secara rinci pada bab IV.

Hasil dari pencarian pola-pola periodik ini adalah informasi-informasi tentang keperiodikan dari data yang sedang dianalisa. Hasil ini perlu diformat sehingga sebelum diberikan kepada *user*, sehingga menjadi mudah untuk dibaca.

Untuk menganalisa data-data lainnya, proses-proses yang telah dijelaskan di atas ini dapat dilakukan kembali, atau bila diiginkan untuk menambah dimensi lokasi yang lain, maka dapat dilakukan pembentukan *data cube* yang baru.

5.2 Pembuatan Perangkat Lunak

Setelah dilakukan perancangan terhadap perangkat lunak, maka tahap selanjutnya adalah mengimplemetasikan rancangan yang telah dibuat di atas menjadi sebuah perangkat lunak dengan spesifikasi seperti telah ditentukan dalam perancangan. Perangkat lunak dibuat dengan menggunakan Borland C++ Builder v3.0 dan ODBC 3.5 dari Microsoft untuk manajemen basis-datanya.

Secara umum dengan mengacu pada rancangan yang telah dibuat, maka dibuat modul-modul program yang akan dijelaskan sebagai berikut :

5.2.1 Modul Masukan

Modul ini meliputi *user interface* untuk pemilihan basis data, pembuatan *data cube*, pembuatan *slice* dari *cube*, dan pembuatan *time series*. Sub-modul pemilihan basis data akan menampilkan basis-data dari barbagai cabang retail yang ada dan kemudian setelah dilakukan pemilihan akan membuat *data cube* yang meliputi empat dimensi yang pada sistem retail, yaitu dimensi kategori, dimensi produk, dimensi lokasi, dan dimensi waktu.

Sub-modul pembuatan *slice* akan menampilkan apa saja "isi" dari keempat dimensi yang ada dalam *data cube*, yaitu bila dimensi lokasi sedang aktif, maka akan tampak lokasi mana saja yang terdapat dalam *data cube*. Hal ini juga berlaku untuk ketiga dimensi yang lain.

Sub-modul pembuatan *time series* akan mengambil masukan dari *slice* yang ada dan kemudian meminta *user* untuk memasukkan data apa yang akan

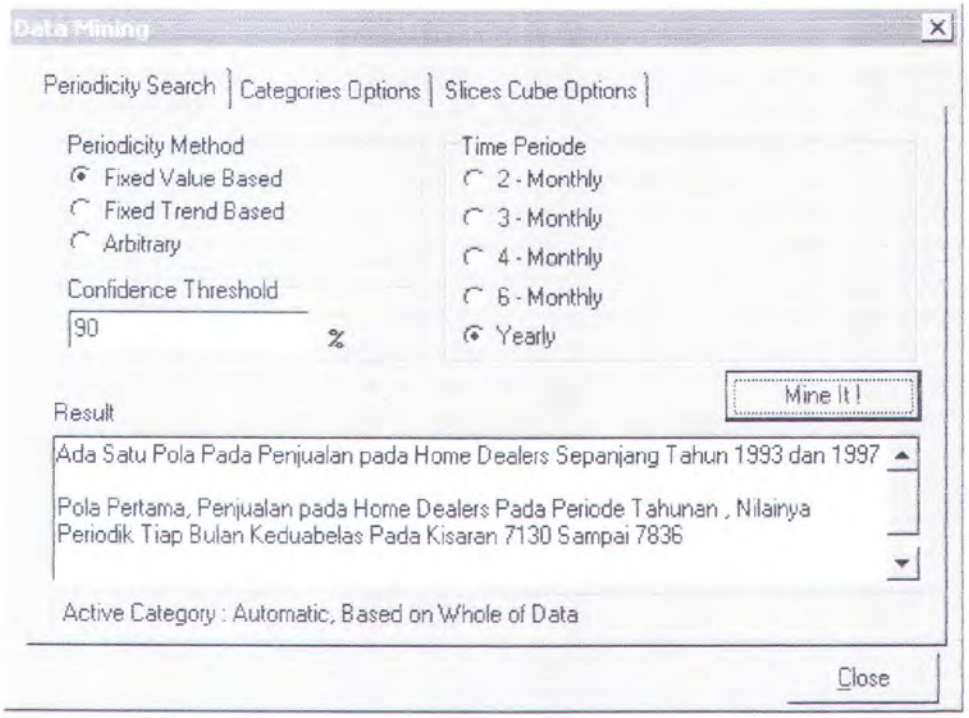
dijadikan dalam *time series*. Setelah *time series* dibentuk, *user* dapat menganalisa, apakah *time series* tersebut stasioner atau tidak untuk dilakukan proses berikutnya.

5.2.2 Modul Proses

Modul ini adalah modul untuk menangani proses pencarian pola periodik dengan ketiga metoda yang telah dijelaskan di bab III. Dalam modul ini, *user* dapat memilih perioda apa yang diinginkannya untuk membentuk *working cube*. Juga dapat memilih *slice-slice* mana dari *working cube* yang akan diikutkan dalam pencarian. Selain itu, dimungkinkan dilakukan operasi OLAP yaitu *drill-down* dan *roll-up* pada *working cube* untuk menurunkan atau menaikkan level agregasi dari *working cube*.

Fasilitas lain yang ada dalam modul ini adalah fasilitas untuk mengubah konsep kategori dari data yang sedang dianalisa. Seperti telah dijelaskan, pola-pola periodik yang dihasilkan adalah pola-pola periodik yang berada dalam suatu konsep kategori tertentu. Dengan adanya fasilitas ini, *user* dapat mengubah konsep kategori dari data sesuai dengan data atau kebijakan yang ada.

Terdapat dua macam cara untuk mengubah konsep kategori pada modul ini, pertama yaitu *customize*. Dengan dengan cara *customize* ini, *user* dapat mengkategorikan data menurut apa yang diiginkannya, baik jumlah kategori maupun *range* kategori dalam tiap kelas kategori. Yang kedua adalah *semi automatic*, yaitu *user* cukup memberikan *input* berapa jumlah kategori yang diinginkannya dan program akan secara otomatis membangkitkan *range* dari tiap-tiap kelas data dengan menggunakan algoritma *k-means clustering*.



Gambar 5.8 Modul Proses

Segera setelah *working cube* dibentuk, *user* dapat memilih jenis metoda keperiodikan yang ada dan memulai proses pencarian pola periodik. Hasil dari pencarian ini adalah informasi-informasi keperiodikan dari data yang sedang dianalisa.

5.2.3 Modul Keluaran

Modul keluaran ini untuk menginformasikan pola-pola periodik apa saja yang dapat ditemukan oleh metoda-metoda pencarian yang telah dipilih dan dalam rancangan perangkat lunak yang telah dibuat ini diintegrasikan menjadi satu dengan model proses. Terlihat pada Gambar 5.8 bahwa keluaran (informasi keperiodikan) dari data langsung tampak (bila ada) dalam satu modul dengan modul proses.

BAB VI

IMPLEMENTASI DAN EVALUASI PERANGKAT LUNAK

BAB VI

Implementasi dan Evaluasi Perangkat Lunak

Pada bab ini, perangkat lunak akan dijalankan dan mengambil data dari data-data retail di beberapa kota di Amerika Serikat yang diambil dari sebuah situs biro sensus AS, yaitu www:\\census.gov\svsd\www\advtable.html. Data-data ini adalah telah data berupa data bulanan selama tahun 1993 sampai 1997 mempunyai dimensi sebagai berikut :

- Lokasi
- Kategori
- Sub Kategori
- Waktu

Untuk mensimulasikan sebuah *data warehouse* tentang sistem retail ini, maka dibuat beberapa basis-data, yang mana tiap basis-data ini mewakili satu kota yang ada di Amerika Serikat tersebut. Kemudian basis-data basis-data ini dikumpulkan dalam sebuah *warehouse* untuk dibuat *data cube* untuk melayani semua analisa yang menjadi topik dalam penyusunan tugas akhir ini, yaiu pencarian pola periodik pada data beratribut waktu.

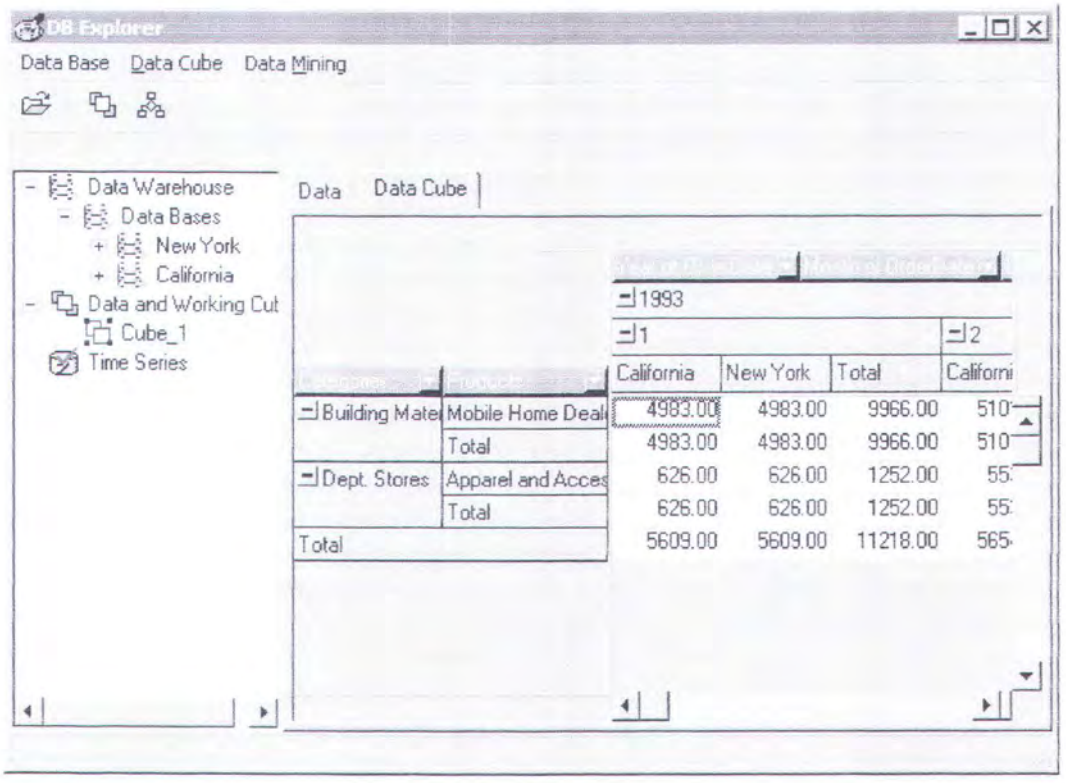
6.1 Pengujian Perangkat Lunak

Perangkat lunak dimulai dengan melakukan pemilihan terhadap basis-data mana yang akan dianalisa. Sesuai dengan konsep di atas, ini artinya adalah dari

kota mana saja data-data penjualan akan dianalisa. Setelah itu baru dibentuk *data cube*-nya. Dalam bab ini, akan dicontohkan basis-data dari dua kota, yaitu New York dan California. Dalam basis-data ini terdapat dua kategori, yaitu

- *Building Material*
- *Dept. Stores*

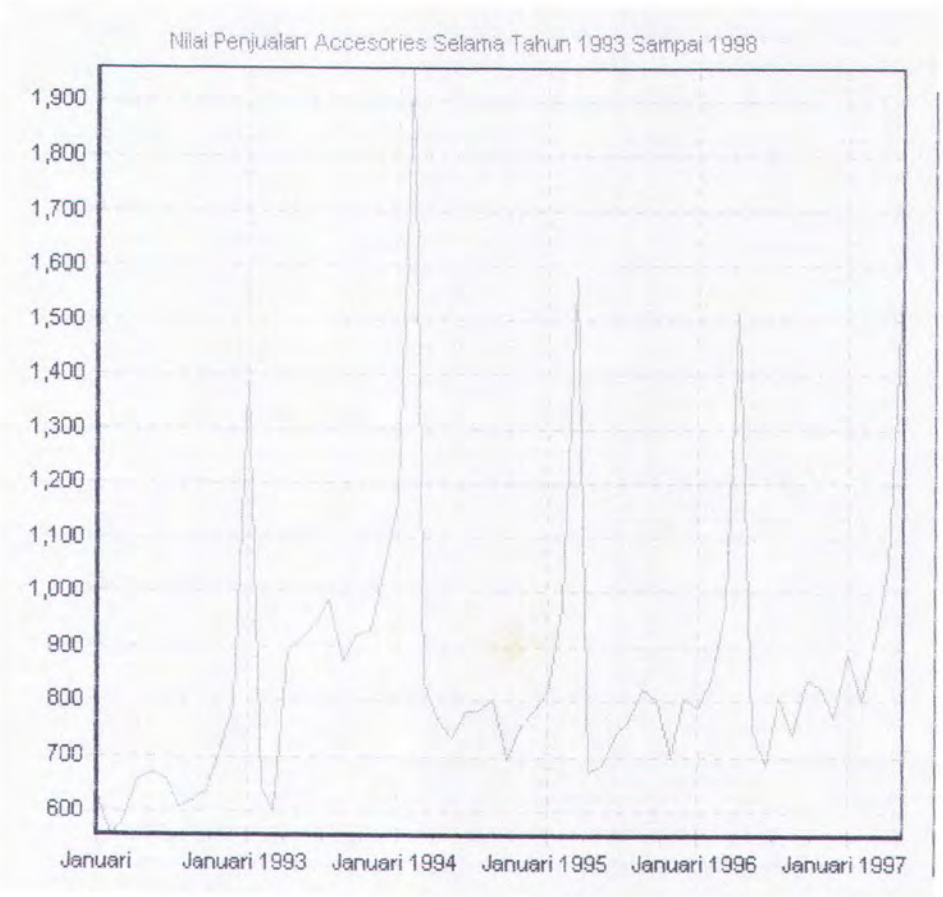
Pada kategori *building material* terdapat sub kategori *Mobile Home Dealer* sedangkan pada kategori *Dept. Stores* terdapat sub kategori *Women Accessories*. Setelah kedua basis-data ini dibuka dan dimensi-dimensinya dipilih, maka terbentuk *data cube* seperti berikut:



Gambar 6-1 Data Cube hasil dari Pembentukan beberapa Database

Seperti tampak pada gambar di atas, maka *data cube* disajikan dalam bentuk tabel multi dimensi sehingga akan memudahkan untuk melihat data-data, misalnya data penjualan *Women Accessories* selama tahun 1993 di California dan sebagainya.

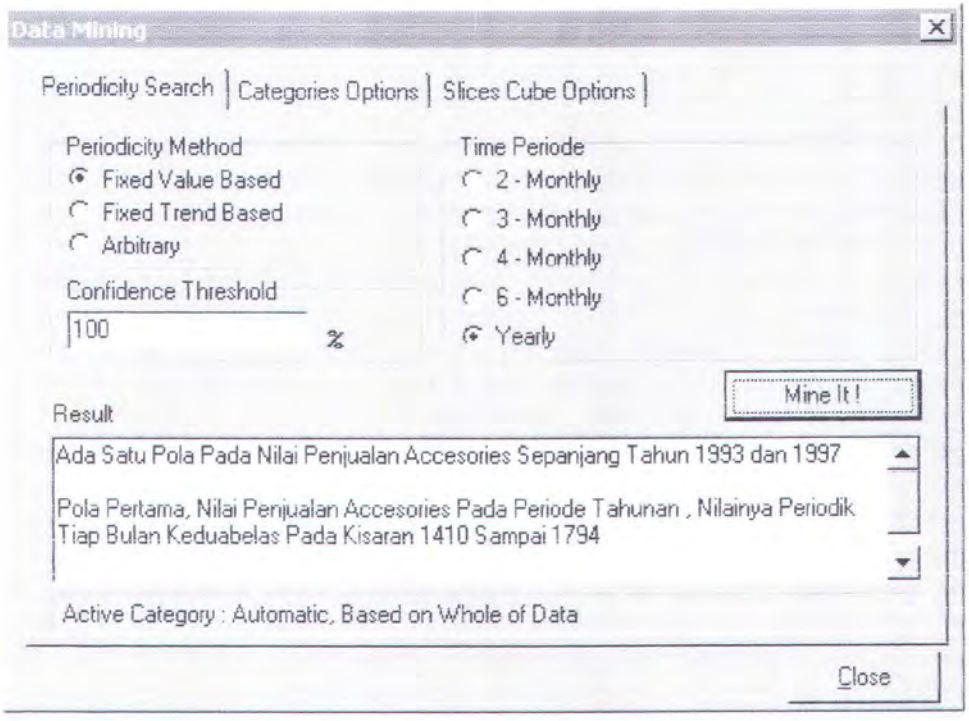
Selanjutnya diinginkan untuk melakukan analisa keperiodikan terhadap data penjualan *Women Accessories* di California selama tahun 1993 sampai 1997, maka *cube* "dipotong" untuk menghasilkan data dengan kriteria lokasi = "California" dan produk = "*Women Accessories*". Lalu dibuat *time series dari* dari data-data itu sehingga menghasilkan data bulanan nilai penjualan *Women Accessories* selama tahun 1993 sampai 1997 sebagai berikut :



Gambar 6-2 Time Series untuk Penjualan Women Accesories

Dari *time series* diatas, hendak dilakukan analisa, apakah data di atas mempunyai pola keperiodikan, jika ya, pada nilai berapa. Dari analisa awal di atas terlihat bahwa nilai penjualan tiap bulan Desember selalu mengalami kenaikan. Dari sini dapat disimpulkan bahwa data di atas mempunyai pola musiman, tetapi apakah nilai dari tiap-tiap bulan yang terdapat dalam pola musiman adalah periodik adalah perlu diselidiki lebih lanjut.

Metoda yang dipilih adalah *fixed value-based search* dengan nilai *threshold* 100% dan dalam periode tahunan, artinya adalah frekuensi keberadaan pola harus sama dengan panjang pola.



Gambar 6-3 Hasil pada Fixed Value Based threshold 100 %

Ternyata, dalam periode tahunan, nilai penjualan memang periodik pada nilai 1410 sampai 1794 (dalam ribu $) tiap bulan keduabelas selama tahun 1993 sampai

1997. Karena nilai *threhold*-nya 100%, maka dapat dipastikan bahwa pola ini adalah benar-benar pola (*exactly patterns*).

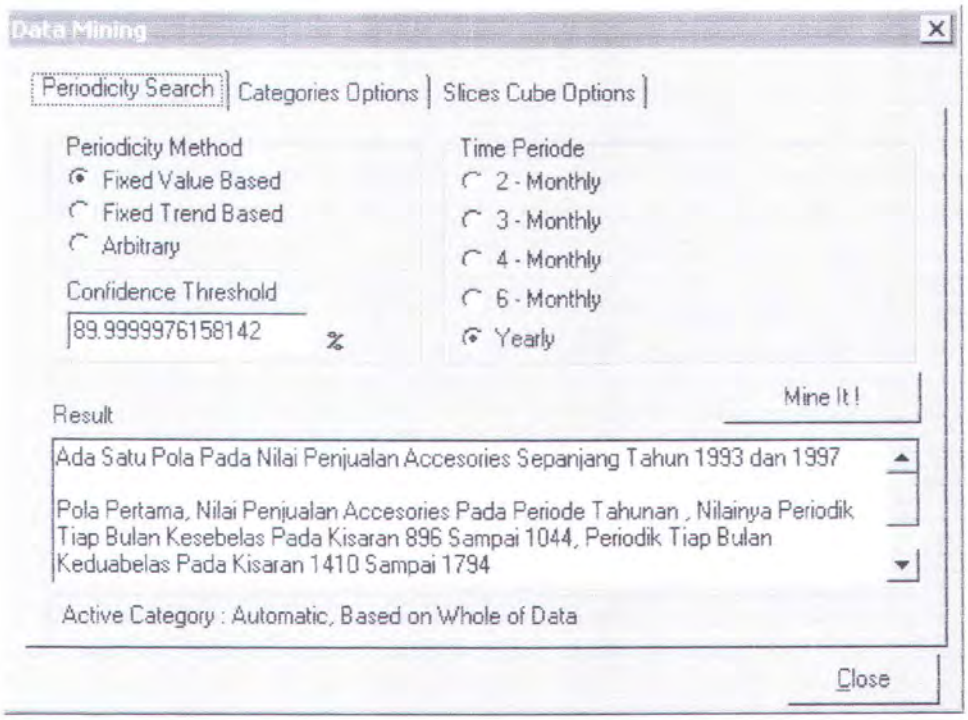
Nilai 1410 sampai 1794 dihasilkan dari fasilitas konsep kategori yang disediakan oleh perangkat lunak. Terdapat dua pilihan pada perangkat lunak, yaitu *custome* dan semi-otomatik. *Custom* artinya *user* dapat memasukkan konsep kategori sesuai dengan pilihannya, baik dalam jumlah kategori maupun nilai kategorinya, sedangkan semi otomatik artinya bahwa *user* hanya memsaukkan jumlah kategori yang diinginkan dan perangkat lunak akan secara otomatis membuat kategori sesuai dengan data yang ada dengan teknik *K-Means Clustering*. Berikut adalah konsep kategori yang dihasilkan dari *K-Means Clustering* dengan jumlah *clusters* adalah 5

Table 6-1 Konsep Kategori untuk Penjualan Women Accesories

Kategori	Nilai
1	579 ~ 633
2	667 ~ 725
3	771 ~ 831
4	896 ~ 1044
5	1410 ~ 1794

Misalkan kita ingin melihat apakah mungkin ada bulan-bulan lain yang juga mirip atau tidak 100% sama, maka kita dapat menurunkan nilai *threshold*-nya, 90% misalkan. Tentu saja hasil dari analisa ini nantinya adalah analisa kemiripan pola periodik, karena dengan nilai *threshold* yang semakin kecil, maka semakin kecil pula nilai minimal frekuensi dari pola yang harus dipenuhi dari data dan ini berarti tidak semua data akan menempati pola hasil analisa ini nantinya.

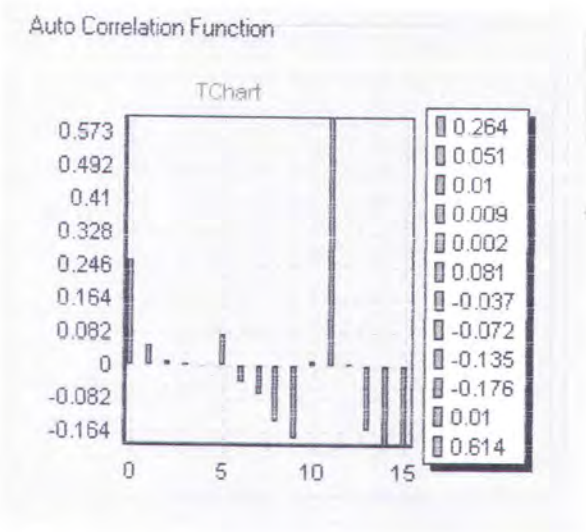
Sekali lagi progran akan dijalankan dengan mengambil periode tahunan untuk membandingakn dengan hasil yang telah diperoleh dari analsia seelumnya yang juga periodenya tahunan tetapi nilai *threshold*nya 100%.



Gambar 6-4 Fixed Value Based dengan Threhold 90%

Hasil dari analisa menunjukkan bahwa dengan *threshold* 90%, ternyata nilai penjualan pada bulan kesebelas juga mengalami keperiodikan pada nilai 896 sampai 1044 disamping pola tiap bulan keduabelas yang antara nilai 1410 sampai 1794. Hasil dari analisa adalah bukan pola sebenarnya, karena nilai *threshold* adalah 90% yang artinya adalah dalam jangka waktu antara 1993 sampai 1997, pola nilai penjualan adalah "mirip" pada keperiodikan di bulan kesebelas dan keduabelas.

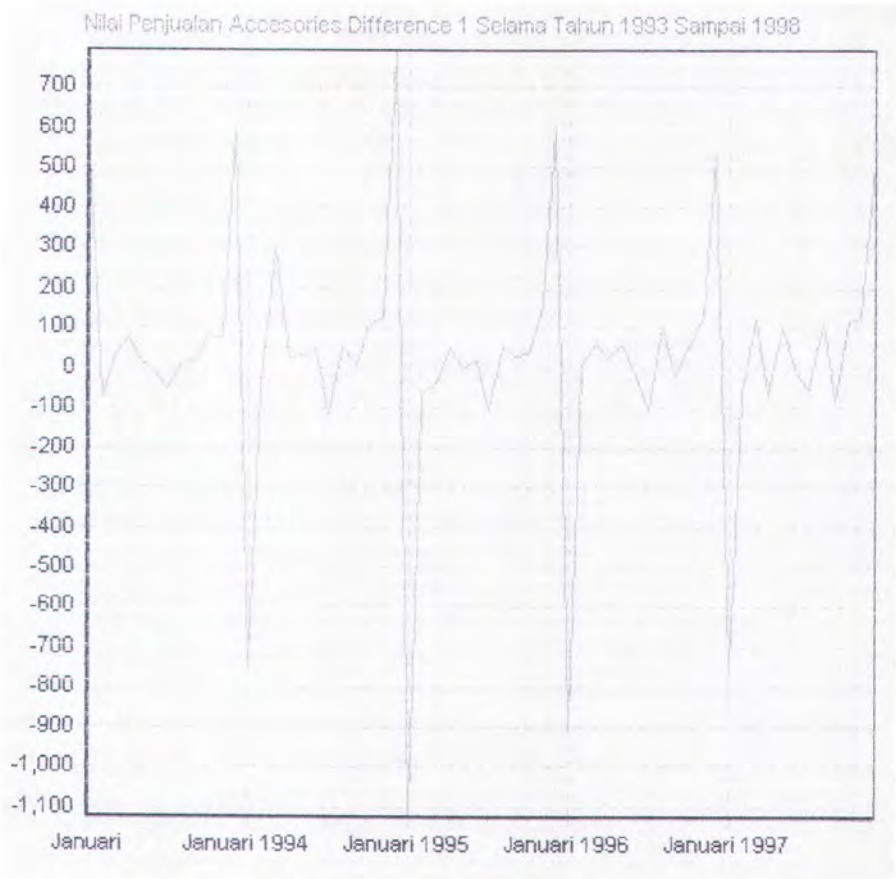
Selanjutnya mungkin diinginkan untuk menganalisa data pada data yang stasioner, untuk itu dilakukan uji stationeritas pada data penjualan *Women Accessories* di atas.



Gambar 6-5 Fungsi Autokorelasi Dari Data Asli

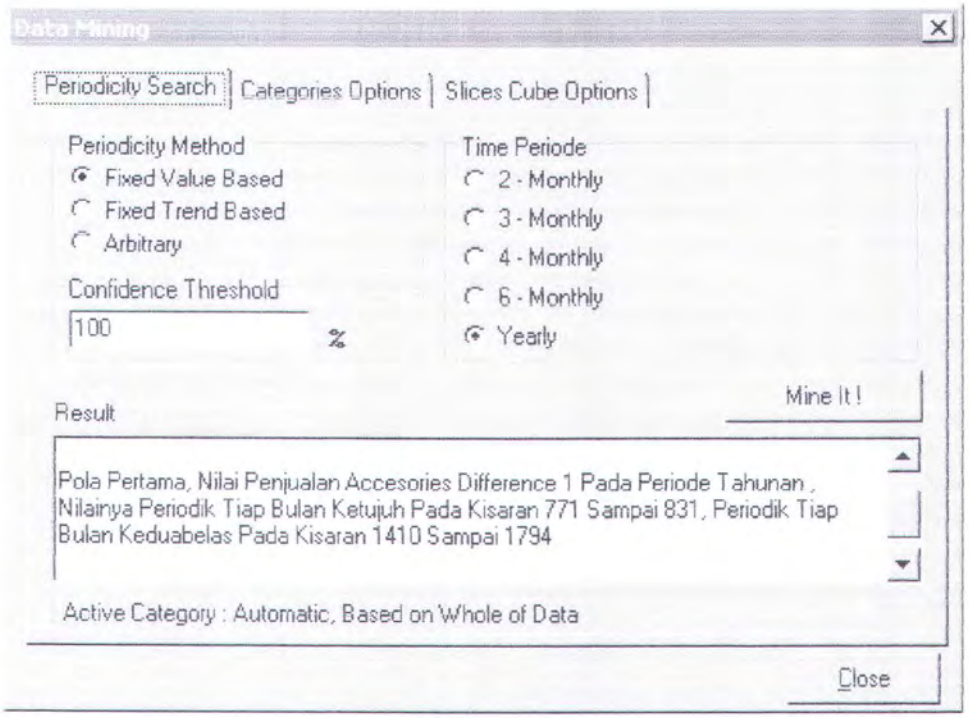
Dari grafik fungsi di atas, terlihat bahwa nilai Autokorelasi menurun dengan lambat ke nilai nol yang menandakan bahwa data asli adalah tidak stasioener. Untuk itu perlu dilakukan proses stationerisasi untuk data di atas. Proses staionerisasi dilakukan dengan melakukan beda hingga orde pertama pada data, setelah itu dianalisa kembali.

Perangkat lunak akan secara otomatis menyesuaikan antara kategori dari data asli dengan kategori dari data hasil dari proses stasionerisasi, karena data setelah mengalami proses stasionerisasi akan berubah dan tentu saja konsep kategori dari data juga akan berubah.



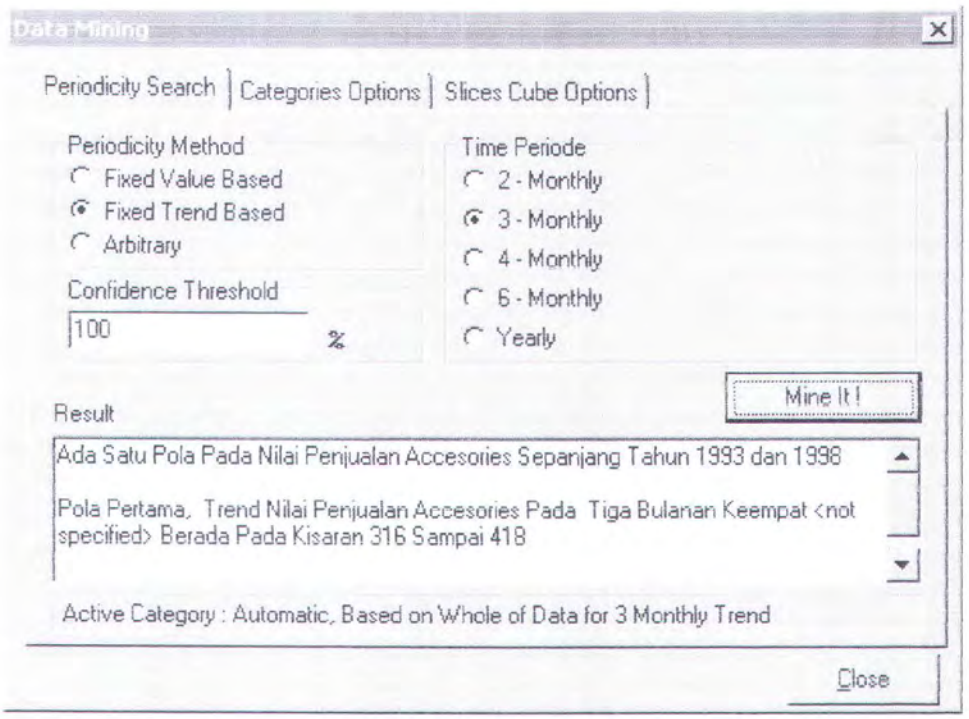
Gambar 6-6 Time Series setelah dilakukan proses Stasionerisasi

Selanjutnya dilakukan kembali analisa keperiodikan dengan *fixed value-based* dan nilai *threhold* 100%. Ternyata, hasilnya adalah sama dengan analisa awal sebelum data terstasionerisasi pada data 100% dan bahkan ada pola lain yang muncul, yaitu pada bulan ketujuh, yaitu antara nilai 771 sampai 831. Dengan nilai *threshold* yang 100%, maka dapat dikatakan bahwa memang itulah pola yang terdapat dalam data walaupun data telah mengalami proses stasionerisasi yang artinya merubah nilai data. Untuk hal demikian, perlu dilakukan pengkajian lebih lanjut bagi pengambil keputusan apakah pola ini bisa diterima.



Gambar 6-7 Hasil Fixed Value-based pada Data Terstasionerisasi

Selanjutnya, mungkin juga diinginkan menganalisa dari sisi trend-nya, maka dapat dilakukan metoda *fixed trend-based* pada data.



Gambar 6-8 Trend Based pada Contoh 1

Masih sama seperti pada contoh-contoh sebelumnya, maka dilakukan metoda *Fixed Trend Based* pada data penjualan *Women Accessories* selama tahun 1993 sampai 1997. Dengan mengambil periode tiga bulanan, maka didapatkan hasil bahwa trend penjualan setiap tiga-bulanan keempat tiap tahunnya adalah berada pada kisaran 316 sampai 418. Tiga-bulanan keempat artinya adalah periode antara bulan Oktober sampai Desember tiap tahunnya.

Pada hasil di atas, nilai 316~418 bisa diberi label untuk menandakan apakah trend saat itu naik, turun, atau stabil. Tentu saja pemberian label ini subyektif dan berdasarkan pengalaman atau kebijakan perusahaan.

Pengujian pada Konsep Kategori yang Berbeda

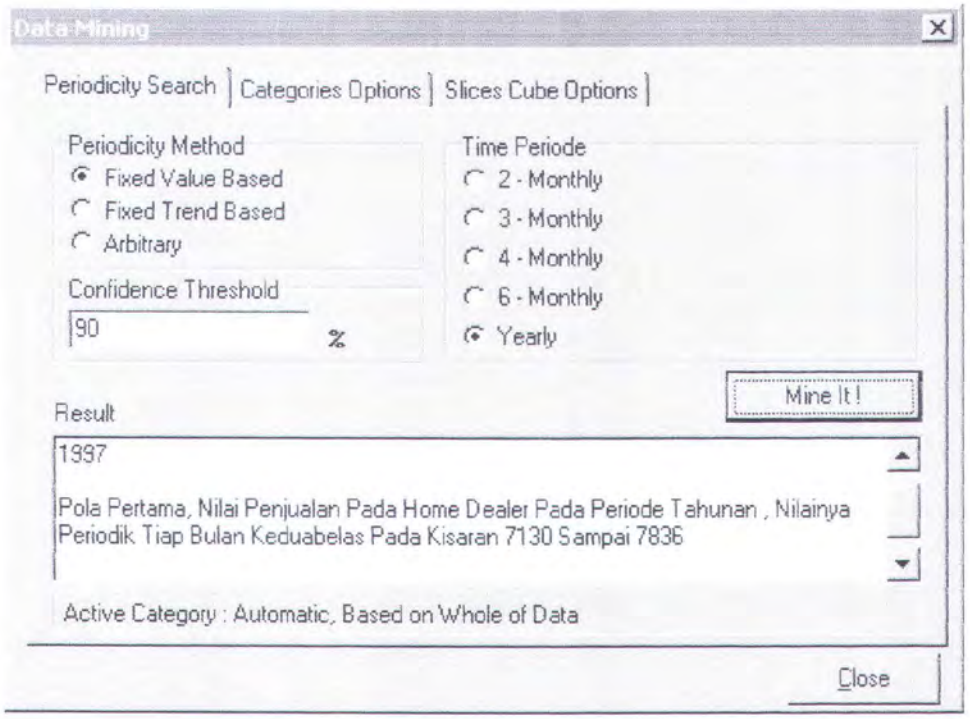
Berikut ini akan disajikan beberapa hasil dari pencarian pola-pola peridodik yang menggunakan contoh data lain, yaitu nilai penjualan pada *Home Dealers* (contoh 2) dengan dilakukan perubahan pada nilai-nilai konsep kategori. Awalnya, jumlah kategori di-*set* sama dengan lima yang ditunjukkan oleh Table 6.2

Table 6-2 Konsep Kategori pada contoh 2 dengan jumlah Kategori = 5

Kategori	Nilai
1	4894 ~ 5100
2	5829 ~ 6455
3	7130 ~ 7836
4	8471 ~ 9163
5	9987 ~ 10955

Lalu diadakan analisa keperiodikan dengan metoda *fixed value based* dengan tingkat kepercayaan 90% dan menghasilkan suatu pola bahwa tiap dua

belas bulan sekali, nilai penjualan pada *home dealers* periodik pada nilai antara 7130 ~ 7836 seperti ditunjukkan pada Gambar 6.9.



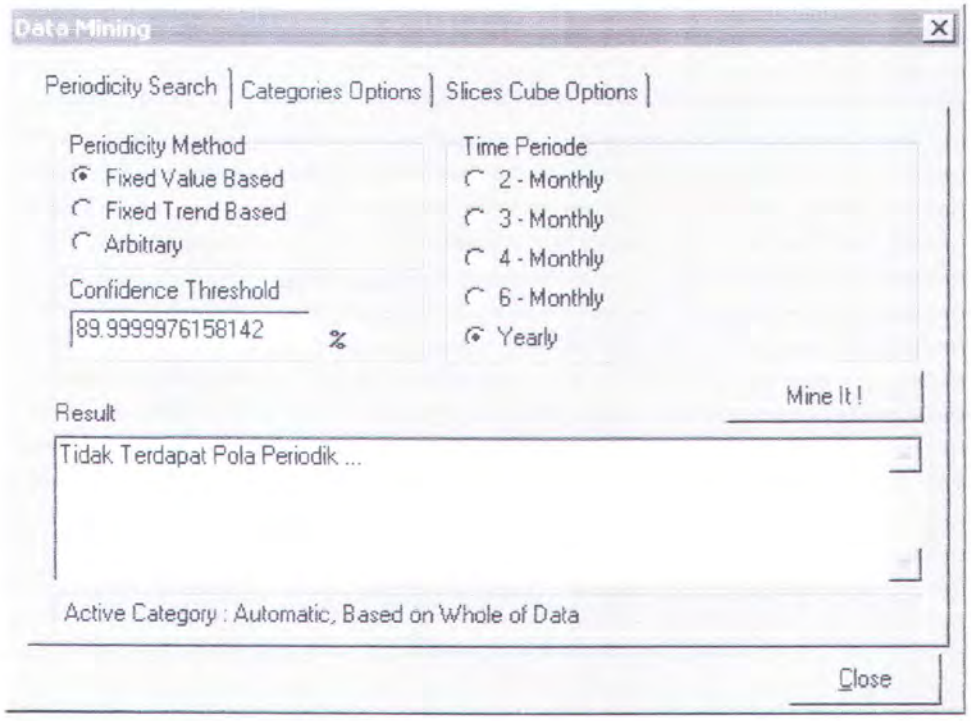
Gambar 6-9 Fixed Value Based pada Contoh 2 dengan Jumlah Kategori = 5

Kemudian, jumlah kategori dirubah menjadi enam, masih sama dengan menggunakan teknik *K Means Clustering* dan menghasilkan konsep kategori sebagai berikut :

Table 6-3 Konsep Kategoripada Contoh dengan Jumlah kategori - 6

Kategori	Nilai
1	4894 ~ 5100
2	5659 ~ 5675
3	6268 ~ 6818
4	7325 ~ 7855
5	8471 ~ 9163
6	9987 ~ 10955

Dengan tingkat kepercayaan yang sama yaitu 90%, ternyata dengan jumlah kategori berubah menjadi enam, tidak ditemukan adanya pola-pola periodik pada data seperti ditunjukkan pada Gambar 6.10.



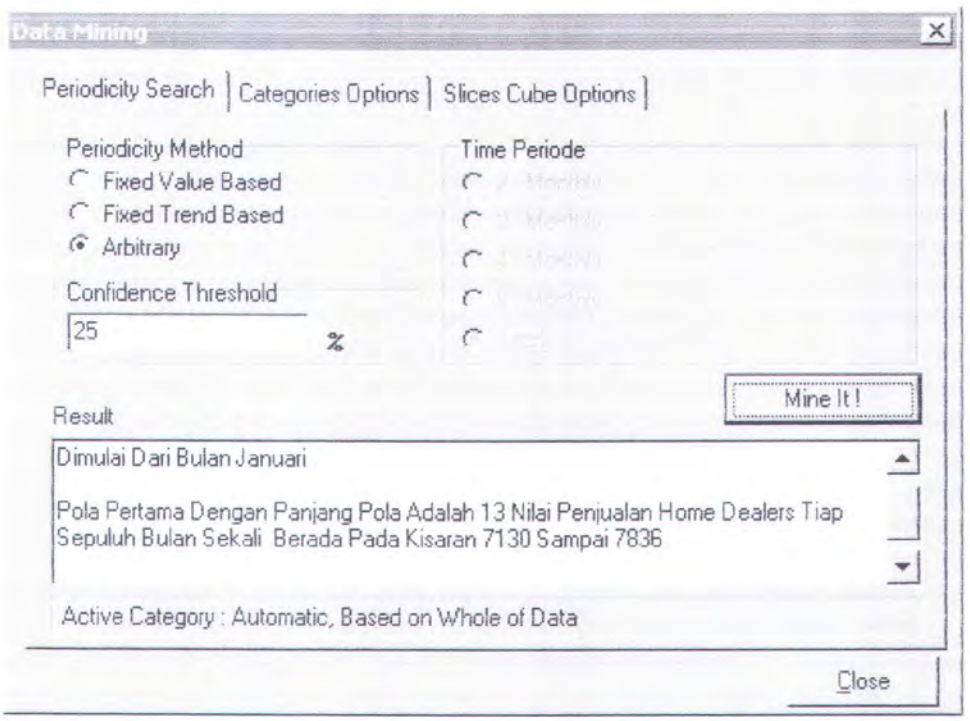
Gambar 6-10 Fixed Value Based pada Contoh 2 dengan jumlah kategori 6

Hasil ini menunjukan bahwa pemilihan konsep kategori sangat menentukan dalam pembentukan pola, dengan mempertimbangkan sebagai berikut :

- Jumlah konsep kategori yang sedikit akan menyebabkan kekurang-presisian kategori karena akan menyebabkab data-data yang seharusnya tidak masuk kategori ke-*i* itu menjadi masuk kategori ke-*i* karena jangkauannya besar
- Jumlah kategori yang banyak juga tidak menguntungkan karena menyebabkan jangkauan kategori menjadi kecil sehingga data terpecah-pecah menjadi kelompok-kelompok yang lebih banyak, ada kemungkinan tidak dapat

menemukan pola keperiodikan karena metoda pencarian pola keperiodikan yang dikerjakan di sini sensitif terhadap konsep kategori.

Arbitrary Search



Gambar 6-11 Arbitrary Search pada Contoh 2

Pengujian selanjutnya adalah untuk menguji algoritma *arbitrary search* dengan contoh kasus yang sama, yaitu untuk data nilai penjualan *home dealers* selama lima tahun. Seperti telah dijelaskan, nilai *threshold* pada algortima *arbitrary search* adalah untuk menentukan panjang pola maksimal yang akan dicari dari data. Pada gambar di atas, pengujian dilakukan dengan mengambil nilai *threshold* 25%, yang artinya adalah panjang pola maksimal yang akan dicari dari data adalah 25% * 60 = 15.

Setelah dilakukan pengujian, ternyata ditemukan sebuah pola, yang mempunyai panjang pola 13, artinya tiap tigas belas bulan sekali dimulai dari bulan Januari nilai penjualan *home dealer* periodik pada nilai 7130 sampai 7386. Pola ini menunjukkan bahwa selain pola-pola yang telah ditemukan sebelumnya dengan menggunakan algoritma-algoritma sebelumnya yang mana pola ditemukan dalam suatu periode-periode yang merupakan periode natural dari waktu (seperti tahunan, tiga bulanan, dan sebagainya.), terdapat juga pola dalam suatu periode yang bukan merupakan periode natural yang, dalam contoh ini adalah tiap tiga belas bulanan.

BAB VII

KESIMPULAN

BAB VII

KESIMPULAN

Dari hasil implementasi dan evaluasi dari perangkat lunak ini, maka berikut adalah beberapa kesimpulan yang dapat dihasilkan

- Metoda *fixed value-based* adalah metoda efisien dan fleksibel dalam mencari pola keperiodikan dari data. Efisien berarti bahwa metoda yang digunakan dapat mencari pola keperiodikan dari data dengan relatif cepat dan akurat karena metoda *cube* untuk mekanisme penyimpanan datanya. Fleksibel berarti bahwa metoda ini mempunyai kemampuan untuk mencari, baik pola yang lengkap maupun pola yang pola yang parsial dan mirip, hal ini dapat dilakukan dengan mengatur nilai *threshold* untuk metoda pencariannya
- Metoda *fixed value-based* cocok untuk data yang mempunyai sifat pola musiman
- Bila data cenderung mempunyai trend yang kuat, maka dapat digunakan metoda *fixed trend based*
- Proses kategorisasi data adalah penting karena pemilihan jumlah kategori yang terlalu sedikit (rentang kelas menjadi lebar) akan mengakibatkan banyak data jatuh ke dalam kategori yang sebenarnya bukan merupakan ketagorinya, sebaliknya bila jumlah kategori terlalu banyak (rentang kelas menjadi sempit) akan mengakibatkan sebaran kategori data menjadi tidak seimbang.

Sebagai penutup dari penyusunan tugas akhir ini, diberikan beberapa saran untuk semakin meningkatkan kinerja dan hasil dari pencarian pola periodik ini, yaitu sebagai berikut :

- Sebaiknya dilakukan analisa frekuensi dari data dengan mengubah *domain* data dari bidang waktu ke bidang frekuensi untuk mengetahui frekuensi dari data. Hal ini berguna untuk menentukan periode awal dari data
- Data yang terkumpul dalam *data warehouse* terkadang tidak lengkap atau ada beberapa data yang hilang, maka sebaiknya dilakukan pemeriksaan atau analisa terhadap data sebelum masuk pada proses *mining* untuk mengembalikan data-data yang hilang ini. Beberapa teknik *signal processing* dapat diaplikasikan untuk mengatasi masalah ini.

DAFTAR PUSTAKA

Daftar Pustaka

1. Agrawal, Rakesh and Srikant, R. "Fast Algoritm for Mining Association Rules". *In Proc 21. Int. Conf. Very Large Databases, Page 487-499,* Santiago, Chile, September 1991
2. Agrawal, Rakesh and Srikant, R. "Mining Sequential Patterns". *In Proc. 1995 Int. Conf. Data Engineering*, pages 3-14, Taipei, 1995
3. Gong, Wan. "Periodic Patterns Search on Time-Related Data Sets". *Paper on Simon Fraser University*, 1995
4. Harinarayan, V. Rajaman, A. and Ullman, J.D. "Implementing Data Cube Efficiently", *In Proc. ACM-SIGMOD Int. Conf. Management of Data.* Pages 205-216, Montreal, Canada, June, 1996
5. Makridakis, Spyros., Wheelright, Steven C., and McGee, Victor E. "*Forecasting: Methods and Applications*". John Wiley and Sons, 1983
6. Pilot Software. "An Introduction to Data Mining, Discovering Hidden Value in Your Data Warehouse". *White Paper*
7. Pressman, Roger S. "*Software Engineering, A Practioners Approach*". McGraw Hill, 1992
8. Singh, Harry. "*Data Warehousing Concepts, Technologies, Implementations, and Management*". Prentice Hall, 1998
9. www:\\olapcouncil.org