

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Perancangan**

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

Menurut (Kusrini: 2009: 11) Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*)

##### **II.1.2. Data**

Menurut (Kusrini: 2009: 3) Data merupakan representasi dari fakta atau gambaran mengenai suatu objek atau kejadian, ambil contoh fakta mengenai biodata mahasiswa yang meliputi nama, alamat, jenis kelamin, agama yang dianut, dan lain-lain. Contoh lain dari fakta mengenai kejadian /transaksi dalam sebuah perusahaan dagangan adalah seperti transaksi penjualan yang meliputi waktu transaksi, pelaku transaksinya (pelanggan, kasir), barang yang ditransaksikan, serta jumlah dan harganya. Data dinyatakan dengan nilai yang berbentuk angka, deretan karakter, atau simbol.

##### **II.1.3. Informasi**

Menurut (Kusrini: 2009: 4) informasi merupakan hasil olahan data, dimana data tersebut sudah diproses dan diinterpretasikan menjadi sesuatu yang bermakna untuk pengambilan keputusan, informasi juga diartikan sebagai himpunan dari data yang relevan dengan satu atau beberapa orang dalam suatu waktu.

Menurut (Kusrini: 2009: 4) Suatu informasi berguna bagi pembuatan keputusan karena informasi bias menurunkan ketidakpastian (meningkatkan pengetahuan) tentang hal yang sedang dipikirkan. Makna dari sebuah informasi tentu berbeda-beda antara seorang dengan lainnya, tergantung pada tingkat kepentingannya, misalnya informasi daftar pelanggan yang potensial akan sangat dibutuhkan oleh bagian marketing di suatu perusahaan guna meningkatkan penjualan produk, tetapi barangkali tidak akan menjadi perhatian dibagian personalia.

Kegunaan informasi bagi seseorang juga sangat tergantung pada waktu, pada suatu waktu tertentu informasi tersebut mungkin sangat diperlukan dilain hari, mungkin saja hal tersebut sudah tidak berguna sama sekali. Contohnya, informasi perbandingan harga barang akan sangat dibutuhkan oleh seseorang yang akan membeli barang tersebut. Namun saat ini dia sedang tidak mempertimbangkan untuk membeli barang tersebut. Informasi tersebut menjadi kurang bermakna.

#### **II.1.4. Kualitas Informasi**

Agar bisa menyediakan keluaran yang berguna untuk membantu manager atau para pengambil keputusan, sebuah sistem informasi harus mampu mengumpulkan data dan mentransformasikan data tersebut kedalam informasi yang memiliki kualitas-kualitas tersebut.

Berikut karakteristik informasi yang berkualitas :

- 1. Relevan.** Informasi yang disajikan sebaiknya terkait dengan keputusan yang akan diambil oleh pengguna informasi tersebut. Misalnya, seorang

manager yang akan memberikan kredit kepada pelanggan bisa melihat laporan keuangan pelanggan tersebut karena laporan tersebut terkait dengan keputusan yang akan dibuat, yaitu memberikan atau tidak memberikan kredit kepada pelanggan tersebut.

2. **Akurat.** Kecocokan antara informasi dengan kejadian-kejadian atau objek-objek yang diwakilinya. Misalnya, laporan inventaris yang tidak akurat menyebutkan bahwa terdapat 15 unit barang di dalam gudang.
3. **Lengkap.** Merupakan derajat sampai seberapa jauh informasi menyertakan kejadian-kejadian atau objek-objek yang berhubungan. Misalnya, penjualan selama satu hari yang seharusnya ada 150 transaksi dilaporkan hanya tercatat sebanyak 145 transaksi.
4. **Tepat Waktu.** Informasi yang tidak tepat waktu akan menjadi informasi yang tidak berguna atau tidak dapat digunakan untuk membantu pengambilan keputusan. Misalnya, informasi jadwal ujian seorang mahasiswa disampaikan setelah kegiatan ujian diselenggarakan. Informasi ini menjadi tidak berguna lagi.
5. **Dapat dipahami.** Hal tersebut terkait dengan bahasa dan cara penyajian informasi agar pengguna lebih mudah mengambil keputusan.
6. **Dapat dibandingkan.** Sebuah informasi yang memungkinkan seorang pemakai untuk mengidentifikasi persamaan dan perbedaan antara dua objek atau kejadian yang mirip. Misalnya, membandingkan laporan pendapatan antara tahun 2006 dan 2007.

### **II.1.5. Sistem Informasi**

Menurut (Kusrini :2009: 11) suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan untuk pihak luar.

Berdasarkan dukungan kepada pemakainnya, sistem informasi dibagi menjadi :

1. Sistem Pemrosesan Transaksi (*Transaksi Processing System*) atau TPS
2. Sistem Informasi Manajemen (*Manajemen Information System*) atau MIS
3. Sistem Otomasi Perkantoran (*Office Automation System/OAS*)
4. Sistem Pendukung Keputusan (*Decision Support System*) atau DSS
5. Sistem Informasi Eksekutif (*Executive Information System*) atau EIS
6. Sistem Pendukung Kelompok (*Group Support System*) atau GSS
7. Sistem Pendukung Cerdas (*Intelegent Support System* ) atau ISS

Mengingat bahwa EIS, DSS, dan MIS digunakan untuk pendukung manajemen, maka ketiga system tersebut sering disebut Sistem Pendukung Manajemen (*management support system*) atau MSS.

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

Menurut (Kusrini : 2009 : 15) Sistem Pendukung Keputusan merupakan Sistem Informasi interaktif yang menyediakan informasi pemodelan dan memanipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur 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. Dss yang seperti itu disebut dengan aplikasi DSS. Aplikasi DSS digunakan dalam pengambilan keputusan. Aplikasi DSS menggunakan CBIS (Computer Based Information Systems) yang fleksibel, interaktif, dan dapat diadaptasi yang dikembangkan untuk mendukung solusi atau masalah manajemen spesifik yang tidak terstruktur.

Aplikasi DSS menggunakan data, memberikan antara muka pengguna yang mudah dan dapat digabungkan pengguna yang mudah, dan dapat digabungkan pemikiran pengambilan keputusan.

DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas.

DSS tidak dimaksudkan untuk mengotomatiskan pengambilan keputusan tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

### **II.2.1. Arsitektur Sistem Pendukung Keputusan**

Menurut (Kusrini :2009: 25) Aplikasi sistem pendukung keputusan bisa berdiri dari beberapa subsistem, yaitu :

1. Subsistem manajemen data

Subsistem manajemen data memasukkan satu database yang berisi data yang relevan untuk suatu situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen data bisa diinterkoneksi dengan data

*warehouse* perusahaan, suatu repository untuk data perusahaan yang relevan dengan pengambilan keputusan.

2. Subsistem manajemen modal

Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lain yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat. Bahasa-bahasa pemodelan untuk membangun model-model kustom juga dimasukkan. Perangkat lunak itu sering disebut sistem manajemen basis model (MBMS), komponen tersebut bisa dikoneksikan ke penyimpanan korporat atau eksternal yang ada pada model.

3. Subsistem antarmuka pengguna

Pengguna berkomunikasi dengan dan memerintahkan sistem pendukung keputusan melalui subsistem tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa kontribusi unik dari sistem pendukung keputusan berasal dari interaksi yang interaktif antara computer dan pembuat keputusan.

4. Subsistem manajemen berbasis-pengatahuan

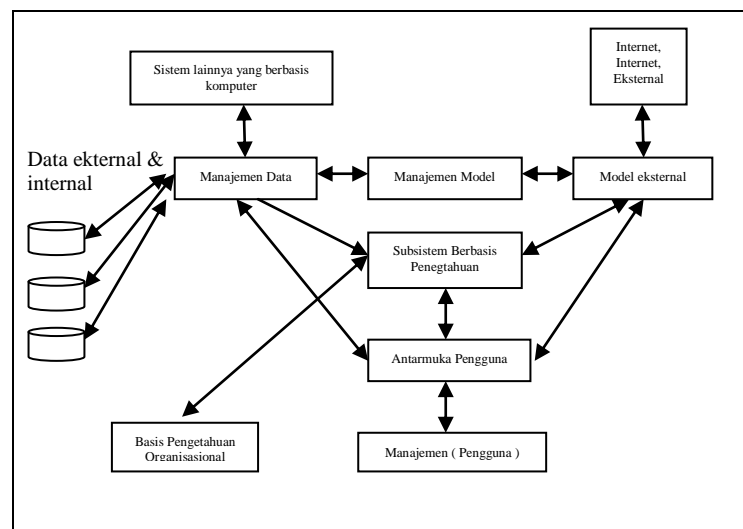
Subsistem tersebut mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen independent dan bersifat opsional.

Selain memberikan intelegensi untuk memperbesar pengetahuan si pengambil keputusan, subsistem tersebut bisa diinterkoneksikan dengan repository pengetahuan perusahaan (bagian dari sistem manajemen

pengetahuan), yang kadang-kadang disebut basis pengetahuan organisasional.

Berdasarkan defenisi, sistem pendukung keputusan harus mencakup tiga komponen utama dari DBMS, MBMS, dan antarmuka pengguna. Subsistem manajemen berbasis pengetahuan adalah opsional, tetapi bisa memberikan banyak manfaat karena memberikan intelegensi bagi ketiga komponen utama tersebut, seperti pada semua sistem pendukung keputusan. Komponen-komponen tersebut membentuk sistem aplikasi sistem pendukung keputusan yang bisa dikoneksikan ke intranet perusahaan, ekstranet, atau internet. Arsitektur dari sistem pendukung keputusan ditunjukkan dalam tabel II.1 berikut.

**Tabel II.1 Arsitektur DSS**



Sumber : Kusrini (2009 : 26)

### **II.3. Metode TOPSIS (*Technique For Others Reference by Similarity to Ideal Solution*)**

Teori TOPSIS (*Technique For Others Reference by Similarity to Ideal Solution*) adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh yoon dan Hwang (1981). 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 negatif-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 relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa tercapai.

Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relative dari alternatif-alternatif keputusan (Tonni Limbong : 2010: 1)

#### **II.3.1. Langkah-langkah penyelesaian masalah dengan metode TOPSIS:**

1. Membuat matriks keputusan yang ternormalisasi;
2. Membuat matriks keputusan yang ternormalisasi terbobot;

3. Menentukan matriks solusi ideal positif & matriks solusi ideal negative;
4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif & matriks solusi ideal negatif;
5. Menentukan nilai preferensi untuk setiap alternatif.

**1. Rating kinerja alternative A1 pada setiap criteria C1 yang ternormalisasi :**

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

dengan  $i = 1, 2, \dots, m$  dan  $j = 1, 2, \dots, n$

**2. Solusi ideal positif dan solusi ideal negative dapat ditentukan berdasarkan rating bobot ternormalisasi (y):**

$$y_{ij} = W_i r_{ij}$$

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+)$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-)$$

dengan  $i = 1, 2, \dots, m$  dan  $j = 1, 2, \dots, n$

Dengan :

$$y_j^+ = \begin{cases} \text{Max}_i y_{ij} & : \text{Jika } j \text{ adalah atribut keuntungan} \\ \text{Min}_i y_{ij} & : \text{Jika } j \text{ adalah atribut biaya} \end{cases}$$

$$y_j^- = \begin{cases} \text{Min}_i y_{ij} & : \text{Jika } j \text{ adalah atribut keuntungan} \\ \text{Max}_i y_{ij} & : \text{Jika } j \text{ adalah atribut biaya} \end{cases}$$

3. Jarak antara alternatif  $A_i$  dengan *solusi ideal positif* dirumuskan sebagai:

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2}$$

4. Jarak antara alternative  $A_i$  dengan *solusi ideal negatif* dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2}$$

5. Nilai preferensi untuk setiap alternatif ( $v_i$ ), diberikan sebagai berikut :

$$V_i = \frac{D_i^+}{D_i^- + D_i^+}$$

Sumber: Tonni Limbong (2010:1)

#### II.4. Microsoft Visual Basic 2012

Pemrograman Microsoft Visual Basic .NET 2012 adalah merupakan peningkatan bahasa pemrograman *BASIC* dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Beberapa bahasa skrip seperti Visual Basic *for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda. Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic Program-program yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi luar tambahan. Pemrograman Visual basic pada Microsoft Visual Studio 2012 juga menyediakan tingkat level fitur yang sama

dengan bahasa pemrograman lainnya, hanya saja jenis bahasa pemrogramannya saja yang berbeda. (Wahana Komputer: 2013: 4)

## **II.5. MySQL Server**

Database MySQL merupakan system manajemen basis data SQL yang sangat terkenal dan bersifat open source, meski dirilis secara open source namun keandalannya dapat dibuktikan. MySQL mempunyai stabilitas dan kecepatan akses yang tinggi, dapat berjalan pada berbagai system operasi, mudah digunakan, dan tersedia dalam berbagai macam bahasa. MySQL dikembangkan untuk menangani database yang besar secara cepat dan telah sukses digunakan selama bertahun-tahun. Konektivitas, kecepatan, dan keamanannya telah menjadikan MySQL sebagai server yang cocok untuk mengakses database di internet. (Budi Raharjo: 2009: 21)

### **II.5.1. Pemodelan Data**

Menurut (Edhy Sutanta :2011: 85) Pada Perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antara data. Hubungan tersebut dapat dinyatakan dalam E-R, karena model E-R adalah dasar penting dalam merancang database maka akan dijelaskan tentang gambaran tentang model E-R, penjelasan mengenai komponen-komponen yang menyusun model E-R, hingga cara penyusunan model E-R.

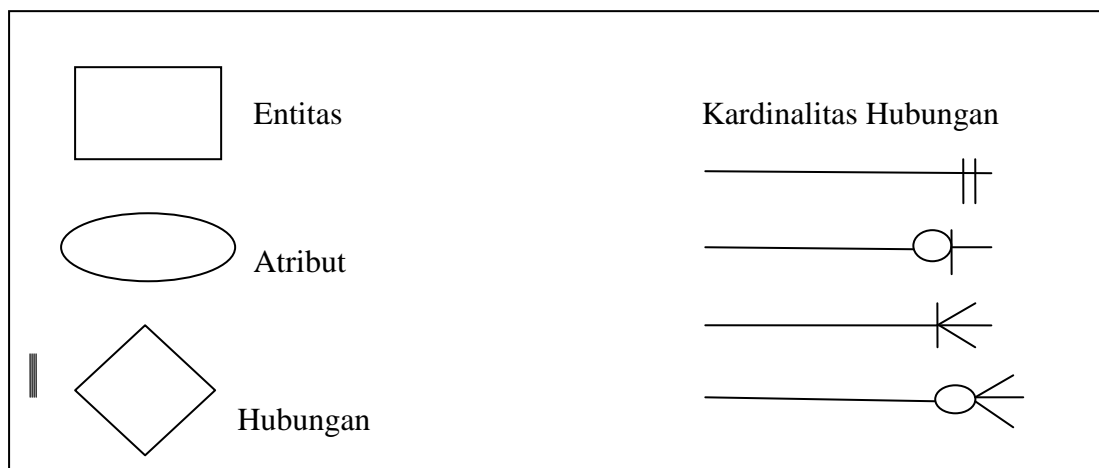
### **II.5.2. Database**

Istilah '*database*' banyak memiliki definisi. Untuk sebagian kalangan, secara sederhana *database* diartikan sebagai kumpulan data (buku, nomor telepon,

daftar pegawai, dan lain sebagainya). Ada juga yang menyebut *database* dengan defenisi lain yang telah formal dan tegas.(Budi Raharjo: 2009: 3)

### II.5.3. Model E-R

Menurut (Edhy Sutanta :2011: 91) Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas. Huruf E sendiri menyatakan entitas dan R menyatakan hubungan (dari kata *Relationship*). Model E-R melibatkan sejumlah notasi, beberapa notasi dasar dalam model E-R ditunjukkan pada gambar II.1, notasi-notasi tersebut diberikan hanya untuk memberikan suatu pengetahuan dasar.



**Gambar II.1 : Sejumlah notasi pada model E-R**

Sumber : Edhy Sutanta (2011: 109)

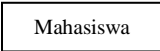
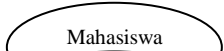

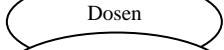
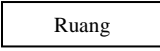
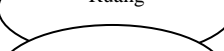
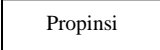

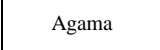

#### 1. Entitas

Yang dimaksud dengan entitas adalah obyek-obyek dasar yang terkait di dalam sistem, dapat berupa orang, benda, atau hal yang keterangannya perlu

disimpan didalam basis data. Entitas dapat digambarkan dengan cara berikut (Sutanta: 2011: 124) :

- a. Entitas dinyatakan dengan symbol persegi panjang atau elips
- b. Nama entitas ditulis di dalam symbol persegi panjang
- c. Nama entitas berupa: kata benda, tunggal
- d. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas

Penggunaan tanda garis bawah (*hyphen/underscore*), pemendekan, dan singkatan juga lazim digunakan untuk memberikan nama entitas sebagai-nama dalam ER-D. Gambar II.2 Contoh-contoh Entitas

Obyek Dasar	Simbol Entitas (pilihan 1)		Simbol Entitas (pilihan 2)
Mahasiswa	 Mahasiswa	atau	 Mahasiswa
Dosen	 Dosen	atau	 Dosen
Ruang	 Ruang	atau	 Ruang
Propinsi	 Propinsi	atau	 Propinsi
Agama	 Agama	atau	 Agama

**Gambar II.2 : Contoh-contoh Entitas**

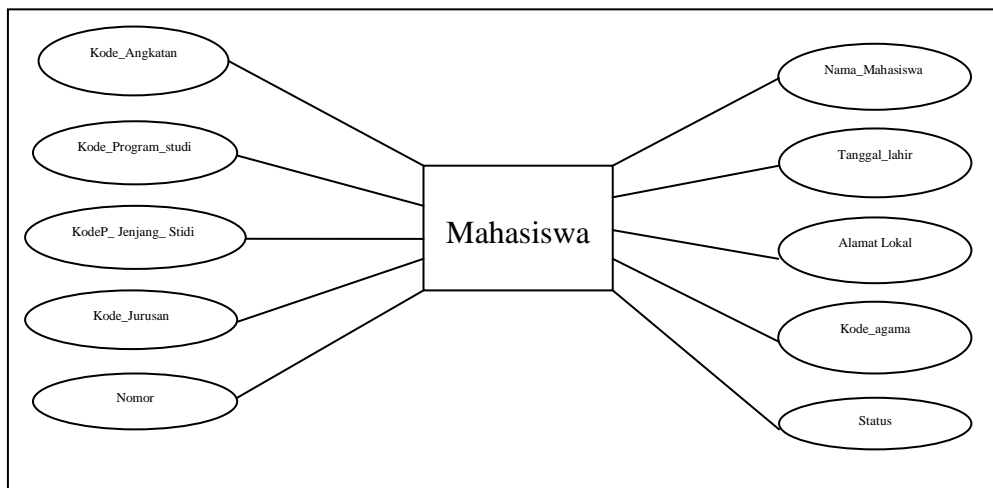
Sumber : Edhy Sutanta (2011: 124)

## 2. Atribut

Atribut atau properti (*property*) merupakan keterangan-keterangan yang berkaitan pada sebuah entitas yang perlu disimpan sebagai basis data. Atribut dapat digambarkan dengan cara sebagai berikut (Sutanta: 2011: 125)

1. Atribut dinyatakan dengan symbol elips
2. Nama atribut ditulis di dalam symbol elips
3. Nama atribut berupa kata benda, tunggal
4. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas
5. Atribut di hubungkan dengan entitas yang bersesuaian dengan menggunakan sebuah garis (seyogianya menggunakan garis lurus, namun dalam kondisi yang tidak memungkinkan dapat tidak menggunakan garis lurus)

Konsep atribut sederhana (*simple attribute*) dan atribut komposit (*composite attribute*) yang dikenal dalam model ER juga tetap berlaku pada *semantic model*. Sebagai contoh, atribut pada entitas mahasiswa ditunjukkan pada gambar II.3.

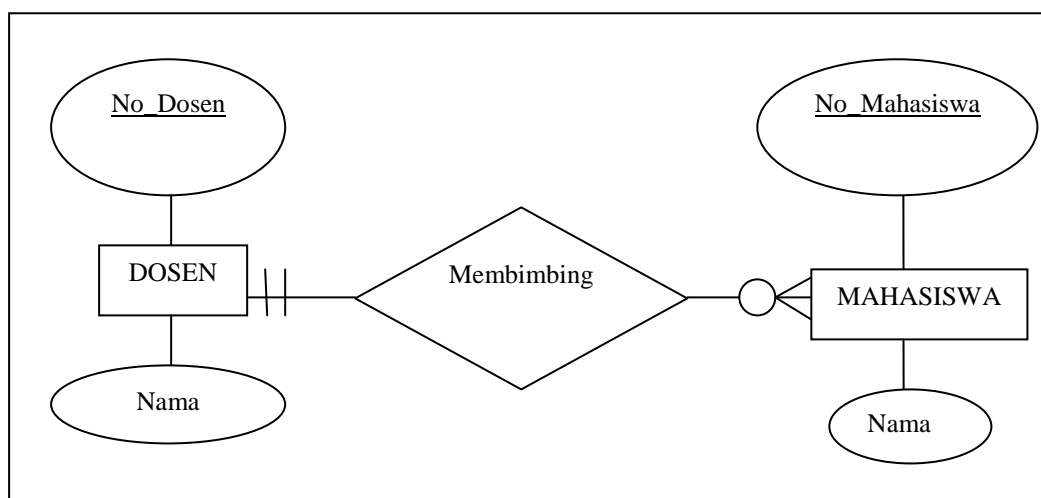


**Gambar II.3 : Contoh atribut pada entitas Mahasiswa**

Sumber : Edhy Sutanta (2011: 125)

### 3. Hubungan (*Relationship*)

Hubungan (*Relationship*) menyatakan ketertarikan antara beberapa tipe entitas. Sebagai contoh, tipe entitas MAHASISWA dan DOSEN mempunyai hubungan yang mencerminkan bahwa seorang mahasiswa memiliki dosen pembimbing akademis. Gambar II.4 menunjukkan hubungan tersebut.



**Gambar II.4 : Contoh Hubungan antara tipe entitas**

Sumber : Edhy Sutanta (2011: 127)

#### II.5.4. Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen data store.

Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem.

Pada tahap analisis, kamus data merupakan alat komunikasi antara user dan analis sistem tentang data yang mengalir di dalam sistem, yaitu tentang data yang masuk ke sistem tentang perancangan sistem kamus data digunakan untuk merancang input, laporan dan data base. (jurnal Analisa Sistem Informasi: 2010:1)

## **II.6. *Unified Modeling Language* (UML)**

Menurut (Whitten, et.al 2010: 1) Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangun perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari system perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan, jadi pengguna UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataan UML paling banyak digunakan pada metode berorientasi objek.

Menurut (Braun, et.al 2010 : 1) UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

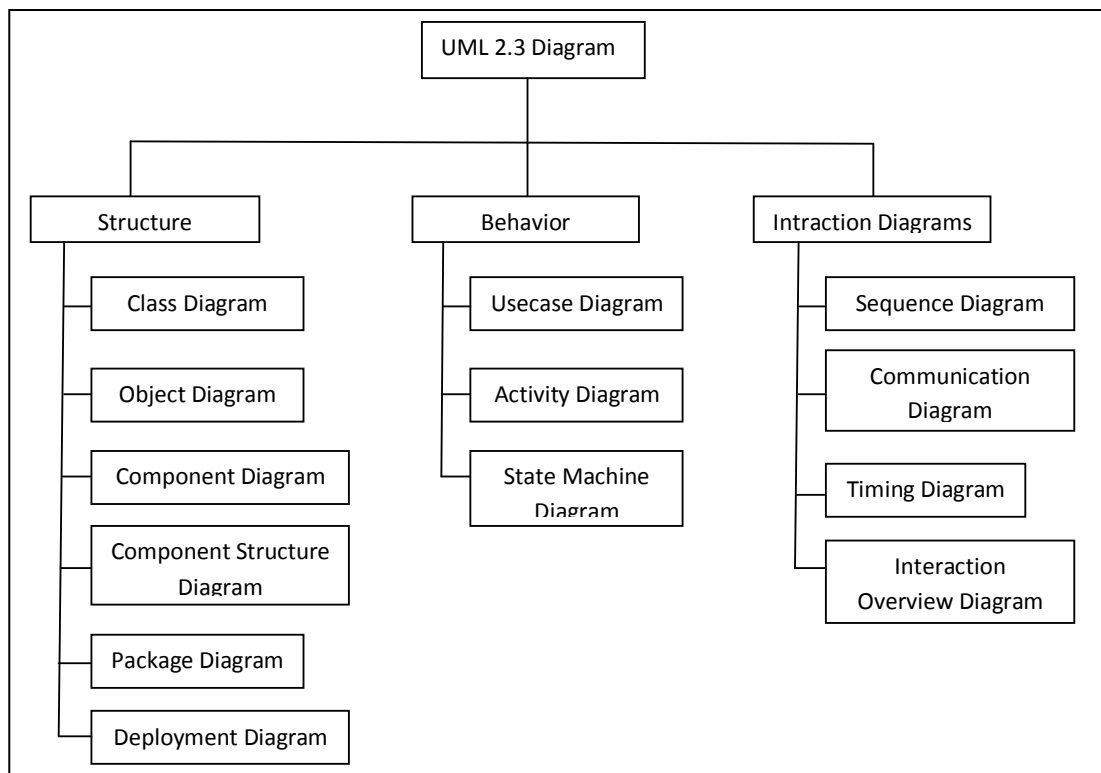
1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.

4. Mendokumentasikan sistem secara yang ada, proses-proses dan organisasinya.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini.

## II.7. Diagram-Diagram UML

Menurut (Sugrue J.2009: 3) Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar II.5 di bawah ini.



**Gambar II.5 : Diagram UML**

Sumber : Sugrue J. (2009: 15)

Berikut ini penjelasan singkat dari pembagian kategori tersebut

1. *Struktur Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

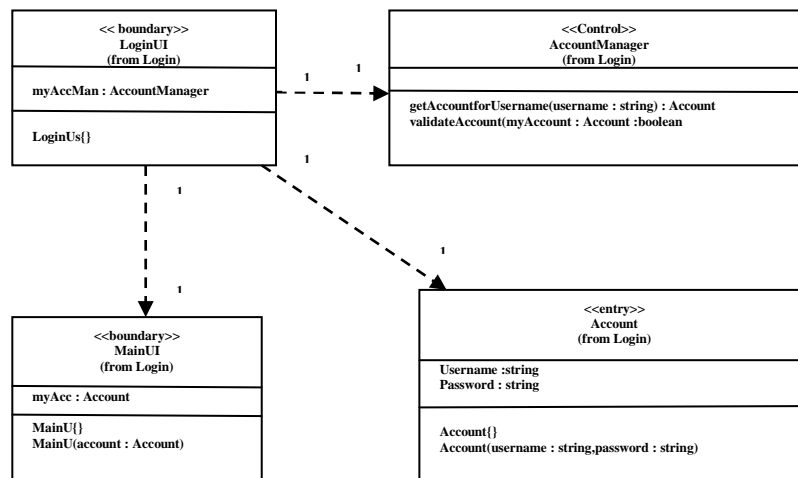
#### A. *Class Diagram*

Diagram kelas atau 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

Berikut ini gambar II.6 diagram kelas :



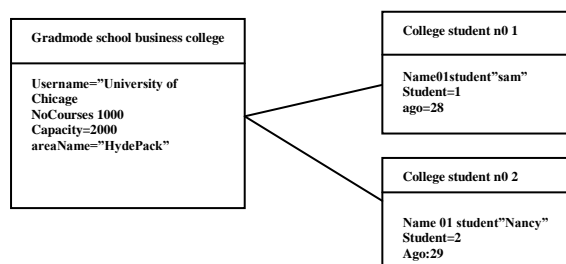
**Gambar II.6 : Class Diagram**

Sumber : Sugrue J. (2009: 3)

## B. Object Diagram

Diagram objek 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 adalah gambar II.7 diagram objek



**Gambar II.7: Diagram Objek**

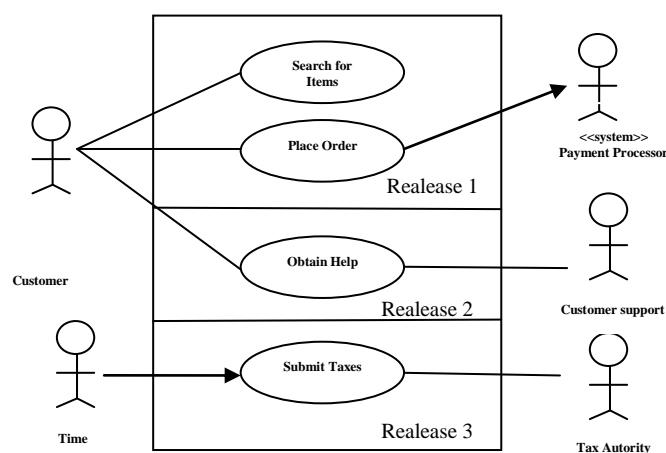
Sumber : Sugrue J. (2009: 3)

### C. 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.

Berikut adalah gambar II.8 Use Case Diagram



**Gambar II.8 : Use Case Diagram**

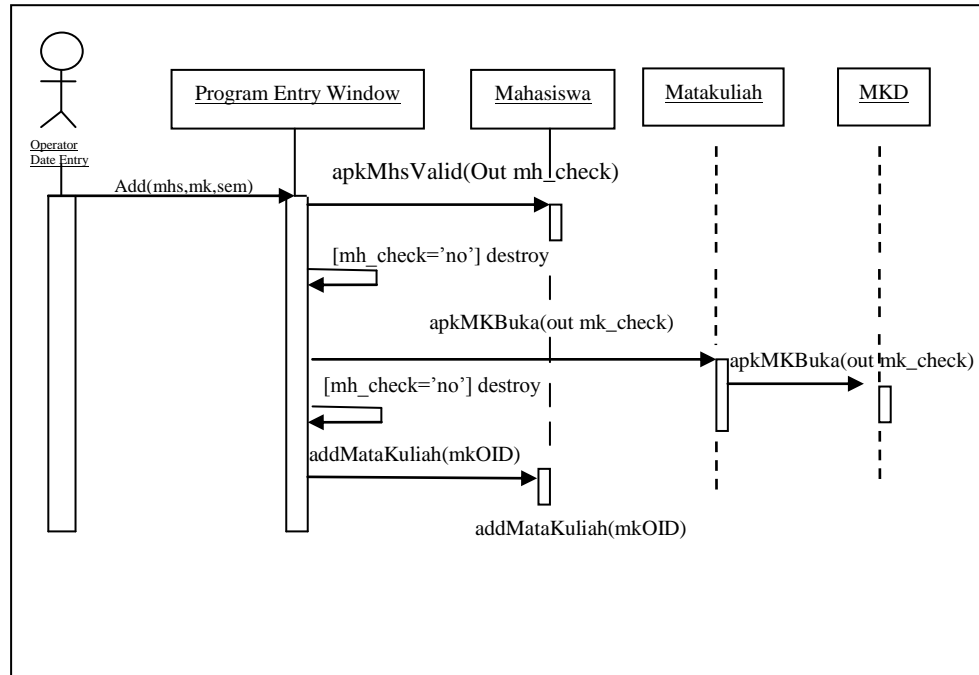
Sumber : Sugrue J. (2009: 4)

### D. 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.

Berikut adalah gambar II.9 Sequence Diagram



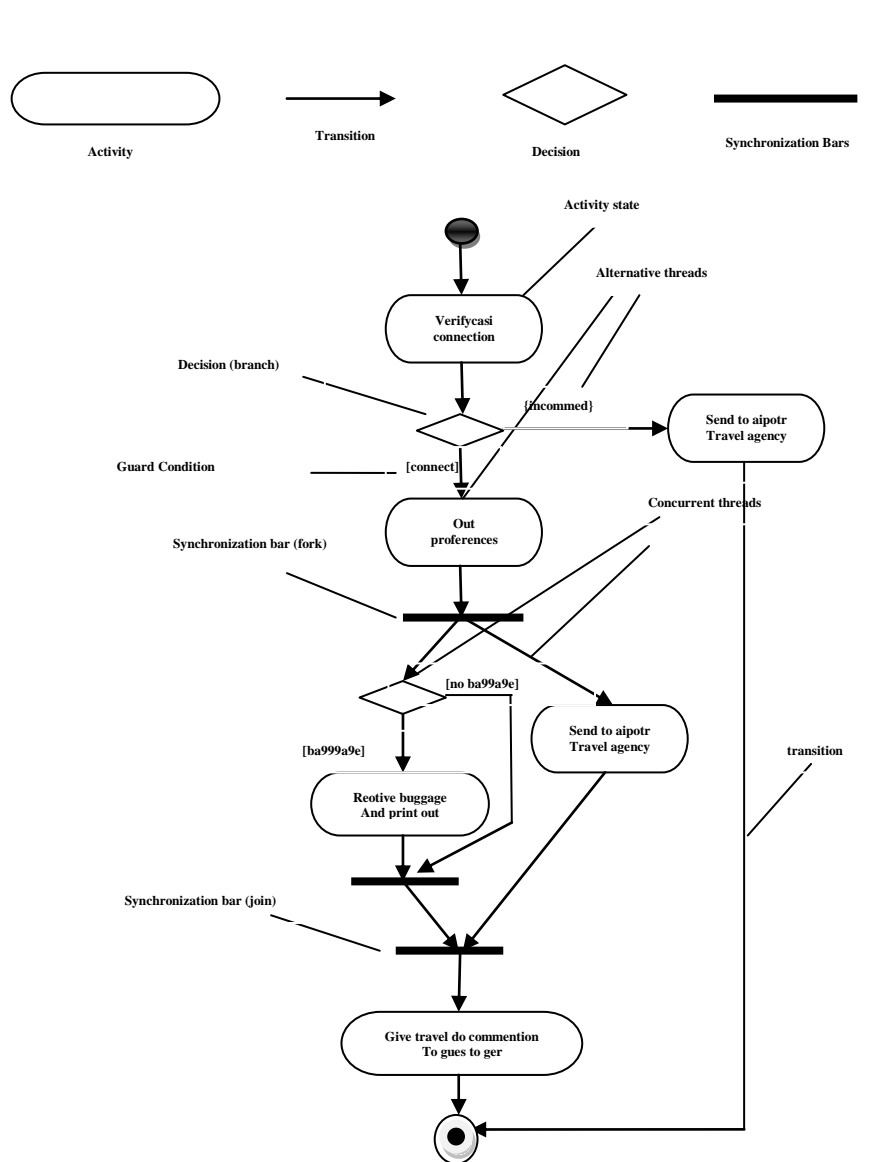
**Gambar II.9 : Sequence Diagram**

Sumber : Sugrue J. (2009: 5)

## E. Activity Diagram

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

Berikut adalah gambar II.10 Use Case Diagram



**Gambar II.10 : activity Diagram**

Sumber : Sugrue J. (2009: 4)

## DAFTAR PUSTAKA

Budi Raharjo, 2009, *Belajar otodidak membuat database menggunakan MySQL*, Penerbit INFORMATIKA

Edhy Sutanta, 2011, *Basis Data dalam Tinjauan Konseptual*, Penerbit Andi, Yogyakarta

Kusrini, M.Kom, 2009, *Konsep dan Aplikasi Sistem Pendukung Keputusan*, Penerbit Andi, Yogyakarta

Sugrue J. 2011, *Getting Started with UML*.  
<http://www.dzone.com/links/index.html>

Wahana Komputer, 2013, *Paling Dicari Visual Basic 2012*, Penerbit Andi, Yogyakarta