

BAB 2

LANDASAN TEORI

2.1 Teori – Teori Dasar / Umum

2.1.1 Definisi *Database*

Menurut Connolly (2005, p15), *database* adalah kumpulan data yang berelasi secara logikal, dan sebuah deskripsi data tersebut, yang didisain untuk memenuhi kebutuhan organisasi.

Menurut O'Brien (2004, p73), *database* adalah kumpulan elemen data yang terintegrasi yang berhubungan secara logikal.

Menurut Turban et al (2005, p446), *database* adalah kumpulan dari file-file yang menyimpan data yang saling berelasi dan berasosiasi satu dengan yang lainnya.

Berdasarkan definisi tersebut diatas, dapat disimpulkan bahwa *database* adalah kumpulan dari elemen, relasi atau *file-file* yang saling terintegrasi untuk digunakan bersama dan dapat digunakan untuk memperoleh informasi yang dibutuhkan oleh perusahaan.

2.1.2 *Data warehouse*

2.1.2.1 Definisi *Data warehouse*

Menurut Connolly dan Begg (2002, p1047), *data warehouse* adalah suatu kumpulan data yang bersifat *subject-oriented*, *integrated*, *time-variant*, dan *non-volatile* dalam mendukung proses pengambilan keputusan.

Menurut Inmon (2005, p29), *data warehouse* adalah koleksi data yang berorientasi subjek, terintegrasi, tidak mengalami perubahan, dan berdasarkan variasi waktu untuk mendukung keputusan manajemen.

Dari definisi – definisi diatas tersebut, adapun tujuan yang ingin dicapai *data warehouse* yaitu :

1. *Data warehouse* menyediakan suatu pandangan (*view*) umum, sehingga *data warehouse* akan memiliki keleluasaan untuk mengakomodasi bagaimana data akan ditafsirkan atau dianalisis selanjutnya.
2. *Data warehouse* merupakan tempat penyimpanan seluruh data historis. *Data warehouse* akan bertumbuh menjadi sangat besar sehingga harus dirancang untuk mengakomodasi pertumbuhan data.
3. *Data warehouse* dirancang untuk menyediakan data bagi berbagai teknologi analisis dalam komunitas bisnis.

2.1.2.2 Karakteristik *Data warehouse*

Menurut Inmon (2002, p31), *data warehouse* bersifat sebagai berikut:

- *Subject Oriented*

Data warehouse dirancang untuk melakukan analisa data berdasarkan subjek-subjek tertentu yang ada dalam organisasi, tidak berorientasi kepada proses atau aplikasi fungsional tertentu.

- *Integrated*

Karakteristik kedua dan terpenting dari *data warehouse* adalah integrasi. Data diambil dari banyak sumber terpisah ke dalam *data warehouse*. Data yang

diambil itu akan diubah, diformat, disusun kembali, diringkas, dan seterusnya. Data yang masuk ke dalam *data warehouse* dengan berbagai cara dan mempunyai ketidakkonsistenan pada tingkat aplikasi tidak akan dimasukkan. Contoh konsistensi data antara lain adalah penamaan, struktur kunci, ukuran atribut, dan karakteristik data secara fisik. Hasilnya adalah data dalam *data warehouse* yang mempunyai satu bentuk.

- *Non-Volatile*

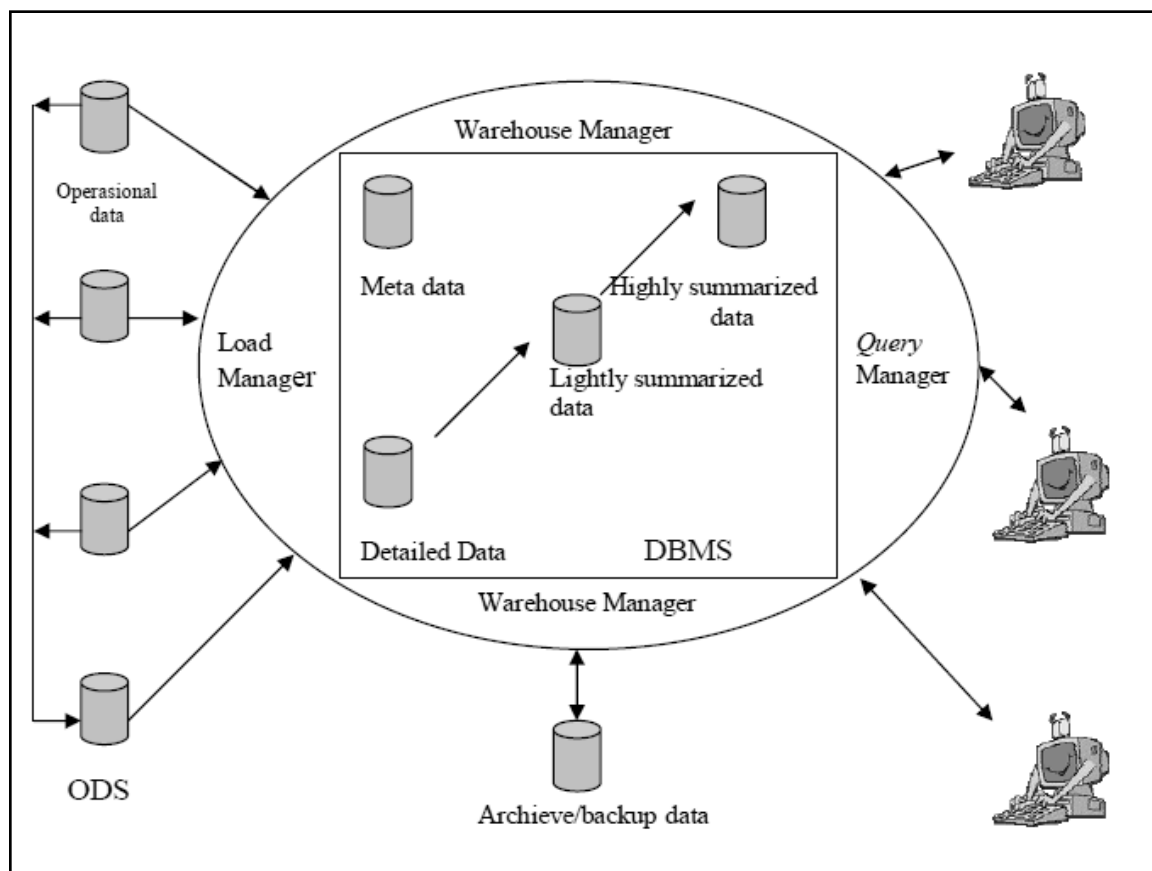
Non-volatile dapat diartikan bahwa data tersebut tidak mengalami perubahan. Data di lingkungan operasional dapat dilakukan perubahan (*update*), dihapus (*delete*), dan dimasukkan data baru (*insert*) seperti, tetapi data dalam *data warehouse* hanya melakukan *loading* dan *accessing*. Dengan ini maka data yang lama tetap tersimpan dalam *data warehouse*.

- *Time-Variant*

Karakteristik ini mengimplikasikan bahwa tiap data dalam *data warehouse* itu selalu akurat dalam periode tertentu. Dalam satu sisi, sebuah *record* dalam *database* memiliki waktu yang telah ditetapkan secara langsung. Di sisi lain, sebuah *record* mempunyai waktu transaksi.

Dalam setiap lingkungan baik operasional maupun *data warehouse*, lingkungan tersebut memiliki *time horizon*. *Time horizon* adalah sebuah parameter waktu yang dipertunjukkan dalam lingkungan tersebut. Batas waktu pada *data warehouse* lebih lama daripada sistem operasional. Karena perbedaan batas waktu tersebut, maka *data warehouse* mempunyai lebih banyak histori daripada lingkungan lainnya.

2.1.2.3 Arsitektur *Data warehouse*



Gambar 2.1 Arsitektur *Data warehouse*

Sumber : Connolly, 2005, p 1053

Arsitektur *data warehouse* menyediakan kerangka dengan mengidentifikasi dan memahami data akan pindah melalui sistem dan digunakan dalam perusahaan. Arsitektur *data warehouse* mempunyai komponen utama yaitu *database* yang hanya dapat dibaca.

Karakteristik arsitektur *data warehouse* :

- Data diambil dari sistem asal (sistem informasi yang ada), *database* dan *file*.
- Data dari sistem asal diintegrasikan dan ditransformasikan sebelum disimpan ke dalam *data warehouse*.
- *Data warehouse* adalah jenis *database read-only* yang diciptakan untuk mengambil keputusan.

- *User* mengakses *data warehouse* via *front-end tool* atau aplikasi.

Berdasarkan Gambar 2.1, komponen utama *data warehouse* menurut Connolly (2005,p1052) yaitu:

- *Operational Data*

Data operasional adalah data yang digunakan untuk mendukung proses bisnis sehari-harinya.

- *Operational Data Store*

Operational Data Store adalah tempat penyimpanan data operasional yang bersifat *current* dan terintegrasi yang digunakan untuk analisis. Atau dengan kata lain, *operational data store* mendukung proses transaksi operasional maupun proses analisis. Dengan adanya *operational data store*, maka pembangunan *data warehouse* menjadi lebih mudah karena *operational data store* dapat menyediakan data yang telah diekstrak dari sumber dan telah dibersihkan sehingga proses pengintegrasian dan restrukturisasi data untuk *data warehouse* menjadi lebih sederhana.

- *Load Manager*

Load manager, disebut juga dengan komponen *front-end*, melakukan semua operasi yang berhubungan dengan ekstraksi dan *load* data ke dalam *data warehouse*. Data dapat diekstrak langsung dari sumber data atau lebih biasa dari *operational data store*. Operasi yang dilakukan oleh *load manager* dapat meliputi transformasi sederhana dari data untuk mempersiapkan data untuk masukan ke dalam *data warehouse*. Ukuran dan kompleksitas dari komponen ini akan bervariasi di antara *data warehouse* dan dapat dibangun dengan

menggunakan kombinasi dari *tool loading data* dari vendor dan program yang *custom-built*.

- *Warehouse Manager*

Warehouse manager melakukan semua operasi yang berhubungan dengan manajemen dari data di dalam *data warehouse*. Komponen ini dibangun menggunakan *tool management data* vendor dan program *custom-built*.

Operasi yang dilakukan oleh *warehouse manager* meliputi:

- analisis data untuk memastikan konsistensi data
- transformasi dan menggabungkan sumber data dari tempat penyimpanan sementara ke dalam tabel *data warehouse*
- pembuatan indeks dan *view* berdasarkan tabel dasar
- melakukan denormalisasi (jika diperlukan)
- melakukan agregasi (jika diperlukan)
- melakukan backup data

Dalam beberapa kasus, *warehouse manager* juga menghasilkan *profile query* untuk menentukan indeks dan agregasi manakah yang sesuai. Sebuah *profile query* dapat dihasilkan untuk tiap *user*, sekelompok *user*, atau *data warehouse* dan didasarkan pada informasi yang menggambarkan karakteristik dari *query* seperti frekuensi, target tabel, ukuran set hasil.

- *Query Manager*

Query manager, disebut juga dengan komponen *back-end*, melakukan semua operasi yang berhubungan dengan manajemen dari *query user*. Komponen ini biasanya dibangun menggunakan *tool akses data end-user* vendor, *tool*

pengawasan *data warehouse*, fasilitas *database* dan program *custom-built*. Kompleksitas dari *query manager* ditentukan oleh fasilitas yang disediakan oleh *tool* akses *end-user* dan *database*. Operasi yang dilakukan oleh komponen ini meliputi mengarahkan *query* ke tabel yang sesuai dan menjadwalkan pengekseskuan *query*. Dalam beberapa kasus, *query manager* juga menghasilkan *profile query* untuk mengizinkan *warehouse manager* menentukan indeks dan agregasi mana yang tepat.

- *Metadata*

Area dari *data warehouse* ini menyimpan semua definisi *metadata* yang digunakan oleh semua proses dalam *data warehouse*. *Metadata* digunakan untuk berbagai tujuan yang meliputi:

- ekstraksi dan *load* proses – *metadata* digunakan untuk memetakan sumber data ke dalam *view* biasa dari data di dalam *warehouse*
- proses *management warehouse* – *metadata* digunakan untuk mengotomatisasi produksi dari tabel ringkasan
- sebagai bagian dari proses *management query* – *metadata* digunakan untuk mengarahkan langsung sebuah *query* ke sumber data yang paling tepat.

Struktur dari *metadata* berbeda di antara tiap proses, karena tujuannya berbeda. Hal ini berarti *copy* dari *metadata* yang menggambarkan *item* data yang sama disimpan dalam *data warehouse*. Sebagai tambahan, kebanyakan *tool* vendor untuk *management copy* dan akses data *end-user* menggunakan *metadata* versi mereka sendiri. Secara spesifik, *tool management copy* menggunakan *metadata* untuk mengerti aturan pemetaan untuk menerapkan dengan tujuan untuk

mengkonversi sumber data ke dalam bentuk yang umum. *Tool* akses *end-user* menggunakan *metadata* untuk mengerti bagaimana membangun sebuah *query*. Manajemen dari *metadata* di dalam *data warehouse* merupakan tugas sangat kompleks yang tidak boleh diremehkan.

- *End-User Access Tool*

Tujuan utama dari *data warehouse* adalah untuk menyediakan informasi kepada *user* bisnis untuk membuat keputusan strategis. *User* ini berinteraksi dengan *data warehouse* menggunakan *tool* akses *end-user*. *Data warehouse* harus mendukung secara efektif analisis *ad hoc* dan analisis rutin. *Performance* yang tinggi dapat dicapai dengan merencanakan terlebih dahulu kebutuhan untuk *join*, *summation*, dan laporan periodik oleh *end-user*.

Ada lima kelompok utama dari *tool* ini:

- *Reporting and Query Tool*

Tool reporting meliputi *tool* produksi laporan dan penulis laporan. *Tool* produksi laporan digunakan untuk menghasilkan laporan operasional regular atau mendukung tugas *batch* berkapasitas tinggi, seperti pesanan pelanggan/faktur dan cek *staff*. Penulis laporan, merupakan *tool desktop* yang murah yang didesain untuk *end-user*. *Tool query* untuk *data warehouse relational* didesain untuk menerima SQL atau menghasilkan pernyataan SQL untuk meng-*query* data yang disimpan di dalam *data warehouse*. *Tool* ini melindungi *user* dari kompleksitas dari SQL dan struktur *database* dengan mengikutsertakan sebuah layer meta di antara *user* dan *database*. Layer meta adalah perangkat lunak yang menyediakan *view subject-oriented* dari

database dan mendukung pembuatan ‘*point and click*’ dari SQL. Sebuah contoh dari *tool query* adalah *Query-By-Example* (QBE). *Tool query* dikenal dengan *user* dari aplikasi bisnis seperti analisis *demographic* dan daftar *mailing* pelanggan.

- *Application Development Tool*

Kebutuhan dari *end-user* seperti kapabilitas *built-in* dari *tool reporting* dan *tool query* tidak sesuai baik karena analisis yang dibutuhkan tidak dapat dilakukan atau karena interaksi *user* membutuhkan *expertise* tingkat tinggi pada *user*. Dalam situasi ini, akses *user* mungkin membutuhkan pengembangan aplikasi *in-house* yang menggunakan *tool* akses data grafikal yang didesain untuk lingkungan *client-server*. Beberapa dari *tool* pengembangan aplikasi ini berintegrasi dengan *tool* OLAP, dan dapat mengakses semua sistem *database*, termasuk *Oracle*, *Sybase*, dan *Informix*.

- *EIS (Executive Information System) tool*

EIS, yang sering disebut dengan ‘sistem informasi semua orang’, awalnya didesain untuk mendukung pembuatan keputusan strategis. Namun, fokus dari sistem ini diperluas untuk mendukung semua tingkat dari manajemen. *Tool* EIS awalnya berhubungan dengan *mainframe* yang memungkinkan *user* untuk membangun aplikasi pembuat keputusan grafikal untuk menyediakan sebuah *overview* dari data organisasi dan mengakses sumber data eksternal.

- *OLAP (Online analytical processing) tool*

Tool OLAP didasarkan pada konsep *database* multidimensi dan memungkinkan *user* untuk menganalisa data menggunakan *view*

multidimensi yang kompleks. Aplikasi bisnis untuk *tool* ini meliputi mempertinggi efektivitas dari kampanye *marketing*, peramalan penjualan produk, dan perencanaan kapasitas. *Tool* ini mengasumsikan data diorganisir dalam model multidimensi yang didukung oleh MDDB (*multi-dimensional database*) khusus atau oleh *database relational* yang didesain untuk memungkinkan *query* multidimensi.

- *Data mining tool*

Data mining adalah sebuah proses menemukan korelasi, pola, dan tren baru dengan menggali sejumlah besar data menggunakan teknik statistik, matematis, intelegensia buatan. *Data mining* memiliki potensial untuk memperluas kapabilitas dari *tool* OLAP karena tujuan utama dari *data mining* adalah untuk membangun model *predictive* dari pada model *retropective*.

2.1.2.4 Manfaat *Data warehouse*

Implementasi *data warehouse* dapat memberikan berbagai keuntungan bagi organisasi (Connolly dan Begg, 2002, p1152) antara lain :

- Potensi tingkat pengembalian atas investasi (ROI) yang tinggi

Sebuah organisasi harus mengeluarkan sumber daya yang besar untuk memastikan keberhasilan implementasi *data warehouse*, dan besar biaya bervariasi bergantung pada solusi teknis yang ada. Namun, rata-rata tingkat ROI yang akan didapatkan perusahaan relatif lebih besar.

- Keunggulan kompetitif

Tingkat pengembalian investasi yang tinggi bagi perusahaan yang sukses mengimplementasikan *data warehouse* tersebut merupakan bukti dari keunggulan kompetitif yang didapatkan perusahaan dari teknologi tersebut. Keunggulan kompetitif diperoleh dengan memungkinkan para pembuat keputusan untuk dapat mengakses data yang sebelumnya tidak tersedia, tidak diketahui, ataupun informasi yang tidak tercatat.

- Peningkatan produktifitas para pengambil keputusan perusahaan

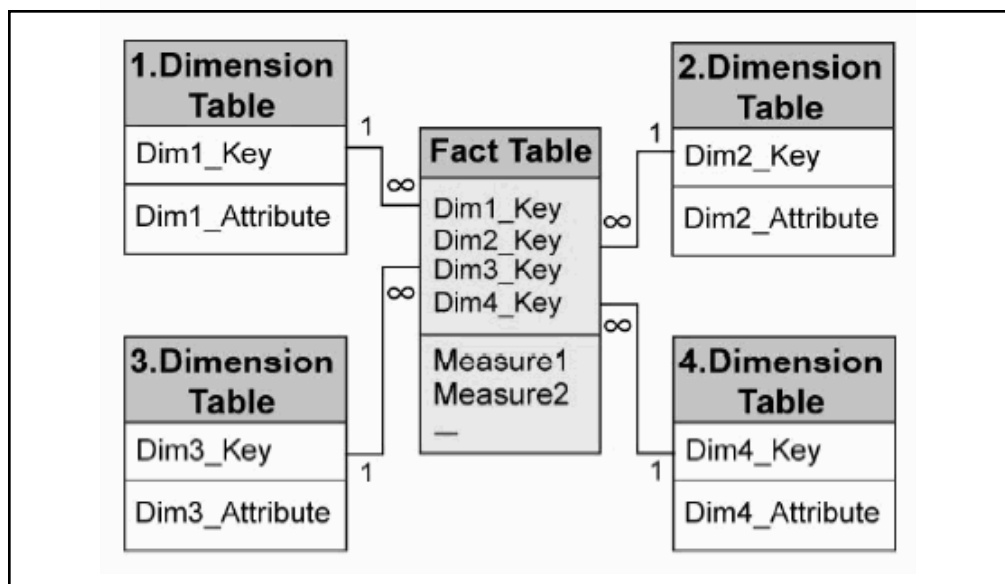
Data warehouse meningkatkan produktifitas para pembuat keputusan dalam perusahaan dengan cara menyediakan database yang terintegrasi dari data-data historis yang berorientasi subjek dan konsisten. *Data warehouse* mengintegrasikan data dari berbagai sistem yang terpisah menjadi suatu bentuk yang menyediakan satu pandangan utuh mengenai organisasi. Melalui proses pengubahan data menjadi informasi yang lebih berarti, *data warehouse* memungkinkan para manajer bisnis untuk melakukan analisis yang lebih substansial, akurat, dan konsisten.

2.1.2.5 Skema Bintang

Menurut Connolly dan Begg (2005, p1183), skema bintang (*star schema*) adalah struktur logikal yang mempunyai sebuah tabel fakta berisi data faktual yang ditempatkan di tengah, dikelilingi oleh tabel dimensi berisi data referensi (yang dapat dinormalisasi). Skema bintang mengeksploitasi karakteristik dari data faktual di mana fakta dibuat dari peristiwa yang muncul di masa lalu dan mustahil untuk berubah, dengan mengabaikan bagaimana mereka dianalisis. Kebanyakan fakta yang digunakan dalam tabel fakta adalah angka dan *additive* karena aplikasi *data warehouse* tidak

pernah diakses sebagai sebuah *record* tunggal, tetapi mereka diakses ratusan, ribuan bahkan jutaan *record* pada suatu waktu dan hal yang paling berguna untuk dilakukan dengan *record* yang begitu banyak tersebut adalah dengan mengagregasikan mereka. Tabel dimensi, berisi deksripsi informasi berupa teks.

Skema bintang dapat digunakan untuk mempercepat kinerja *query* dengan denormalisasi informasi ke dalam sebuah tabel dimensi. Denormalisasi tepat ketika terdapat sejumlah *entity* yang berhubungan dengan tabel dimensi yang sering diakses, menghindari *overhead* dari penggabungan tabel tambahan untuk mengakses atribut. Denormalisasi tidak tepat di mana data tambahan tidak sering diakses, karena *overhead* tabel dimensi yang diperluas tidak mungkin *offset* oleh berbagi perolehan dalam *query*.



Gambar 2.2 Skema Bintang

Sumber : <http://www.juergen-konicek.de/Pictures/DWHSchemas.gif>

Gambar 2.2 di atas menggambarkan karakteristik dari komponen skema bintang:

1. Tabel dimensi

Menurut Inmon (2005,p495) tabel dimensi adalah tempat di mana data-data yang tidak berhubungan yang berelasi dengan tabel fakta ditempatkan di dalam tabel multidimensional

Disebut juga tabel kecil (*minor table*), biasanya lebih kecil dan memegang data deskriptif yang mencerminkan dimensi suatu bisnis. Tabel dimensi berisi data yang merupakan deskripsi lebih lanjut dari data yang ada pada tabel fakta.

2. Tabel fakta

Menurut Inmon (2005,p497) tabel fakta adalah tabel pusat dari *star join* di mana data yang sering muncul akan ditempatkan.

Disebut juga tabel utama (*major table*), merupakan inti dari skema bintang dan berisi data aktual yang akan dianalisis (data kuantitatif dan transaksi). *Field-field* tabel fakta sering disebut *measure* dan biasanya dalam bentuk numerik. Selalu berisi *foreign key* dari masing-masing tabel dimensi. Tabel ini dapat terdiri dari banyak kolom dari ribuan baris data.

Keuntungan yang diperoleh dari skema bintang yaitu :

1. Mudah dipahami pengguna.

Skema bintang menggambarkan dengan jelas bagaimana pengguna berpikir dan memerlukan data untuk *query* dan analisa. Skema bintang menggambarkan hubungan antar tabel sama seperti cara pengguna melihat hubungan tersebut secara normal.

2. Mengoptimalkan navigasi.

Skema bintang mengoptimalkan navigasi melewati *database* sehingga lebih mudah dilihat. Meskipun hasil *query* terlihat kompleks, tetapi navigasi itu memudahkan pengguna.

3. Paling cocok untuk pemrosesan *query*.

Skema bintang paling cocok untuk pemrosesan *query* karena skema ini berpusat pada *query*. Tanpa bergantung pada banyak dimensi dan kompleksitas *query*, setiap *query* akan dengan mudah dijalankan, pertama dengan memilih baris dari tabel dimensi dan kemudian menemukan baris yang sama di tabel fakta.

2.1.3 *Data Mart*

2.1.3.1 Definisi *Data Mart*

Menurut Imhoff, Galemno, dan Geiger (2003, p14-p15), *data mart* merupakan sebuah subset dari *data warehouse*, dimana hampir seluruh aktivitas analitik dalam lingkungan *business intelligent* dilakukan. Data di dalam setiap *data mart* biasanya disesuaikan untuk melakukan fungsi tertentu, misalnya digunakan untuk *product profitability analysis*, *KPI analysis*, *customer demographic analysis*, dsb.

Sedangkan Inmon (2005, p494) mendefinisikan *data mart* sebagai data yang dibentuk berdasarkan departemen, yang sumber datanya berasal dari *data warehouse*, dimana proses denormalisasi dilakukan terhadap data berdasarkan kebutuhan informasi departemen tersebut.

Ada beberapa karakteristik yang membedakan antara *data mart* dengan *data warehouse*, yaitu (Connolly dan Begg, 2005, p1171-p1173) :

- *Data mart* lebih berfokus kepada ketentuan maupun permintaan dari pengguna yang berkaitan dengan sebuah departemen ataupun fungsi-fungsi bisnis organisasi.
- Secara normal *data mart* tidak mencakup data operasional yang mendalam tidak seperti halnya dengan *data warehouse*
- Karena *data mart* memiliki jumlah data yang lebih sedikit dibandingkan dengan *data warehouse*, *data mart* lebih mudah dimengerti dan diaplikasikan

2.1.3.2 Alasan Membangun *Data Mart*

Ada beberapa alasan untuk membangun sebuah *data mart* (Connolly dan Begg, 2005, p1173), diantaranya :

- Memberi akses kepada *user* akan data yang diperlukan untuk melakukan analisis
- Menyediakan data dalam bentuk yang disesuaikan dengan berbagai sudut pandang atas data oleh sekelompok pemakai dalam sebuah departemen atau fungsi bisnis
- Mengurangi waktu respon dari *end-user*, sehubungan dengan berkurangnya volum data yang diakses
- Menyediakan struktur data yang sesuai dengan kebutuhan *tools* milik *end-user*, seperti OLAP ataupun *data mining*. *Tools* tersebut mungkin membutuhkan struktur basis data internal sendiri. Pada praktiknya, *tools* tersebut biasanya membentuk desain *data mart* sendiri untuk mendukung fungsionalitas tertentu secara spesifik.

- *Data mart* umumnya menggunakan data lebih sedikit sehingga berbagai proses seperti *cleansing*, *loading*, transformasi dan integrasi menjadi jauh lebih mudah, oleh karena itu pembuatan serta implementasi *data mart* menjadi lebih sederhana bila dibandingkan dengan *data warehouse*.
- Biaya implementasi *data mart* yang dibutuhkan umumnya jauh lebih kecil daripada biaya implementasi sebuah *data warehouse*.
- Para pengguna *data mart* yang potensial dapat didefinisikan dengan lebih jelas, serta dengan mudah dapat ditetapkan sebagai sasaran untuk memperoleh dukungan terhadap sebuah proyek *data mart* daripada sebuah proyek *data warehouse* perusahaan.

2.1.4 Data mining

2.1.4.1 Definisi Data mining

Menurut Berry (2004, p7), *data mining* adalah kegiatan mengeksplorasi dan menganalisis data dalam jumlah yang besar untuk menemukan *pattern* dan *rule* yang berarti. *Data mining* digunakan untuk mencari informasi bisnis yang berharga dari basis data yang sangat besar, yang dipakai untuk memprediksi trend dan sifat-sifat bisnis serta menemukan pola- pola yang tidak diketahui sebelumnya.

Menurut Witten dan Frank (2005, p5), *data mining* adalah proses untuk menemukan pola-pola dalam data. Proses yang dilakukan dapat berupa proses otomatis ataupun semi-otomatis. Pola-pola yang ditemukan harus memiliki arti dan memberikan manfaat tertentu, biasanya manfaat ekonomis.

Menurut Prescott, Hoffer dan McFadden (2005, p482), *data mining* adalah penemuan pengetahuan dengan menggunakan teknik-teknik yang tergabung dari statistic tradisional, artificial intelligence dan grafik komputer.

Berdasarkan beberapa pengertian diatas, dapat ditarik kesimpulan bahwa *data mining* adalah suatu proses analisis untuk menggali informasi yang berharga yang tersembunyi dengan menggunakan statistik, *artificial intelligence* di dalam suatu koleksi data (*database*) dengan ukuran sangat besar sehingga ditemukan suatu pola dari data yang sebelumnya tidak diketahui dan pola tersebut direpresentasikan dengan grafik komputer agar mudah dimengerti.

2.1.4.2 Metode Pembelajaran *Data mining*

Teknik-teknik *data mining* berhubungan dengan proses penemuan dan pembelajaran, dimana proses tersebut dapat dikategorikan ke dalam tiga metode utama, yaitu *supervised*, *unsupervised*, serta *reinforcement learning*. (Berson, Smith, dan Thearling, 2000, p36-p37).

- *Supervised learning*

Metode ini melibatkan fase pembelajaran yang terjadi ketika data-data historis yang karakteristiknya dipetakan ke hasil keluaran, diproses melalui algoritma *data mining*. Proses tersebut akan melatih algoritma untuk mengenali variabel kunci dan nilai-nilai yang akan dijadikan sebagai dasar pembuatan prediksi.

- *Unsupervised learning*

Metode ini tidak melibatkan tahap pembelajaran melainkan bergantung pada penggunaan algoritma untuk mendeteksi pola-pola, seperti asosiasi, *sequences*,

yang ada pada data masukan berdasarkan kriteria penting yang telah ditentukan. Pendekatan ini mengarah kepada pembuatan aturan-aturan yang menggambarkan asosiasi, *cluster*, dan segmen yang telah ditemukan.

- *Reinforcement learning*

Metode ini tidak banyak digunakan bila dibandingkan dengan metode yang lainnya, namun memiliki penerapan dalam optimisasi berdasarkan waktu dan pengendalian adaptif. Metode ini menyerupai realita, dimana pembelajaran didapatkan dari konsekuensi tindakan yang telah dipilih sebelumnya, bukan berdasarkan aturan yang secara eksplisit dinyatakan. Karena prosesnya yang tidak menyediakan tindakan korektif secara seketika, maka metode ini dapat digunakan untuk menyelesaikan permasalahan yang sangat sulit dan tidak bergantung pada waktu.

2.1.4.3 Tugas *Data mining*

Ada banyak permasalahan dalam bidang intelektual, ekonomi, dan bisnis yang dapat diekspresikan dalam tugas-tugas *data mining*. Berry dan Linoff (2004, p8-p12) mengemukakan beberapa tugas yang dapat diselesaikan menggunakan *data mining*, antara lain :

- *Classification* (Klasifikasi)

Klasifikasi merupakan salah satu tugas *data mining* yang paling umum. Klasifikasi melibatkan proses pemeriksaan karakteristik suatu objek dan kemudian memasukkannya ke dalam salah satu kelas yang sudah didefinisikan sebelumnya. Proses klasifikasi biasanya melibatkan pendefinisian kelas-kelas

dan data sampel yang berisi contoh objek-objek yang sudah diklasifikasi sebelumnya. Tujuan dari tugas ini adalah untuk membuat sebuah model yang dapat diaplikasikan pada data yang belum terklasifikasi. Beberapa contoh tugas klasifikasi yang ditujukan untuk menyelesaikan permasalahan bisnis misalnya seperti klasifikasi tingkat resiko dari pihak yang mengajukan kredit dan mengetahui klaim asuransi yang palsu, dsb.

- *Estimation* (Estimasi)

Estimasi berkaitan hasil keluaran yang berupa nilai kontinu. Berangkat dari data yang digunakan sebagai masukan, proses estimasi akan menghasilkan sebuah nilai untuk variabel kontinyu yang sebelumnya tidak diketahui nilainya. Estimasi juga sering digunakan dalam melakukan tugas-tugas klasifikasi. Contoh permasalahan yang melibatkan proses estimasi misalnya seperti memperkirakan jumlah anak dalam sebuah keluarga, total pendapatan sebuah keluarga, atau *lifetime value* (nilai hidup) seorang pelanggan, dsb.

- *Prediction* (Prediksi)

Prediksi sama seperti tugas klasifikasi dan estimasi, hanya saja data diklasifikasikan berdasarkan perilaku atau nilai yang diperkirakan pada masa yang akan datang. Dalam proses prediksi, satu-satunya cara untuk memeriksa ketepatan hasil adalah dengan menunggu dan memperhatikan. Teknik-teknik yang digunakan untuk melakukan tugas klasifikasi dan estimasi dapat diadaptasikan dalam melakukan prediksi dengan menggunakan data sampel dimana nilai dari variabel yang akan diprediksikan sudah diketahui, begitu pula dengan data historis untuk data sample tersebut. Data historis digunakan untuk

membuat sebuah model yang menjelaskan perilaku yang sedang diamati. Apabila model ini diaplikasikan pada data masukan, akan menghasilkan prediksi di masa yang akan datang. Contoh prediksi yang dapat dilakukan dengan *data mining* misalnya memprediksikan pelanggan mana yang akan beralih ke pesaing dalam waktu enam bulan ke depan, pelanggan mana yang akan memesan layanan atau produk tertentu, dsb.

- *Affinity grouping* (Pengelompokan berdasarkan persamaan)

Tujuan *affinity grouping* adalah untuk menentukan hal-hal yang akan terjadi bersamaan, misalnya perusahaan *retail* dapat melakukan *market basket analysis*, yakni menentukan produk-produk apa saja yang dibeli oleh pelanggan secara bersamaan. *Affinity grouping* dapat juga digunakan untuk mengidentifikasi kesempatan melakukan *cross-selling* atau untuk mengelompokan produk atau layanan-layanan yang saling sesuai. *Affinity grouping* merupakan salah satu contoh sederhana dalam menghasilkan aturan-aturan dari data.

- *Clustering* (Pengelompokan)

Clustering adalah proses untuk melakukan segmentasi atas sebuah populasi yang heterogen menjadi beberapa sub kelompok atau *cluster* yang homogen. *Clustering* berbeda dengan proses klasifikasi, karena tidak bergantung pada kelas-kelas yang sudah ditetapkan sebelumnya ataupun data sampel. Data-data akan dikelompokkan berdasarkan kemiripan karakteristik. *Clustering* seringkali digunakan sebagai langkah awal sebelum melakukan tugas *data mining* lainnya.

- *Description and profiling*

Data mining dapat digunakan untuk menemukan dan mendeskripsikan apa yang ada atau terjadi berdasarkan data yang dimiliki, sehingga meningkatkan pemahaman tentang berbagai hal.

2.1.4.4 Teknik *Data mining*

Berson, Smith, dan Thearling (2000, p36-p37) menyatakan bahwa teknik *data mining* dapat didasarkan pada hal-hal berikut ini :

- Representasi dari model dan hasil (misalnya : pohon keputusan, aturan (*rule*), penyimpangan (*deviation*), dan korelasi)
- Tipe dari data yang dipakai (misalnya : data yang berkesinambungan (*continuous*), data yang periodik, data yang nominal)
- Tipe aplikasi yang digunakan (misalnya : ekonomi, biologi)
- Atribut – atribut dari pola yang dihasilkan (misalnya : keakuratan, ketelitian)

Penggunaan *data mining* untuk memecahkan masalah bisnis menurut Berson, Smith, dan Thearling (2000, p37-p38) dapat dikategorikan pula berdasar orientasi masa waktu analisis, yaitu :

- Analisis retrospektif (*Retrospective analysis*)
 Memfokuskan pada kegiatan yang dilakukan dimasa lalu atau sekarang. Contohnya sebuah perusahaan perlu analisa tentang unjuk kerja dan hasil penjualan perusahaan yang berkaitan dengan lokasi geografi yang berbeda, demo grafi dan tipe produk.
- Analisis prediktif (*Predictive analysis*)

Memfokuskan pada prediksi (peramalan) untuk kejadian tertentu atau karakteristik pelanggan berdasarkan informasi histori. Contohnya sebuah perusahaan ingin membangun sebuah model yang mampu memprediksi tingkat kehilangan pelanggan dari perusahaan serta menentukan langkah-langkah yang mengurangi tingkat kehilangan pelanggan tersebut.

Menurut Berson, Smith, dan Thearling (2000, p124-p200), teknik *data mining* terbagi menjadi dua, yaitu teknik klasik (*classical techniques*) dan teknik generasi berikut (*next generation*). Teknik klasik terdiri dari teknik statistik, *nearest neighbor*, dan *clustering*, sedangkan teknik generasi berikut terdiri dari *trees*, *networks*, dan *rules*.

2.1.4.4.1 Teknik Klasik

- Statistik

Pada dasarnya teknik ini bukanlah *data mining*. Teknik ini telah banyak digunakan pada banyak aplikasi bisnis lama sebelum *data mining* muncul. Bagaimanapun teknik ini dipacu oleh data dan digunakan untuk menemukan pola dan membangun model prediksi. Namun, pemakaian teknik ini telah banyak ditinggalkan karena teknik *data mining* lainnya seperti CART, *neural networks* dan *nearest neighbor* cenderung lebih *powerfull* bagi data bisnis yang kompleks dan mendukung untuk digunakan oleh *user* yang kurang ahli dalam statistik.

- *Nearest Neighbor*

Teknik *clustering* dan *nearest neighbor* merupakan teknik pertama yang digunakan dalam *data mining*. *Nearest neighbor* adalah suatu teknik untuk prediksi yang hampir mirip dengan *clustering* yaitu melakukan pengelompokan

record yang sejenis. Dalam memprediksikan nilai prediksi pada sebuah *record*, melihat *record-record* lain yang memiliki kemiripan dengan nilai yang dijadikan nilai prediksi (*predictor*) dalam basis data historis dan menggunakan nilai prediksi dari *record* yang paling mendekati *record* yang tidak sejenis. Contoh penggunaannya adalah dalam *text retrieval*, yang masalahnya adalah dalam pencarian dokumen yang mirip atau sejenis dengan suatu dokumen atau jurnal tertentu. Teknik ini akan mencari dokumen lain yang memiliki kemiripan karakteristik penting yang ditentukan sebagai karakteristik yang paling menarik dan sangat penting untuk ditemukan.

- *Clustering*

Teknik *clustering* adalah metode yang digunakan untuk mengelompokkan *records* yang ada. Biasanya, hal ini dilakukan untuk memberikan pengguna akhir (*end-user*) sebuah *high-level view* tentang apa yang terjadi dalam *database*. *Clustering* kadang digunakan untuk membuat segmentasi, yang banyak digunakan oleh bagian pemasaran sebagai *birds-eye view* dari bisnis. Algoritma *nearest neighbor* merupakan peningkatan dari *clustering*, dalam hal mereka menggunakan *distance* untuk membuat struktur dalam data atau prediksi. Algoritma *nearest neighbor* dikatakan sebagai peningkatan karena algoritma ini menentukan bobot *importance* dari prediktor dan bagaimana *distance* akan diukur, secara otomatis. Sedangkan, dalam *clustering* setiap prediktor masih diperlakukan sama.

2.1.4.4.2 Teknik Generasi Berikutnya

- *Decision Tree*

Decision tree merupakan model prediktif yang dapat direpresentasikan seperti bentuk pohon. Setiap cabang dari pohon merupakan sebuah pertanyaan klasifikasi dan akar dari pohon merupakan bagian dari *dataset* berdasarkan klasifikasinya masing-masing.

- *Neural Networks*

Neural Network merupakan teknik *data mining* yang paling umum. Teknik ini mempelajari *training set*, kemudian membentuk suatu pola di dalamnya untuk melakukan klasifikasi dan prediksi. Keuntungan utama dari *neural network* adalah dapat diterapkan secara luas. Teknik ini sangat menarik karena dapat mendeteksi pola data yang dapat dianalogikan seperti jalan pikiran manusia. Namun *neural network* memiliki dua kekurangan utama. Pertama, sulit untuk memahami model yang dihasilkan, karena model tersebut direpresentasikan dengan nilai numerik yang berasal dari proses kalkulasi yang kompleks, dan hasil dari *neural network* tersebut juga berupa numerik yang perlu dikonversikan apabila nilai prediksi yang diberikan merupakan nilai kategori. Kedua, teknik ini sangat sensitif terhadap format data yang masuk. Representasi data yang berbeda dapat memberikan hasil yang berbeda pula, sehingga penting sekali dilakukan pengaturan data sebelum menggunakannya.

- *Rule Induction* atau *Association Rule*

Rule Induction merupakan salah satu bentuk utama dari *data mining* dan merupakan bentuk paling umum mengenai proses penemuan pengetahuan dalam

sistem *unsupervised learning*. Teknik ini juga merupakan bentuk *data mining* yang paling mendekati apa yang dipikirkan orang mengenai *data mining*, yaitu menambang emas dari sebuah data yang besar. Yang menjadi emas adalah sebuah aturan yang menarik, dan dapat memberikan suatu gambaran tentang data yang mungkin tidak diketahui dan dapat dengan mudah dikemukakan. Teknik *rule induction* melibatkan pemrosesan data secara besar-besaran, dimana semua pola yang memungkinkan secara sistematis akan diambil dari data, kemudian dilakukan pengukuran terhadap pola tersebut untuk mengetahui kemungkinan pola tersebut muncul kembali.

2.1.4.5 Metodologi Data Mining

Banyak *vendor* atau organisasi yang menspesifikasikan sebuah *model* proses yang dirancang untuk memandu user dalam melakukan *Data mining*.

Pada tahun 1996, sebuah konsorsium dari *vendor* dan *user* yang terdiri dari NCR Systems Engineering Copenhagen (Denmark), Daimler – Benz AG (Germany), SPSS/Integral Solutions Ltd. (England), dan OHRA Verzekeringen en Bank Groep BV (The Netherlands) membangun atau merumuskan sebuah spesifikasi sebagai standarisasi metodologi *data mining* dengan nama *Cross Industry Standard Process for Data mining (CRISP-DM)* (Connolly dan Begg, 2005, p1239).

Tahapan tersebut terdiri dari (<http://www.crisp-dm.org>):

a. Business Understanding

Tujuan dari tahap ini adalah untuk mengerti tujuan proyek dan kebutuhan dasar (*requirements*) dari perspektif bisnis, dan kemudian mengkonversikan

pengetahuan menjadi definisi masalah *data mining* dan kemudian membuat perencanaan awal.

b. *Data Understanding*

Dimulai dengan mengumpulkan data, kemudian mengenal data serta mengerti data dengan baik, dengan tujuan:

- Untuk mengidentifikasi masalah kualitas data
- Untuk menemukan arti dari data
- Untuk membuat hipotesis dari data mengenai informasi yang tersembunyi

c. *Data Preparation*

Mencakup semua kegiatan yang dibutuhkan untuk membangun *data set* akhir dari data-data mentah. Ada pun *data set* adalah data yang akan digunakan untuk dimasukkan ke dalam *modeling tool*. Kegiatan ini meliputi: memilih tabel, *case/record* dan atribut untuk kemudian di transformasi dan dibersihkan. Setelah itu data dapat digunakan untuk *modeling tool*.

d. *Modeling*

Memilih dan menerapkan berbagai variasi dari teknik *modeling* dan membuat *standar parameter* dari *modeling tool* sampai nilai *optimalnya*. Ada beberapa teknik yang dapat digunakan dalam kasus *data mining* yang sama. Beberapa teknik mempunyai *requirement* yang berbeda terhadap data yang akan digunakan. Kalau data yang akan digunakan belum siap, maka kita harus kembali ke tahap *data preparation*.

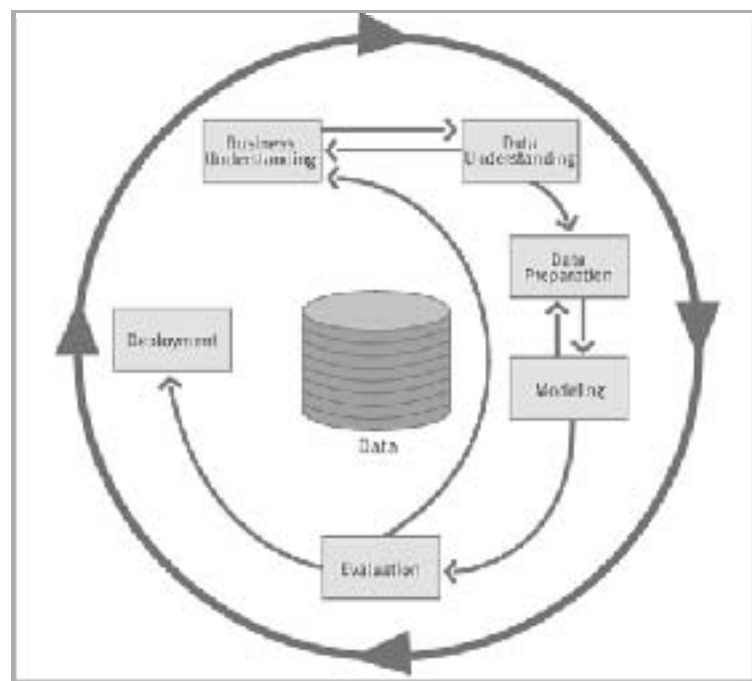
e. *Evaluation*

Evaluasikan model secara menyeluruh dan lihat kembali langkah-langkah membuat model yang telah dikerjakan, untuk meyakinkan apakah model telah

mencapai *goal* dari bisnis. Tentukan juga beberapa faktor bisnis penting yang tidak ditangani oleh model. Pada akhir dari tahap ini, keputusan penggunaan hasil *data mining* telah ditentukan.

f. *Deployment*

Mengorganisir dan mempresentasikan hasil dari *data mining*. *Deployment* dapat saja semudah membuat *report* otomatis, ataupun susah mengimplementasikan proses *data mining* secara berulang.



Gambar 2.3 Metodologi *Data mining* CRISP-DM

Sumber : <http://www.crisp-dm.org/Images/Crisp-dmchartnew.gif>

Gambar 2.3 di atas menggambarkan metodologi *data mining* CRISP-DM mulai dari *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation* dan *deployment*.

2.1.4.6 Perbedaan *Data mining* dengan OLAP

Online Analytical Processing (OLAP) merupakan sebuah istilah untuk tool yang terdiri dari basis data (*database*) dan antar-muka pengguna (*user interface*), dimana pengguna dapat menavigasikan data. OLAP merupakan sebuah tool untuk melihat dengan cepat “gunung” data yang dimiliki. Namun tidak dapat membedakan mana yang bernilai dan mana yang tidak dalam “gunung” tersebut (Berson, Smith, dan Thearling, 2000, p91). Kelemahan OLAP ini akan dilaksanakan oleh *data mining*.

2.1.4.7 Perbedaan *Data mining* dengan Statistik

Statistik mempunyai penggunaan dan hasil yang sama dengan *data mining*. Regresi sering digunakan dalam statistik untuk menciptakan model yang bersifat prediktif. Model ini dibuat dari data historis yang juga berkarakter sama dengan data yang digunakan oleh *data mining*. Perbedaan utama antara statistik dengan *data mining* adalah *data mining* digunakan oleh pebisnis, bukan ahli statistik. *Data mining* mengotomatisasi proses statistik, sehingga pengguna tidak diharuskan untuk belajar statistik terlebih dahulu (Berson, Smith, dan Thearling, 2000, p91)

2.1.5 Teknik *Clustering*

Clustering adalah proses membuat pengelompokan sehingga semua anggota dari setiap partisi mempunyai persamaan berdasarkan matrik tertentu. Sebuah *cluster* adalah sekumpulan obyek yang digabung bersama karena persamaan atau kedekatannya. *Clustering* berdasarkan persamaanya merupakan sebuah teknik yang sangat berguna. Karena akan mentranslasi ukuran persamaan yang intuitif menjadi ukuran yang kuantitatif. Ada banyak pendekatan untuk membuat *cluster*, diantaranya adalah

membuat aturan yang mendikte keanggotaan dalam group yang sama berdasarkan tingkat persamaan diantara anggota-anggotanya. Pendekatan lainnya adalah dengan membuat sekumpulan fungsi yang mengukur beberapa properti dari pengelompokan tersebut sebagai fungsi dari beberapa parameter dari sebuah *clustering*.

Analisis *Clustering* adalah proses pengelompokan obyek ke dalam *subsets* yang mempunyai arti dalam konteks masalah tertentu. Obyek dengan demikian diorganisir ke dalam suatu penyajian efisien dan bermanfaat. Tidak sama dengan klasifikasi, *clustering* tidak bersandar pada kelas sudah ada. *Clustering* dikenal sebagai suatu metode pelajaran pembelajaran *unsupervised* karena tidak ada informasi disajikan tentang “jawaban yang benar“ untuk obyek yang manapun. Ini dapat menemukan hubungan yang sebelumnya tidak diketahui didalam suatu *dataset* yang kompleks.

Analisis *cluster* adalah suatu teknik analisa multivariate untuk mencari dan mengorganisir informasi tentang variabel sehingga secara relatif dapat dikelompokkan dalam kelompok yang homogen atau “*cluster*” dapat dibentuk. *Cluster* dibentuk dengan metode kedekatan yang secara internal harus homogen (anggota adalah serupa untuk satu sama lain) dan sangat secara eksternal tak sejenis (anggotanya tidak seperti anggota dari *cluster* yang lain).

Analisis *cluster* dapat menerima suatu data masukan yang beragam. Ini biasanya disebut pengukuran “kesamaan“, dapat juga disebut “kedekatannya“, dan “kemiripannya“. Beberapa ahli merekomendasikan penggunaan standardisasi data, *cluster* dapat dihitung dalam skala yang berbeda dan standardisasi akan memberi pengukuran dengan menggunakan unit yang berbeda.

Seperti teknik yang lain, analisis *cluster* menghadapi permasalahan dalam beberapa banyak faktor, atau dimensi, atau berapa banyak *cluster* yang akan dihasilkan. Untuk itu

akan dipilih suatu tempat dimana struktur *cluster* yang stabil untuk jarak yang jauh. Beberapa kemungkinan lain akan mencari pengelompokan grup dengan struktur cocok atau yang diharapkan.

2.1.5.1 Fungsi Jarak

Pengukuran *proximity* yang paling umum digunakan, sedikitnya untuk rasio skala adalah matrik *Minkowski*, yang mana adalah suatu generalisasi jarak antara titik di dalam *Euclidean Space*.

- **Fungsi *Euclidean Distance* :**

Jarak *Euclidean* dapat dianggap sebagai jarak yang paling pendek antar dua poin, dan pada dasarnya sama halnya dengan persamaan *Pythagoras* ketika digunakan di dalam dua dimensi. Secara matematis dapat dituliskan di dalam persamaan berikut :

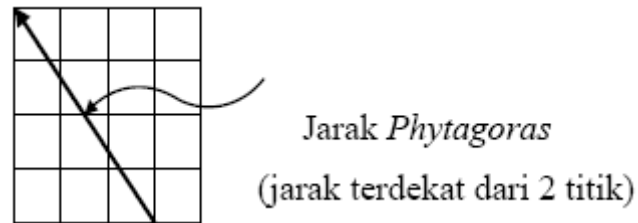
$$d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2}$$

Ketika menggunakan fungsi jarak *Euclidean* untuk membandingkan jarak, tidak diperlukan untuk mengkalkulasi akar dua sebab jarak selalu merupakan angka positif. Untuk dua jarak, d_1 dan d_2 , jika :

$$\sqrt{d_1} > \sqrt{d_2} \Leftrightarrow d_1 > d_2$$

Jika sebagian dari suatu atribut obyek diukur dengan skala berbeda, maka ketika menggunakan fungsi jarak *Euclidean*, atribut dengan skala yang lebih besar boleh meliputi atribut yang terukur pada skala yang lebih kecil. Untuk mencegah masalah ini, nilai-nilai atribut dinormalisasi untuk terletak diantara 0 dan 1. Fungsi jarak lain mungkin lebih sesuai untuk beberapa data. Lebih

jelasanya berikut di bawah ini ditampilkan gambar representasi dari jarak terdekat dari 2 titik, seperti yang terlihat pada gambar 2.4 berikut ini :



Gambar 2.4 Fungsi *Euclidean*

Suatu komponen penting pada algoritma *cluster* adalah mengukur jarak antara poin-poin data. Jika komponen dari data adalah semua termasuk dalam unit yang sama, mungkin jarak *Euclidean* yang sederhana cukup sukses mengelompokkan data serupa.

Bagaimanapun, bahkan dalam hal ini jarak *Euclidean* kadang-kadang dapat salah. Di samping kedua-duanya pengukuran diambil di dalam unit yang sama, suatu keputusan harus dibuat berkaitan dengan skala. Skala yang berbeda dapat menyebabkan perbedaan *clustering*.

2.1.5.2 Algoritma *K-Means*

Algoritma *K-Means* adalah metode *clustering* berbasis jarak yang membagi data ke dalam sejumlah *cluster* dan algoritma ini hanya bekerja pada atribut numerik.

Pada dasarnya penggunaan algoritma dalam melakukan proses *clustering* tergantung dari data yang ada dan konklusi yang ingin dicapai. Untuk itu digunakan Algoritma *K-Means* yang di dalamnya memuat aturan sebagai berikut :

- Jumlah *cluster* perlu diinputkan.

- Hanya memiliki atribut bertipe numerik.
- Jumlah atribut sedikit (<100)

Algoritma *K-Means* merupakan metode *nonheirarchial* yang pada awalnya mengambil sebagian dari banyaknya komponen dari populasi untuk dijadikan pusat *cluster* awal. Pada step ini pusat *cluster* dipilih secara acak dari sekumpulan populasi data. Berikutnya *K-Means* menguji masing-masing komponen di dalam populasi data dan menandai komponen tersebut ke salah satu pusat *cluster* yang telah didefinisikan tergantung dari jarak minimum antar komponen dengan tiap-tiap pusat *cluster*. Posisi pusat *cluster* akan dihitung kembali sampai semua komponen data digolongkan ke dalam tiap-tiap pusat *cluster* dan terakhir akan terbentuk posisi pusat *cluster* baru. Algoritma *K-Means* pada dasarnya melakukan dua proses yakni proses pendeteksian lokasi pusat tiap *cluster* dan proses pencarian anggota dari tiap-tiap *cluster*.

Penggunaan metode yang berbeda untuk menentukan persamaan *cluster* mempunyai pengaruh yang berbeda pula pada teknik *clustering*. Suatu jalan/cara standard untuk menghitung persamaan antar tiap *cluster* dengan populasi data adalah menggambarkan suatu fungsi yang mengukur jarak antar tiap *cluster* dengan populasi data. Definisi fungsi jarak akan mengukur jarak sekelompok populasi data.

Dalam perancangan ini ukuran jarak adalah jarak *Euclidean* untuk menandai adanya persamaan antar tiap *cluster* dengan jarak minimum dan mempunyai persamaan atau kemiripan yang lebih tinggi. *Euclidean Matrix* antara titik $a=(a_1,a_2,a_3,a_4,\dots,a_n)$ dan titik $b=(b_1,b_2,b_3,b_4,\dots,b_n)$ adalah :

$$d(a,b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$

Proses awal dalam algoritma *K-Means* adalah populasi data kriteria pembayaran PT. Enseval Putera Megatrading yang merupakan data yang diproses dan input parameter k , jumlah maksimal dari *cluster* harus ada di dalam algoritma nanti.

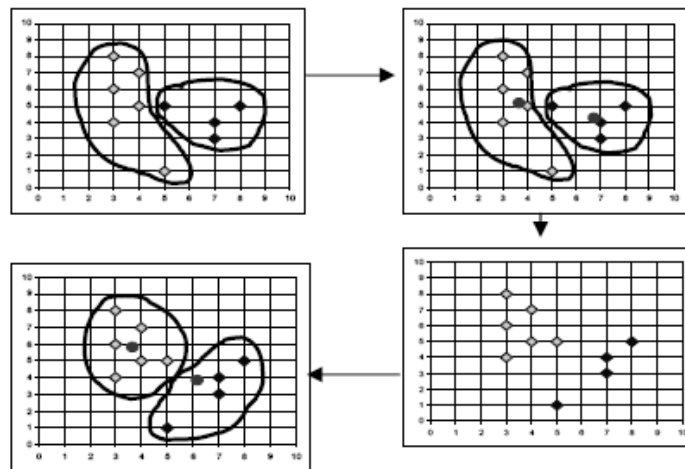
Algoritma K-Means :

1. Tentukan jumlah K *cluster*.
2. Inisialisasi k pusat *cluster* sebagai seed points, Pusat ini dapat diperoleh secara acak.
3. Untuk setiap data pembayaran PT. Enseval Putera Megatrading, cari data yang lebih dekat dengan pusat *cluster* dan tandai titik data tersebut di pusat *cluster* yang terdekat dan posisi pusat *cluster* dihitung kembali dengan rata-rata anggota dari setiap *cluster*.
4. Cek semua data sekali lagi dan taruh setiap data yang terdekat dengan pusat *cluster* (pusat *cluster* tidak dihitung lagi). Jika anggota dari tiap pusat *cluster* tidak berubah, berhenti dan jika masih berubah kembali ke Step ke-2.

Dari algoritma diatas dapat diambil dua operasi yang penting sepanjang dilakukan proses *clustering* dengan algoritma *K-Means* yakni menghitung jarak antar obyek data dengan pusat *cluster* dan menghitung rata-rata dari pusat *cluster* tersebut. Selama ada proses iterasi untuk menentukan pusat *cluster* baru dengan menghitung jarak dari setiap obyek data, disini dapat diketahui bahwa jika ada dua obyek yang memiliki tingkat kedekatan / kemiripan yang sama dengan melihat ukuran jarak sama, maka bisa dimungkinkan dua obyek tersebut dikelompokkan bersama-sama ke dalam pusat *cluster*. Ketika menghitung rata-rata pada setiap *cluster*, semua obyek di dalam *cluster* dijumlahkan dan dibagi dengan total jumlah obyek yang terkait di masing-masing pusat *cluster*.

Karena di dalam algoritma ini dibutuhkan penyelesaian dengan meminimalisasi jumlah atau rata-rata jarak yang besar, varian, maka selama proses *clustering* penting untuk mengkalkulasi ulang total jarak dari matrik data tersebut. Dengan kata lain jika terdapat matrik data yang jaraknya jauh dari pusat *cluster*, maka matrik tersebut mempunyai pengaruh yang besar daripada matrik data lain yang dekat dengan pusat *cluster*. Untuk memindahkan matrik yang jauh dari pusat *cluster* ini perlu untuk dilakukan standardisasi nilai-nilai setiap matrik yang jauh dari pusat *cluster*. Jadi algoritma *K-Means* untuk menentukan pusat *cluster* yang optimal dengan meminimalisasi total jarak antara populasi data dengan pusat *cluster* berdasarkan iterasi yang akan menyeleksi total jarak yang terkecil.

Hasil dari proses *clustering* yang menggunakan Metode *K-Means Clustering* dapat digambarkan pada gambar 2.5 di bawah ini:



Gambar 2.5 Contoh *K-Means Clustering*

2.1.6 Teknik *Association Rule*

Analisis asosiasi atau *Association Rule mining* adalah teknik *data mining* untuk menemukan aturan asosiatif antara suatu kombinasi item. Contoh dari aturan asosiatif

dari analisa pembelian di suatu pasar swalayan adalah dapat diketahuinya berapa besar kemungkinan seorang pelanggan membeli roti bersamaan dengan susu. Dengan pengetahuan tersebut pemilik pasar swalayan dapat mengatur penempatan barangnya atau merancang kampanye pemasaran dengan memakai kupon diskon untuk kombinasi barang tertentu.

Karena analisis asosiasi menjadi terkenal karena aplikasinya untuk menganalisa isi keranjang belanja di pasar swalayan, analisis asosiasi juga sering disebut dengan istilah *market basket analysis*.

Analisis asosiasi dikenal juga sebagai salah satu teknik *data mining* yang menjadi dasar dari berbagai teknik *data mining* lainnya. Khususnya salah satu tahap dari analisis asosiasi yang disebut analisis pola frekuensi tinggi (*frequent pattern mining*) menarik perhatian banyak peneliti untuk menghasilkan algoritma yang efisien.

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, *support* (nilai penunjang) yaitu persentase kombinasi item tersebut dalam database dan *confidence* (nilai kepastian) yaitu kuatnya hubungan antar item dalam aturan asosiatif.

Aturan asosiatif biasanya dinyatakan dalam bentuk :

$\{\text{roti, mentega}\} \rightarrow \{\text{susu}\} \{\text{support} = 40\%, \text{confidence} = 50\%\}$
--

Yang artinya : "50% dari transaksi di database yang memuat item roti dan mentega juga memuat item susu. Sedangkan 40% dari seluruh transaksi yang ada di database memuat ketiga item itu." Dapat juga diartikan : "Seorang konsumen yang membeli roti dan mentega punya kemungkinan 50% untuk juga membeli susu. Aturan ini cukup signifikan karena mewakili 40% dari catatan transaksi selama ini."

Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan asosiatif yang memenuhi syarat minimum untuk *support* (*minimum support*) dan syarat minimum untuk *confidence* (*minimum confidence*).

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap :

a. **Analisa pola frekuensi tinggi**

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam database. Nilai *support* item diperoleh dengan rumus 1 berikut:

$$Support(A) = \frac{Jumlah_transaksi_mengandung_A}{Total_transaksi} \dots\dots\dots [1]$$

sedangkan nilai *support* dari 2 item diperoleh dari rumus 2 berikut :

$$Support(A, B) = P(A \cap B) = \frac{Jumlah_transaksi_mengandung_A_dan_B}{Total_transaksi} \dots [2]$$

Sebagai contoh, disediakan database dari transaksi belanja pasar swalayan seperti ditunjukkan dalam Tabel 2.1 berikut :

Tabel 2.1 Contoh Transaksi

Transaction	Item Purchased
1	Broccoli, green peppers, corn
2	Asparagus, squash, corn
3	Corn, tomatoes, beans, squash
4	Green peppers, corn, tomatoes, beans
5	Beans, asparagus, broccoli
6	Squash, asparagus, beans, tomatoes
7	Tomatoes, corn
8	Broccoli, tomatoes, green peppers
9	Squash, asparagus, beans
10	Beans, corn
11	Green peppers, broccoli, beans, squash
12	Asparagus, beans, squash
13	Squash, corn, asparagus, beans
14	Corn, green peppers, tomatoes, beans, broccoli

Data tersebut diatas dalam database transaksional biasa direpresentasikan dalam

bentuk seperti tampak pada tabel 2.2 berikut :

Tabel 2.2 Representasi Data Transaksi dalam Database Transaksional

Transaction	Items
1	Broccoli
1	Green peppers
1	Corn
2	Asparagus
2	Squash
:	:
:	:

Misalkan D adalah himpunan transaksi yang direpresentasikan dalam tabel 2.1, dimana masing-masing transaksi T dalam D merepresentasikan himpunan *item* yang berada dalam I . Misalkan kita memiliki himpunan *items* A (misal beans dan squash) dan himpunan item lain B (misal asparagus). Kemudian aturan asosiasi akan berbentuk :

Jika A , maka B ($A \rightarrow B$)

Dimana *antecedent* A dan *consequent* B merupakan *subset* dari I , dan A dan B merupakan *mutually exclusive*. Definisi ini tidak berlaku untuk aturan *trivial* seperti :

Jika Beans dan Squash maka Beans

Sebuah *itemset* adalah himpunan item-item yang ada dalam I , dan *kitemset* adalah *itemset* yang berisi k item. Misalnya {squash, beans} adalah sebuah *2-itemset* dan {broccoli, green peppers, corn} merupakan *3-itemset*. *Frequent Itemset* menunjukkan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang telah ditentukan (ϕ). Misalkan $\phi = 4$, maka semua *itemset* yang frekuensi kemunculannya lebih dari 4 kali disebut *frequent*. Himpunan dari *frequent k-itemset* dilambangkan dengan F_k .

Tabel 2.3 berikut ini menunjukkan calon *2-itemset* dari data transaksi pada Tabel 2.1

Tabel 2.3 Calon 2-*itemset*

Combination	Count	Combination	Count
Asparagus, beans	5	Broccoli, corn	2
Asparagus, broccoli	1	Broccoli, green peppers	4
Asparagus, corn	2	Broccoli, squash	1
Asparagus, green peppers	0	Broccoli, tomatoes	2
Asparagus, squash	5	Corn, green peppers	3
Asparagus, tomatoes	1	Corn, squash	3
Beans, broccoli	3	Corn, tomatoes	4
Beans, corn	5	Green peppers, squash	1
Beans, green peppers	3	Green peppers, tomatoes	3
Beans, squash	6	Squash, tomatoes	2
Beans, tomatoes	4		

Dari data tersebut diatas, jika ditetapkan nilai $\phi = 4$ maka

$$F2 = \{ \{ \text{asparagus, beans} \}, \{ \text{asparagus, squash} \}, \{ \text{beans, corn} \}, \{ \text{beans, squash} \}, \\ \text{beans, tomatoes} \}, \{ \text{broccoli, green peppers} \}, \{ \text{corn, tomatoes} \} \}$$

Kombinasi dari *itemset* dalam F2 dapat kita gabungkan menjadi calon 3-*itemset*.

Itemset-itemset dari F2 yang dapat digabungkan adalah *itemset-itemset* yang memiliki kesamaan dalam $k-1$ item pertama. Calon 3-*itemset* yang dapat dibentuk dari F2 seperti tampak pada Tabel 2.4 berikut:

Tabel 2.4 Calon 3-*itemset*

Combination	Count
asparagus, beans, squash	4
beans, corn, squash	2
beans, corn, tomatoes	2
beans, squash, tomatoes	2

Dengan demikian $F3 = \{ \{ \text{asparagus, beans, squash} \} \}$, karena hanya kombinasi inilah yang memiliki frekuensi kemunculan $\geq \phi$.

b. **Pembentukan aturan asosiatif**

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif $A \rightarrow B$.

Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dari rumus 3 berikut:

$$Confidence = P(B|A) = \frac{Jumlah_transaksi_mengandung_A_dan_B}{Jumlah_transaksi_mengandung_A} \dots\dots[3]$$

Dari F3 yang telah ditemukan, dapat dilihat besarnya nilai *support* dan *confidence* dari calon aturan asosiasi seperti tampak pada Tabel 2.5 berikut:

Tabel 2.5 Calon Aturan Asosiasi dari F3

<i>If Antecedent, then Consequent</i>	<i>Support</i>	<i>Confidence</i>
If buy asparagus, then buy squash	4/14 = 28.6 %	4/5 = 80 %
If buy asparagus and squash, then buy beans	4/14 = 28.6 %	4/5 = 80 %
If buy beans and squash, then buy asparagus	4/14 = 28.6 %	4/6 = 66.7 %

Misalkan ditetapkan nilai *confidence* minimal adalah 80% maka aturan yang bisa terbentuk adalah aturan dengan dua *antecedent* berikut:

If buy asparagus, then buy squash

If buy asparagus and squash, then buy beans

Sementara itu calon aturan asosiasi dari F2 bisa dilihat pada Tabel 2.6 berikut:

Tabel 2.6 Calon Aturan Asosiasi dari F2

<i>If Antecedent, then Consequent</i>	<i>Support</i>	<i>Confidence</i>
If buy asparagus, the buy beans	5/14 = 35.7 %	5/6 = 83.3 %
If buy beans, then buy asparagus	5/14 = 35.7 %	5/10 = 50 %
If buy asparagus, then buy squash	5/14 = 35.7 %	5/6 = 83.3 %
If buy squash, then buy asparagus	5/14 = 35.7 %	5/7 = 71.4 %
If buy beans, then buy corn	5/14 = 35.7 %	5/10 = 50 %
If buy corn, then buy beans	5/14 = 35.7 %	5/8 = 62.5 %
If buy beans, then buy squash	6/14 = 42.9 %	6/10 = 60 %

<i>If Antecedent, then Consequent</i>	<i>Support</i>	<i>Confidence</i>
If buy squash, then buy beans	6/14 = 42.9 %	6/7 = 85.7 %
If buy beans, then buy tomatoes	4/14 = 28.6 %	4/10 = 40 %
If buy tomatoes, then buy beans	4/14 = 28.6 %	4/6 = 66.7 %
If buy broccoli, then buy green peppers	4/14 = 28.6 %	4/5 = 80 %
If buy green peppers, then buy broccoli	4/14 = 28.6 %	4/5 = 80 %
If buy corn, then buy tomatoes	4/14 = 28.6 %	4/8 = 50 %
If buy tomatoes, then buy corn	4/14 = 28.6 %	4/6 = 66.7 %

Algoritma Apriori

Algoritma *Apriori* adalah algoritma klasik untuk belajar aturan asosiasi. *Apriori* dirancang untuk beroperasi pada database yang berisi transaksi (misalnya, koleksi item yang dibeli oleh pelanggan, atau rincian dari sebuah frekuensi kunjungan website). Logika *Apriori* akan diuraikan berikut ini.

- Jika (A, B, C) adalah sebuah *set item* yang besar, maka (A, B) adalah sebuah *set item* yang besar
 - $P\{A,B,C\} < P\{A,B\}$
 - Jika (A, B) adalah *set item* yang kecil, maka (A, B, C) tidak dapat menjadi *set item* yang besar
- Pertimbangkan (A, B, C) hanya jika semua *subsetnya* adalah *set item* yang besar.

Jika ada transaksi pembelian di mana seseorang berbelanja (susu bayi, popok bayi, selai kacang) maka sudah pasti dapat disimpulkan bahwa daftar belanjanya mengandung (susu bayi, popok bayi). Dengan logika *Apriori* didapat data dari toko atau supermarkets bahwa ternyata elemen (susu bayi, popok bayi) mempunyai frekuensi cukup tinggi dibeli bersamaan.

Jadi secara mendasar logika *Apriori* dan teknik *Association Rule* adalah suatu pasangan teknik untuk mencapai tingkat akurasi data yang tinggi dan teknik pengolahan

data yang tersusun secara sistematis dan dapat dipetakan secara algoritma komputasi (Agrawal, Srikant, 1994).

Algoritma *Apriori* pada program aplikasi :

1. $L_1 = \{ \text{large 1-itemsets} \}, k = 1$
2. Pembentukan *set* kandidat

Set kandidat C_k : *set item* besar ke- k yang potensial .

$\{A, B, C\}$ adalah *set* kandidat jika semua *subsetnya* $\{A, B\}$, $\{B, C\}$ dan $\{A, C\}$

adalah *set item* yang besar. Bentuk *set* kandidat C_{k+1} menggunakan L_k

3. *Scanning*

Memeriksa apakah *set* kandidat adalah benar-benar besar.

4. Tambahkan k dengan 1, dan ulangi langkah ke-2

Gambar 2.6 menunjukkan *pseudocode* yang dilakukan algoritma *Apriori* untuk menghasilkan aturan asosiasi.

```

Apriori( $T, \epsilon$ )
 $L_1 \leftarrow \{ \text{large 1-itemsets that appear in more than } \epsilon \text{ transactions} \}$ 
 $k \leftarrow 2$ 
while  $L_{k-1} \neq \emptyset$ 
     $C_k \leftarrow \text{Generate}(L_{k-1})$ 
    for transactions  $t \in T$ 
         $C_t \leftarrow \text{Subset}(C_k, t)$ 
        for candidates  $c \in C_t$ 
             $\text{count}[c] \leftarrow \text{count}[c] + 1$ 
     $L_k \leftarrow \{ c \in C_k \mid \text{count}[c] \geq \epsilon \}$ 
     $k \leftarrow k + 1$ 
return  $\bigcup_k L_k$ 

```

Gambar 2.6 Pseudocode Apriori

2.1.7 Predictive Analysis

Analisa prediktif adalah suatu teknologi yang *capture* proses *data mining* menjadi lebih sederhana. Analisa prediktif disebut sebagai "*one-click data mining*".

Analisa prediktif menganalisa data *input* yang merupakan data *historical* dan membuat model *mining* dengan mencari algoritma yang paling sesuai dari data yang *diinput*. Model-model ini *ditraining* dan dites kemudian digunakan untuk mencari faktor yang menentukan hasil yang akurat kemudian memprediksi hasil yang paling mungkin terjadi dan mengidentifikasi keakuratan hasil prediksi tersebut. Tabel 2.7 merupakan penjelasan operasi dari analisa prediktif *Oracle* :

Tabel 2.7 Oracle Predictive Analytics Operations

Operasi	Keterangan
EXPLAIN	Menjelaskan bagaimana suatu atribut mempengaruhi variasi dari hasil pada <i>target column</i>
PREDICT	Memprediksi hasil dari <i>target column</i>
PROFILE	Membuat rule untuk kasus-kasus yang sama dengan <i>target value</i>

Berikut ini merupakan contoh prediksi pelanggan yang akan berbelanja lebih banyak dengan kartu kreditnya. Data pelanggan seperti *history* pemakaian kartu kredit, jenis kelamin, pendidikan dan status pernikahan yang disimpan dalam sebuah view dengan nama `MINING_DATA_APPLY_V`. Berikut ini query DMX yang akan dijalankan :

```

DECLARE
p_accuracy NUMBER(10,9);
BEGIN
  DBMS_PREDICTIVE_ANALYTICS.PREDICT(
    accuracy          => p_accuracy,
    data_table_name   => 'mining_data_apply_v',
    case_id_column_name => 'cust_id',
    target_column_name => 'affinity_card',
    result_table_name => 'p_result_tbl');
  DBMS_OUTPUT.PUT_LINE('Accuracy: ' || p_accuracy);
END;
/
Accuracy: .492433267

```

Berikut ini query untuk menampilkan jenis kelamin dan usia rata-rata pelanggan yang akan berbelanja lebih banyak dengan kartu kreditnya :

```

SELECT cust_gender, COUNT(*) as cnt, ROUND(AVG(age)) as avg_age
FROM mining_data_apply_v a, p_result_tbl b
WHERE a.cust_id = b.cust_id AND b.prediction = 1
GROUP BY a.cust_gender
ORDER BY a.cust_gender;

```

Dari query di atas dihasilkan usia rata-rata wanita dan pria yang akan berbelanja lebih banyak dengan kartu kredit adalah 45 tahun.

C	CNT	AVG_AGE
-	-----	-----
F	90	45
M	443	45

2.2 Teori – Teori Khusus yang Berhubungan dengan Topik yang Dibahas

2.2.1 *Customer Relationship Management (CRM)*

2.2.1.1 Pengertian CRM

Berikut ini adalah kutipan beberapa penjelasan tentang *Customer Relationship Management (CRM)* :

- CRM adalah sebuah proses yang mengelola interaksi antara perusahaan dan pelanggannya (Berson, 2000, p10).
- CRM dimaksudkan untuk menciptakan dan mengatur lingkungan dinamis dalam usaha meningkatkan hubungan dan interaksi perusahaan dengan pelanggan secara *continue*. Tugas utamanya adalah mengenali karakter pelanggan (Beck dan Summer, 2001, p1-p2).

Jadi dapat disimpulkan bahwa CRM adalah strategi bisnis yang didukung oleh teknologi dengan pendekatan layanan pelanggan untuk membangun hubungan jangka panjang sehingga memberikan nilai tambah bagi keduanya, baik perusahaan maupun pelanggan.

2.2.1.2 Fase CRM

Tujuan dari CRM adalah untuk mengoptimasi profitabilitas perusahaan dengan pelanggan, melalui pengenalan *if, what* dan *how* dari pelanggan yang berinteraksi dengan perusahaan, sehingga dapat menemukan cara untuk memperoleh kesetiaan pelanggan, yang nantinya diasosiasikan dengan meningkatnya profitabilitas yang diperoleh perusahaan.

Menurut Kalakota dan Robinson (2001, p174-175) ada tiga fase CRM, yaitu:

- Mendapatkan pelanggan baru (*Acquire*)

Mendapatkan pelanggan baru dilakukan dengan mengajukan kepemimpinan prosuk/jasa yang mendorong batasan kinerja mengenai kenyamanan dan inovasi.

Nilai yang diusulkan untuk pelanggan adalah penawaran produk yang *superior* dengan ditopang oleh layanan yang unggul.

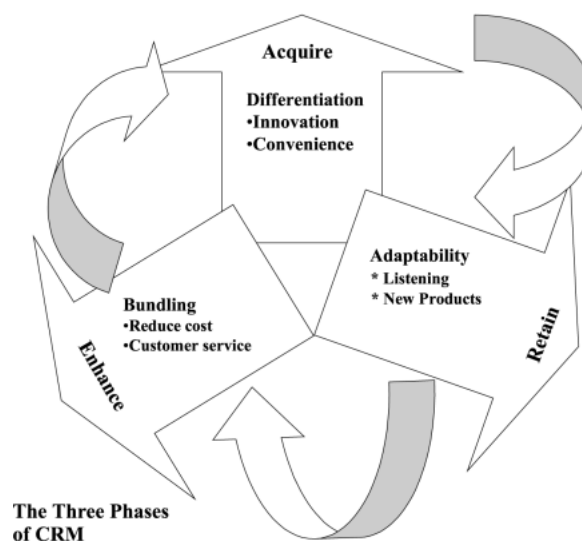
- Mempertahankan pelanggan untuk selamanya (*Retain*)

Penyimpanan informasi pelanggan berfokus pada kemampuan beradaptasi dalam pelayanan, mengirimkan bukan apa yang diinginkan pasar, namun apa yang diinginkan pelanggan. Sekarang, lebih banyak perusahaan yang berfokus pada strategi mempertahankan pelanggan dari pada menarik pelanggan baru. Alasan dibalik strategi ini sebenarnya sederhana : *If you want to make money, hold onto your good customers.*

- Meningkatkan nilai transaksi pelanggan (*Enhance*)

Meningkatkan profitabilitas pelanggan atau hubungan yang telah ada dapat dilakukan dengan cara mendorong keunggulan dalam *cross-selling* dan *up-selling*, hal ini menambah kedalaman hubungan. Nilai yang diusulkan untuk pelanggan adalah penawaran kenyamanan yang lebih baik dengan biaya yang rendah.

Gambar 2.7 menggambarkan ketiga fase CRM tersebut.

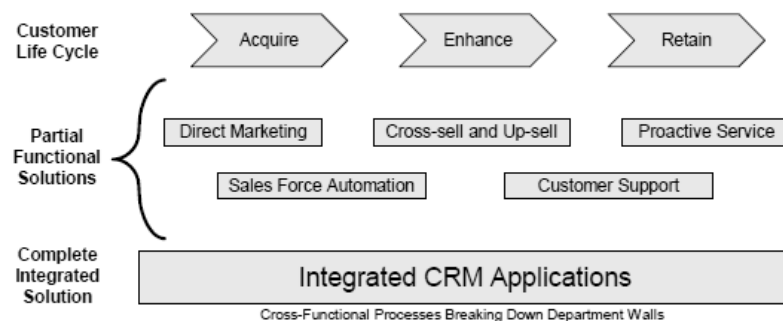


Gambar 2.7 *The Three Phases* of CRM

Sumber : Kalakota dan Robinson (2001, p 175)

2.2.1.3 Arsitektur CRM

Sebuah arsitektur CRM generasi baru mampu dengan baik mengintegrasikan proses pelayanan terhadap pelanggan serta mampu mengolah daur hidup dan interaksi pelanggan dengan perusahaan. Bentuk arsitektur CRM yang terintegrasi tampak apada Gambar 2.8. Gambar tersebut menjelaskan bahwa arsitektur CRM adalah sebuah portofolio dari kompetensi proses silang fungsional yang dibutuhkan untuk menciptakan hubungan dengan pelanggan yang sukses.



Gambar 2. 8 *Integrated CRM*

Sumber : Kalakota dan Robinson, 2001, p179

2.2.1.4 Tipe– Tipe CRM

Menurut Turban et al, (2004, p148), aktivitas CRM dibagi menjadi tiga tipe, antara lain :

- *Operational CRM*, terkait fungsi bisnis tertentu yang meliputi layanan pelanggan, manajemen pemesanan, penagihan, atau otomatisasi dan manajemen penjualan serta pemasaran.
- *Analytical CRM*, mencakup aktivitas mengumpulkan, menyimpan, menggali, mengolah, menginterpretasikan, dan melaporkan data pelanggan kepada pengguna yang akan menganalisanya sesuai dengan yang diperlukan.

- *Collaborative CRM*, mencakup semua kebutuhan komunikasi, koordinasi, dan kolaborasi antara penjual dan pelanggan.

CRM yang terdiri dari ketiga komponen diatas memiliki tujuan secara umum yaitu untuk memaksimalkan keuntungan pelanggan sejalan dengan meningkatkan dan memelihara kepuasan pelanggan.

2.2.2 Analytical CRM

Tantangan bagi perusahaan sekarang ini adalah untuk mengerti apa yang menjadi permintaan pelanggan dan memberikan respon, secara lebih baik, mengantisipasi kebutuhan mereka. Namun, banyak perusahaan hanya berfokus untuk mengimplementasikan penggunaan CRM hanya sebatas untuk “mendengar” kebutuhan dan keluhan dari pelanggan, yang pada praktisnya hanya terkonsentrasi pada komponen *operational* dan *collaborative CRM*, sehingga analisis pemahaman dan pengenalan terhadap pelanggan yang sebenarnya belum terjadi. Diperlukan komponen *Analytical CRM* untuk mengoptimisasi hubungan perusahaan dengan pelanggannya.

Analytical CRM memungkinkan perusahaan untuk mendapatkan suatu pengetahuan akan pelanggan dan melakukan penaksiran atau estimasi terhadap pelanggan berdasarkan data-data analisis yang dipakai. Pengetahuan yang didapat perusahaan adalah tentang nilai pelanggan, yang dapat mendukung *channel* pelayanan interaksi dengan lebih baik dan mendukung berbagai keputusan dalam mensinergi penerapan *operational* dan atau *collaborative CRM* dalam proses penjualan dan pemasaran produk, sehingga kedudukannya menjadi sangat esensial bagi komponen CRM lainnya (Beck dan Summer , 2001, p2).

2.2.2.1 Pengertian *Analytical CRM*

Analytical CRM merupakan *feedback loop* antara interaksi pelanggan yang *real time* yang terjadi pada *front-end* atau *back-end scorecard* untuk menganalisis apa yang telah terjadi dan bagaimana meningkatkan cara perusahaan berinteraksi di kemudian hari. Beck dan Summer (2001, p5) mengatakan bahwa *analytical CRM* adalah penggunaan data pelanggan untuk analisis, pemodelan, dan evaluasi yang ditujukan untuk mendukung perusahaan untuk membentuk suatu hubungan yang *profitable* antara perusahaan dan pelanggannya. *Analytical CRM* terdiri dari semua *programming* yang menganalisis data tentang pelanggan perusahaan, sehingga keputusan yang lebih cepat dan lebih baik dapat dihasilkan (Anonim, 2001, p1-p2). *Analytical CRM* dapat menyediakan hal-hal sebagai berikut :

- Kelompok segmentasi pelanggan
- Analisis profitabilitas, mengetahui pelanggan mana yang *profitable* selama kurun waktu tertentu.
- Personalisasi, kemampuan untuk memasarkan kepada pelanggan secara personal berdasarkan data-data yang ada tentang pelanggan tersebut.
- *Event monitoring*, yaitu segala aktivitas yang dilakukan pelanggan yang berkaitan dengan perusahaan dan begitu pula sebaliknya tindakan perusahaan terhadap segala yang berhubungan dengan pelanggannya, seperti promosi, transaksi, dsb.

Analytical CRM mampu melakukan segmentasi pelanggan, membedakan pelanggan yang memiliki profitabilitas tinggi dengan tingkat akurasi yang lebih matang, maka *Return On Investment (ROI)* akan suatu pelanggan dapat diprediksi dengan baik.

(Meta Group, 2000, p 1). Dengan pengetahuan tersebut, penawaran yang tepat dan harga yang tepat dapat ditawarkan pada saat yang tepat, kepada pelanggan yang memang berpotensi untuk membelinya. Hal ini akan mengoptimalkan pelanggan dan perusahaan. Dengan kombinasi informasi tentang pelanggan dari semua sumber dan sarana informasi vital lainnya yang berinteraksi dengan pelanggan, perusahaan dapat memperoleh gambaran yang pasti tentang pelanggan dan perilakunya. Dengan demikian dukungan perusahaan dalam melakukan personalisasi terhadap pelanggan dan menyesuaikan diri seiring dengan kemungkinan perubahan-perubahan yang terjadi dari permintaan pelanggan.

2.2.2.2 Perbedaan *Operational CRM* dengan *Analytical CRM*

Tabel 2.8 berikut ini menjelaskan perbedaan *operational CRM* dan *analytical CRM*.

Tabel 2.8 Perbedaan *Operational CRM* dan *Analytical CRM*

Sumber : <http://www.ebizasia.com/0217-2004/enterprise>

<i>OPERATIONAL CRM</i>	<i>ANALYTICAL CRM</i>
Fokus pada transaksi yang menguntungkan	Fokus pada nilai jangka panjang pelanggan
Menekankan pada akuisisi pelanggan	Menekankan pada mempertahankan pelanggan
Mengukur kepuasan pelanggan	Mengukur nilai pelanggan dan loyalitas
Terorganisasi berdasarkan fungsi dan unit produk	Terorganisasi berdasarkan segmentasi pelanggan
Bergantung pada informasi mengenai pelanggan	Bergantung pada informasi dari pelanggan
Interaksi proaktif dengan pelanggan	Interaksi personal seketika (<i>real-time</i>) dengan pelanggan
Dalam hal peningkatan, fokus ke dalam perusahaan	Dalam hal peningkatan, focus ke luar, pelanggan
Penerapan dan pembelajaran jangka panjang (<i>long-loop</i>)	Penerapan dan pembelajaran jangka pendek (<i>short-loop</i>)

2.2.3 Sistem Distribusi

2.2.3.1 Saluran Distribusi

Menurut Yunarto (2006, p42) menyatakan bahwa dalam saluran distribusi dikenal tiga komponen utama yaitu *Intermediary* (perantara), *Agent* (agen), dan *Facilitator* (fasilitator):

a. *Intermediary* (perantara)

Adalah pihak-pihak seperti *wholesaler* (grosir/pedagang besar) dan *retailer* (pengecer) yang membeli barang, memilikinya dan menjual kembali barang tersebut. *Wholesaler* dan *retailer* sering disebut juga dengan istilah *merchant* (pedagang)

b. *Agent* (agen)

Adalah pihak-pihak seperti *broker* (pedagang perantara yang biayanya dibayar dengan imbalan komisi) dan sales agent (agen penjualan). *Broker* dan *agent* akan mencari pembeli, bertindak dipihak penjual, negosiasi dengan pembeli, tetapi tidak memiliki barang yang diperantarakan atau diperdagangkan

c. *Facilitator* (fasilitator)

Adalah pihak-pihak lain yang memfasilitasi atau membantu proses distribusi dalam hal pengiriman barang secara fisik, pengiriman informasi, ataupun proses pembayaran. Fasilitator adalah pihak ketiga yang tidak terlibat proses jual beli barang dan tidak memiliki barang yang dikirim atau diperdagangkan tersebut.

Menurut Sastradipoera (2003, p172), sedikitnya ada tiga arti saluran distribusi (*channel of distribution*). Ketiga definisi saluran distribusi itu adalah seperti berikut:

1. Saluran distribusi adalah saluran yang dipergunakan untuk dilewati oleh arus pemilikan (*flow of title*) atas barang atau jasa yang diperjualbelikan.

2. Saluran distribusi adalah suatu gabungan lembaga sebagai tempat yang dilalui oleh penjual dalam proses pemilikan ketika penjual itu menjajakan barang-barangnya hingga tiba di tangan pemakai atau konsumen.
3. Saluran distribusi adalah suatu jaringan organisasi yang menata perubahan-perubahan dalam pemilikan atas barang-barang karena barang-barang itu bergerak dari pabrikan kepada konsumen.

2.2.3.2 Fungsi Saluran Distribusi

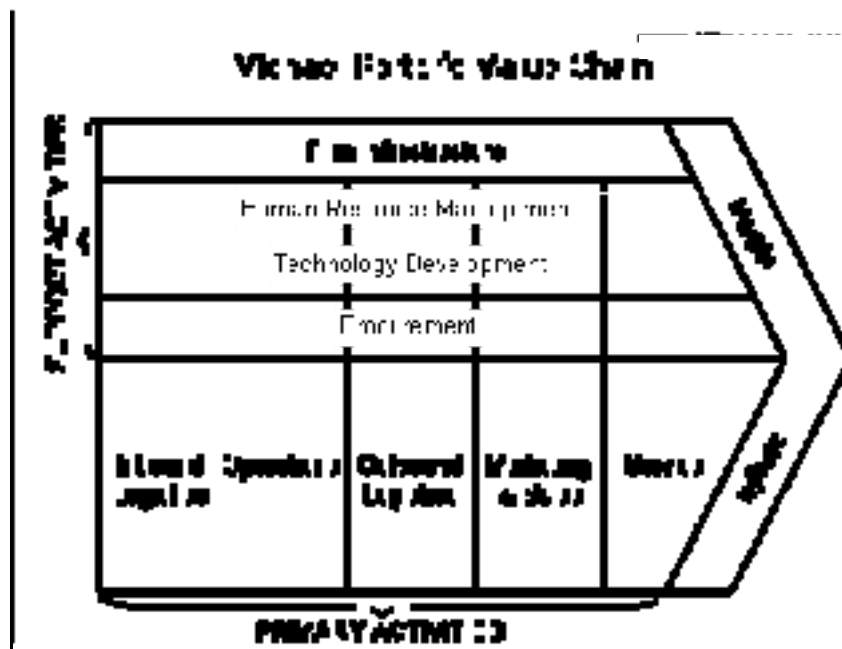
Menurut Kotler (2005, p183), Anggota-anggota saluran pemasaran melaksanakan sejumlah fungsi utama:

- Mereka mengumpulkan informasi mengenai pelanggan, pesaing, dan pelaku dan kekuatan lainnya dalam lingkungan pemasaran tersebut.
- Mereka mengembangkan dan menyebarkan komunikasi persuasive untuk merangsang pembelian.
- Mereka mencapai kesepakatan mengenai harga dan ketentuan-ketentuan lain sehingga peralihan kepemilikan dapat terlaksana.
- Mereka melakukan pemesanan kepada produsen.
- Mereka memperoleh dana untuk membiayai persediaan pada tingkat yang berbeda dalam saluran pemasaran.
- Mereka menanggung risiko yang berhubungan dengan fungsi saluran.
- Mereka mengatur kesinambungan penyimpanan dan perpindahan produk fisik.
- Mereka mengatur pelunasan tagihan mereka kepada pembeli melalui bank dan lembaga keuangan lainnya.

- Mereka mengawasi peralihan kepemilikan aktual dari suatu organisasi atau orang kepada organisasi atau orang lainnya.

2.2.4 Value Chain Analysis

Value Chain merupakan model yang ditemukan oleh Michael Porter yang digunakan untuk membantu menganalisis aktivitas-aktivitas spesifik yang dapat menciptakan nilai dan keuntungan kompetitif bagi organisasi.



Gambar 2.9 Rantai Nilai Porter

Sumber : http://www.12manage.com/images/picture_porter_value_chain.gif

Rantai nilai menggambarkan nilai total, dan terdiri atas aktivitas nilai (*value activities*) dan margin. Aktivitas nilai adalah kegiatan fisik dan teknologis yang diselenggarakan perusahaan. Ini merupakan batu-batu pembangunan (*building blocks*) pembelinya. Margin adalah selisih antara nilai total dengan biaya kolektif untuk menyelenggarakan aktivitas nilai. Margin dapat diukur dengan berbagai cara. Rantai nilai pemasok dan penyalur juga mengandung suatu margin yang penting untuk dikenali guna memahami sumber posisi biaya perusahaan, karena margin pemasok dan penyalur merupakan bagian dari biaya total yang dipikul pembeli.

Setiap aktivitas nilai menggunakan masukan yang dibeli, sumber daya manusia (tenaga kerja dan manajemen), dan bentuk teknologi tertentu untuk menyelenggarakan fungsinya. Setiap aktivitas nilai juga menggunakan dan menciptakan informasi, seperti data pembeli (pesanan yang masuk), parameter kinerja (pengujian), dan statistik kegagalan produk. Aktivitas nilai mungkin juga menghasilkan kekayaan (*asset*) seperti persediaan dan piutang, atau kewajiban, misalnya utang lancar.

Aktivitas nilai dapat dibagi ke dalam dua golongan besar, aktivitas primer dan aktivitas pendukung. Aktivitas primer, tercantum dibagian bawah Gambar 2.8, merupakan aktivitas yang dilakukan dalam membuat produk secara fisik serta menjual dan menyampaikannya kepada pembeli selain juga aktivitas dalam bentuk layanan purna jual, pada setiap perusahaan, aktivitas primer dapat dibagi menjadi lima kelompok generik seperti tampak pada Gambar 2.9. Aktivitas pendukung menunjang aktivitas primer dan aktivitas pendukung lainnya dengan menyediakan masukan yang dibeli, teknologi, sumber daya manusia, serta sejumlah fungsi dalam perusahaan lainnya. Pembelian, pengembangan teknologi, dan manajemen sumber daya manusia dapat dikaitkan dengan aktivitas primer tertentu selain juga menunjang keseluruhan rantai. Infrastruktur perusahaan tidak terkait dengan aktivitas primer tertentu tetapi menunjang keseluruhan rantai.

Oleh karena itu, aktivitas nilai merupakan batu-batu pembangun keunggulan bersaing. Cara melakukan masing-masing aktivitas dan seberapa efisien melakukannya akan menentukan apakah suatu perusahaan berbiaya tinggi atau rendah dibandingkan dengan pesaingnya. Bagaimana masing-masing aktivitas nilai dilakukan juga akan menentukan kontribusinya terhadap kebutuhan pembeli dan karenanya juga diferensiasi.

Memperbandingkan rantai nilai pesaing akan mengungkapkan perbedaan yang menentukan keunggulan bersaing.

Analisis terhadap rantai nilai dan bukan terhadap nilai tambahlah yang merupakan cara yang tepat untuk menelaah keunggulan bersaing. Nilai tambah (harga jual dikurangi dengan biaya bahan baku yang dibeli) terkadang digunakan sebagai titik fokus analisis biaya karena dipandang sebagai bidang tempat perusahaan dapat mengendalikan biaya. Tetapi, nilai tambah bukanlah dasar yang layak untuk analisis biaya karena secara keliru membedakan bahan baku.

A. Aktivitas Primer

1. *Inbound logistic*

Aktivitas yang berhubungan dengan penerimaan, penyimpanan, dan penyebaran masukan ke produk, seperti penanganan material, pergudangan, pengendalian persediaan, penjadwalan kendaraan pengangkut, dan pengembalian barang kepada pemasok.

2. *Operations*

Aktivitas yang menyangkut perubahan masukan menjadi produk akhir, seperti masinasi, pengemasan, perakitan, pemeliharaan alat-alat, pengujian, pencetakan, dan pengoperasian fasilitas.

3. *Outbound logistics*

Aktivitas yang berhubungan dengan pengumpulan, penyimpanan, dan pendistribusian fisik produk kepada pembeli, seperti pergudangan barang jadi, penanganan material, operasi kendaraan pengirim, pengolahan pesanan, dan penjadwalan.

4. *Marketing and sales*

Aktivitas yang menyangkut penyediaan sarana agar pembeli dapat membeli produk dan aktivitas yang mempengaruhi pembeli agar mereka mau membelinya, seperti periklanan, promosi, wiraniaga, penentuan kuota, pemilihan penyalur, hubungan dengan penyalur, dan penetapan harga.

5. *Service*

Aktivitas yang menyangkut penyediaan layanan untuk memperkuat atau menjaga nilai produk, seperti pemasangan, perbaikan, pelatihan, pasokan suku cadang, dan penyesuaian produk.

B. Aktivitas Pendukung

1. *Procurement*

Pembelian mengacu pada fungsi pembelian masukan yang digunakan dalam rantai nilai perusahaan, bukan pada masukan yang dibeli itu sendiri. Masukan yang dibeli meliputi bahan baku, bahan pendukung, serta bahan-bahan lain selain juga barang modal seperti mesin, peralatan laboratorium, peralatan kantor, dan bangunan.

2. *Human Resources Management*

Manajemen sumber daya manusia terdiri atas beberapa aktivitas yang meliputi perekrutan, penerimaan, pelatihan, pengembangan, dan kompensasi untuk semua jenis tenaga kerja. Manajemen sumber daya manusia mendukung aktivitas nilai primer dan pendukung secara individual maupun keseluruhan rantai nilai. Manajemen sumber daya manusia mempengaruhi keunggulan bersaing pada setiap perusahaan, melalui perannya dalam menentukan ketrampilan dan motivasi karyawan serta biaya penerimaan dan pelatihan karyawan.

3. *Technological Development*

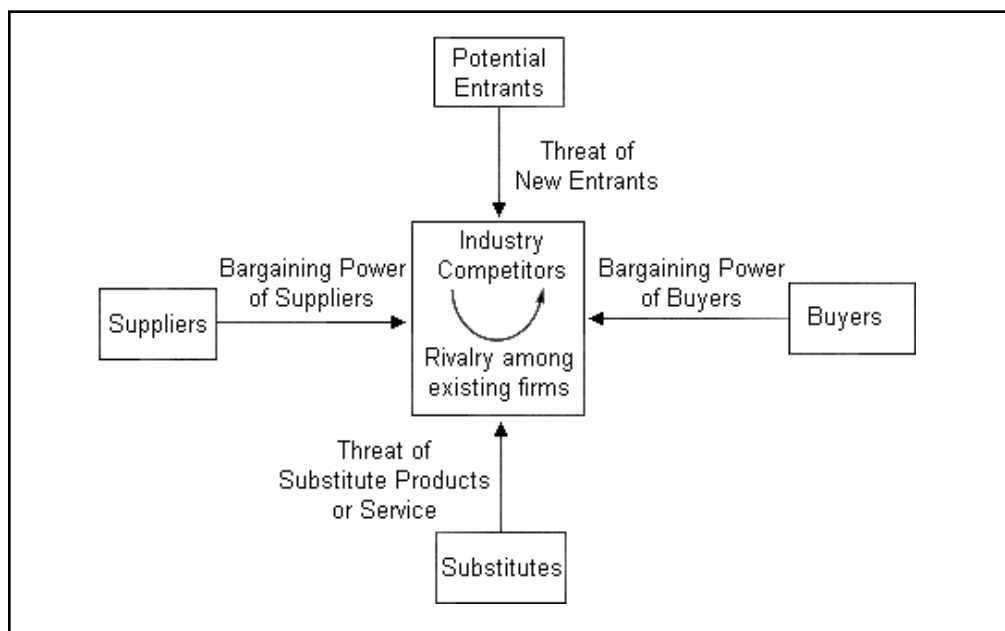
Setiap aktivitas nilai mengandung teknologi, apakah itu berupa pengetahuan, prosedur, atau teknologi yang terlekat dalam peralatan proses. Ragam teknologi yang digunakan pada banyak perusahaan sangat luas, mulai dari teknologi yang digunakan dalam menyiapkan dokumen dan mengangkut barang sampai ke teknologi yang terlekat dalam produk yang dihasilkan itu sendiri. Lebih jauh lagi, kebanyakan aktivitas nilai menggunakan teknologi yang menggabungkan sejumlah subteknologi berbeda yang melibatkan berbagai disiplin ilmu.

4. *Infrastructure*

Infrastruktur perusahaan terdiri atas sejumlah aktivitas yang meliputi manajemen umum, perencanaan, keuangan, akuntansi, hukum, hubungan dengan pemerintah, dan manajemen mutu. Infrastruktur, tidak seperti aktivitas pendukung lainnya, biasanya menunjang keseluruhan rantai dan bukan aktivitas tertentu.

2.2.5 Analisis Porter

David (2006, p130-135) mengutip Professor Harvard Michael E. Porter menyatakan Analisis Porter: *The Five Forces of Competition Analysis* digunakan untuk menganalisis daya tarik dari suatu industri. Adapun pengertian industri itu sendiri adalah sekumpulan pasar yang terdiri dari berbagai pasar usaha yang memiliki model bisnis yang sejenis namun berbeda fokus usaha di masing-masing pasar ini.



Gambar 2.10 Porter's Five Forces Model

Sumber : <http://faculty.css.edu/dswenson/web/525ARTIC/porter5forces.gif>

Saat mempelajari suatu persaingan, perusahaan harus berkonsentrasi untuk dapat bersaing secara langsung. Namun demikian, perusahaan harus mencari dan mengidentifikasi pola potensial para pesaingnya dengan melakukan hal yang sama untuk *customer potential* mereka. Model lima kekuatan tersebut mengenali bahwa *supplier* dapat menjadi kompetitor.

Dalam dunia industri, faktor persaingan merupakan salah satu elemen yang penting di dalam melakukan proses bisnis yang simultan dan berkelanjutan. Ketatnya persaingan akan menekan perusahaan untuk mampu melakukan terobosan-terobosan baik dari sisi *product resourcing*, *supply resources*, *operational cost*, dan lainnya sehingga memberikan nilai tambah (*added value*) untuk mengalahkan pesaingnya. Analisis kompetitor Michael Porter seperti pada gambar 2.10 di atas meliputi:

1. Ancaman masuknya pendatang baru (*Potential New Entrants*)

Pendatang baru dalam industri biasanya dapat mengancam pesaing yang ada. Hal ini disebabkan karena pendatang baru seringkali membawa kapasitas baru, keinginan untuk merebut pangsa pasar, serta seringkali pula memiliki sumber daya yang besar. Akibatnya harga dapat menjadi turun atau biaya meningkat sehingga mengurangi laba yang dapat dihasilkan oleh organisasi. Selain itu, adanya pendatang baru dapat memaksa perusahaan yang sudah ada untuk lebih efektif dan efisien serta belajar untuk bersaing dalam dimensi baru.

Secara sederhana, kemungkinan perusahaan akan memasuki suatu industri adalah fungsi dari dua faktor, yaitu: hambatan memasuki industri dan reaksi dari perusahaan yang sudah ada. Apabila hambatan-hambatan untuk masuk adalah tinggi dan pendatang baru mendapatkan reaksi yang tajam dari pemain lama dalam industri, sudah pasti pendatang baru tersebut tidak menimbulkan suatu ancaman masuk yang serius.

Terdapat beberapa hambatan untuk memasuki industri (*entry barriers*). Diantara adalah sebagai berikut:

a. Skala Ekonomi (*Economies of Scale*)

Skala ekonomi adalah bertambahnya jumlah barang yang diproduksi dalam suatu periode sehingga mengakibatkan biaya produksi per unit menjadi turun.

b. Diferensiasi Produk (*Product Differentiation*)

Diferensiasi produk artinya perusahaan tertentu mempunyai identifikasi merek dan loyalitas pelanggan, yang disebabkan oleh periklanan, pelayanan pelanggan, perbedaan produk di masa lampau, atau sekedar merupakan perusahaan pertama yang memasuki industri.

c. Persyaratan Modal (*Capital Requirement*)

Kebutuhan untuk melakukan investasi sumber daya keuangan yang besar agar dapat bersaing menciptakan suatu hambatan masuk, khususnya apabila modal yang diperlukan untuk pengeluaran tidak dapat diterima kembali.

d. Biaya Peralihan Pemasok (*Switching Cost*)

Biaya yang harus dikeluarkan pembeli bilamana berpindah dari produk pemasok tertentu ke produk pemasok lainnya, seperti: biaya pelatihan karyawan, biaya untuk mendesain ulang produk, dan sebagainya.

e. Akses ke saluran distribusi

Hambatan masuk dapat ditimbulkan dengan adanya kebutuhan dari pendatang baru untuk mengamankan distribusi produknya bilamana saluran distribusi untuk produk tersebut telah ditangani oleh perusahaan yang sudah mapan, perusahaan baru harus membujuk saluran tersebut agar menerima produknya melalui penurunan harga, kerja sama periklanan, dan sebagainya yang tentu saja berimplikasi terhadap turunnya laba.

f. Kebijakan Pemerintah

Pemerintah dapat membatasi atau bahkan menutup masuknya industri dengan melakukan pengendalian dan pengawasan, seperti: perjanjian lisensi dan batasan-batasan pada akses ke bahan baku. Pemerintah juga dapat memainkan peranan tidak langsung seperti: perjanjian lisensi dan batasan-batasan pada akses ke bahan baku. Pemerintah juga dapat

memainkan peranan tidak langsung seperti: standar polusi udara dan peraturan keamanan.

2. Ancaman persaingan diantara perusahaan sejenis (*Rivalry among Competing Sellers*)

Persaingan di kalangan pesaing yang ada berbentuk suatu kompetisi untuk mendapatkan keunggulan posisi dengan menggunakan taktik-taktik seperti: persaingan harga, pengenalan produk, perang iklan, dan peningkatan pelayanan atau jaminan kepada pelanggan. Persaingan terjadi karena satu atau lebih pesaing merasakan adanya tekanan atau melihat peluang untuk memperbaiki posisi. Perubahan strategi oleh satu perusahaan akan mendapatkan serangan balasan dari pesaing sejenis. Pada kebanyakan industri, gerakan persaingan oleh satu perusahaan mempunyai pengaruh yang besar terhadap para pesaingnya. Dengan demikian, dapat mendorong perlawanan atau usaha untuk menandingi gerakan tersebut. Intensitas persaingan antar perusahaan merupakan fungsi dari beberapa faktor, seperti:

- Adanya beberapa pesaing yang seimbang
- Pertumbuhan industri yang lambat,
- Kurangnya diferensiasi atau switching cost,
- Pertambahan kapasitas yang tinggi,
- Pesaing yang berbeda-beda, dan
- Hambatan pengunduran diri yang tinggi.

3. Ancaman dari produk pengganti (*Substitute Products*)

Walaupun karakteristiknya berbeda, barang pengganti dapat memberikan fungsi atau jasa yang sama. Acaman dari produk pengganti ini kuat jika konsumen dihadapkan pada sedikitnya biaya peralihan pemasok dan jika produk pengganti tersebut mempunyai harga yang lebih murah atau kualitasnya sama bahkan lebih tinggi dari produk-produk suatu industri.

4. Kekuatan tawar-menawar pembeli (*Buyers*)

Para pembeli biasanya berusaha membeli barang dengan harga termurah yang dapat diperolehnya. Untuk mengurangi biaya mereka, biasanya pembeli meminta kualitas yang lebih tinggi, pelayanan yang lebih baik serta yang lebih penting harga yang lebih murah. Tindakan ini akan menyebabkan persaingan yang kuat diantara perusahaan yang ada dalam suatu industri yang sama.

Biasanya kekuatan tawar-menawar pembeli meningkat jika:

- Pembeli membeli dalam jumlah besar
- Produk yang dibeli adalah produk standar atau tidak terdiferensiasi
- Pembeli memperoleh laba yang rendah
- Produk industri adalah tidak terlalu penting untuk produk atau jasa pembeli

5. Kekuatan tawar-menawar pemasok (*Suppliers*)

Pemasok dapat saja menekan perusahaan yang sudah ada dalam suatu industri dengan cara menaikkan harga serta menurunkan kualitas barang yang dijualnya disaat jumlah pemasok di industri berjumlah sedikit. Jika perusahaan tidak dapat menutupi kenaikan biaya melalui struktur harganya, maka kemampuan untuk menghasilkan laba pada perusahaan tersebut dapat menurun

karena tindakan pemasok tadi. Namun bisa juga seringkali pemasok saling memberikan harga yang masuk akal, memperbaiki kualitas dan mengurangi biaya persediaan di saat jumlah pemasok besar. Dengan demikian akan memperbaiki profitabilitas jangka panjang untuk semua pihak.

Pemasok memiliki kekuatan tawar-menawar jika :

- Didominasi oleh sedikit perusahaan
- Produknya adalah unik
- Industri tersebut bukanlah pelanggan penting dari pemasok

2.2.6 Analisis SWOT

2.2.6.1 Pengertian Analisis SWOT

Pengertian analisis SWOT (Rangkuti, 2006) adalah identifikasi berbagai faktor secara sistematis untuk merumuskan strategi perusahaan. Analisis ini didasarkan pada logika yang dapat memaksimalkan kekuatan (*strengths*) dan peluang (*opportunities*), namun secara bersamaan dapat meminimalkan kelemahan (*weaknesses*) dan ancaman (*threats*). Proses pengambilan keputusan strategis selalu berkaitan dengan pengembangan misi, tujuan, strategi, dan kebijakan perusahaan. Dengan demikian perencana strategis (*strategic planner*) harus menganalisis faktor-faktor strategis perusahaan (kekuatan, kelemahan, peluang, dan ancaman) dalam kondisi yang ada saat ini.

Proses pengambilan keputusan strategis selaku berkaitan dengan pengembangan misi, tujuan, strategi, dan kebijakan perusahaan. Metode analisis SWOT bisa dianggap sebagai metode analisis yang paling dasar, yang berguna untuk melihat suatu topik atau

permasalahan dari empat sisi yg berbeda. Hasil analisis biasanya merupakan arahan atau rekomendasi untuk mempertahankan kekuatan dan menambah keuntungan dari peluang yang ada, sekaligus mengurangi kekurangan dan menghindari ancaman.

Teknik ini diciptakan oleh Albert Humphrey, yang memimpin proyek riset pada Universitas Stanford pada dasawarsa 1960-an dan 1970-an dengan menggunakan data dari perusahaan-perusahaan Fortune 500. Berikut ini adalah penjelasan dari masing-masing komponen SWOT:

- Kekuatan (*Strengths*) adalah kegiatan-kegiatan organisasi yang berjalan dengan baik atau sumber daya yang dapat dikendalikan.
- Kelemahan (*Weakness*) adalah kegiatan-kegiatan organisasi yang tidak berjalan dengan baik atau sumber daya yang dibutuhkan oleh organisasi tetapi tidak dimiliki oleh organisasi.
- Kesempatan (*Opportunities*) adalah faktor-faktor lingkungan luar yang positif.
- Ancaman (*Threats*) adalah faktor-faktor lingkungan luar yang negatif.

2.2.6.2 Matrik SWOT

Matrik SWOT menurut Rangkuti (2006, p31-p32) adalah alat yang dipakai untuk menyusun faktor-faktor strategi perusahaan. Matrik ini dapat menggambarkan secara jelas bagaimana peluang dan ancaman eksternal yang dihadapi perusahaan dapat disesuaikan dengan kekuatan dan kelemahan internal yang dimilikinya. Matrik ini dapat menghasilkan empat kemungkinan strategi, antara lain :

- Strategi SO (*Strength - Opportunity*)

Strategi ini akan menggunakan kekuatan internal perusahaan untuk mencapai peluang-peluang yang ada di perusahaan. Pada umumnya perusahaan berusaha melaksanakan strategi-strategi WO, ST, atau WT untuk menerapkan strategi SO. Dengan demikian, apabila perusahaan memiliki banyak kelemahan, perusahaan harus mengatasi kelemahan yang ada agar menjadi kuat. Sementara jika perusahaan menghadapi banyak ancaman, perusahaan harus berusaha menghindarinya dan berusaha untuk konsentrasi pada peluang yang ada.

- Strategi ST (*Strength – Threat*)

Melalui strategi ini perusahaan berusaha untuk menghindari atau mengurangi dampak dari ancaman-ancaman eksternal.

- Strategi WO (*Weakness – Opportunity*)

Strategi ini bertujuan untuk meminimalkan kelemahan-kelemahan internal perusahaan dengan memanfaatkan peluang-peluang eksternal. Salah satu alternatif untuk mengatasi masalah kesenjangan teknologi ini adalah melalui strategi WO, yakni dengan mengadakan suatu kerja sama (*joint venture*) dengan perusahaan lain yang memiliki kompetensi.

- Strategi WT (*Weakness – Threat*)

Strategi ini merupakan taktik untuk bertahan dengan cara mengurangi kelemahan internal serta menghindari ancaman. Perusahaan dapat bertahan dengan menggunakan strategi-strategi seperti *merger*, *declared bankruptcy*, *retrench*, atau *liquidation*.

Tabel 2.9 Matrik TOWS

Sumber: Rangkuti, 2006, p31

IFAS	Strength (S) Tentukan faktor-faktor kekuatan internal	Weaknesses (W) Tentukan faktor-faktor kelemahan internal
EFAS		
Opportunities (O) Tentukan faktor – faktor peluang eksternal	Strategi SO Ciptakan strategi yang menggunakan kekuatan untuk memanfaatkan peluang	Strategi WO Ciptakan strategi yang meminimalkan kelemahan untuk memanfaatkan peluang
Threats (T) Tentukan faktor – faktor ancaman eksternal	Strategi ST Ciptakan strategi yang menggunakan kekuatan untuk mengatasi ancaman	Strategi WT Ciptakan strategi yang meminimalkan kelemahan dan menghindari ancaman

2.2.6.3 Matrik Faktor Strategi Eksternal

Sebelum membuat penilaian EFAS (*External Strategic Analysis Summary*), faktor strategi eksternal perlu diketahui terlebih dahulu. Berikut ini adalah cara-cara penentuan faktor strategi eksternal (EFAS) (Rangkuti, 2006, p22-p23) :

- Susunlah peluang dan ancaman dalam kolom satu
- Beri bobot masing-masing faktor dalam kolom dua, mulai dari 1,0 (sangat penting) sampai dengan 0,0 (tidak penting). Faktor – faktor tersebut kemungkinan dapat memberikan dampak terhadap faktor strategis
- Hitung *rating* (dalam kolom tiga) untuk masing-masing faktor dengan memberikan skala mulai dari empat (*outstanding*) sampai dengan satu (*poor*) berdasarkan pengaruh faktor tersebut terhadap kondisi yang bersangkutan.

Pemberian nilai *rating* untuk faktor peluang bersifat positif (peluang yang semakin besar diberi *rating* +4, tetapi jika peluangnya kecil, diberi *rating* +1).

Pemberian nilai *rating* ancaman adalah kebalikannya. Misalnya, jika nilai ancamannya sangat besar, *rating*-nya adalah satu. sebaliknya, jika nilai ancamannya sedikit *rating*-nya empat

- Kalikan bobot pada kolom dua dengan *rating* pada kolom tiga, untuk memperoleh faktor pembobotan dalam kolom empat. Hasilnya berupa skor pembobotan untuk masing-masing faktor yang nilainya bervariasi mulai dari 4,0 (*outstanding*) sampai dengan 1,0 (*poor*)
- Jumlahkan skor pembobotan (pada kolom empat), untuk memperoleh total skor pembobotan bagi perusahaan yang bersangkutan.

Tabel 2.10 Penilaian EFAS

Sumber : Rangkuti, 2006, p24

FAKTOR-FAKTOR STRATEGI EKSTERNAL	BOBOT	RATING	BOBOT X RATING
PELUANG: 1. Peluang satu 2. Peluang dua			
TOTAL			
ANCAMAN: 1. Ancaman satu 2. Ancaman dua			
TOTAL			

2.2.6.4 Matrik Faktor Strategi Internal

Sebelum membuat penilaian IFAS (*Internal Strategic Analysis Summary*), faktor strategi internal perlu diketahui terlebih dahulu. Berikut ini adalah cara – cara penentuan faktor strategi internal (IFAS) (Rangkuti, 2006, p24-p26) :

- Tentukan faktor – faktor yang menjadi kekuatan serta kelemahan perusahaan dalam kolom satu
- Beri bobot masing – masing faktor tersebut dengan skala mulai dari 1,0 (paling penting) sampai 0,0 (tidak penting), berdasarkan pengaruh faktor–faktor tersebut terhadap posisi strategis perusahaan. (semua bobot tersebut jumlahnya tidak boleh melebihi skor total 1,00)
- Hitung *rating* (dalam kolom tiga) untuk masing–masing faktor dengan memberikan skala mulai dari empat (*outstanding*) sampai dengan satu (*poor*), berdasarkan pengaruh faktor tersebut terhadap kondisi perubahan yang bersangkutan. Variabel yang bersifat positif (semua variabel yang masuk kategori kekuatan) diberi nilai +1 sampai dengan +4 (sangat baik) dengan membandingkannya dengan rata–rata industri atau dengan pesaing utama, sedangkan variabel yang bersifat negatif, kebalikannya.
- Kalikan bobot pada kolom dua dengan *rating* pada kolom tiga, untuk memperoleh faktor pembobotan dalam kolom empat. Hasilnya berupa skor pembobotan untuk masing–masing faktor yang nilainya bervariasi mulai dari 4,0 (*outstanding*) sampai dengan 1,0 (*poor*)
- Jumlahkan skor pembobotan (pada kolom empat), untuk memperoleh total skor pembobotan bagi perusahaan yang bersangkutan.

Tabel 2.11 Penilaian IFAS

Sumber: Rangkuti, 2006, p25

FAKTOR-FAKTOR STRATEGI INTERNAL	BOBOT	RATING	BOBOT X RATING
KEKUATAN: 1. Kekuatan satu 2. Kekuatan dua			
TOTAL			
KELEMAHAN : 1. Kelemahan satu 2. Kelemahan dua			
TOTAL			

2.2.6.5 Penilaian EFAS dan IFAS

Setelah melakukan perhitungan EFAS dan IFAS, posisi perusahaan dalam diagram analisis SWOT dapat ditentukan. Diagram ini (Gambar 2.11) memiliki beberapa kuadran, yaitu:

- Kuadran satu

Ini merupakan situasi yang sangat menguntungkan. Perusahaan tersebut memiliki peluang dan kekuatan sehingga dapat memanfaatkan peluang yang ada. Strategi yang harus diterapkan dalam kondisi ini adalah mendukung kebijakan pertumbuhan yang agresif (*growth-oriented strategy*).

- Kuadran dua

Meskipun menghadapi berbagai ancaman, perusahaan ini masih memiliki kekuatan dari segi internal. Strategi yang harus diterapkan adalah menggunakan

kekuatan untuk memanfaatkan peluang jangka panjang dengan cara strategi diversifikasi produk atau pasar.

- Kuadran tiga

Perusahaan menghadapi peluang pasar yang sangat besar, tetapi dilain pihak perusahaan menghadapi beberapa kendala/kelemahan internal. Kondisi bisnis pada kuadran tiga ini mirip dengan *Question Mark* pada BCG matrik. Fokus strategi perusahaan ini adalah meminimalkan masalah–masalah internal perusahaan sehingga dapat merebut peluang pasar yang lebih baik

- Kuadran empat

Ini merupakan situasi yang sangat tidak menguntungkan, perusahaan tersebut menghadapi berbagai ancaman dan kelemahan internal.



Gambar 2.11 Diagram Analisis SWOT

Sumber : Rangkuti, 2006, p 19

2.2.7 Critical Succes Factor (CSF)

Critical Success Factor (CSF) adalah beberapa area kritis di dalam perusahaan yang harus berjalan dengan baik sehingga perusahaan dapat berkembang. Faktor–faktor ini menjadi penentu bagi perusahaan dalam mencapai tujuan yang dilakukannya (Tozer, 1996, p41). Sedangkan menurut Martin, *Critical Success Factor* (CSF) adalah sejumlah area terbatas dimana hasil yang memuaskan akan menjamin peforma yang kompetitif untuk suatu individu, departemen atau suatu organisasi. (Martin, 1990, p89)

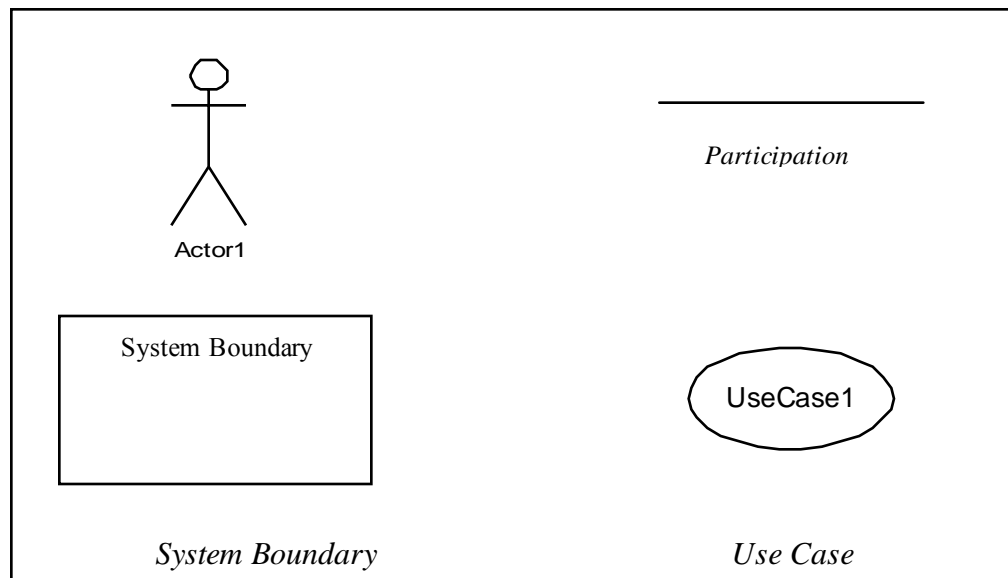
2.2.8 Rich Picture

Menurut Mathiassen., et. al., (2000, p26), *rich picture* merupakan gambaran informal yang menggambarkan pengertian dari ilustrator akan situasi sistem. *Rich picture* berfokus pada aspek penting akan situasi yang ditentukan oleh ilustrator. *Rich picture* harus mampu memberikan gambaran umum tentang situasi yang memungkinkan beberapa alternatif intrepretasi.

2.2.9 Use Case Diagram

Menurut Mathiassen (2000, p343), *use case diagram* adalah suatu gambaran interaksi antara sistem, sistem eksternal dan pengguna sistem dan dengan cara apa pengguna dapat berinteraksi dengan sistem.

Adapun notasi yang digunakan dalam *use case diagram* dapat dilihat pada gambar 2.12 dibawah ini :



Gambar 2.12 Notasi Pada *Use Case Diagram*

Sumber : Mathiassen., et. al., (2000, p343)

2.2.10 *Navigation Diagram*

Menurut Mathiassen., et. al., (2000, p344), *navigation diagram* adalah diagram yang menggambarkan alur keseluruhan sistem dalam bentuk *interface*.

2.2.11 *Oracle Data Miner*

Salah satu *data mining tools* yang khusus diperuntukkan untuk *database Oracle* adalah *Oracle Data Miner (ODM)*. ODM merupakan *software* yang disediakan oleh *Oracle* sendiri untuk membantu pengguna dalam melakukan proses *data mining* itu sendiri.

ODM telah memuat beberapa fungsi dasar yang harus dimiliki oleh *data mining tools*, yang terdiri dari :

- *Data Source*, untuk menentukan sumber data untuk di-*mining* yang dapat berasal dari *database Oracle*, maupun di-*import* dari sumber yang lain, berupa file teks (*.txt) dan juga *Excel Binary Workbook (*.csv)* yang sebelumnya harus

dikonversi terlebih dahulu ke dalam *database Oracle*. Selain itu, terdapat juga fasilitas untuk mempersiapkan data, yang dikelompokkan dalam fungsi *data transformation*, dengan tujuan agar data yang akan di-*mining* dapat dipersiapkan sebaik mungkin.

- *Activity Builder*, dimana terdapat tiga jenis aktivitas yang dapat dibentuk, yaitu :
 - *Build model* : terdiri dari serangkaian proses pembuatan *model*, yang terdiri dari pemilihan teknik algoritma yang hendak digunakan dan pemilihan *data source* yang akan di-*mining*. *Output* dari *build model* ini adalah sebuah *model data* yang merupakan pola dari data yang di-*mining*. *Model* dari data ini disajikan kepada pengguna dalam bentuk teks dan grafik.
 - *Test model* : terdiri dari proses percobaan untuk mengecek apakah *model* yang telah dibuat mempunyai tingkat *accuracy* yang baik. Tujuannya adalah mencari *model* yang terbaik yang dapat diterapkan kepada data yang akan datang.
 - *Apply model* : terdiri dari proses mengaplikasikan *model* pada data baru, sehingga didapatkan hasil tertentu yang dicari dalam data.