

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. (Asbon hendra : 2012 : 157).

##### **II.1.1. Karakteristik Sistem**

Menurut (Asbon hendra : 2012 : 158-160) Ada beberapa karakteristik sistem adalah sebagai berikut :

a. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang memengaruhi operasi dari suatu sistem.

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya.

e. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi.

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem meliputi output yang berguna.

g. Pengolah Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang di inginkan.

h. Tujuan Sistem (*Goal*)

Suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya.

## **II.2. Informasi**

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat.

(Asbon hendra : 2012 : 167).

### **II.3. Sistem informasi**

Sistem informasi adalah sekumpulan prosedur manual atau terkomputerisasi yang mengumpulkan/mengambil, mengolah, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan. (Asbon hendra : 2012 : 168-169).

### **II.4. Sistem Informasi Akuntansi**

Menurut Anastasia Diana dan Lilis Setiawati (2011 : 4) Sistem informasi akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari sistem informasi akuntansi pada sebuah toko baju, seperti pada contoh sebelumnya adalah transaksi penjualan.

### **II.5. Biaya Operasional**

Biaya operasional atau biaya komersial merupakan biaya yang memiliki peran besar dalam mempengaruhi keberhasilan perusahaan mencapai tujuannya, yaitu memperoleh laba usaha. Tanpa aktivitas operasional yang terarah maka seluruh produk yang dihasilkan tidak akan memiliki manfaat apapun bagi perusahaan. Secara umum, biaya operasional dibagi ke dalam 2 kelompok besar, yaitu biaya pemasaran (*marketing cost*) dan biaya administrasi dan umum. (Jurnal EMBA : 2013 : 695)

## **II.6. Metode *Multiple Step* (Langkah Bertahap)**

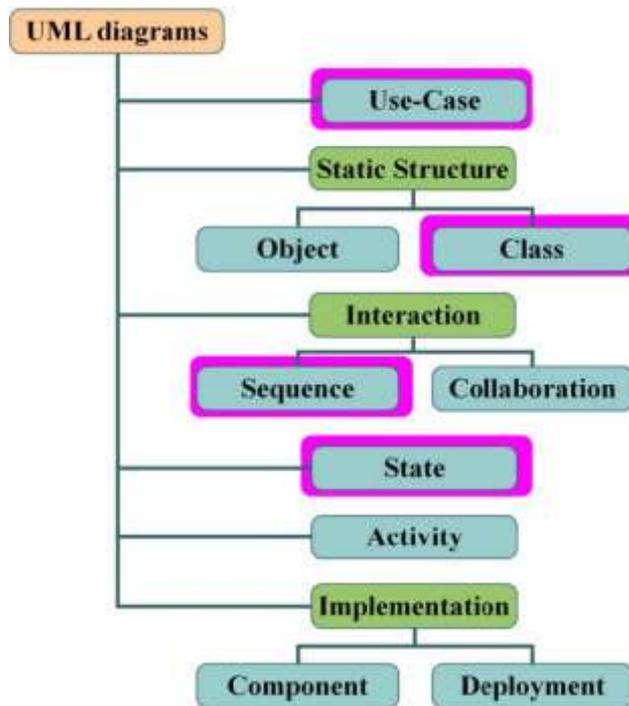
Pada langkah bertahap, beban penjualan, beban umum dan administrasi, serta pendapatan dan beban lain-lain, dibedakan satu sama lain dan dikelompokkan sesuai dengan kelompok masing-masing, sehingga dalam laporan laba rugi terlihat dengan jelas jumlah beban usaha dan beban di luar usaha, serta pendapatan usaha dan pendapatan di luar usaha. Metode pengelompokan beban dan pendapatan semacam ini dapat memudahkan pihak manajemen menganalisis penyimpangan beban dan pendapatan. (www.e-booksPDF.org)

## **II.7. UML ( *Unified Modeling Language* )**

*Unified Modelling Language* (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek.

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. (Haviluddin; 2011: 01).

Berikut gambar dari diagram UML



**Gambar II.1. Diagram UML**

Sumber : Haviluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 02

### II.7.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schmuller J. (2004), *“The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model”*.

Berikut tujuan utama dalam desain UML adalah (Haviluddin; 2011: 02) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.

3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

### **II.7.2. Komponen-komponen UML**

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik.

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*).

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu (Haviluddin; 2011: 03) :

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

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

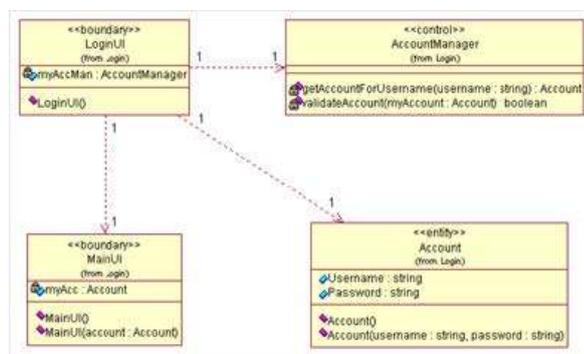
Struktur diagram dalam UML terdiri atas :

### 1) *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



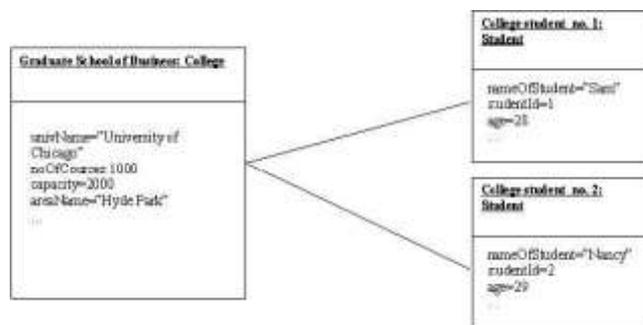
**Gambar II.2. Notasi class diagram**

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 03

## 2) *Object diagram*

*Object diagram* menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.

Berikut notasi *object diagram*.

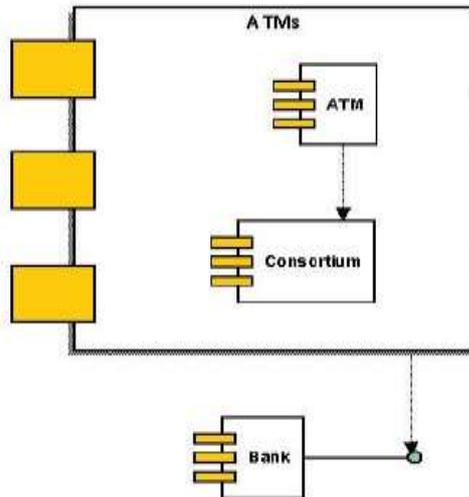


**Gambar II.3. Notasi *object diagram***

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 03

## 3) *Component diagram*

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



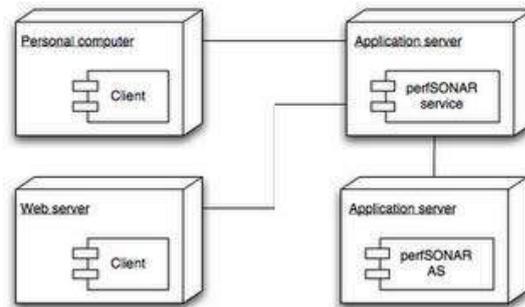
**Gambar II.4. Notasi *component diagram***

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling*

*Language*); 2011 : 03

4) *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.

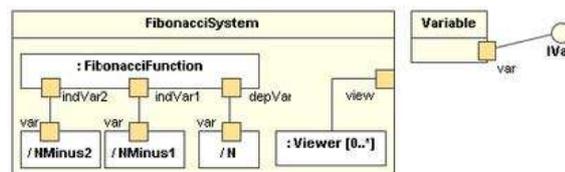


**Gambar II.5. Notasi *deployment diagram***

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 04

5) *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.

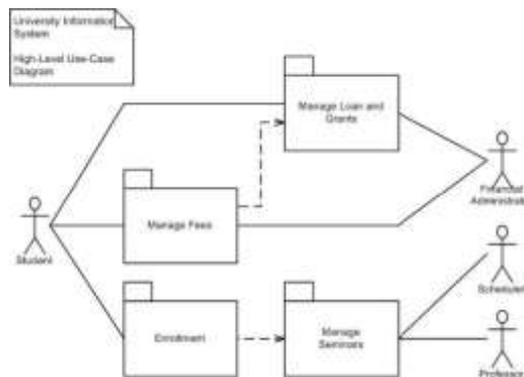


**Gambar II.6. Notasi *composite diagram***

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 04

## 6) *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.



**Gambar II.7. Notasi *Package diagram***

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 04

## 2. *Behavior Diagram*

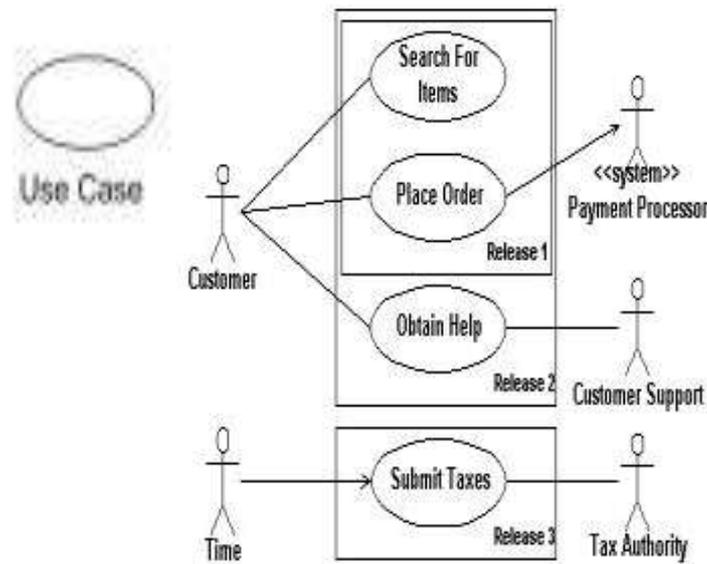
Menggambarkan ciri-ciri *behavior/metode/* fungsi dari sebuah sistem atau *business process*. *Behavior diagram* dalam UML terdiri atas

### 7) *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 :

1. *System use case*; interaksi dengan sistem.
2. *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata

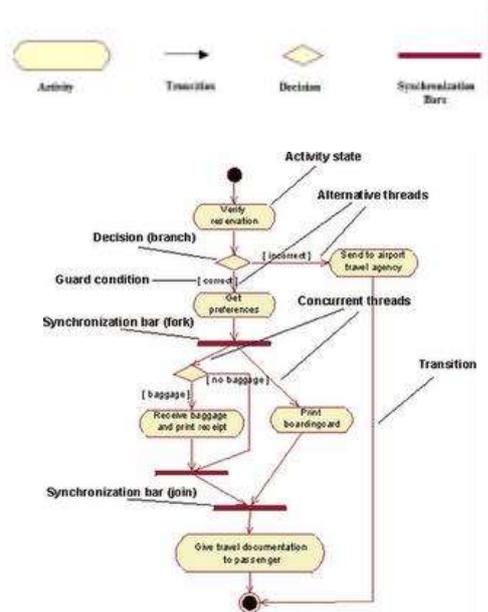


**Gambar II.8. Notasi *Use case diagram***

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 04

#### 8) *Activity diagram*

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

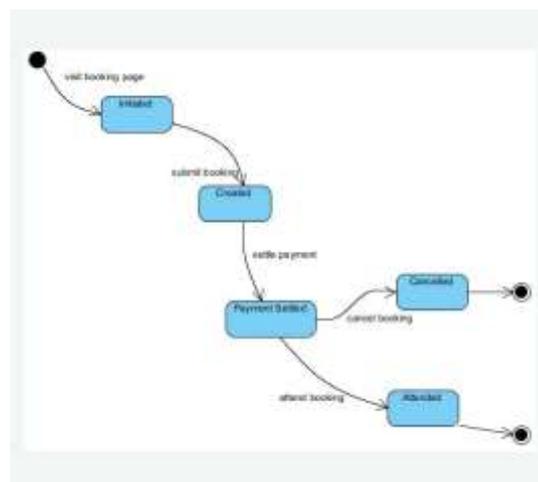


**Gambar II.9. Notasi Activity diagram**

Sumber : Haviluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 04

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

Menggambarakan *state*, transisi *state* dan *event*.



**Gambar II.10. Notasi State Machine diagram**

Sumber : Haviluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 05

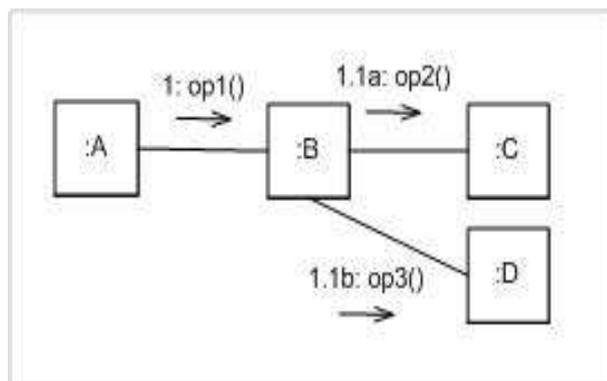
### 3. Interaction diagram

Bagian dari behavior diagram yang menggambarkan interaksi objek.

Interaction diagram dalam UML terdiri atas :

#### 10) *Communication diagram*

Serupa dengan sequence diagram, tetapi diagram komunikasi juga digunakan untuk memodelkan perilaku dinamis dari use case. Bila dibandingkan dengan Sequence diagram, diagram komunikasi lebih terfokus pada menampilkan kolaborasi benda daripada urutan waktu.

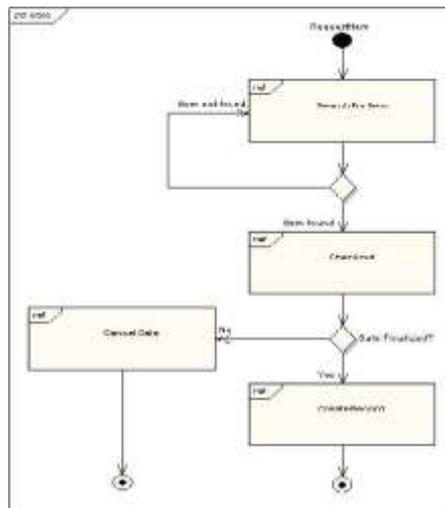


**Gambar II.11. Notasi *communication diagram***

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 05

### 11) *Interaction Overview diagram*

*Interaksi overview diagram* berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi.

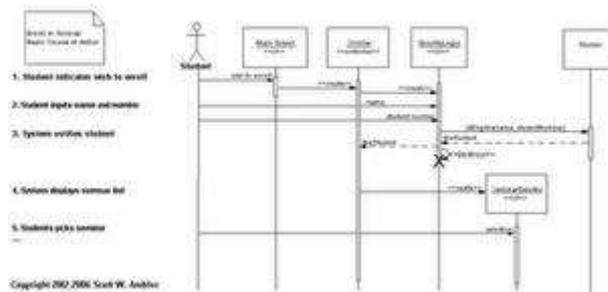


**Gambar II.12. Notasi overview diagram**

Sumber : Havaluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 05

### 12) *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*.

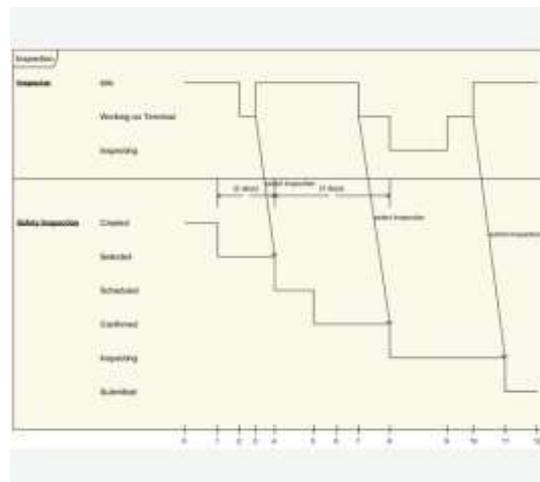


**Gambar II.13. Notasi *Sequence diagram***

Sumber : Haviluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 05

### 13) *Timing diagram*

*Timing diagram* di UML didasarkan pada diagram waktu hardware awalnya dikembangkan oleh para insinyur listrik.



**Gambar II.14. Notasi *Timing diagram***

Sumber : Haviluddin, Memahami Penggunaan UML (*Unified Modelling Language*); 2011 : 05

## **II.8. Microsoft Visual Basic 2010**

Wahana Komputer (2011 : 2-3) Visual basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh Microsoft, yaitu Microsoft Visual Studio 2010. Visual studio merupakan produk pemrograman andalan dari Microsoft Cooperation, dimana di dalamnya berisi beberapa jenis IDE pemrograman seperti Visual Basic, Visual C++, Visual Web Developer, Visual C#, dan Visual F#.

Semua IDE pemrograman tersebut sudah mendukung penuh implementasi .Net Framework terbaru, yaitu .Net Framework 4.0 yang merupakan pengembangan dari .Net Framework 3.5. Adapun database standart yang disertakan adalah Microsoft SQL Server 2008 express. Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu Visual Basic 2008.

## **II.9. Microsoft SQL Server 2008**

Wahana Komputer (2010 : 2) SQL Server 2008 adalah sebuah terobosan baru dari Microsoft 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 dibuat pada saat kemajuan dalam bidang hardware sedemikian pesat. Oleh karena itu, sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data.

Database atau basis data adalah sekumpulan data yang memiliki hubungan secara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah representasi dari semua fakta yang ada pada dunia nyata. Database sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu.

### **II.9.1. Entity Relationship Diagram**

Wahana Komputer (2010 : 30) Pada dasarnya ERD (Entity Relationship Diagram) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD diatas, ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity tersebut.

### **II.9.2. Normalisasi Data**

Wahana Komputer (2010 : 32-35) Normalisasi data adalah proses dimana tabel-tabel pada database dites dalam hal kesalingtergantungan di antara field-field pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu field dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel. Banyaknya tabel pecahannya tergantung pada seberapa banyak ketergantungannya. Tiap tabel hanya boleh memiliki sebuah field kunci yang menjadi ketergantungan dari field lainnya dalam tabel tersebut.

Pada proses normalisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci.