# BAB II

# LANDASAN TEORI

## II.1. Kecerdasan Buatan *(Artificial Intelegence)*

Menurut Muhammad Dahria (2008) Kecerdasan Buatan(Artificial Intelligence) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik dari pada yang dilakukan manusia.

Menurut John McCarthy, 1956, AI : untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik.

Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melkukan penalaran,mengambil kesimpilan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidask akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Demikian juga agar mesin bisa cerdas (bertindak seperti dan sebaik manusia) maka harus diberi bekal pengetahuan, sehingga mempunyai kemampuan untuk menalar. Untuk membuat aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan :

1. Basis Pengetahuan(Knowledge base), bersifat fakta-fakta, teori, pemikiran dan hubungan antar satu dengan yang lainnya.
2. Motor Inferensi(Inference Engine), kemampuan menarik kesimpulan pengetahuan dan pengalaman.

**II.2. 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”.

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.

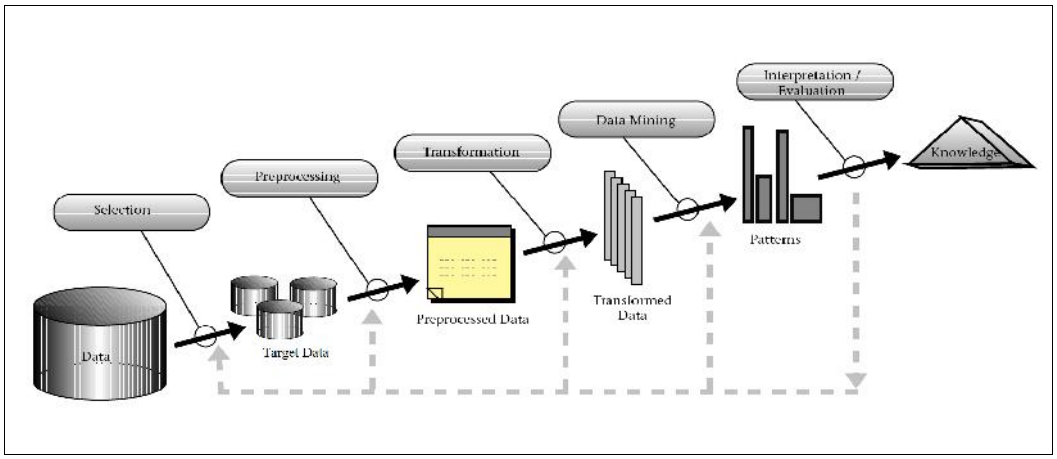
### II.2.1 Tahapan Data *Mining*

Menurut Liliana Swastina (2013:3), 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 memliki 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*

Proses *KDD* digambarkan pada Gambar 2.1 berikut:




**Gambar II.1 Proses *KDD***
***Sumber : Liliana Swastina (Penerapan Algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa)***

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 (2013:12),

### II.2.2. 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 kecelekaan 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 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.

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 :

Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respons positif terhadap penawaran *upgrade* layanan yang diberikan.

### II.2.3. Algoritma Decision Tree

Menurut Budanis Dwi Meilani Achmad (2012:18), Metode ini merupakan salah satu metode yang ada pada teknik klasifikasi dalam data mining. Metode pohon keputusan mengubah fakta yang sangat besar menjadi pohon keputusan yang merepresentasikan aturan. Pohon keputusan juga berguna untuk mengekplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target.

Data dalam pohon keputusan biasanya dinyatakan dalam bentuk tabel dengan atribut dan *record*. Atribut menyatakan suatu parameter yang disebut sebagai kriteria dalam pembentukan pohon. Misalkan untuk menentukan main tenis, kriteria yang diperhatikan adalah cuaca, angin, dan suhu. Salah satu atribut merupakan atribut yang menyatakan data solusi per *item* data yang disebut atribut hasil. Banyak algoritma yang dapat dipakai dalam pembentukan pohon keputusan, antara lain ID3, C4.5, CART.

Metode *Decision Tree* menggunakan Algoritma C4.5. Algoritma C4.5 merupakan pengembangan dari algoritma ID3. Algoritma C4.5 dan ID3 diciptakan oleh seorang peneliti dibidang kecerdasan buatan bernama j. Rose quinlan pada akhir tahun 1970-an. Algoritma C4.5 membuat pohon keputusan dari atas ke bawah, dimana atribut paling atas merupakan akar, dan yang paling bawah dinamakan daun.

Secara umum, algoritma C4.5 untuk membangun sebuah pohon keputusan adalah sebagai berikut:

1. Hitung jumlah data, jumlah data berdasarkan anggota atribut hasil dengan syarat tertentu. Untuk proses pertama syaratnya masih kosong.
2. Pilih atribut sebagai *Node*.

3. Buat cabang untuk tiap-tiap anggota dari Node.
4. Periksa apakah nilai *entropy* dari anggota Node ada yang bernilai nol. Jika ada, tentukan daun yang terbentuk. Jika seluruh nilai *entropy* anggota Node adalah nol, maka proses pun berhenti.
5. Jika ada anggota Node yang memiliki nilai *entropy* lebih besar dari nol, ulangi lagi proses dari awal dengan Node sebagai syarat sampai semua anggota dari Node bernilai nol.

*Node* adalah atribut yang mempunyai nilai *gain* tertinggi dari atribut-aribut yang ada. Untuk menghitung nilai *gain* suatu atribut digunakan rumus seperti yang tertera dalam persamaan berikut:

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

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|Si| : Jumlah kasus pada partisi ke-i

|S| : Jumlah kasus dalam S

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

$$Entropy(S) = \sum_{i=1}^{n} - pi * \log_2 pi \quad \text{..... (2)}$$

n : Jumlah partisi S

$p_i$ : Proporsi dari Si terhadap S

**II.3. Penjualan**

Menurut Sugiono Sugiharto, S.E., M.M (2013:2), penjualan adalah "Bagaimana menciptakan hubungan jangka panjang dengan pelanggan melalui

produk atau jasa perusahaan. Dalam hal ini, *selling* berarti sebuah taktik yang dapat mengintegrasikan perusahaan, pelanggan, dan relasi antara keduanya"

"*Selling* adalah suatu kegiatan yang ditujukan untuk mencari pembeli, mempengaruhi dan memberi petunjuk agar pembeli dapat menyesuaikan kebutuhannya dengan produk yang ditawarkan serta mengadakan perjanjian mengenai harga yang menguntungkan bagi kedua belah pihak"

Dalam prakteknya, kegiatan penjualan dipengaruhi oleh :

a. Kondisi dan kemampuan menjual

   Penjual harus dapat meyakinkan kepada pembelinya agar dapat berhasil mencapai sasaran penjualan yang diharapkan. Penjual harus memahami jenis karakteristik produk yang ditawarkan, harga produk, dan syarat penjualan seperti pembayaran, penghantaran, pelayanan purna jual, dan garansi.

b. Kondisi pasar

   Pasar sebagai kelompok pembeli atau pihak yang menjadi sasaran dalam penjualan. Faktor-faktor kondisi pasar yang perlu diperhatikan adalah jenis pasar, kelompok pembeli, segmen pasar, daya beli, frekuensi pembelian, keinginan dan kebutuhannya.

c. Modal

   Penjual harus memperkenalkan dulu atau membawa produknya kepada pembeli, diperlukan adanya sarana serta usaha seperti alat transport, tempat peragaan baik dalam perusahaan maupun di luar perusahaan, usaha promosi, dan lain-lain, dimana semuanya itu disebut dengan modal.

d. Kondisi organisasi perusahaan

Pada perusahaan kecil, jumlah tenaga kerjanya lebih sedikit, sistem organisasinya lebih sederhana, masalah-masalah yang dihadapi serta sarana yang dimilikinya tidak sekompleks perusahaan besar. Masalah penjualan ditangani sendiri oleh pimpinan dan tidak diberikan pada orang lain.

e. Faktor lain

Faktor-faktor lain umumnya seperti periklanan, peragaan, kampanye, pemberian hadiah sering mempengaruhi penjualan.

## II.4. *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 dipelajaran 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 system/perangkat lunak aplikasi) sebagai sarana analisis, pemasahaman visualisasi dan komunikasi antar anggota tim pengembang.

### II.4.1. Pengenalan UML

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 bias digunakan untuk memodelkan suatu system. 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 memungkin 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 system dan mempertahankan konsistensi antara desain dan implementasi dalam bahasa pemrograman.

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

### II.4.2. 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.

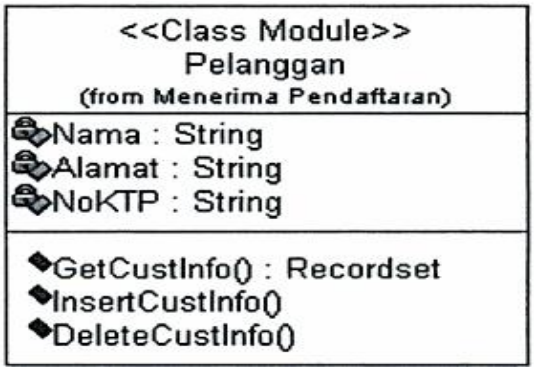
1. Aktor

   Aktor adalah segala sesuatu yang berinteraksi dengan system aplikasi computer. Jadi actor ini bisa berupa orang, perangkat keras atau juga objek lain dalam system yang sama. Biasanya yang dilakukan oleh Aktor adalah memberikan informasi pada system dan\atau memerintahkan system untuk melakukan sesuatu.

**Gambar II.2. Notasi Aktor**
***Sumber : Julius Hermawan (Analisis Desain dan Pemrograman Berorientasi Objek dengan UML dan Visual Basic.Net)***

2. *Class*

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




**Gambar II.3. Notasi Class**
***Sumber : Julius Hermawan (Analisis Desain dan Pemrograman Berorientasi Objek dengan UML dan Visual Basic.Net)***

   *Class* digunakan untuk mengabstraksikan elemen-elemen dari system 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.4. Notasi Interface**
***Sumber : Julius Hermawan (Analisis Desain dan Pemrograman Berorientasi Objek dengan UML dan Visual Basic.Net)***

4. *Use Case*

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

**Gambar II.5 Notasi *Use Case***
***Sumber : Julius Hermawan (Analisis Desain dan Pemrograman Berorientasi Objek dengan UML dan Visual Basic.Net)***

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 actor dan system 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 system 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*

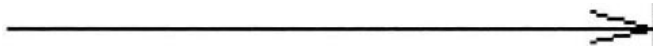
Menjelaskan persyaratan yang harus dipenuhi sebelum *use case* bisa dimulai.

*g. Post-Condition*

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

5. *Interaction*

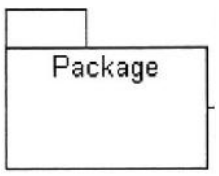
Digunakan untuk menunjukan baik aliran pesan maupun informasi antara objek maupun antara hubungan objek. Biasanya *Interaction* dilengkapi juga dengan teks bernama *operation signature* yang tersusun dari mana operasi, parameter yang dikirim dan type parameter yang dikembalikan.



**Gambar II.6. Notasi *Interaction***
***Sumber : Julius Hermawan (Analisis Desain dan Pemrograman Berorientasi Objek dengan UML dan Visual Basic.Net)***

6. *Package*

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



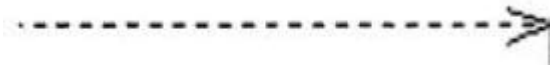

**Gambar II.7. Notasi *Package***
***Sumber : Julius Hermawan (Analisis Desain dan Pemrograman Berorientasi Objek dengan UML dan Visual Basic.Net)***

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*

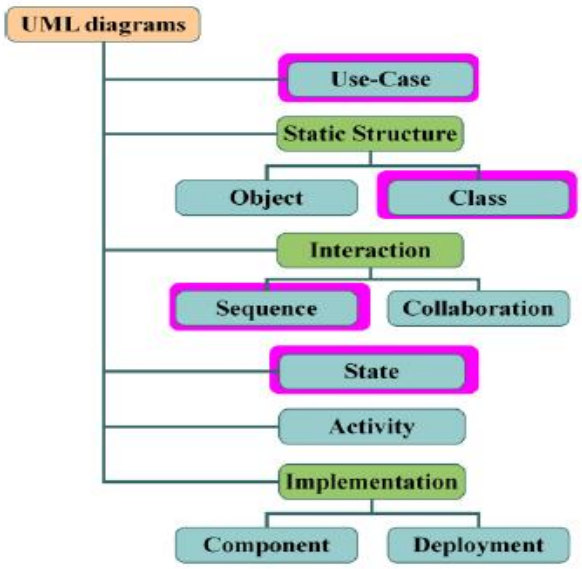
   *Dependency* merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain.



**Gambar II.8. Notasi *Dependency***
***Sumber : Julius Hermawan (Analisis Desain dan Pemrograman Berorientasi Objek dengan UML dan Visual Basic.Net)***

### II.4.3. Diagram UML

Menurut Haviluddin (2011:2), Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. Berikut gambar dari diagram UML



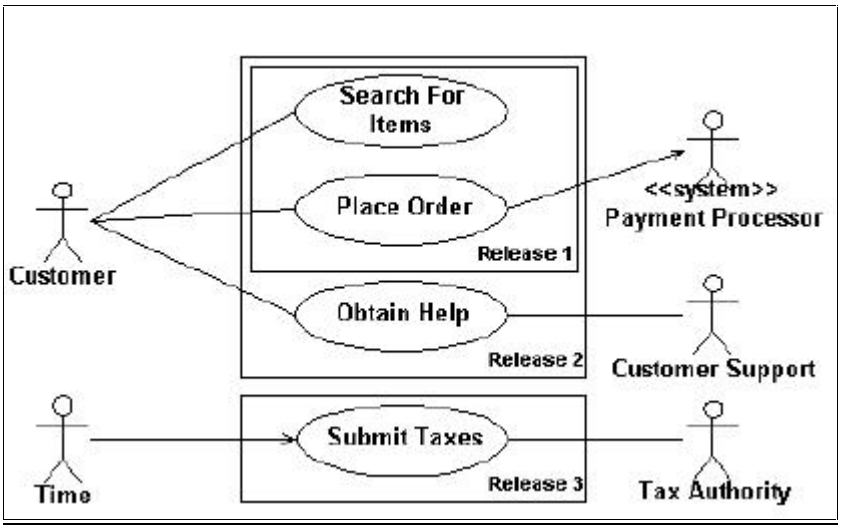
**Gambar II.9 Diagram UML**
***Sumber : Haviluddin(Memahami Penggunaan UML (Unified Modelling Language)***

1. *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 :

a. *System use case*; interaksi dengan sistem.

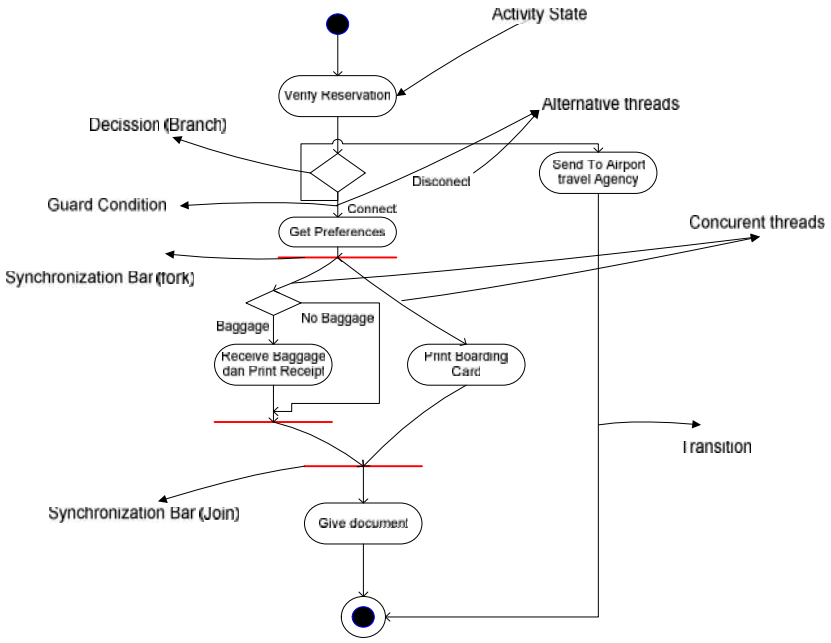
b. *Business use case;* interaksi bisnis dengan konsumen atau kejadian nyata




**Gambar II.10. Contoh *use case diagram*.**
***Sumber : Haviluddin(Memahami Penggunaan UML (Unified Modelling Language)***

2. *Activity* diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas




**Gambar II.11 Contoh *Activity* diagram.**
***Sumber : Haviluddin(Memahami Penggunaan UML (Unified Modelling Language)***

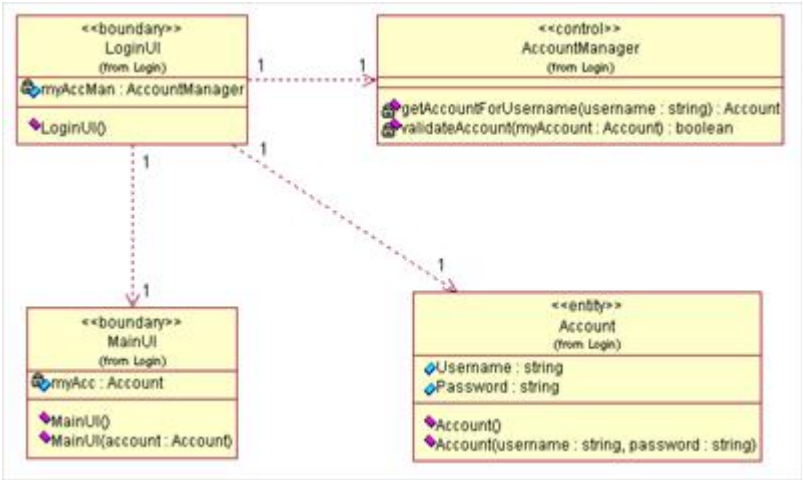
3. *Class* diagram

*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 :

a. Nama (dan *stereotype*)

b. Atribut

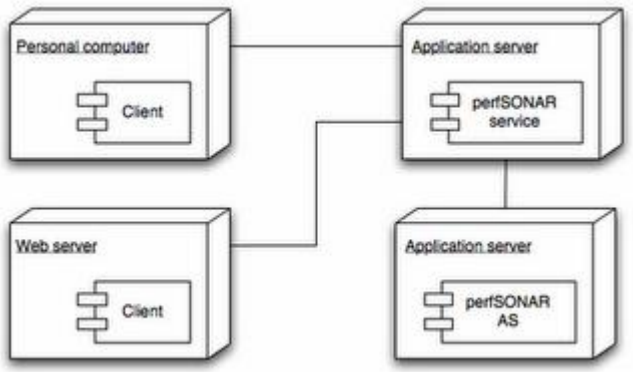
c. Metoda




**Gambar II.12 Contoh *Class* diagram.**
***Sumber : Haviluddin(Memahami Penggunaan UML (Unified Modelling Language)***

4. *Deployment* diagram

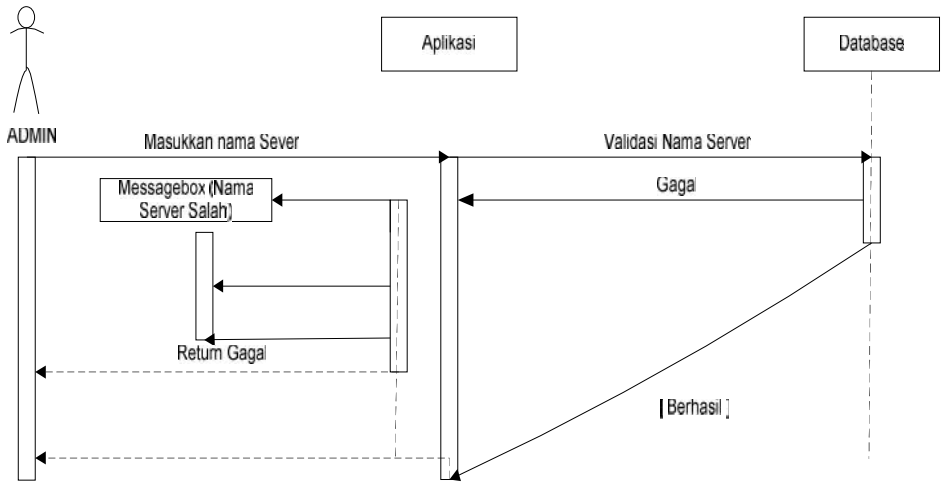
*Deployment* diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.




**Gambar II.13 Contoh *Deployment* diagram.**
***Sumber : Haviluddin(Memahami Penggunaan UML (Unified Modelling Language)***

5. *Sequence* diagram

*Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram




**Gambar II.14 Contoh *Sequence* diagram.**
***Sumber : Haviluddin(Memahami Penggunaan UML (Unified Modelling Language)***

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

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

Menurut Kusrini (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 symbol).

Basis data dapat didefinisikan dalam berbagai susut 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 Kusrini (2010, p2), Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan dan kecepatan dalam pengambilan kembali. Untuk mencapat tujuan, syarat basisdata 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 lainya, 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 DBMNnya.

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

### II.5.2. Manafaat/Kelebihan Basis Data

Menurut Kusrini (2010, p5), Banyak manfaat yang diperoleh dengan menggunakan basis data, Manafaat/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 memilki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan metematika. 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. Efesiensi 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 ketidakakuratan 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 kemmapuan dari DBMS. Sehingga membuat aplikasi tidak perlu mengurusi penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

i. Pemakaian secara langsung

Basis data memiliki fasilitas yang lengkap untuk melihat datnya 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 penguna.

### II.5.3. Operasi Dasar Database

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

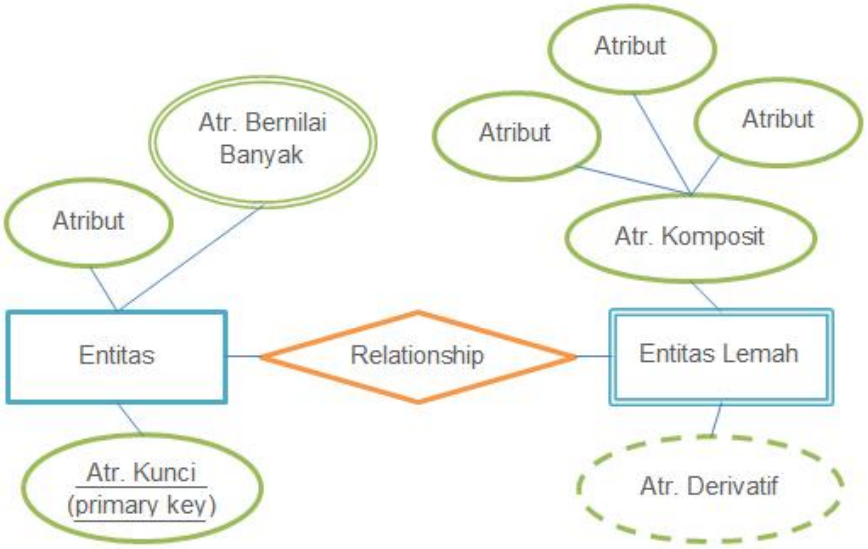
a. Pembuatan basis data
b. Penghapusan basis data
c. Pembuatan file/tabel
d. Penghapusan file/tabel
e. Pengubahan tabel
f. Penambahan/pengisian
g. Pengambilan data
h. Penghapusan data

### II.5.4. 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 variable sebelum diaplikasikan ke system yang berjalan. Untuk merancang suatu aplikasi basis data alat yang biasa

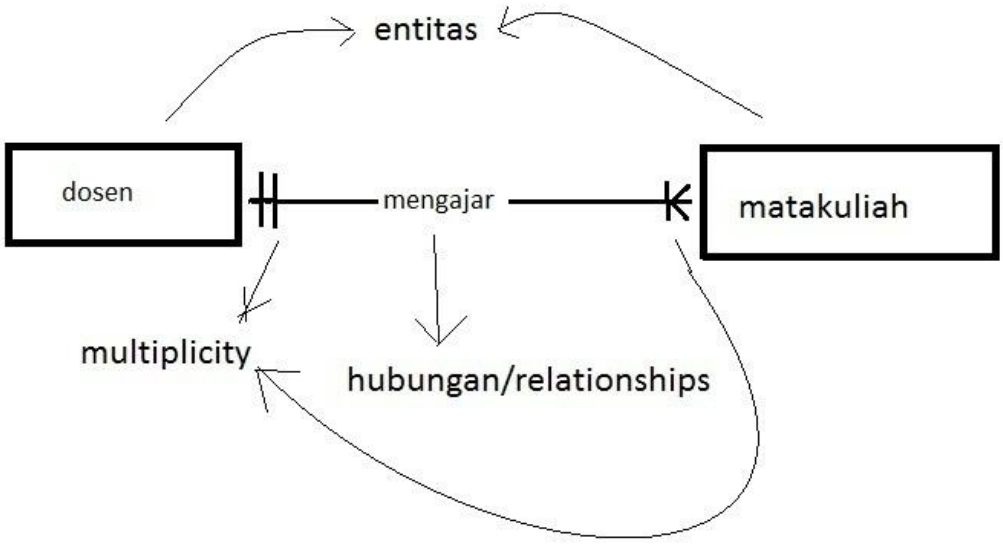
digunakan adalah *Entity Relationship Diagram (ERD).* ERD didasarkan dari artikel yang dipublikasikan oleh Peter Phin Shan Chen.

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



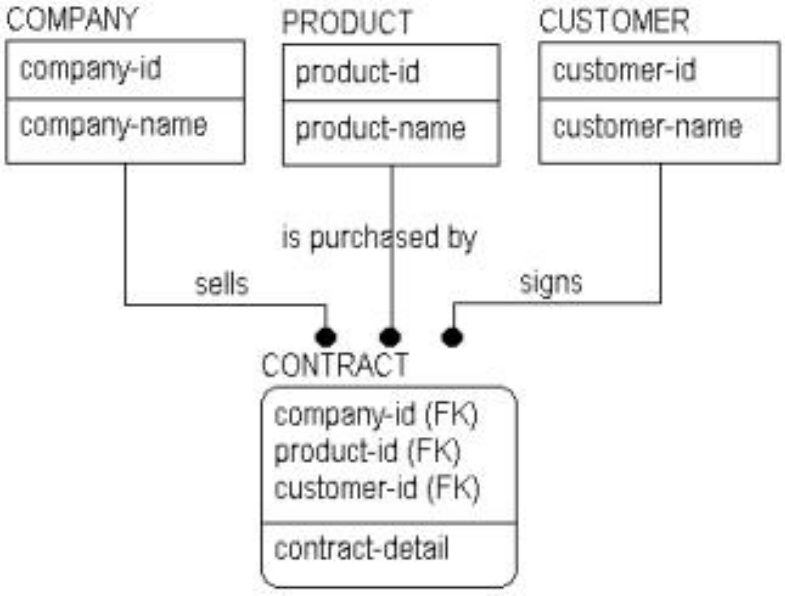

**Gambar II.15. Diagram Dengan Notasi Chen ERD**
***Sumber : Samiaji Sarosa (Sistem Informasi Akuntansi)***

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



**Gambar II.16. Diagram Dengan Notasi Crows Foot**
***Sumber : Samiaji Sarosa (Sistem Informasi Akuntansi)***

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




**Gambar II.17. Diagram Dengan Notasi Crows Foot**
***Sumber : Samiaji Sarosa (Sistem Informasi Akuntansi)***

### II.5.5. 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 Nornal 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 table. Pada dasarnya sebuah table selamat tidak ada kolom yang sama merupakan bentuk table dengan 1NF.

b. *Second Nornal Form (2NF)*

   Untuk menjadi 2NF suatu table 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 Nornal Form (3NF)*

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

## II.6. Visual Basic 2010

Menurut Edi Winarno ST, M.Eng dkk (2010:1), Visual Basic adalah bahasa pemrogram 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.

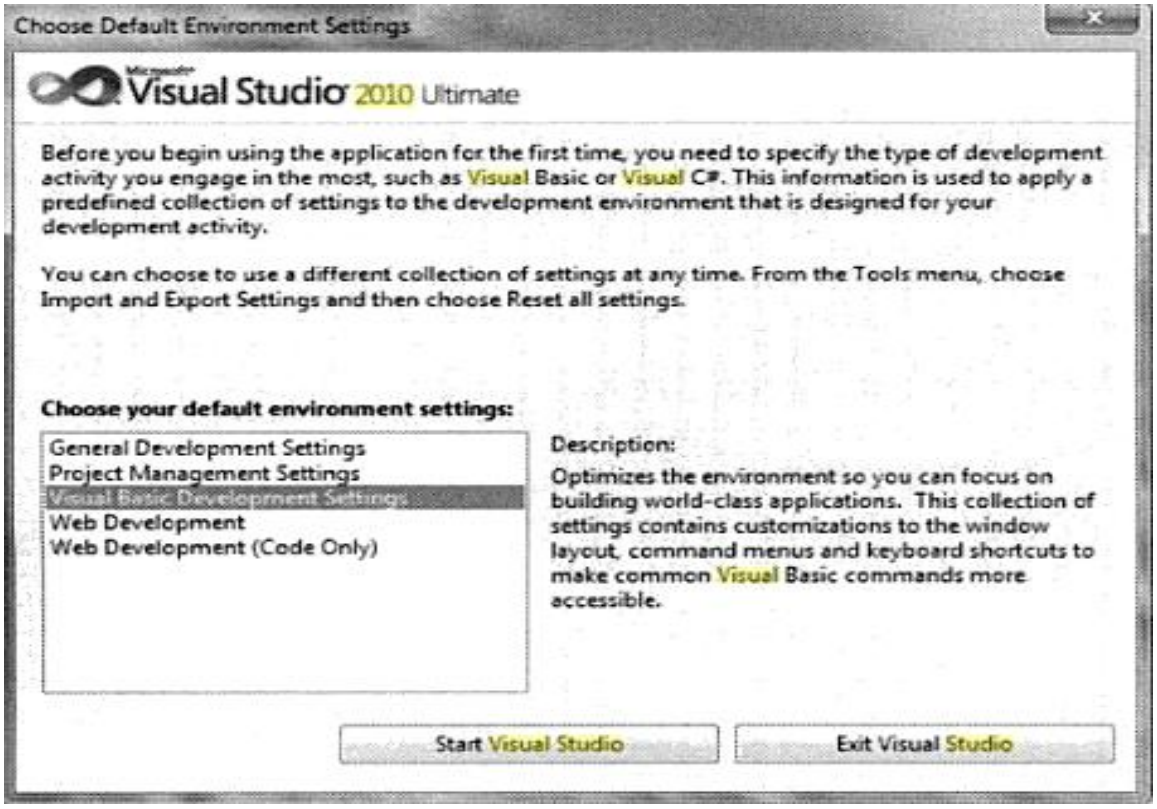
Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 selain disebut dengan bahasa pemrograman, juga sering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

Beberapa kemampuan atau manfaat dari Visual Studio 2010 diantaranya seperti :

1. Untuk membuat program aplikasi berbasiskan 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 *chose default environtment settings.* Disini bisa memilih apakah ingin memilih antar muka di Visual Studio. Untuk *programmer* Visual Basic lebih baik memilih *Visual Basic Development Centre.*

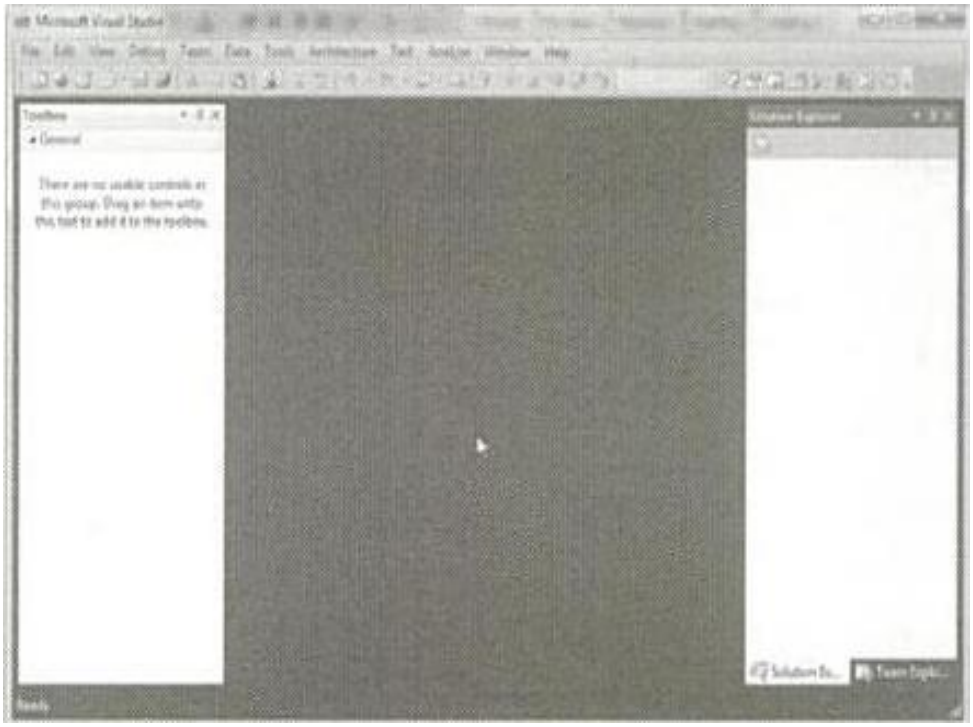



**Gambar II.18 Form Chose Default Environtment Settings**
***Sumber : Edi Winarno(Dasar-Dasar Pemrograman Dengan Visual Basic 2010)***

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.*

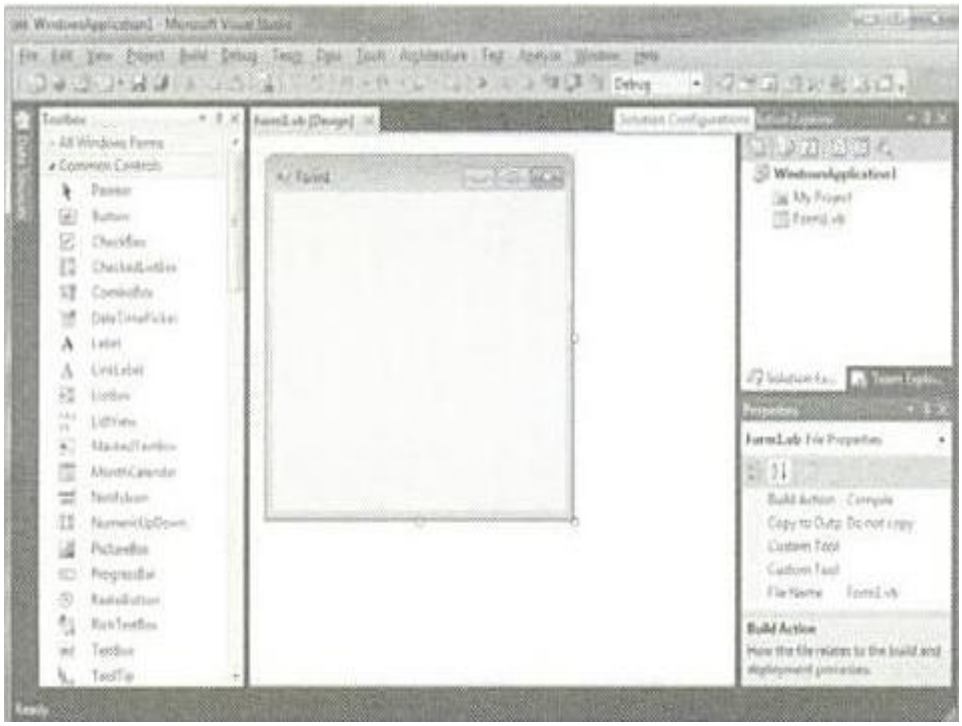
**Gambar II.19. Start Page Visual Basic 2010**
***Sumber : Edi Winarno(Dasar-Dasar Pemrograman Dengan Visual Basic 2010)***

Jika *start page* ditutup terlihat tampilan sebagai berikut :



**Gambar II.20. Tampilan IDE (*Integrated Development Environtment)* setelah *Start Page* ditutup**
***Sumber : Edi Winarno(Dasar-Dasar Pemrograman Dengan Visual Basic 2010)***

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



**Gambar II.21. Tampilan lengkap IDE**
***Sumber : Edi Winarno(Dasar-Dasar Pemrograman Dengan Visual Basic 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 ST, M.Eng 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 adalah sebagai berikut :

a. Processcor minimal 1 GHz
b. Memori minimal 512 MB
c. Sistem Operasi Windows

Biar dapat diinstal pada system computer dengan memoti 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.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 adalag sebagai berikut :

a. Menurut cara pemrosesan data pada prosesor makan 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 computer yang lebih sari satu processor (Misalnya *Core 2 duo)* dan system operasi 64 bit, Vista dan Windows 7.
b. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut :
    1. Versi *Compact* ini adalah versi “tipis” dari semu versi yang ada
    2. Versi *Express* ini adalah versi “ringan”

### II.7.3 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 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 (Wahana Komputer, 2010:2)