

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur. Sistem Pendukung Keputusan biasanya dibangun untuk mendukung solusi atau suatu masalah atau mengevaluasi suatu peluang. (Kusrini;2007:15)

Sistem Pendukung Keputusan digunakan dalam pengambilan keputusan. Aplikasi Sistem Pendukung Keputusan menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Aplikasi Sistem Pendukung Keputusan menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan. (Kusrini;2007:16)

Adapun tujuan Sistem Pendukung Keputusan sebagai berikut:

- a. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
- b. Memberi Mendukung atas pertimbangan manajer bukan dimaksudkan untuk mengantikan fungsi manajer.
- c. Meningkatkan efektivitas pengambilan keputusan manajer daripada efisiensinya

Tujuan-tujuan ini berhubungan dengan empat prinsip dasar dari konsep SPK yaitu struktur masalah, dukungan keputusan, dan efektivitas keputusan. SPK terdiri dari tiga subsistem utama yaitu :

a. *Data Management*

Yaitu Data manajemen meliputi database, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh software yang disebut *Database Management Systems* (DBMS).

b. *Model Management*

Yaitu Model manajemen melibatkan model finansial, statistikal, manajemen science, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen software yang diperlukan.

c. *Communication (dialog subsystem)*

Yaitu pengguna dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini, yang berarti menyediakan antarmuka.

d. *Knowledge Management*

Yaitu subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri. (Kusrini;2007:16)

II.2. Biaya Perawatan Mesin Produksi Makanan

Biaya perawatan mesin produksi makanan adalah biaya suku cadang (*spareparts*), biaya bahan habis pakai (*factory supplies*), dan harga jasa yang perlu dikeluarkan perusahaan untuk keperluan perbaikan dan perawatan mesin produksi

makanan dalam masa operasinya. Tujuan adanya perawatan mesin produksi makanan adalah untuk merawat mesin produksi makanan dan mengendalikan biaya sehingga perawatan mesin produksi makanan harus dirancang dan dipelihara untuk mencapai standar mutu dan kinerja yang diharapkan.

Perawatan mesin produksi makanan meliputi segala aktifitas yang terlibat dalam penjagaan peralatan mesin produksi makanan dalam aturan kerja. Kebanyakan dari mesin produksi makanan pasti dipelihara, diperbaiki jika terjadi kegagalan, dan suatu kegiatan dilakukan atas mesin tersebut agar tetap dapat bekerja. Untuk mencapai tingkat biaya perawatan mesin produksi makanan serendah mungkin, dengan melaksanakan kegiatan perawatan secara efektif dan efisien untuk keseluruhannya. (Tita;2015:2)

II.3. Metode AHP (*Analaytical Hierarchy Process*)

Metode *Analaytical Hierarchy Process* (AHP) merupakan sebuah hirarki fungsional dengan input utamanya persepsi manusia. Dengan hirarki, suatu masalah kompleks dan tidak terstruktur dipecahkan salah satu model pengambilan keputusan yang sering digunakan. *Analaytical Hierarchy Process* atau AHP dikembangkan oleh Prof. Thomas L. Saat sebagai algoritma pengambilan keputusan untuk permasalahan multikriteria (*Multi Criteria Decision Making* atau MCDM). Permasalahan multikriteria dalam AHP disederhanakan dalam bentuk hierarki yang terdiri dari 3 komponen utama. Yaitu tujuan atau goal dari pengambilan keputusan, kriteria penilaian dan alternatif pilihan. (Kusrini;2007:133)

AHP digunakan dengan tujuan untuk menyusun prioritas dari berbagai alternatif atau pilke dalam kelompok-kelompok tersebut diatur menjadi suatu bentuk hirarki. *Analaytical Hierarchy Process* (AHP) merupakan salah satu metode untuk membantu menyusun suatu prioritas dari berbagai pilihan dengan menggunakan beberapa kriteria (multi criteria). Karena sifatnya yang multi kriteria, AHP cukup banyak digunakan dalam penyusunan prioritas. Disamping bersifat multi kriteria, AHP juga didasarkan pada suatu proses yang terstruktur dan logis. (Kusrini;2007:133)

Adapun langkah-langkah dalam penerapan metode AHP sebagai berikut:

- a. Mendefenisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi.
- b. Menentukan prioritas elemen
- c. Sintesis
- d. Mengukur konsistensi
- e. Hitung *Consistency Index* (CI) / *Consistency Ratio* (CR) dan memeriksa konsistensi hierarki.

$$CI = \frac{\lambda_{maks-n}}{n-1}$$

Keterangan :

n = banyak kriteria atau subkriteria

CI = indeks konsisten (*Consistent Index*)

$$CR = \frac{CI}{RI}$$

Table Nilai RI (*Random Index*)

N										0	1
R	.00	.00	.58	.90	.12	.24	.32	.41	.45	.49	.51
I											

Gambar II.1. Rumus metode AHP

Sumber : (Kusrini;2007:136)

II.4. Alat Bantu Pengembangan Sistem

Alat bantu pengembangan sistem yang digunakan dalam penulisan skripsi ini adalah sebagai berikut :

II.4.1. Sistem Basis Data

Basis data adalah penyimpanan kumpulan informasi secara sistematik dalam sebuah komputer sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sedangkan sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi

suatu proses pekerjaan. Sehingga bisa dikatakan bahwa sistem basis data adalah sistem yang terdiri atas kumpulan file-file yang saling berhubungan dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi data tersebut. (Politeknik Telkom ; Sistem Manajemen Basis Data ; 2009:2)

Menyimpan data dalam bentuk file yang berbeda-beda, memiliki kekurangan-kekurangan:

1. Data *redundancy* dan *inconsistency*.

Dikarenakan programer yang berbeda membuat file dan aplikasi masing-masing, menyebabkan beragam format dan aplikasi yang dibuat. Bahkan, aplikasi pun dibuat menggunakan bahasa pemrograman yang berbeda-beda. Lebih jauh lagi, data atau informasi yang sama bisa terdapat dalam beberapa file yang berbeda. Ini yang disebut dengan redundancy. Redundancy data ini lama kelamaan akan menyebabkan inconsistency dari data.

2. Kesulitan dalam pengaksesan data.

Dikarenakan setiap aplikasi memiliki file tersendiri untuk penyimpanan dan pengambilan data, maka jika suatu bagian dari institusi membutuhkan data dari bagian lain, akan menemui kesulitan. Hal ini dikarenakan aplikasi yang dimiliki bagian tersebut, tidak dapat membaca file yang terdapat di bagian lain.

3. Isolasi data.

Dikarenakan data tersebar dalam berbagai macam file, dan file tersebut dalam beragam format, pembuatan aplikasi baru akan terasa sulit ketika harus membaca format dari masing-masing file tersebut.

4. Masalah integritas.

Data yang disimpan harus memenuhi hal yang dinamakan dengan *consistency constraint*. Jika sebuah constraint berubah, maka seluruh aplikasi yang digunakan harus mengakomodasinya. Masalah akan muncul, jika constraint melibatkan beberapa data dari file yang berbeda-beda.

5. Masalah keamanan.

Tidak semua pengguna dari basis data dapat mengakses semua data. Hal ini akan sulit dilakukan jika menggunakan gaya penyimpanan data dalam file. (Politeknik Telkom ; Sistem Manajemen Basis Data ; 2009:3)

II.4.2. *Entity Relationship Diagram (ERD)*.

Pengertian dari ERD (*Entity Relationship Diagram*) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.

Pada dasarnya ada tiga komponen yang digunakan, yaitu :

a. Entitas

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda, derajat relasi atau kardinalitas rasio menjelaskan jumlah maksimum hubungan antara satu entitas dengan entitas lainnya

1) *One to One (1:1)*

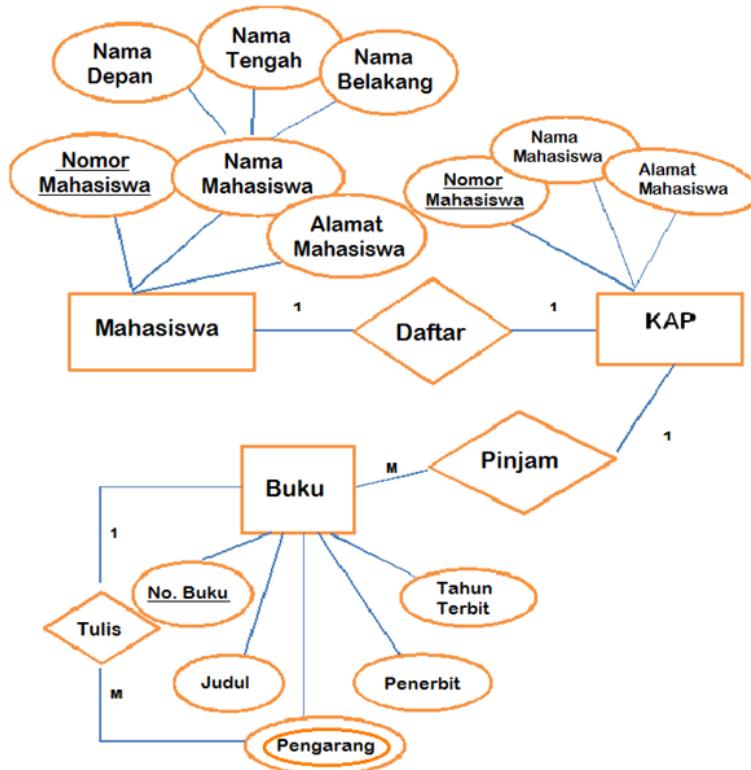
Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

2) *One to many (1:M / Many)*

Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

3) *Many to Many (M:M)*

Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya. (Kusrini ; 2007:21-23)



Gambar II.2. Entity Relationship Diagram

Sumber : (Kusrini ;2007:23)

II.4.3. Kamus Data.

Kamus data (*data dictionary*) adalah suatu penjelasan tertulis tentang suatu data yang berada di dalam database atau 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 strore.

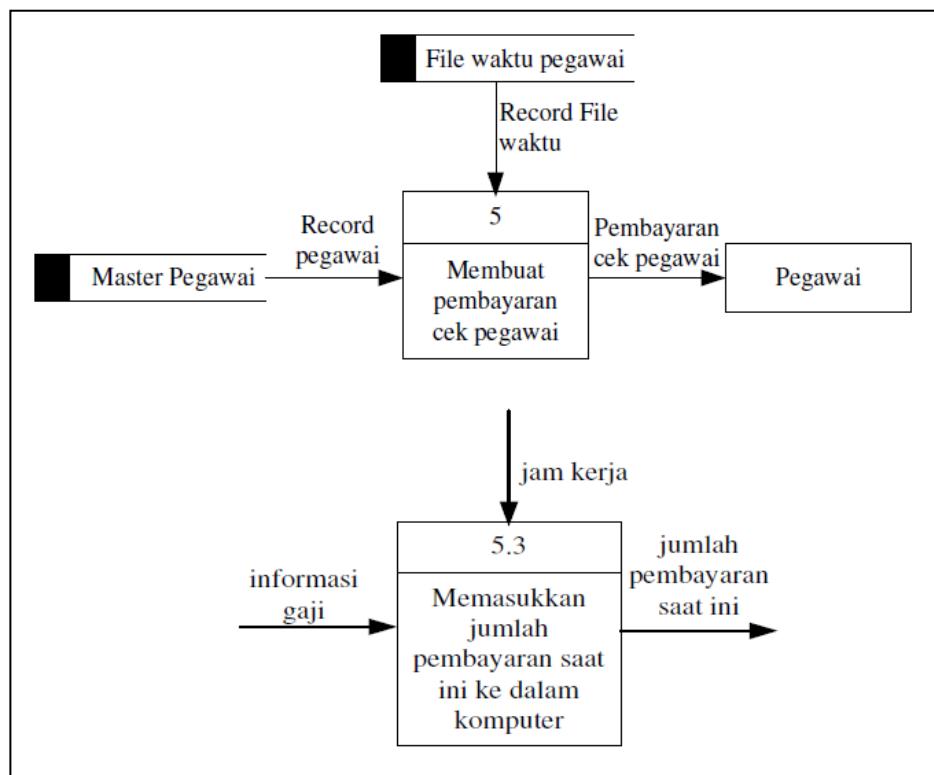
Kamus data pertama berbasis kamus dokumen tersimpan dalam suatu bentuk hard copy dengan mencatat semua penjelasan data dalam bentuk yang dicetak. Walau sejumlah kamus berbasis dokumen masih ada, praktik yang umum saat ini ialah mempergunakan kamus data yang berbasis komputer. Pada kamus data berbasis komputer, penjelasan data dimasukkan ke dalam komputer dengan memakai Data Description Language (DDL) dari sistem manajemen database, sistem kamus atau peralatan CASE. Kamus data tidak perlu dihubungkan dengan diagram arus data dan formulir-formulir kamus data dirancang untuk mendukung diagram arus data.

Kamus Data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redudansi, 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 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. (Kusrini;2007:63)

Dua diagram alir data di bawah ini dan masukan-masukan kamus data yang berhubungan untuk memproduksi pembayaran cek pegawai.



Gambar II.3. Pengelolaan Kamus Data

Sumber : (Kusrini;2007:67)

Contoh Kamus Data:

Record Pegawai = Nomor Pegawai +

Informasi Pribadi +

Informasi Gaji +

Informasi Pembayaran Saat Ini +

Informasi Gaji Tahunan Sampai Hari Ini

Record File Waktu = Nomor Pegawai +

Nama Pegawai +

Jam Kerja

Pembayaran Cek Gaji = Nomor Pegawai +

Nama Pegawai +
 Alamat +
 Jumlah Pembayaran Saat Ini +
 Jumlah Gaji Tahunan Sampai Saat Ini
 Informasi Gaji = Perhitungan Pembayaran + Jumlah Tanggungan
 Jumlah Pembayaran Saat Ini = Gaji Kotor +
 Potongan Pajak Pemerintah +
 Potongan Pajak Negara Bagian +
 Potongan Pajak Jaminan Sosial + Gaji Bersih
 (Kusrini; 2007:67)

II.4.4. Normalisasi.

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redudansi).

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan.

Tujuan normalisasi data adalah untuk:

- a. Untuk menghilang kerangkapan data.
- b. Untuk mengurangi kompleksitas.
- c. Untuk mempermudah pemodifikasi data.
- d. Proses normalisasi adalah sebagai berikut:
- e. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.

- f. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. (Kusrini;2007:39)

Tahapan dalam Normalisasi Data.

Tahap Normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF.

1. Bentuk 1NF (First Normal Form)

Aturan :

Tidak adanya atribut multi-value, atribut komposit atau kombinasinya.

Mendefinisikan atribut kunci.

Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi).

2. Bentuk 2NF (Second Normal Form)

Aturan :

Sudah memenuhi dalam bentuk normal kesatu (1NF).

Semua atribut bukan kunci hanya boleh tergantung (functional dependency) pada atribut kunci.

Jika ada ketergantungan parsial maka atribut tersebut harus dipisah pada tabel yang lain.

Perlu ada tabel penghubung ataupun kehadiran foreign key bagi atribut-atribut yang telah dipisah tadi.

3. Bentuk 3NF (3th Normal Form)

Aturan :

Sudah berada dalam bentuk normal kedua (2NF).

Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya).

4. Bentuk 4NF (4th Normal Form)

Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF, dan tabel tersebut tidak boleh memiliki lebih dari sebuah multivalued attribute.

Untuk setiap multivalued dependencies (MVD) juga harus merupakan functional dependencies.

5. Bentuk 5NF (5th Normal Form)

Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah lossless decomposition menjadi tabel-tabel yg lebih kecil.

Jika 4 bentuk normal sebelumnya dibentuk berdasarkan functional dependency, 5NF dibentuk berdasarkan konsep join dependence.

Yakni apabila sebuah tabel telah di-dekomposisi menjadi tabel-tabel lebih kecil, harus bisa digabungkan lagi (join) untuk membentuk tabel semula (Kusrini;2007:40-41)

Contoh penerapan normalisasi data

NoProyek	NamaProyek	NoPegawai	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
		Peg02	Paula	B	900.000
		Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
		Peg12	Sita	B	900.000
		Peg14	Yusni	B	900.000

Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama. Field-field tabel di atas yang merupakan group berulang : NoPegawai, NamaPegawai, Golongan, BesarGaji.

Solusinya hilangkan duplikasi dengan mencari ketergantungan parsial. menjadikan field-field menjadi tergantung pada satu atau beberapa field. Karena yang dapat dijadikan kunci adalah NoProyek dan NoPegawai, maka langkah kemudian dicari field-field mana yang tergantung pada NoProyek dan mana yang tergantung pada NoPegawai. (Kusrini;2007:41)

<i>Noproyek</i>	<i>NamaProyek</i>	<i>Nopegawai</i>	<i>NamaPegawai</i>	<i>Golongan</i>	<i>BesarGaji</i>
NP001	BRR	Peg01	Anton	A	1.000.000
		Peg02	Paula	B	900.000
		Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
		Peg12	Sita	B	800.000
		Peg14	Yusni	B	900.000

Field-field yang tergantung pada satu field haruslah dipisah dengan tepat, misalnya NoProyek menjelaskan NamaProyek dan NoPegawai menjelaskan NamaPegawai, Golongan dan Besar Gaji.

Tabel 1

<i>Noproyek</i>	<i>NamaProyek</i>
NP001	BRR
NP002	PEMDA

Tabel 2

<i>Nopegawai</i>	<i>NamaPegawai</i>	<i>Golongan</i>	<i>BesarGaji</i>
Peg01	Anton	A	1.000.000
Peg02	Paula	B	900.000
Peg06	<u>Koko</u>	C	750.000
Peg12	<u>Sita</u>	B	800.000
Peg14	<u>Yusni</u>	B	900.000

<i>Noproyek</i>	<i>NoPegawai</i>
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

Pada tabel diatas masih terdapat masalah, bahwa BesarGaji tergantung kepada Golongan nya. Padahal disini Golongan bukan merupakan field kunci. Artinya kita harus memisahkan field non-kunci Golongan dan BesarGaji yang tadinya tergantung secara parsial kepada field kunci NoPegawai, untuk menghilangkan ketergantungan transitif. (Kusrini;2007:42)

II.4.5. MySQL

MySQL (My Strukture Query Language) adalah program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *Multi User* (Banyak Pengguna). Saat ini database *MySQL* telah digunakan hampir semua *programer database*, apalagi dalam pemrograman *Web*. (Bunafit

Nugroho;2004:29). *MySQL (My Strukture Query Language)* juga merupakan sebuah program pembuat database yang bersifat open source, artinya siapa saja boleh menggunakannya dan tidak dicekal. Saat kita mendengar open source, kita ingat dengan sistem operasi handal keturunan Unix ,yaitu Linux. *MySQL* sebenarnya produk yang berjalan pada platform Linux, karena sifatnya yang open source, dia dapat dijalankan pada semua platform baik Windows maupun Linux.

Kelebihan lain dari *MySQL* adalah menggunakan bahasa *Query standar* yang dimiliki *SQL (Structure Query Language)*. Sebagai sebuah program penghasil *database*, *MySQL* tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (*interpace*).

Fitur-fitur *MySQL* antara lain :

1. *Relational Database System.*

Seperti halnya software database lain yang ada di pasaran, *MySQL* termasuk RDBMS.

2. *Arsitektur Client-Server.*

MySQL memiliki arsitektur *client-server* dimana server *database MySQL* terinstal di server. *Client MySQL* dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet. Mengenal perintah SQL standar. *SQL (Structured Query Language)* merupakan suatu bahasa standar yang berlaku di hampir semua software database. *MySQL* mendukung SQL versi SQL:2003.

3. Mendukung *Sub Select*.

Mulai versi 4.1 *MySQL* telah mendukung select dalam select (*sub select*).

4. Mendukung *Views*.

MySQL mendukung views sejak versi 5.0

5. Mendukung *Stored Prosedured* (SP).

MySQL mendukung SP sejak versi 5.0

6. Mendukung *Triggers*.

MySQL mendukung trigger pada versi 5.0 namun masih terbatas.

Pengembang *MySQL* berjanji akan meningkatkan kemampuan trigger pada versi 5.1. (Bunafit Nugroho;2004:30)

II.4.6. PHP (*Page Hypertext Preprocessor*)

PHP merupakan bahasa standar yang digunakan dalam dunia *website*. *PHP* adalah bahasa program yang berbentuk script yang diletakkan didalam server web. Jika kita lihat dari sejarah, mulanya *PHP* diciptakan dari ide Rasmus Lerdorf yang membuat sebuah script perl. Script tersebut sebenarnya dimaksudkan digunakan sebagai program untuk dirinya sendiri. Akan tetapi, kemudian dikembangkan lagi sehingga sebuah bahasa yang disebut “*Personal Home Page*”. Inilah awal mula munculnya *PHP* sampai saat ini.

PHP adalah bahasa program yang baru dibangun sekitar tahun 1994/1995 oleh Rasmus Lerdorf telah meletakkan bersama perl script untuk membuat siapa yang telah melihat resumennya terkesan. Kemudian sedikit demi sedikit pengguna

atau programer mulai menyukai script ini. Adapun beberapa cara menuliskan script PHP :

```
<? dan ?>  
<?php dan ?>  
<% dan %>  
<script language="php"> dan </script>
```

Setiap halaman yang mengandung script PHP harus disimpan dengan EXTENSI PHP sesuai dengan program PHP yang mendukung dan setiap script PHP harus didahului dengan pembuka PHP (<?php dan lain-lain) dan kemudian diakhiri dengan penutup (?>). (Bunafit Nugroho;2004:140-144)

II.4.7. 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. (Jurnal Informatika Mulawarman 2011;Haviluddin:1)

II.4.7.1. 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. (Jurnal Informatika Mulawarman 2011;Haviluddin:3).

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

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu :

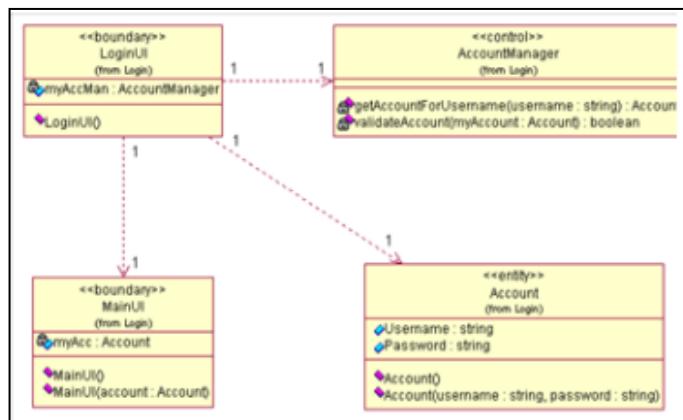
1. *Structure Diagram.*

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

- a. *Class diagram* 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

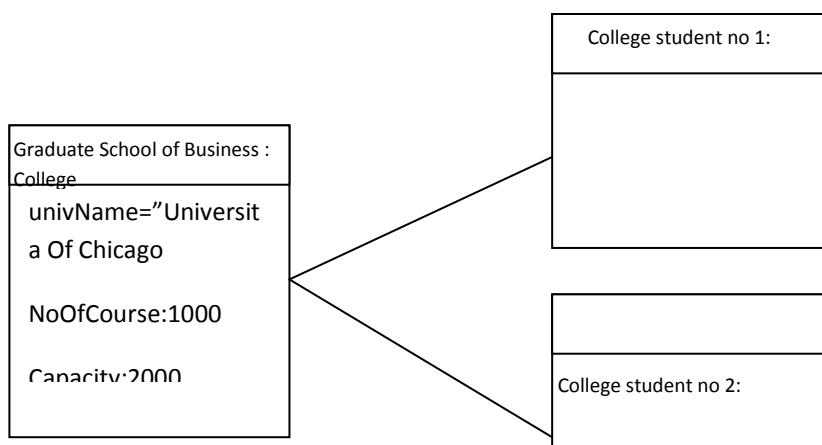


Gambar II.4 Class Diagram

Sumber : (Jurnal Informatika Mulawarman 2011;Haviluddin:3)

b. 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.

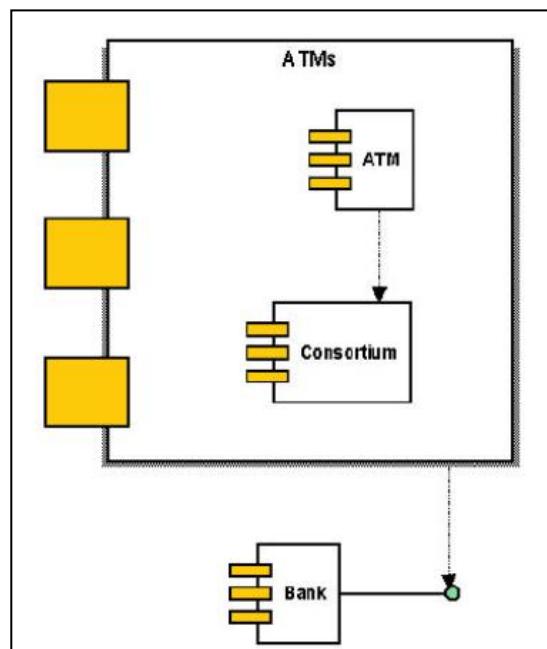


Gambar II.5. Object

Sumber : (Jurnal Informatika Mulawarman 2011;Haviluddin:3)

c. Component diagram

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



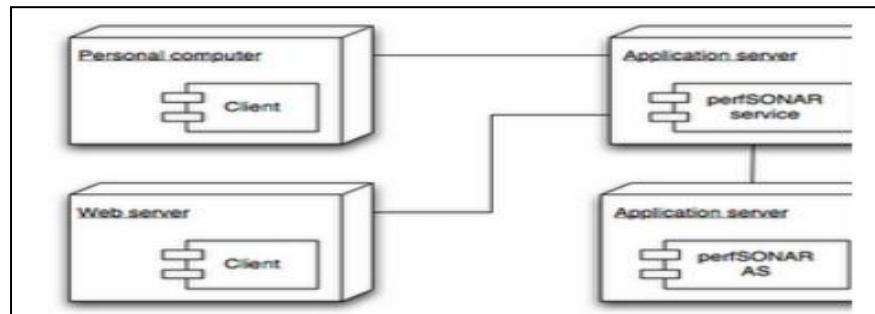
Gambar II.6. Component Diagram

Sumber : (Jurnal Informatika Mulawarman 2011: Hayiluddin:3)

d. *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.

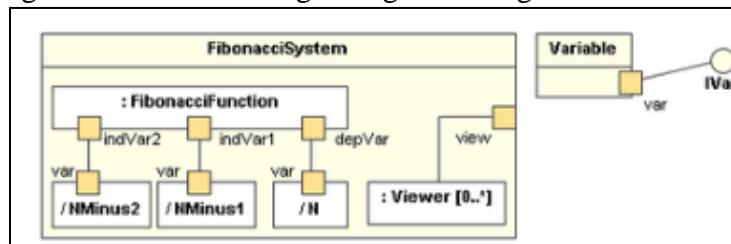


Gambar II.7. Composite Diagram.

Sumber : (Jurnal Informatika Mulawarman 2011;Haviluddin:4)

e. *Composite structure diagram*

Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi.

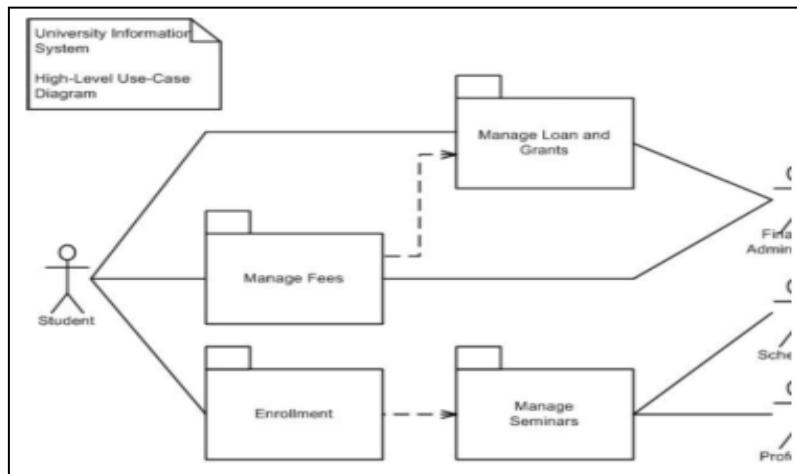


Gambar II.8. Composite Diagram.

Sumber : (Jurnal Informatika Mulawarman 2011;Haviluddin:4)

f. *Package diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek software. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.



Gambar II.9. Package Diagram.

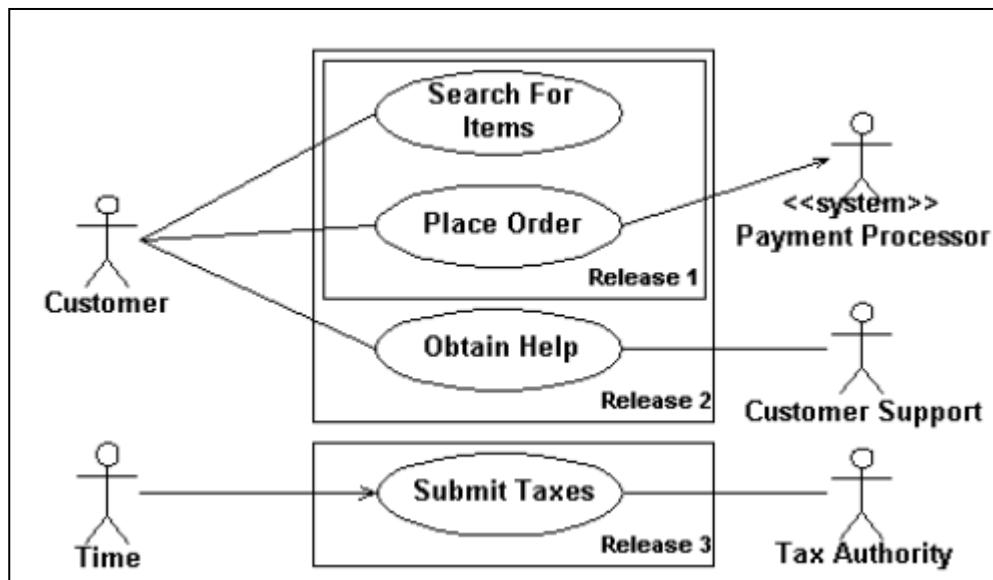
Sumber : (Jurnal Informatika Mulawarman 2011;Haviluddin:4)

2. Behavior Diagram

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

a. 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*.

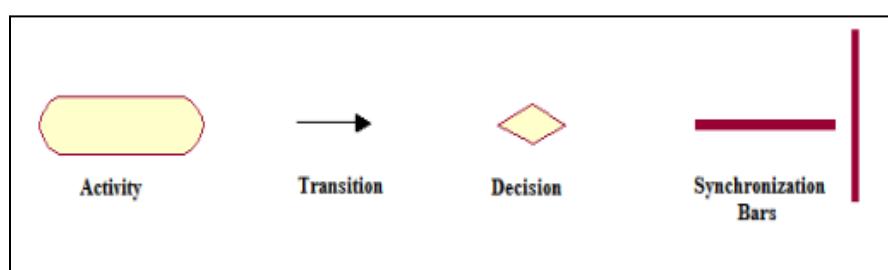


Gambar II.10. Use Case Diagram

Sumber : (Jurnal Informatika Mulawarman 2011;Haviluddin:4)

b. Activity diagram

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

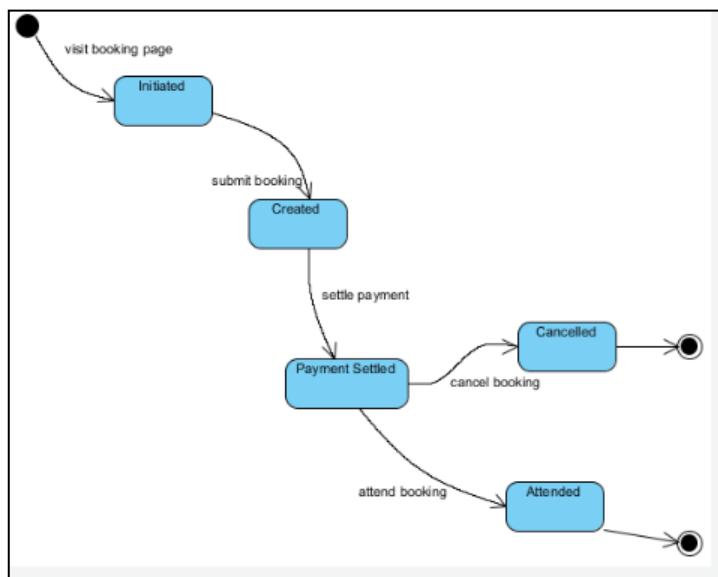


Gambar II.11. Activity Diagram

Sumber : (Jurnal Informatika Mulawarman 2011;Haviluddin:4)

c. State Machine diagram (State chart diagram in version 1.x)

Menggambarkan state, transisi state dan event.



Gambar II.12. *State Machine diagram*