

BAB II

LANDASAN TEORI

II.1. *Data Mining*

Menurut Liliana Swastina (2013:2), *Data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Secara umum *Data Mining* memiliki beberapa kajian. *Data Mining* merupakan pusat dari beberapa kajian, diantaranya adalah estimasi, seleksi variabel, *clustering*, visualisasi, *market basket analysis* dan klasifikasi. Semua kajian tersebut termasuk ke dalam *data mining*.

”*Data Mining* merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dan pembelajaran mesin, pengenalan pola, statistik, *database* dan *visualisasi* untuk penanganan permasalahan pengambilan informasi dari *database* yang besar”. (Liliana Swastina,2013:12).

Hal-hal penting yang terkait dengan *Data Mining* adalah:

- a. *Data Mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
- b. Data yang akan diproses berupa data yang sangat besar.
- c. Tujuan *Data Mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat.

Hubungan yang dicari dalam *Data Mining* dapat berupa hubungan antara dua atau lebih objek dalam satu dimensi yang sama. Misalnya dalam dimensi produk dapat melihat keterkaitan pembelian suatu produk dengan produk yang lain. Selain itu, hubungan juga dapat dilihat antara dua atau lebih atribut dan dua

atau lebih objek. Masalah-masalah yang sesuai untuk diselesaikan dengan teknik *Data Mining* dapat dicirikan dengan :

- a. Memerlukan keputusan yang bersifat *knowledge-based*.
- b. Mempunyai lingkungan yang berubah.
- c. Metode yang ada sekarang bersifat sub-optimal.
- d. Tersedia data yang bisa diakses, cukup dan relevan.
- e. Memberikan keuntungan yang tinggi jika keputusan yang diambil tepat.

Kata *Mining* mempunyai arti yaitu usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar. *Data Mining* memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelegent*), *machine learning*, statistik dan *database*. Beberapa metode yang sering disebut-sebut dalam literatur *Data Mining* antara lain *clustering*, *classification*, *association rules mining*, *neural network*, *genetic algoritma* dan lain-lain.

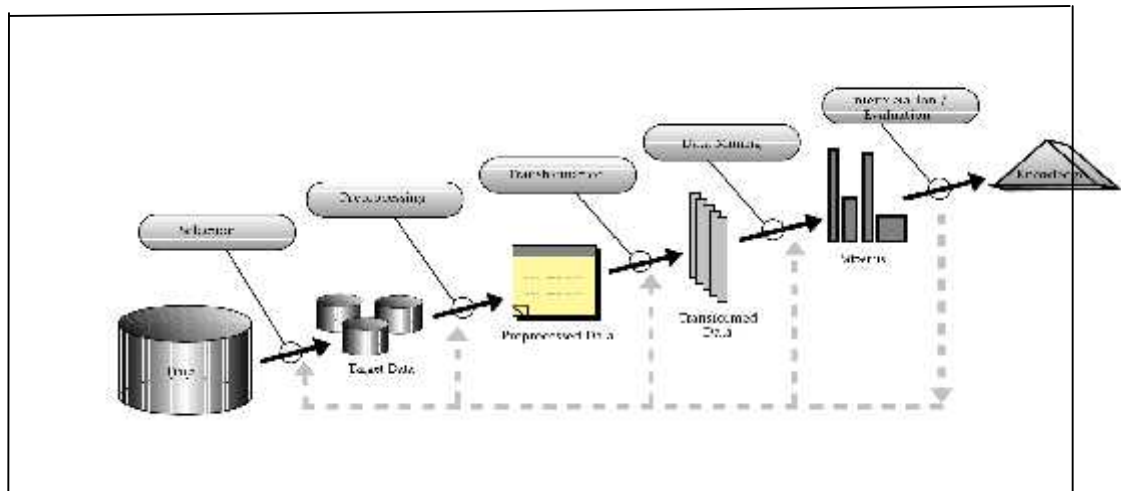
II.1.1 Tahapan *Data Mining*

Istilah *Data Mining* dan *Knowledge Discovery In Databases (KDD)* sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses *KDD* adalah *Data Mining*.

Data yang ada, tidak dapat langsung diolah dengan menggunakan sistem *Data Mining*. Data tersebut harus dipersiapkan terlebih dahulu agar hasil yang diperoleh dapat lebih maksimal, dan waktu komputasinya lebih minimal. Proses persiapan data ini sendiri dapat mencapai 60 % dari keseluruhan proses dalam *Data Mining*. Proses *KDD* secara garis besar terdiri dari 5 tahapan yaitu data

selection, pre-processing/cleaning, transformation, data mining dan interpretation/evaluation Liliana Swastina (2013:3).

Proses *KDD* digambarkan pada Gambar 2.1 berikut:



Gambar II.1 Proses *KDD*
(Sumber : Liliana Swastina ; 2013 : 3)

Berikut ini merupakan tahapan dari proses *KDD* :

a. *Data Selection*

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam *KDD* dimulai. Data hasil seleksi yang akan digunakan untuk proses *Data Mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

b. *Pre-processing/Cleaning*

Sebelum proses *Data Mining* dapat dilaksanakan, perlu dilakukan proses *cleaning* pada data yang menjadi fokus *KDD*. Proses *cleaning* mencakup antara lain membuang duplikasi data, memeriksa data yang inkonsisten dan memperbaiki kesalahan pada data, seperti kesalahan cetak (*tipografi*). Juga dilakukan proses *enrichment*, yaitu proses “memperkaya” data yang

sudah ada dengan data atau informasi lain yang relevan dan diperlukan untuk *KDD*, seperti data atau informasi eksternal.

c. *Transformation*

Coding adalah proses transformasi pada data yang telah dipilih, sehingga data tersebut sesuai untuk proses *Data Mining*. Proses *coding* dalam *KDD* merupakan proses kreatif dan sangat tergantung pada jenis atau pola informasi yang akan dicari dalam basis data.

d. *Data Mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode atau algoritma dalam *data mining* sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses *KDD* secara keseluruhan.

e. *Interpretation/Evaluation*

Pola informasi yang dihasilkan dari proses *Data Mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini merupakan bagian dari proses *KDD* yang disebut *interpretation*. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesis yang ada sebelumnya
Liliana Swastina (2013:3).

II.1.2. Pengelompokan *Data Mining*

Menurut Liliana Swastina (2013:4), *Data Mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu :

a. Deskripsi

Terkadang penelitian analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

b. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

c. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa datang. Contoh prediksi dalam bisnis dan penelitian adalah :

1. Prediksi harga beras dalam tiga bulan yang akan datang.

2. Prediksi presentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan. Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

d. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

Contoh lain klasifikasi dalam bisnis dan penelitian adalah :

1. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
2. Mendiagnosis penyakit seorang pasien untuk mendapatkan kategori penyakit apa.

e. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. *Cluster* adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidak miripan dengan *record-record* dalam *cluster* lain.

Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target. Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (*homogen*), yang mana kemiripan *record* dalam satu kelompok akan bernilai

maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

f. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

Contoh asosiasi dalam bisnis dan penelitian adalah :

1. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.
2. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan.

II.2 Metode *K-Means Cluster*

Menurut Ediyanto, dkk (2013:134), *Cluster Analysis* merupakan salah satu metode *objek mining* yang bersifat tanpa latihan (*unsupervised analysis*), sedangkan *K-Means Cluster Analysis* merupakan salah satu metode *cluster analysis* non hirarki yang berusaha untuk mempartisi objek yang ada kedalam satu atau lebih *cluster* atau kelompok objek berdasarkan karakteristiknya, sehingga objek yang mempunyai karakteristik yang sama dikelompokkan dalam satu *cluster* yang sama dan objek yang mempunyai karakteristik yang berbeda dikelompokkan kedalam *cluster* yang lain. Tujuan pengelompokan adalah untuk meminimalkan *objective function* yang di set dalam proses *clustering*, yang pada dasarnya

berusaha untuk meminimalkan variasi dalam satu *cluster* dan memaksimalkan variasi antar *cluster*.

Metode *cluster* ini meliputi *sequential threshold*, *parallel threshold* dan *optimizing threshold*. *Sequential threshold* melakukan pengelompokan dengan terlebih dahulu memilih satu objek dasar yang akan dijadikan nilai awal *cluster*, kemudian semua *cluster* yang ada dalam jarak terdekat dengan *cluster* ini akan bergabung, lalu dipilih *cluster* kedua dan semua objek yang mempunyai kemiripan dengan *cluster* ini akan digabungkan, demikian seterusnya sehingga terbentuk beberapa *cluster* dengan keseluruhan objek yang terdapat didalamnya.

Jika diberikan sekumpulan objek $X=(X_1, X_2, \dots, X_n)$ maka algoritma *K-Means Cluster Analysis* akan mempartisi X dalam k buah *cluster*, setiap *cluster* memiliki *centroid* dari objek-objek dalam *cluster* tersebut. Pada tahap awal algoritma *K-Means Cluster Analysis* dipilih secara acak k buah objek sebagai *centroid*, kemudian jarak antara objek dengan *centroid* dihitung dengan menggunakan jarak *euclidian*, objek ditempatkan dalam *cluster* yang terdekat dihitung dari titik tengah *cluster*. *Centroid* baru ditetapkan jika semua objek sudah ditempatkan dalam *cluster* terdekat. Proses penentuan *centroid* dan penempatan objek dalam *cluster* diulangi sampai nilai *centroid* konvergen (*centroid* dari semua *cluster* tidak berubah lagi). Secara umum metode *K-Means Cluster Analysis* menggunakan algoritma sebagai berikut :

1. Tentukan k sebagai jumlah *cluster* yang di bentuk.

Untuk menentukan banyaknya *cluster* k dilakukan dengan beberapa pertimbangan seperti pertimbangan teoritis dan konseptual yang mungkin diusulkan untuk menentukan berapa banyak *cluster*.

2. *Bangkitkan k Centroid* (titik pusat *cluster*) awal secara *random*.

Penentuan *centroid* awal dilakukan secara *random*/acak dari objek-objek yang tersedia sebanyak *k cluster*, kemudian untuk menghitung *centroid cluster ke-i* berikutnya, digunakan rumus sebagai berikut :

$$v = \frac{\sum_{i=1}^n x_i}{n} ; i = 1, 2, 3, \dots, n$$

Gambar II.2 Rumus Menghitung Centroid Cluster ke-i
(Sumber : Ediyanto ; 2013 : 135)

3. Hitung jarak setiap objek ke masing-masing *centroid* dari masing-masing *cluster*.

Untuk menghitung jarak antara objek dengan *centroid* penulis menggunakan *Euclidian Distance*.

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} ; i = 1, 2, 3, \dots, n$$

Gambar II.3 Rumus Menghitung Jarak Antara Objek Dengan Centroid
(Sumber : Ediyanto ; 2013 : 135)

4. Alokasikan masing-masing objek ke dalam *centroid* yang paling terdekat.

Untuk melakukan pengalokasian objek kedalam masing-masing *cluster* pada saat iterasi secara umum dapat dilakukan dengan dua cara yaitu dengan *hard k-means*, dimana secara tegas setiap objek dinyatakan sebagai anggota *cluster* dengan mengukur jarak kedekatan sifatnya terhadap titik pusat *cluster* tersebut, cara lain dapat dilakukan dengan *fuzzy C-Means*.

5. Lakukan iterasi, kemudian tentukan posisi *centroid* baru
6. Ulangi langkah 3 jika posisi *centroid* baru tidak sama.

Pengecekan *konvergensi* dilakukan dengan membandingkan *matriks group assignment* pada iterasi sebelumnya dengan *matriks group assignment* pada iterasi yang sedang berjalan. Jika hasilnya sama maka *algoritma k-means cluster analysis* sudah *konvergen*, tetapi jika berbeda maka belum *konvergen* sehingga perlu dilakukan iterasi berikutnya.

II.3. *Unified Modeling Language (UML)*

Adi Nugroho (2010:6), *Unified Modeling Language (UML)* adalah bahasa pemodelan untuk system atau perangkat lunak yang berparadigma “berorientasi objek” . Pemodelan (*Modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Dalam hal ini sasaran model sesungguhnya adalah abstraksi segala sesuatu yang ada diplanet bumi menjadi gambaran-gambaran umum yang lebih mudah dipahami dan dipelajari. Adapun tujuan pemodelan (dalam rangka pengembangan sistem/perangkat lunak aplikasi) sebagai sarana analisis, pemasahaman visualisasi dan komunikasi antar anggota tim pengembang.

II.3.1. Pengenalan *UML (Unified Modeling Language)*

Menurut Julius Hermawan (2010:7), *UML (Unified Modeling Language)* adalah bahasa standard yang digunakan untuk menjelaskan dan memvisualisasikan artifak dan proses analisis dan desain berorientasi objek. *UML* Menyediakan standar pada notasi dan diagram yang biasa digunakan untuk

memodelkan suatu sistem. *UML* dikembangkan oleh tiga pendekar “berorientasi objek” yaitu *Gradi Booch*, *Jim Rumbaugh* dan *Ivar Jacobson*. *UML* menjadi bahasa yang bias digunakan untuk berkomunikasi dalam prespektif objek antara *user* dengan *developer*, antara *developer* analisis dengan *developer* desain dan antara *developer* desain dengan *developer* pemrograman.

UML memungkinkan *developer* melakukan pemodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Pemodelan visual membantu untuk menangkap struktur dan kelakuan dari si objek, mempermudah penggambaran interaksi antara elemen dalam sistem dan mempertahankan konsistensi antara desain dan implementasi dalam bahasa pemrograman.

Namun karena *UML* hanya merupakan bahasa pemodelan maka *UML* bukanlah rujukan bagaimana melakukan analisis dan desain berorientasi objek. Untuk mengetahui bagaimana melakukan analisis dan desain berorientasi objek secara baik, sudah terdapat beberapa metodologi yang bisa diikuti.

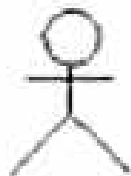
II.3.2. Notasi dan Artifak dalam UML

Menurut Julius Hermawan (2010:13), *UML* menyediakan beberapa notasi dan artifak standard yang bias digunakan sebagai alat komunikasi bagi para proses analisis dan desain. Artifak didalam *UML* didefenisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.

1. Aktor

Aktor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi computer. Jadi aktor ini bisa berupa orang, perangkat keras atau juga objek lain dalam sistem yang sama. Biasanya yang dilakukan oleh Aktor adalah

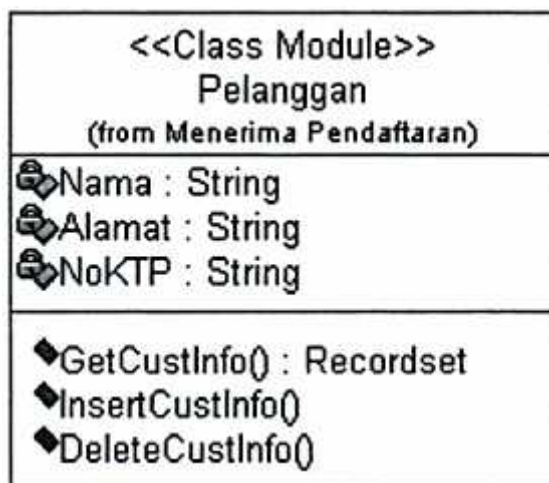
memberikan informasi pada sistem atau memerintahkan sistem untuk melakukan sesuatu.



Gambar II.4. Notasi Aktor
(Sumber : Julius Hermawan ; 2010 ; 14)

2. Class

Class merupakan pembentuk utama dari system berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*



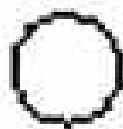
Gambar II.5. Notasi Class
(Sumber : Julius Hermawan ; 2010 ; 14)

Class digunakan untuk mengabstraksikan elemen-elemen dari sistem yang dibangun. *Class* bisa untuk direpresentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata. Atribut digunakan untuk menyimpan informasi. Nama atribut menggunakan kata benda yang bisa dengan jelas direpresentasikan informasi yang disimpan didalamnya. Operasi

menunjukkan sesuatu yang bisa dilakukan oleh objek, dan menggunakan kata kerja.

3. *Interface*

Interface merupakan kumpulan informasi tanpa implementasi dari suatu *class*. Implementasi operasi dari suatu *interface* dijabarkan oleh operasi didalam *class*. Oleh karena itu keberadaan *interface* selalu disertai oleh *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



Gambar II.6. Notasi *Interface*
(Sumber : Julius Hermawan ; 2010 ; 15)

4. *Use Case*

Use Case menjelaskan urutan kegiatan yang dilakukan Aktor dan sistem untuk mencapai tujuan tertentu. Walaupun menjelaskan kegiatan namun *use case* hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana sistem melakukan kegiatan tersebut.



Gambar II.7 Notasi *Use Case*
(Sumber : Julius Hermawan ; 2010 ; 16)

Didalam *use case* terdapat teks untuk menjelaskan urutan kegiatan yang disebut *use case specification*. *use case specification* terdiri dari :

a. Nama *Use Case*

Mencantumkan nama dari *use case* yang bersangkutan. Sebaiknya diawali dengan kata kerja untuk menunjukkan suatu aktivitas

b. Deskripsi singkat

Menjelaskan secara singkat dalam 1 atau 2 kalimat tentang tujuan dari *use case* ini.

c. Aliran Normal (*Basic Flow*)

Ini adalah jantung dari *use case*. Menjelaskan interaksi antara aktor dan sistem dalam kondisi normal, yaitu segala sesuatu berjalan dengan lancar tiada halangan atau hambatan dalam mencapai tujuan dalam *use case*.

d. Aliran Alternatif (*Alternative Flow*)

Merupakan pelengkap dari *basic flow* tidak ada yang sempurna dalam setiap kali *use case* berlangsung. Didalam *Alternative Flow* ini dijelaskan dalam apa yang terjadi bila suatu halangan atau hambatan terjadi sewaktu *use case* berlangsung. Ini terutama berhubungan dengan *error* yang mungkin terjadi terutama karena sistem kekurangan data untuk diolah.

e. *Special Requirement*

Berisi kebutuhan lain yang belum tercakup dalam kebutuhan normal dan alternatif. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan fungsional dari *use case* sementara *Special Requirement* yang tidak berhubungan dengan kebutuhan fungsional, misalnya kecepatan transaksi maksimum artinya

berapa cepat dan berapa lama, kapasitas akses yaitu jumlah *user* yang akan mengakses dalam waktu bersamaan.

f. Pre-Condition

Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

g. Post-Condition

Menjelaskan kondisi yang berubah atau terjadi saat *use case* selesai dieksekusi.

5. *Interaction*

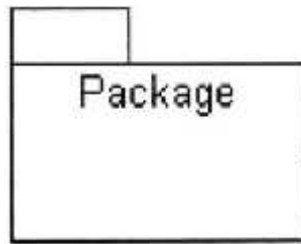
Digunakan untuk menunjukkan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.



Gambar II.8. Notasi *Interaction*
(Sumber : Julius Hermawan ; 2010 ; 18)

6. *Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat model menjadi lebih sederhana. Tujuannya adalah untuk mempermudah pengelihatian dari model yang sedang dibangun.



Gambar II.9. Notasi *Package*
(Sumber : Julius Hermawan ; 2010 ; 19)

7. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditempelkan ke semua elemen notasi yang lain.

8. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain.



Gambar II.10. Notasi *Dependency*
(Sumber : Julius Hermawan ; 2010 ; 20)

II.4. Pengertian Basis Data (*Database*)

Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulan-kumpulan *file* yang saling berkaitan.

Menurut Kusriani (2010, p2), pengertian Basis Data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpan elektronik.

II.4.1. Tujuan Basis Data

Menurut Kusrini (2010, p2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basis data yang baik adalah sebagai berikut :

- a. Tidak adanya redudansi dan inkonsistensi data

Redudansis terjadi jika suatu informasi disimpan dibeberapa tempat. Misalnya ada data mahasiswa yang memuat nim, nama, alamat dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya terdapat NIM, nama, mata kuliah dan nilai. Pada kedua data tersebut kita temukan atribut nama.

- b. Kesulitan pengaksesan data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan *query* ataupun dari tool yang melibatkan tabelnya. Dengan fasilitas ini, bisa segera langsung melihat data dari *software* DBMSnya.

c. *Multiple User*

Basis data memungkinkan penggunaan data secara bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian *server* yang bisa diakses dari banyak *client*, sudah menyediakan akses ke semua pengguna dari komputer *client* ke sumber informasi yaitu basis data.

II.4.2. Manfaat/Kelebihan Basis Data

Menurut Kusri (2010, p5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

a. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan perancangan yang benar maka penyajian informasi dapat dilakukan dengan cepat dan mudah.

b. Kebersamaan pemakai (*sharability*)

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencacatan dimasing-masing bagian/orang, tetapi cukup dengan satu basis data untuk dipakai bersama.

c. Pemusatan kontrol data

Karena cukup satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

d. Efisiensi ruang penyimpanan

Dengan pemakaian bersama, tidak perlu menyediakan tempat penyimpanan diberbagai tempat tetapi cukup satu saja, sehingga ini dapat menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

e. Keakuratan (*Accuracy*)

Penerapan secara tepat acuan tipe data, domain data, keunikan data, hubungan antar data, dan lain-lain, dapat menekan ketidak akuratan dalam pemasukan/penyimpanan data.

f. Ketersediaan (*Availability*)

Dengan basis data, semua data dapat dibackup, memilah-milah data mana yang masih diperlukan yang perlu disimpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi sebuah organisasi dari lain waktu ke waktu membutuhkan penyimpanan yang semakin besar.

g. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Pengguna diberi hak akses yang berbeda-beda sesuai dengan kepentingan dan posisinya. Basis data bisa diberikan *password* untuk membatasi orang yang diaksesnya.

h. Kemudahan dalam pembuatan program aplikasi baru

Penggunaan basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

i. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datanya secara langsung dengan *tools* yang disediakan oleh DBMS.

j. Kebebasan data

Perubahan dapat dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

k. *User View*

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna.

II.4.3. Operasi Dasar *Database*

Menurut Kusrini (2010, p9), Beberapa operasi dasar basis data yaitu :

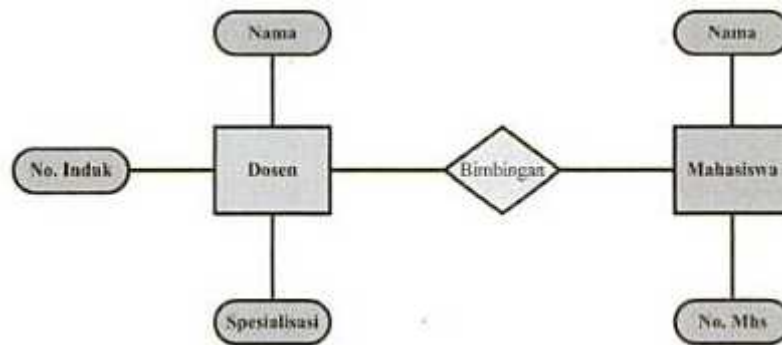
- a. Pembuatan basis data
- b. Penghapusan basis data
- c. Pembuatan file/tabel
- d. Penghapusan file/tabel
- e. Pengubahan tabel
- f. Penambahan/pengisian
- g. Pengambilan data
- h. Penghapusan data

II.4.4. Pemodelan Basis Data

Menurut Samiaji Sarosa (2010:4), Model diperlukan untuk mendapatkan penyederhanaan dari kenyataan dan memungkinkan desainer program program aplikasi bereksperimen dengan berbagai macam variabel sebelum diaplikasikan ke

sistem yang berjalan. Untuk merancang suatu aplikasi basis data alat yang biasa digunakan adalah *Entity Relationship Diagram (ERD)*. ERD didasarkan dari artikel yang dipublikasikan oleh *Peter Phin Shan Chen*.

Ada beberapa case tool menamakan notasi ERD yang digunakan sebagai chen ERD. *Entity Relationship Model* adalah abstraksi konseptual yang mewakili struktur dari suatu basis data.

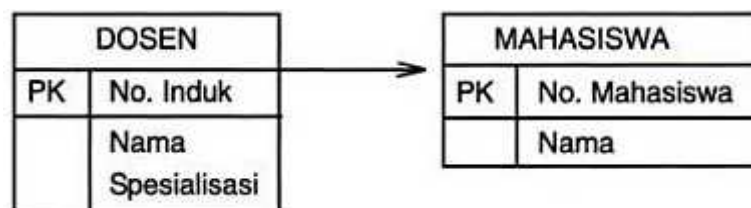


Gambar II.11. Diagram Dengan Notasi Chen ERD
(Sumber : Samiaji Sarosa ; 2010 ; 4)

Dalam perkembangannya banyak diciptakan notasi ERD yang berbeda-beda seperti terlihat gambar dibawah ini.



Gambar II.12. Diagram Dengan Notasi Crows Foot
(Sumber : Samiaji Sarosa ; 2010 ; 4)



Gambar II.13. Diagram Dengan Notasi Relational
(Sumber : Samiaji Sarosa ; 2010 ; 4)

II.4.5. Normalisasi

Menurut Samiaji Sarosa (2010:5), Normalisasi adalah teknik yang dirancang untuk merancang tabel basis data relasional untuk meminimalkan duplikasi data dan menghindarkan basis data tersebut anomali. Suatu basis data dikatakan tidak normal jika terjadi 3 (tiga) anomali berikut :

a. *Insertion Anomaly*

Anomali yang terjadi jika ada data yang tidak bisa disisipkan kedalam table.

b. *Update/Modification anomaly*

Anomali yang terjadi jika ada perubahan pada suatu item data maka harus mengubah lebih dari satu baris data.

Langkah-langkah normalisasi sampai pada bentuk 3NF adalah sebagai berikut :

a. *First Normal Form (1NF)*

Untuk menjadi 1NF suatu table harus memenuhi dua syarat. Syarat pertama tidak ada kelompok data atau *field* yang berulang. Syarat kedua harus ada *primary key (PK)* atau kunci unik, atau kunci yang membedakan satu bari dengan baris yang lain dalam satu tabel. Pada dasarnya sebuah tabel selamat tidak ada kolom yang sama merupakan bentuk tabel dengan 1NF.

b. *Second Normal Form (2NF)*

Untuk menjadi 2NF suatu tabel harus berada dalam kondisi 1NF dan tidak memilik *partial dependencies*. *Partial dependencies* adalah suatu

kondisi jika atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

c. *Third Normal Form (3NF)*

Untuk menjadi 3NF suatu table harus berada dalam kondisi 2NF dan tidak memiliki *transitive dependencies*. *Transitive dependencies* adalah suatu kondisi dengan adanya ketergantungan fungsional antara 2 atau lebih atribut non kunci (Non PK).

II.5. *Visual Basic 2010*

Menurut Edi Winarno ST, M.Eng dkk (2010:1), *Visual Basic* adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer didunia. Pemograman ini dipakai oleh jutaan programmer dan tercatat sebagai program yang paling disukai oleh mayoritas orang.

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. *Visual Studio 2010* selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis *windows*.

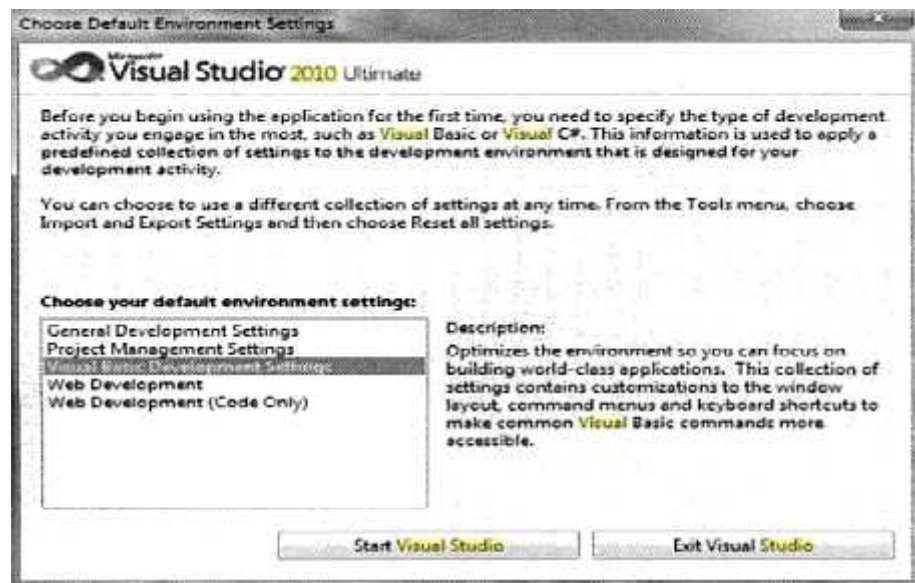
Beberapa kemampuan atau manfaat dari *Visual Studio 2010* diantaranya seperti :

1. Untuk membuat program aplikasi berbasis *windows*.
2. Untuk membuat objek-objek pembantu program seperti, misalnya : kontrol *ActiveX*, *file Help*, aplikasi Internet dan sebagainya.

3. Menguji program (*debugging*) dan menghasilkan program berakhiran *EXE* yang bersifat *executable* atau dapat langsung dijalankan.

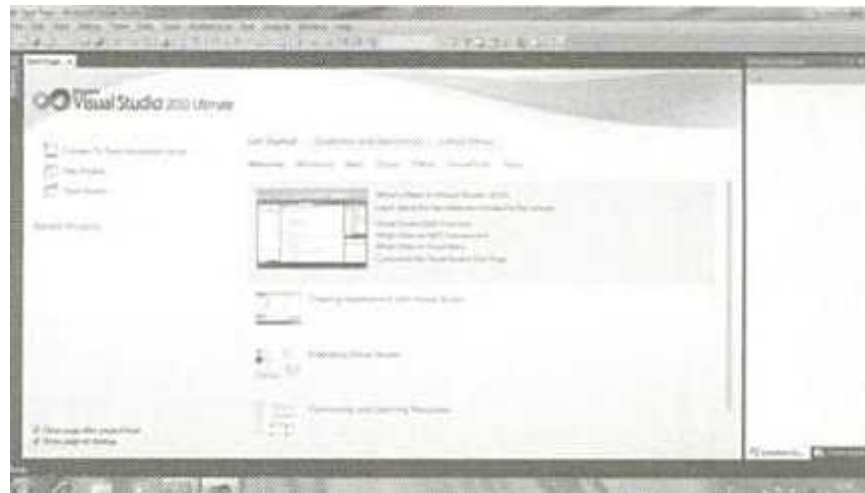
II.5.1. Antar Muka *Visual Basic 2010*

Saat menjalankan *Visual Basic 2010* pertama kali muncul jendela *chosedefault environment settings*. Disini bisa memilih apakah ingin memilih antar muka di *Visual Studio*. Untuk *programmer Visual Basic* lebih baik memilih *Visual Basic Development Centre*.



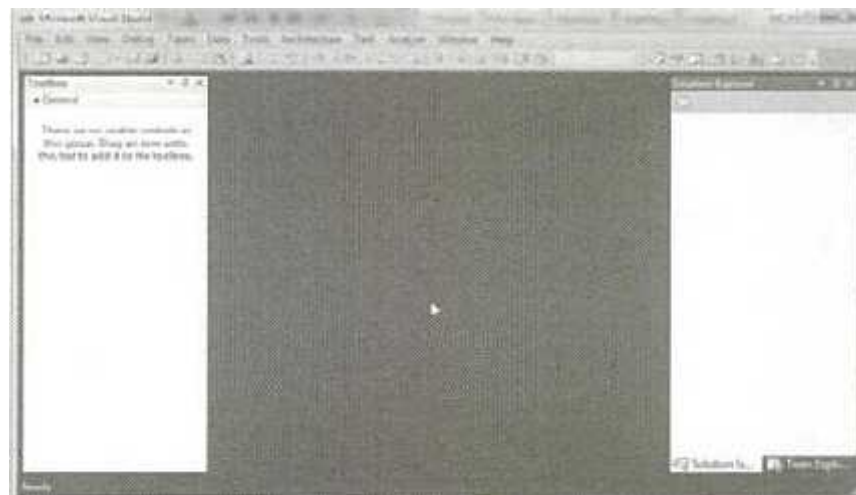
**Gambar II.14 Form Chose Default Environment Settings
(Sumber : Edi Winarno ST, M.Eng dkk ; 2010 ; 2)**

Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



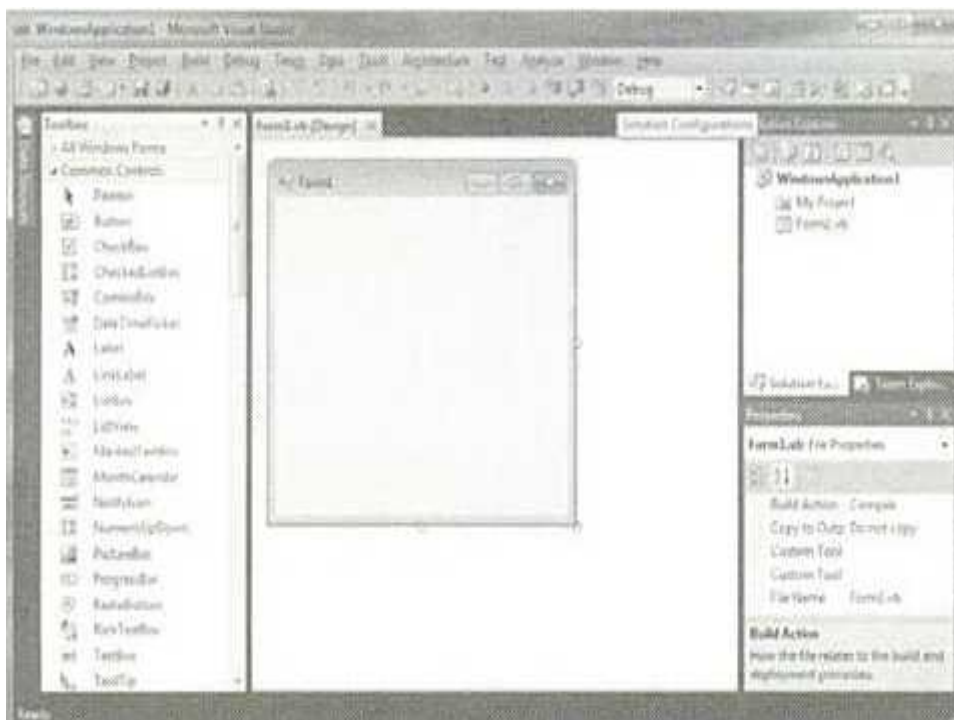
Gambar II.15. Start Page Visual Basic 2010
(Sumber : Edi Winarno ST, M.Eng dkk ; 2010 ; 3)

Jika *start page* ditutup terlihat tampilan sebagai berikut :



Gambar II.16. Tampilan IDE (*Integrated Development Environment*) setelah *Start Page* ditutup
(Sumber : Edi Winarno ST, M.Eng dkk ; 2010 ; 4)

Jika ada sebuah *form* yang terlihat, tampilan lengkap IDE seperti gambar berikut ini.



Gambar II.17. Tampilan lengkap IDE

(Sumber : Edi Winarno ST, M.Eng dkk ; 2010 ; 7)

Komponen-komponen dari IDE adalah :

1. Dibagian kiri terdapat *toolbox* yang menampilkan semua objek *tool* yang bisa dimasukkan kedalam *form* untuk membuat program.
2. Dibagian tengah terdapat tempat meletakkan *form* dan kode, baik disaat desain ataupun pada saat program dijalankan.
3. Dibagian kanan terdapat *solution explorer* yang merupakan *explorer* untuk melihat file-file disebuah objek.
4. Dikanan bawah terdapat *propertis* untuk melihat properti dari nilai-nilai pada objek yang dipilih dibagian tengah. (Edi Winarno ST, M.Eng dkk, 2010:1)

II.6. *SQL Server 2008 Express Edition*

Menurut Wahana Komputer (2010:2), *SQL Server 2008 Express Edition* sebuah terobosan baru dalam bidang *database*, *SQL Server* adalah sebuah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti *IBM* dan *Oracle*. *SQL Server 2008 Express Edition* dibuat pada saat kemajuan dalam bidang *hardware* semakin pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server 2008 Express Edition* membawa terobosan dalam bidang pengolahan dan penyimpanan data.

II.6.1 *Kebutuhan Hardware*

Adapun *hardware* yang diperlukan untuk instalasi *SQL Server 2008 Express Edition* minimal adalah sebagai berikut :

- a. *Procesccor* minimal 1 GHz
- b. Memori minimal 512 MB
- c. Sistem Operasi *Windows*

Biar dapat diinstal pada sistem computer dengan memori 512 MB, tetapi disarankan menggunakan memori 1 GB. Sedangkan untuk jaringannya diperlukan adalah :

- a. *Sharer Memory*
- b. *TCP/IP*
- c. *Named Pipes*
- d. *Virtual Interface Adapter (VIA)*(Wahana Komputer, 2010:2).

II.6.2 Versi *SQL Server 2008 Express Edition*

Microsoft merilis *SQL Server 2008 Express Edition* dalam beberapa versi yang disesuaikan dengan segmen-segmen pasar yang dituju. Versi-versi tersebut adalah sebagai berikut :

- a. Menurut cara pemrosesan data pada prosesor maka *Microsoft* mengelompokkan produk ini berdasarkan dua jenis yaitu :
 1. Versi 32 Bit (x86), yang biasanya digunakan untuk komputer *single processor* (Pentium 4) atau lebih tepatnya *processor* 32 bit atau *Windows XP*
 2. Versi 64 Bit (x64), yang biasanya digunakan oleh komputer yang lebih dari satu *processor* (Misalnya *Core 2 duo*) dan sistem operasi 64 bit, *Vista* dan *Windows 7*.
- b. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
 1. Versi *Compact* ini adalah versi “tipis” dari semua versi yang ada
 2. Versi *Express* ini adalah versi “ringan”

II.6.3 Instalasi *SQL Server 2008 Express Edition*

Proses instalasi *SQL Server 2008 Express Edition* tidak sama dengan instalasi versi-versi sebelumnya. Proses *SQL Server 2008 Express Edition* agak panjang melalui beberapa tahapan. Tahapan yang dilakukan akan membawa beberapa pilihan yang akan diisi dalam *setting* sebuah *server database*. Berikut ini adalah pilihan-pilihan yang akan dijumpai dalam proses instalasi *SQL Server 2008 Express Edition*.

1. Tempat direktori utama dan penyimpanan file *database*

Direktori utama adalah direktori dimana semua file program akan ditempatkan dan file-file tersebut tidak akan berubah selama anda menjalankan *SQL server*. Direktori utama secara standard akan berada dalam direktori “*C:\Program Files\Microsoft SQL Server*”.

2. Penggunaan *Multiple instance*

Instance adalah sebuah turunan dari *server database SQL Server*. Karena sebuah tiruan maka sebuah *Instance* memiliki fungsi yang sama dengan database server aslinya. Arti sebenarnya *Instance SQL Server* adalah sebuah *server database* yang tidak *men-sharing* sistemnya dan *database user* dengan *database server* lainnya yang ada dalam komputer yang sama.

3. Jasa *Autentification User* (Menggunakan *Windows* atau *mixed*)

Autentification User diperlukan supaya server tidak dapat dipergunakan oleh orang yang tidak bertanggung jawab dan tidak berhak. Dalam *SQL server* ada dua *Autentification User* yang dapat digunakan yaitu :

- a. Mode *Windows*, Pada mode ini *SQL Server* akan melakukan autentifikasi dengan menggunakan level login pada sistem operasi.
- b. Mode *Mixel* atau campuran, mode ini mengizinkan *user* untuk masuk kedalam *sistem SQL server* dengan menggunakan *Account* yang dibuat di sistem operasi *windows* atau juga menggunakan *account* yang di *set up* pada *SQL Server* (Wahana Komputer, 2010:2).