

## **BAB II**

### **LANDASAN TEORI**

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

Sistem merupakan sekumpulan elemen-elemen yang saling terintegrasi serta melaksanakan fungsinya masing-masing untuk mencapai tujuan yang telah ditetapkan. Karakteristik sistem terdiri dari :

##### 1. Komponen Sistem

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

##### 2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

##### 3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

#### 4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

#### 5. Masukan Sistem

Masukkan sistem adalah energi yang di masukkan ke dalam sistem. Masukkan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

#### 6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

#### 7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

#### 8. Sasaran Sistem

Suatu sistem dapat mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu

sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.(Sulindawati : 2010 ; 1-2).

## **II.2. Sistem Pendukung Keputusan**

Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan memanipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Sistem pendukung keputusan biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. Sistem pendukung keputusan yang seperti itu disebut aplikasi Sistem pendukung keputusan. Aplikasi Sistem pendukung keputusan digunakan dalam pengambilan keputusan. Aplikasi menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen yang tidak terstruktur.

Aplikasi Sistem pendukung keputusan menggunakan data, memberikan antar muka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan. Sistem pendukung keputusan lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas. Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan

keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

1. Membantu manajer dalam pengambilan keputusan atas masalah semistruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang diambil lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya rendah.
5. Peningkatan produktivitas.
6. Dukungan kualitas.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. (Sylvia Hartati Saragih: 2013; 2)

### **II.2.1. Karakteristik SPK (Sistem Pendukung Keputusan)**

Dari pengertian Sistem Pendukung Keputusan maka dapat ditentukan karakteristik antara lain :

1. Mendukung proses pengambilan keputusan, menitik beratkan pada *managment by perception*.

2. Adanya *interface* manusia atau mesin dimana manusia (*user*) tetap memegang control proses pengambilan keputusan.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur dan tak terstruktur.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan.
5. Memiliki subsistem-subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan item.
6. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manajemen.

## **II.2.2. Komponen-Komponen Sistem Pendukung Keputusan**

Komponen-komponen sistem pendukung keputusan dapat diuraikan dalam beberapa komponen yang terdiri dari :

1. *Data Management* termasuk *database* yang mendukung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System (DBMS)*.
2. *Model Management* melibatkan model finansial, statiscal, *managment science* atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis dan *manajemen software* yang diperlukan.
3. Dialog subsistem (*Communication*), *user* dapat berkomunikasi dan memberikan perintah pada DSS (*Decision Support System*) melalui subsistem ini, ini berarti menyediakan antarmuka.

4. *Knowledge Management*, subsistem *optional* ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

### **II.2.3. Keuntungan Sistem Pendukung Keputusan**

Dengan berbagai karakter khusus yang dimiliki Sistem Pendukung Keputusan, SPK (Sistem Pendukung Keputusan) dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah :

1. SPK memperluas kemampuan pengambilan keputusan dalam memproses data atau informasi bagi pemakainya.
2. SPK membantu pengambil keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu SPK, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun ia dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan.

### **II.3. Metode *Technique For Order Preference by Similarity to Ideal Solution* (Topsis)**

TOPSIS adalah salah satu metode pengambilan keputusan multi kriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang. TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi

ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negative - ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.

TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relative terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan. Hal ini disebabkan konsepnya sederhana, mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relatif dari alternatif - alternatif keputusan.

Berikut langkah-langkah proses perhitungan metode TOPSIS :

1. Membangun *normalized decision matrix*

Elemen  $r_{ij}$  hasil dari normalisasi *decision matrix*  $R$  dengan metode

*Euclidean length of a vector* adalah:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Dimana:

$r_{ij}$ =Hasil dari normalisasi matriks keputusan  $R$

$i=1,2,3,\dots,m;$

$j=1,2,3,\dots,n$

## 2. Membangun *weighted normalized decision matrix*

Dengan bobot  $W=(w_1, w_2, \dots, w_n)$ , maka normalisasi bobot matriks  $V$  adalah:

$$V = \begin{bmatrix} w_{11}r_{11} & \cdots & w_{1n}r_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1}r_{m1} & \cdots & w_{nm}r_{nm} \end{bmatrix}$$

## 3. Menentukan solusi ideal positif dan solusi ideal negatif.

Solusi ideal positif dinotasikan dengan  $A^+$  dan solusi ideal negatif dinotasikan dengan  $A^-$ , sebagai berikut:

Menentukan Solusi Ideal (+) & (-)

$$A^+ = \left\{ \left( \max v_{ij} \right) \left( \min v_{ij} \mid j \in J' \right), i = 1, 2, 3, \dots, m \right\} = \{v_1^+, v_2^+, \dots, v_m^+\}$$

$$A^- = \left\{ \left( \max v_{ij} \right) \left( \min v_{ij} \mid j \in J' \right), i = 1, 2, 3, \dots, m \right\} = \{v_1^-, v_2^-, \dots, v_m^-\}$$

Dimana :

$v_{ij}$  = elemen matriks  $V$  baris ke- $i$  dan kolom ke- $j$

$J = \{j=1, 2, 3, \dots, n \text{ dan } j \text{ berhubungan dengan } \textit{benefit criteria}\}$

$J' = \{j=1, 2, 3, \dots, n \text{ dan } j \text{ berhubungan dengan } \textit{cost criteria}\}$

## 4. Menghitung Separasi

*Separation measure* ini merupakan pengukuran jarak dari suatu alternative ke solusi ideal positif dan solusi ideal negatif. *Separation measure* untuk solusi ideal positif.

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}, \text{ dengan } i=1, 2, 3, \dots,$$

Dimana:

$$J = \{j=1,2,3,\dots,n \text{ dan } j \text{ merupakan } \textit{benefit criteria}\}$$

$$J' = \{j=1,2,3,\dots,n \text{ dan } j \text{ merupakan } \textit{cost criteria}\}$$

5. Menghitung kedekatan relative terhadap solusi ideal.

Kedekatan relatif dari alternatif  $A^+$  dengan solusi ideal  $A^-$  direpresentasikan dengan:

$$C_i = \frac{S_i^-}{S_i^- + S_i^+}, \text{ dengan } 0 < C_i < 1 \text{ dan } i=1,2,3,\dots,m$$

6. Merangking Alternatif

Alternatif dapat dirangking berdasarkan urutan  $C_i$ . Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terpendek terhadap solusi ideal dan berjarak terjauh dengan solusi ideal negatif.

#### II.4. Metode *Weighted Product*

Metode WP merupakan salah satu metode penyelesaian yang ditawarkan untuk menyelesaikan masalah *Multi Attribute Decision Making* (MADM). Metode WP mirip dengan Metode *Weighted Sum* (WS), hanya saja metode WP terdapat perkalian dalam perhitungan matematikanya. Metode WP juga disebut analisis berdimensi karena struktur matematikanya menghilangkan satuan ukuran. Metode WP adalah himpunan berhingga dari alternatif keputusan yang dijelaskan dalam beberapa hal kriteria keputusan. Jadi metode ini tidak perlu dinormalisasikan. Preferensi  $A_i$  diberikan pada Rumus 1 :

$$S_i = \prod_{j=1}^n x_{ij}^{w_j}, \text{ dengan } i = 1, 2, \dots, m \quad \dots\dots\dots(1)$$

dimana  $\sum w_j = 1$ .  $w_j$  adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.

Preferensi relative dari setiap alternatif, diberikan pada Rumus 2 :

$$V_i = \frac{\prod_{j=1}^n x_{ij}^{w_j}}{\prod_{j=1}^n (x_j^w)^{w_j}} ; \text{ dengan } i = 1, 2, \dots, m. \quad \dots\dots\dots(2)$$

Langkah-langkah menggunakan metode WP :

1. Mengalikan seluruh atribut bagi sebuah alternatif dengan bobot sebagai pangkat positif untuk atribut manfaat dan bobot berfungsi sebagai pangkat negatif pada atribut biaya.
2. Hasil perkalian dijumlahkan untuk menghasilkan nilai pada setiap alternatif.
3. Mencari nilai alternatif dengan melakukan langkah yang sama seperti langkah satu, hanya saja menggunakan nilai tertinggi untuk setiap atribut tertinggi untuk setiap atribut manfaat dan terendah untuk atribut biaya.
4. Membagi nilai  $V$  bagi setiap alternatif dengan nilai standar ( $V(A^*)$ ) yang menghasilkan  $R$ .
5. Ditemukan urutan alternatif terbaik yang akan menjadi keputusan.

## II.5. SQL Server

Menurut (Emma Utami & Sukrisno 2008 : 34) *SQL (Structured Query Language)* adalah bahasa komputer standar yang ditetapkan untuk mengakses dan memanipulasi sistem *database*. *SQL Server 2008* adalah teknologi yang

mendukung *development* dan administrasi dari *Business Intelligence* (BI) *Application*. SQL Server Reporting and Integration servis adalah element dari BI, tapi inti dari BI tersebut adalah SQL Server 2008 Analysis Servis (SSAS). *SQL Server 2008* adalah sebuah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. (GI MDP : 2011 ; 2)

## **II.6. Microsoft Visual Studio**

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

### **II.6.1. Mengenal Microsoft Visual Basic 2010**

*Visual Basic 2010* merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat di dalamnya antara lain dukungan terhadap library terbaru *Microsoft*, yaitu *.Net Framework 4.0*, dukungan terhadap pengembangan aplikasi menggunakan *Microsoft SilverLight*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap database-database, baik *standalone* maupun *database server*. Bahasa *Visual Basic 2010* sendiri awalnya berasal dari bahasa pemrograman yang sangat populer di kalangan programmer

komputer, yaitu bahasa *BASIC*, yang oleh Microsoft diadaptasi dalam program *Microsoft Quick BASIC*. Seiring dengan berkembangnya teknologi komputasi dan desain, Microsoft mengeluarkan produk yang dinamakan *Microsoft Visual Studio* dengan *Visual Basic* di dalamnya. Saat ini versi *Microsoft Visual Studio* yang beredar adalah versi 10 yang populer dengan nama *Microsoft Visual Studio 2010*, yang di dalamnya termasuk *Microsoft Visual Basic 2010*. (Wahana Komputer, 2011 : 3)

## II.7. Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi, juga dapat digunakan untuk :

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan.
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan.
3. Menentukan muatan data yang disimpan dalam *file-file*.
4. Mengembangkan logika untuk proses-proses diagram aliran data.

Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pihak pemakai

sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang input, merancang laporan-laporan dan *database*.

Kamus data mendefenisikan elemen data dengan fungsi sebagai berikut :

1. Menjelaskan arti aliran data dan penyimpanan data dalam DFD.
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran (misalnya alamat diuraikan menjadi kota, negara dan kode pos)
3. Mendeskripsikan komposisi penyimpanan data.
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran.
5. Mendeskripsikan hubungan detail antar penyimpanan (yang akan menjadi titik perhatian dalam *entity-relationship diagram*).  
(Muhammad Arhami : 2010 ; 7)

## **II.8. Normalisasi**

Normalisasi adalah suatu teknik untuk memecah/ mendekomposisi data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data dalam bisnis data. Proses normalisasi terdiri dari beberapa level, yaitu :

1. Bentuk Tidak Normal (Un Normalized Form/UNF). Kriteria dari bentuk ini adalah :
  - a. Jika relasi mempunyai bentuk nonflat file.
  - b. Jika relasi memuat set atribut berulang.
  - c. Jika relasi memuat atribut non atomic value.

2. Bentuk Normal Pertama (First Normal Form/ 1NF). Kriteria dari bentuk ini adalah :

- a. Jika seluruh atribut dalam relasi bernilai atomik.
- b. Jika seluruh atribut dalam relasi bernilai tunggal.
- c. Jika relasi tidak memuat set atribut berulang.
- d. Jika semua *record* mempunyai sejumlah atribut yang sama.

Permasalahan dalam adalah :

- a. Tidak dapat menyisipkan informasi parsial.
- b. Terhapusnya informasi ketika menghapus sebuah record.
- c. Pembaharuan atribut non kunci mengakibatkan sejumlah record harus diperbaharui.

Untuk mengubah relasi UNF menjadi bentuk 1NF dilakukan dengan cara :

- a. Melengkapi nilai-nilai dalam atribut.
- b. Mengubah struktur relasi.

3. Bentuk Normal Kedua (Second Normal Form / 2NF). Kriteria dari bentuk ini adalah :

- a. Jika memenuhi kriteria 1NF
- b. Jika semua atribut non kunci memiliki ketergantungan fungsional terhadap atribut kunci.

Permasalahan dalam 2NF adalah :

- a. Kerangkapan data.
- b. Pembaharuan yang tidak benar dapat menimbulkan inkonsistensi data.

- c. Proses pembaharuan data tidak efisien.
- d. Penyimpanan pada saat penyisipan, penghapusan dan pembaharuan.

Untuk mengubah relasi 1NF menjadi bentuk 2NF dilakukan dengan cara :

- a. Identifikasi ketergantungan fungsional pada relasi 1NF.
  - b. Berdasarkan ketergantungan fungsional, dekomposisi relasi 1NF menjadi relasi-relasi baru sesuai dengan ketergantungan fungsionalnya.
4. Bentuk normal Ketiga (Third Normal Form/3NF). Kriteria dari bentuk ini adalah :
- a. Jika memenuhi kriteria 3NF.
  - b. Jika semua atribut non kunci tidak memiliki ketergantungan transitif kunci. (Putu Manik Prihatin: 2012; 5-6).

## **II.9. *Unified Modelling Language(UML)***

Pengembangan sistem adalah aktivitas manusia. Tanpa adanya kemudahan untuk memahami sistem notasi, proses pengembangan kemungkinan besar akan mengalami kesalahan. UML adalah sistem notasi yang sudah dibakukan di dunia pengembangan sistem, hasil kerjasama dari Grady Booch, James Rumbaugh dan Ivar Jacobson. UML yang terdiri dari serangkaian diagram memungkinkan bagi sistem analis untuk membuat cetak biru sistem yang komperhesif kepada klien, programmer dan tiap orang yang terlibat dalam proses pengembangan tersebut. Dengan UML akan dapat menceritakan apa yang seharusnya dilakukan oleh sistem bukan bagaimana yang seharusnya dilakukan oleh sebuah sistem.

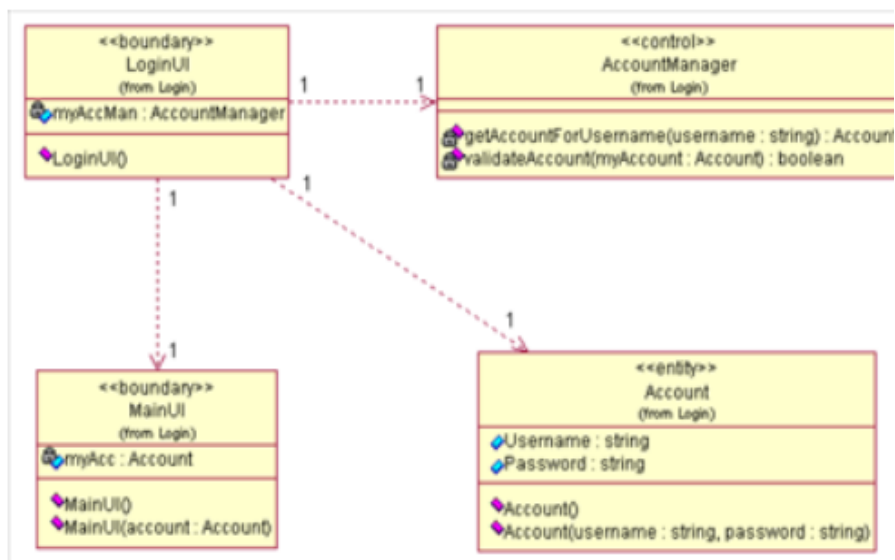
UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman seperti JAVA, C++, Visual Basic atau bahkan dihubungkan secara langsung ke dalam sebuah *object oriented database*. Begitu juga mengenai pendokumentasian dapat dilakukan seperti *requirements*, arsitektur, *design*, *source*, *project plan*, *tests* dan *prototypes*. UML memiliki 8 tipe diagram, namun pada penulisan skripsi ini penulis akan menggunakan 4 tipe diagram UML yaitu *use case diagram*, *sequence diagram*, *activity diagram* dan *class diagram*.

### **II.9.1. Tipe Diagram UML**

Adapun 8 tipe diagram UML adalah :

#### **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 : nama, atribut, metoda.



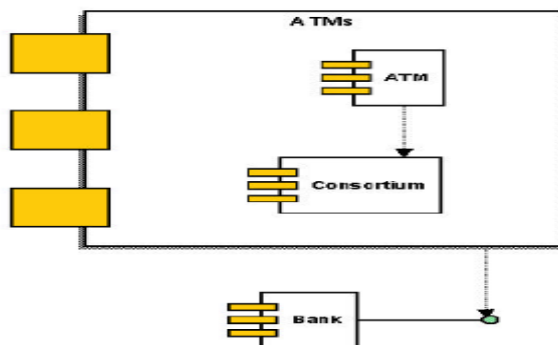
**Gambar II.1 Use Case Model**  
(Sumber : Haviluddin:2011; 3)

## 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.

## 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.2. Notasi Component Diagram**  
(Sumber : Haviluddin:2011; 3)

#### 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.

#### 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.

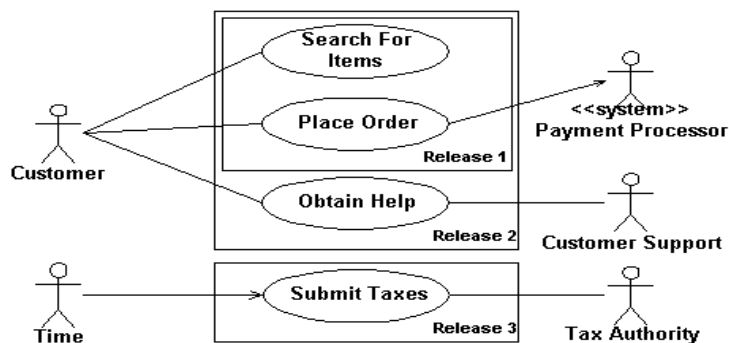
#### 6. *Package Diagram*

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.

### 7. Use Diagram

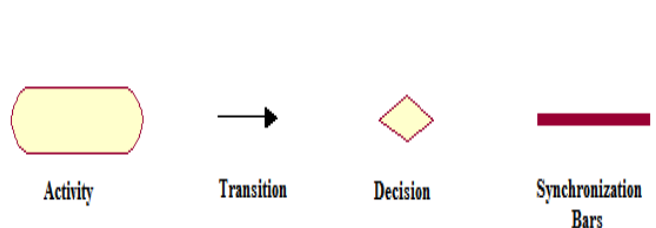
Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*.



**Gambar II.3. Notasi Component Diagram**  
(Sumber : Haviluddin:2011; 4)

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