

## **BAB II**

### **LANDASAN TEORI**

#### **II.1 Kecerdasan Buatan**

Menurut Anita Desiani dan Muhammad Arhami (2006:1), *Artificial Intelligence* (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang koncern dengan pengautomalisasi tingkah laku cerdas. Pernyataan tersebut juga dapat dijadikan defenisi dari AI. Defenisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada *Sound Theoretical* (teori suara) dan juga prinsip-prinsip aplikasi dari bidangnya. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritma yang diperlukan untuk mengaplikasi pengetahuan tersebut, serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya.

Namun defenisi di atas belum memadai karena pada kenyataannya istilah kecerdasan itu sendiri tidak didefenisikan atau dipahami dengan baik. Kerap kali kita meyakini telah mengetahui tingkah laku cerdas saat melihatnya sendiri. Akan tetapi keraguan akan muncul ketika orang lain kecerdasan dalam cara khusus menurut cara mereka yang cukup membantu dalam evaluasi kecerdasan program komputer. Dengan demikian belum ada suatu defenisi yang cocok untuk *Artificial Intelligence* itu sendiri.

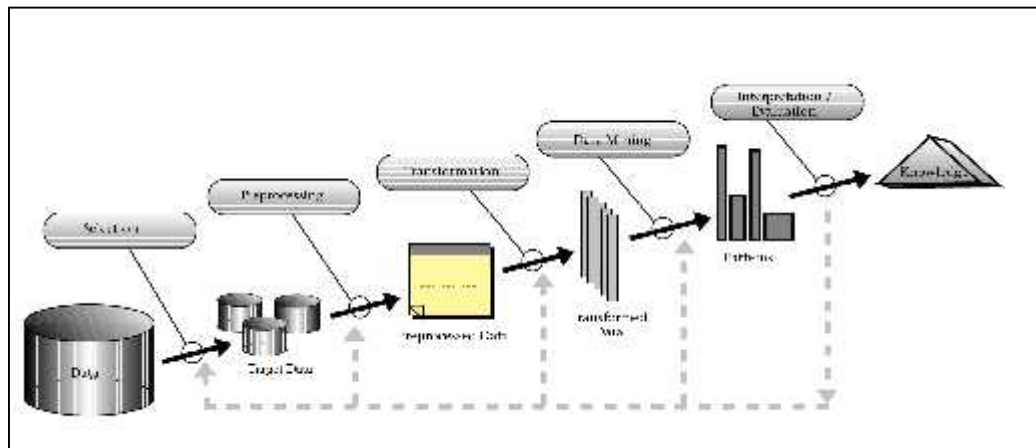
## II.2 Data Mining

Istilah *Data Mining* memiliki beberapa padanan, seperti *Knowledge Discovery* ataupun *Pattern Recognition*. Kedua istilah tersebut sebenarnya memiliki ketepatannya masing-masing. Istilah *Knowledge Discovery* atau penemuan pengetahuan tepat digunakan karena tujuan utama dari *Data Mining* memang untuk mendapatkan pengetahuan yang masing tersembunyi di dalam bongkahan data. Istilah *Pattern Recognition* atau disebut juga sebagai pengenalan pola pun tepat untuk digunakan karena pengetahuan yang hendak digali memang berbentuk pola-pola yang mungkin juga masih perlu digali dari dalam bongkahan data yang tengah dihadapi. Jadi, apakah sebenarnya *Data Mining* itu? Banyak definisi bagi istilah ini dan belum ada yang dibakukan atau disepakati semua pihak. Namun demikian, istilah ini memiliki hakikat (notion) sebagai disiplin ilmu yang tujuan utamanya adalah untuk menemukan, menggali, atau menambang pengetahuan dari data atau informasi yang kita miliki. Kegiatan inilah yang menjadi garapan atau perhatian utama dari disiplin ilmu *Data Mining* menurut Sani Susanto dan Dedy Suryadi (2010:2).

### II.2.1 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* Liliana Swastina (2013:3).

Dalam suatu *Knowledge Discovery In Databases (KDD)* ada beberapa tahapan yang dapat dilihat secara garis besar dan tahapan-tahapan tersebut dapat kita lihat pada Gambar II.1. :



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

Berikut ini merupakan tahapan dari proses KDD :

### 1. *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.

### 2. *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.

### 3. *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.

### 4. *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.

### 5. *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.

## II.2.2. Pengelompokan *Data Mining*

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

### 1. 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.

### 2. 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.

### 3. 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 :

- a. Prediksi harga beras dalam tiga bulan yang akan datang.
- b. 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.

### 4. 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 :

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

### 5. 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 ketidakmiripan 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.

#### 6. 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 :

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

### II.3 Algoritma C4.5

Menurut Muhammad Yunus, et al (2014:49), Algoritma yang dapat digunakan untuk memprediksi atau mengklarifikasi suatu kejadian dengan pembentukan pohon keputusan antara lain C4.5, yang merupakan salah satu algoritma induksi pohon keputusan yang dikembangkan oleh J. Ross Quinlan. Secara umum alur proses algoritma C4.5 untuk membangun pohon keputusan dalam *data mining* adalah:

1. Pilih atribut sebagai simpul akar.
2. Buat cabang untuk tiap-tiap nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Menurut Mahmud Yunus, et al (2014:49), Pemilihan atribut sebagai simpul, baik simpul akar (*root*) atau simpul internal didasarkan pada nilai *Gain* tertinggi dari atribut-atribut yang ada.

1. Penghitungan nilai *Gain* digunakan rumus seperti dalam Persamaan 1.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \dots (1)$$

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|S<sub>i</sub>| : Jumlah kasus pada partisi ke-i

|S| : Jumlah kasus dalam S



2. Untuk menghitung nilai *Entropy* dapat dilihat pada Persamaan 2.

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i \dots\dots\dots (2)$$

$n$  : Jumlah partisi  $S$

$p_i$  : Proporsi dari  $S_i$  terhadap  $S$

#### II.4 Pengertian *Unified Modeling Language (UML)*

Menurut Haviluddin (2011;1), *Unified Modelling Language* merupakan alat perancangan sistem yang berorientasi pada objek. Secara filosofikemunculan UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented (OO)*, karenakonsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik maka OO memiliki proses standart dan bersifat independen.

*Unified Modelling Language* diagram memiliki tujuan utama untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

*Unified Modelling Language* mempunyai tiga kategori utama yaitu struktur diagram, *behaviour* diagram dan *interaction* diagram. Dimana masing-

masing kategori tersebut memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi.

#### **II.4.1. Evolusi *Unified Modeling Language (UML)***

Menurut Prabowo Pudjo Widodo dan Herlawati (2005;8), para pengguna notasi yang berlainan ini saling berebut pengaruh agar notasi yang digunakan menjadi notasi standart. Walaupun dijumpai beberapa notasi yang cukup jelas dan sangat cocok untuk menyelesaikan problem pembuatan perangkat lunak *Object Oriented Program* (OOP), tetapi belum diketahui oleh pengembang sistem yang lain.

Pada Bulan Oktober 1994, Jim Rumbaugh, penemu notasi *Object Modeling Technique* (OMT) dan Gray Booch, penemu *Booch Method* (Metode Booch) bersama-sama menyamakan notasi mereka. Ditahun yang sama, Ivar Jacobson Penemu (*Objectory Method*) ikut bergabung. Ketiga orang itu sering disebut “*Thee Amigos*” bersama-sama membangun notasi standart OOP untuk software Rational IBM. Jim Rumbaugh banyak member masukkan dalam hal Pembuatan notasi dan analisa *Unified Modelling Language* (UML), Grady Booch merancang secara detail kapabilitas *Unified Modelling Language* (UML) sedang Ivar Jacobson berusaha membuat *Unified Modelling Language* (UML) cocok dengan model bisnis dan mencoba menegembangkan *Use Case Diagram* lebih lanjut.

Ketiga pengembang *Unified Modelling Language* (UML) tersebut mengalami kesulitan akibat kompleksnya permasalahan OOP yang ada, namun mereka dibantu oleh *Object Management Group* (OMG). OMG adalah gabungan

dari kurang lebih 800 perusahaan pengembang perangkat lunak berorientasi objek. Setelah perdebatan sengit yang cukup lama, consensus tentang notasi berhasil dicapai dengan sukses pada bulan November 1997 berkat bantuan OMG.

Menurut Haviluddin (2011;3), Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*) dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

#### **II.4.2. Diagram-Diagram *Unified Modeling Language (UML)***

Menurut Prabowo Pudjo Widodo dan Herlawati (2005;10), beberapa literature menyebutkan bahwa *Unified Modelling Language (UML)* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan jenis, karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi.

Menurut Haviluddin (2011;3), Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram, dan diantaranya adalah sebagai berikut :

##### **1. Struktur Diagram**

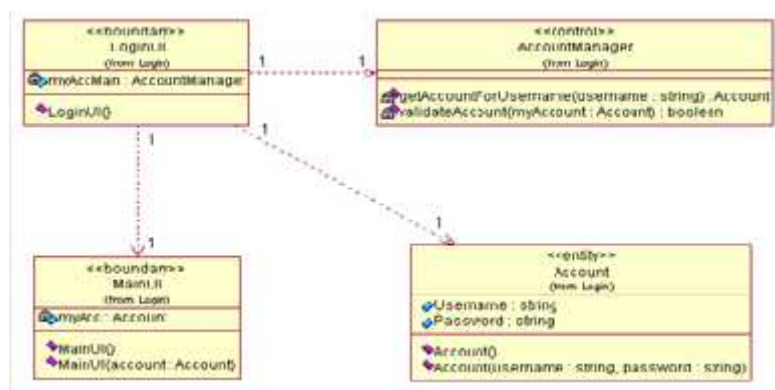
Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas :

##### **a. *Class diagram***

*Class diagram* menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan

hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok :

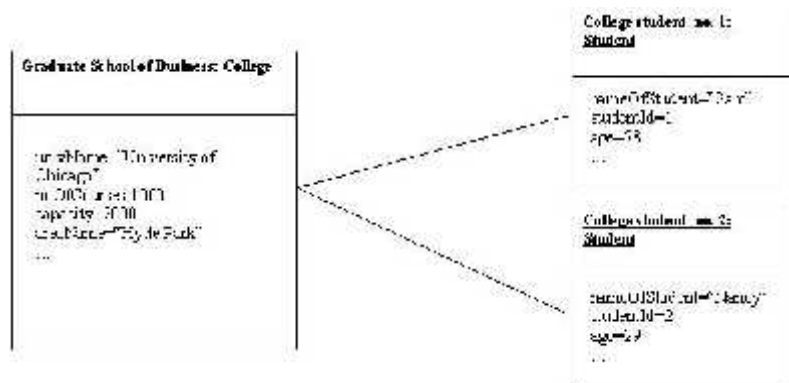
- 1). Nama (dan *stereotype*)
- 2). Atribut
- 3). Metoda



**Gambar II.2. Notasi Class Diagram**  
(Sumber : Haviluddin ; 2011)

#### b. *Object Diagram*

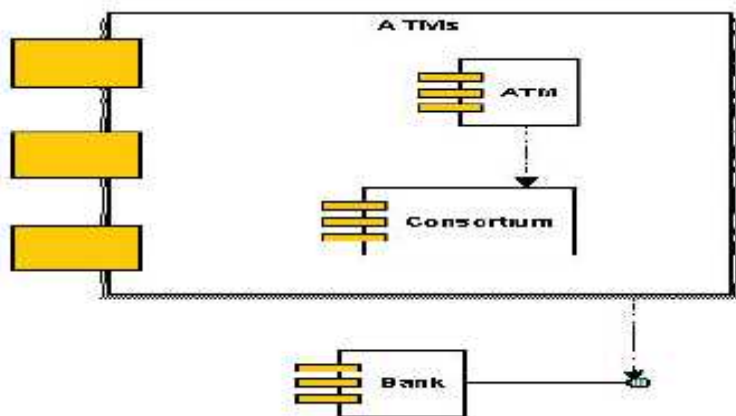
*Object diagram* menggambarkan kejelasan kelas dan warisan dan kadangkadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak. Berikut notasi *object diagram* dilihat pada gambar II.3.



**Gambar II.3. Notasi *Object Diagram***  
(Sumber : Haviluddin ; 2011)

c. *Component diagram*

*Component diagram* menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan dan gambaran bentuk *Component Diagram* dilihat pada gambar II.4.

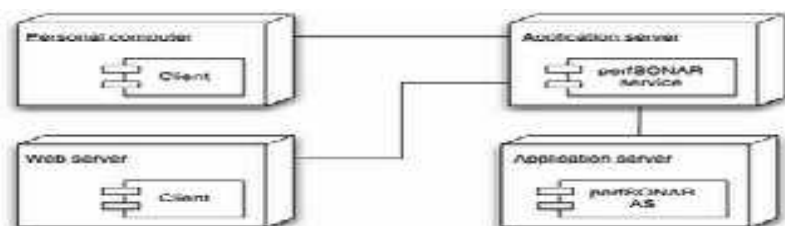


**Gambar II.4. Notasi *Component Diagram***  
(Sumber : Haviluddin ; 2011)

d. *Deployment diagram (Collaboration diagram in version 1.x)*

*Deployment diagram* memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari

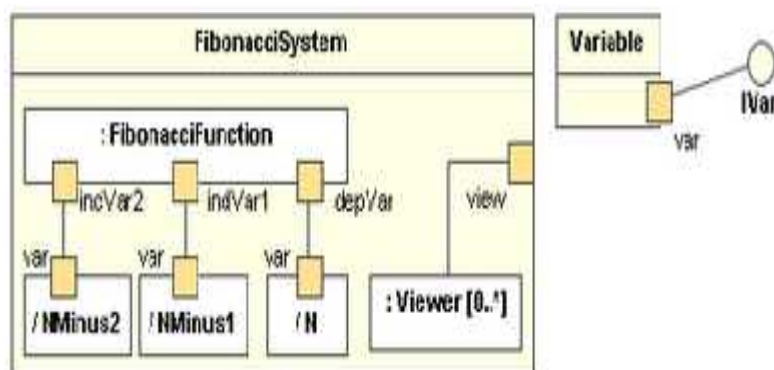
sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem dan diagram ini dapat dilihat pada Gambar II.5.



**Gambar II.5. Notasi *Deployment* Diagram**  
(Sumber : (Sumber : Havaluddin ; 2011))

e. *Composite structure* diagram

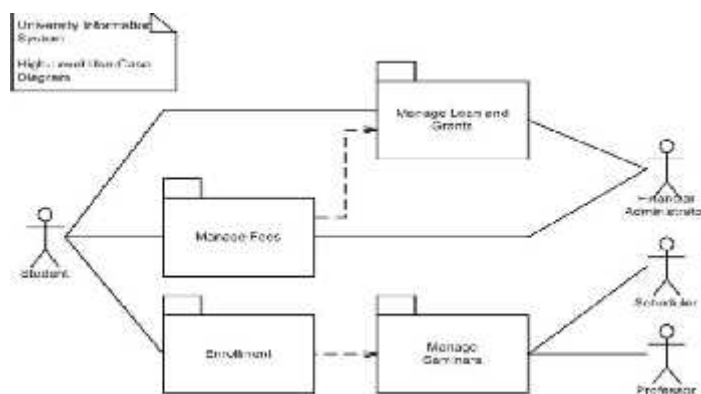
Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi dan bentuk dari diagram ini dapat dilihat pada Gambar II.6.



**Gambar II.6. Notasi *Composite structure* Diagram**  
(Sumber : (Sumber : Havaluddin ; 2011))

#### f. *Package diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model dan bentuk dari diagram ini dapat dilihat pada Gambar II.7.



**Gambar II.7. Notasi *Package Diagram***  
(Sumber : (Sumber : Haviluddin ; 2011))

## 2. *Behavior Diagram*

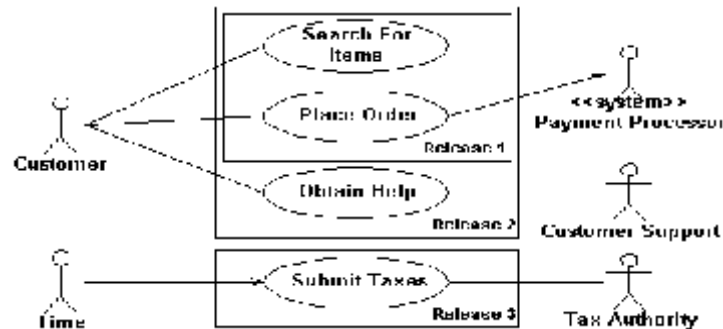
Menggambarkan ciri-ciri *behavior*/metode/ fungsi dari sebuah sistem atau *business process*. *Behavior* diagram dalam UML terdiri atas :

#### g. *Use case diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. *Use Case* memiliki dua istilah yaitu :

- 1). *System use case*; interaksi dengan sistem.

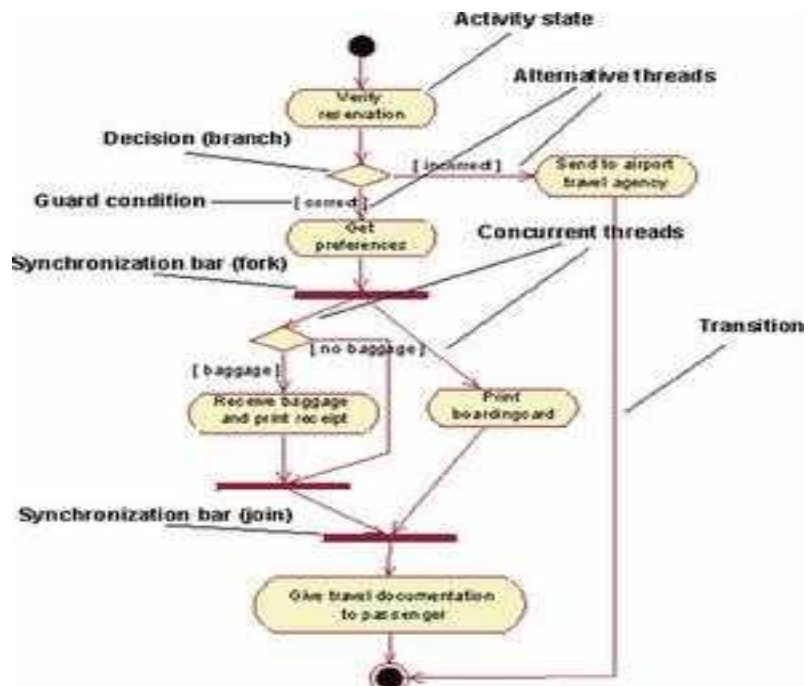
2). *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata.



**Gambar II.8. Notasi Use Case Diagram**  
(Sumber : (Sumber : Haviluddin ; 2011))

#### h. Activity Diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk segala aktivitasnya, dan gambarannya dapat dilihat pada Gambar II.9.

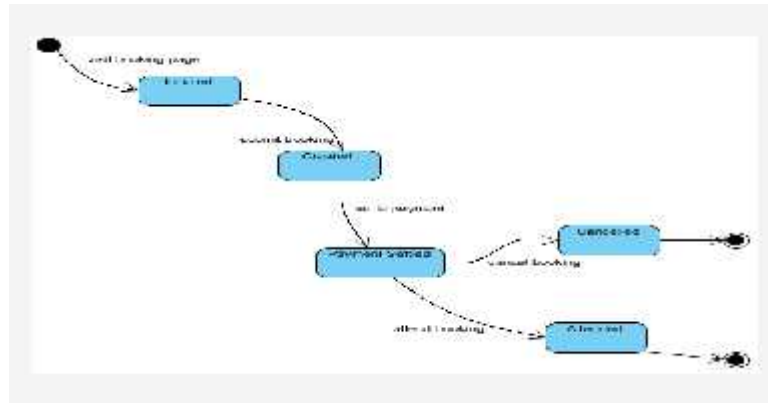


**Gambar II.9. Notasi Activity Diagram**  
(Sumber : (Sumber : Haviluddin ; 2011))



i. *State Machine* diagram (*State chart* diagram in version 1.x)

Menggambarakan *state*, transisi *state* dan *event* dan untuk gambaran dari diagram ini dapat dilihat pada Gambar II.10.



**Gambar II.10. Notasi *State Machine* Diagram**  
(Sumber : (Sumber : Haviluddin ; 2011))

## II.5 Pengertian Basis Data (*Database*)

Menurut Kusri (2010;2), 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 symbol).

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.5.1. Tujuan Basis Data

Menurut Kusriani (2010:2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat basisdata yang baik adalah sebagai berikut :

1. Tidak adanya redudansi dan inkonsistensi data

Redudansi terjadi jika suatu informasi disimpan di beberapa 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.

2. 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 DBM Nya.

### 3. 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 kesemua pengguna dari komputer client ke sumber informasi yaitu basis data.

#### **II.5.2. Manfaat/Kelebihan Basis Data**

Menurut Kusrini (2010:5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manfaat/Kelebihan Basis Data dan kelebihan basis data diantaranya adalah :

##### 1. 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.

##### 2. Kebersamaan pemakai (*shareability*)

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.

### 3. Pemusatan kontrol data

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

### 4. 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.

### 5. Keakuratan (*Accuracy*)

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

### 6. 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.

### 7. 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.

8. 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.

9. Pemakaian secara langsung

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

10. Kebebasan data

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

11. User View

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

### **II.5.3. Operasi Dasar Database**

Menurut Kusrini (2010:9) ada beberapa operasi dasar basis data diataranya yaitu:

1. Pembuatan basis data
2. Penghapusan basis data
3. Pembuatan file/tabel
4. Penghapusan file/tabel
5. Pengubahan tabel

6. Penambahan/pengisian

7. Pengambilan data

8. Penghapusan data

#### **II.5.4. 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) anomaly sebagai berikut :

##### *1. Insertion Anomaly*

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

##### *2. 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 :

##### *1. 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 baris dengan baris yang lain dalam satu table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

## 2. *Second Normal Form (2NF)*

Untuk menjadi 2NF suatu table harus berada dalam kondisi 1NF dan tidak memiliki *partial dependencies*. *Partial dependencies* adalah suatu kondisi jika atribut non kunci (Non PK) tergantung sebagian tetapi bukan seluruhnya pada PK.

## 3. *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.6. Visual Basic 2010

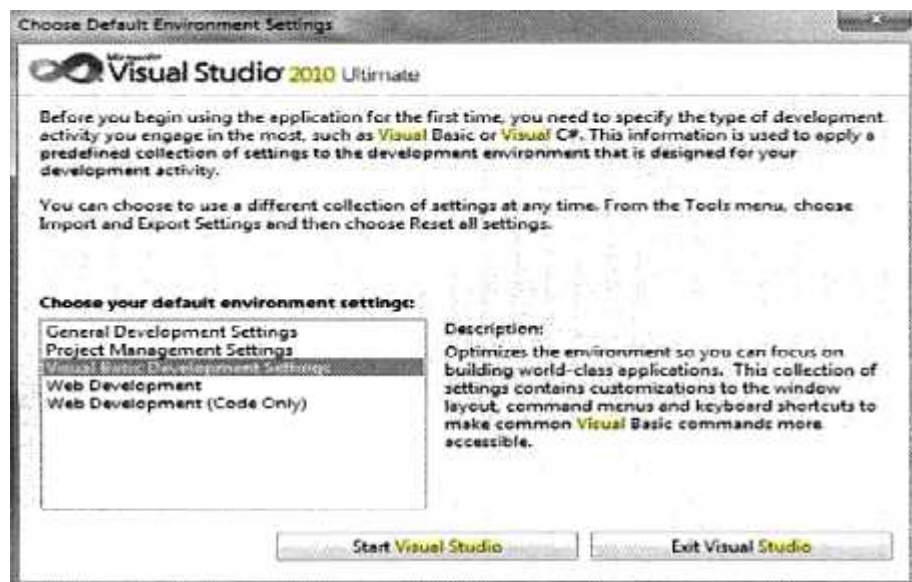
Menurut Edi Winarno, et al (2010:1), Visual Basic adalah bahasa pemrograman klasik, legendaris yang paling banyak dipakai oleh programmer di dunia. Pemrograman ini dipakai oleh jutaan programmer dan tercatat sebagai program yang paling disukai oleh mayoritas orang. Visual Studio 2010 pada dasarnya adalah 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.6.1. Antar Muka Visual Basic 2010

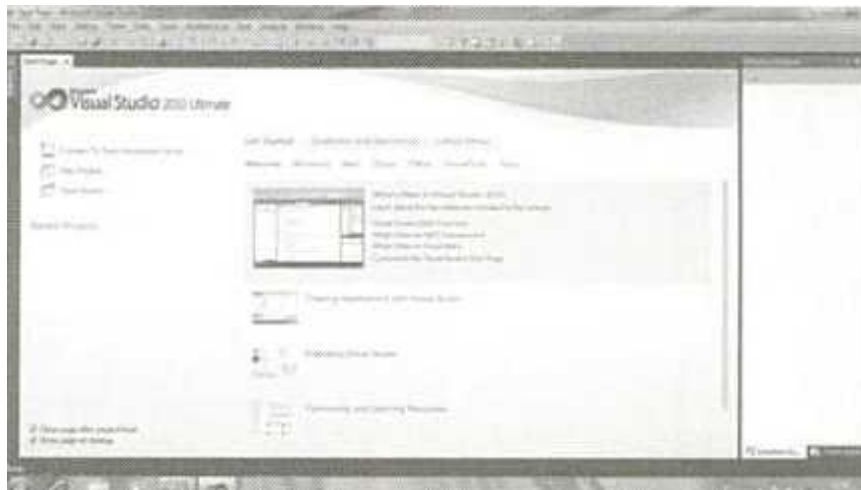
Saat menjalankan Visual Basic 2010 pertama kali muncul jendela *choose default environment settings*. Disini bisa memilih apakah ingin memilih antar muka di Visual Studio. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre*.



**Gambar II.11 Form Chose Default Enviromtent Settings**  
(Sumber : Edi Winarno, et al ; 2010)

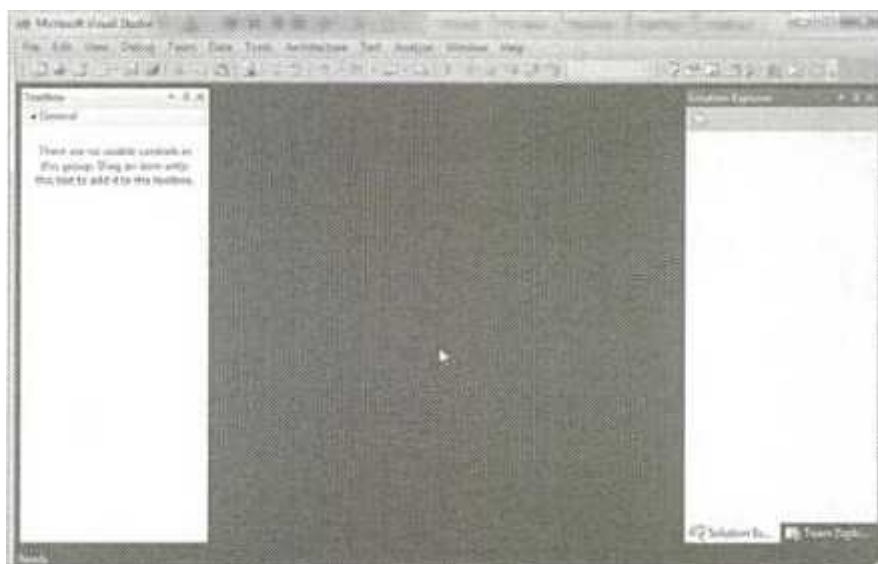


Dibagian awal visual basic, bisa memilih *Start Page*. *Start Page* adalah halaman yang mencantumkan informasi-informasi seputar program dan juga informasi RSS dari sumber tertentu. Jika tidak ingin menampilkan hal ini hilangkan tanda centang pada *Show Page On Startup*.



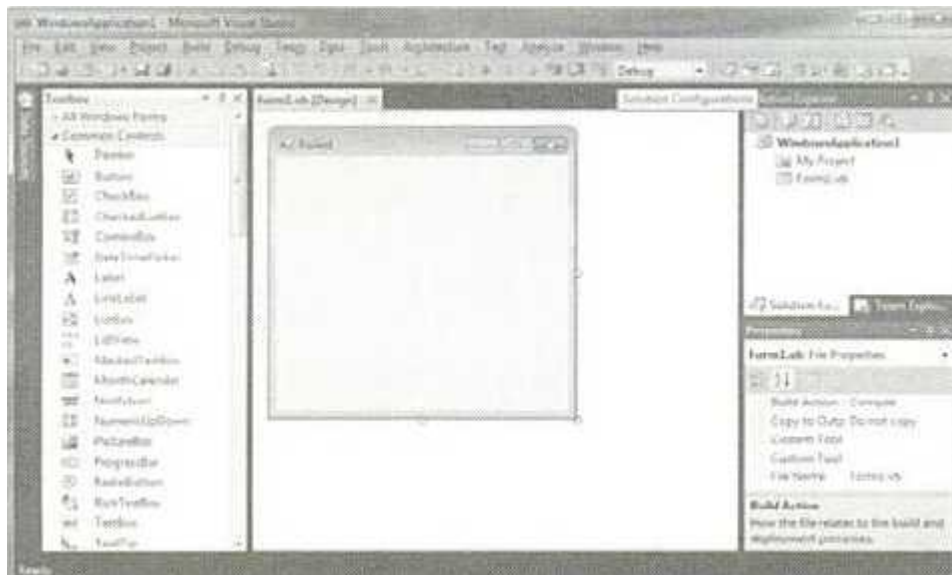
**Gambar II.12. Start Page Visual Basic 2010**  
(Sumber : Edi Winarno, et al ; 2010)

Jika *start page* ditutup terlihat tampilan sebagai berikut :



**Gambar II.13. Tampilan IDE (*Integrated Development Environment*) setelah *Start Page* ditutup**  
(Sumber : Edi Winarno, et al; 2010)

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



**Gambar II.14. Tampilan lengkap IDE  
(Sumber : Edi Winarno, et al; 2010)**

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 dkk, 2010:1)

## **II.7. 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.7.1 Kebutuhan *Hardware***

Adapun *hardware* yang diperlukan untuk instalasi SQL Server 2008 *Express Edition* minimal menurut Wahana Komputer (2010:2) adalah sebagai berikut :

1. Prosescor minimal 1 GHz
2. Memori minimal 512 MB
3. Sistem Operasi Windows

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

1. *Sharer Memory*
2. TCP/IP
3. *Named Pipes*
4. *Virtual Interface Adapter (VIA).*

### II.7.2 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 :

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

### II.7.3 Instalasi SQL Server 2008 *Express Edition*

Menurut Wahana Komputer (2010:2), 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 bertanggungjawab 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 system operasi.
- b. Mode Mixel atau campuran, mode ini menginjinkan *user* untuk masuk kedalam system SQL server dengan menggunakan *Account* yang dibuat di system operasi windows atau juga menggunakan *account* yang di *set up* pada SQL Server.