



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Aplikasi

Menurut Asropudin (2013 : 35), “pengertian aplikasi adalah *software* yang dibuat oleh suatu perusahaan computer untuk mengerjakan tugas-tugas tertentu, misalnya Ms-Word, Ms-Excel”.

2.1.2 Pengertian Software

Menurut Ladjamudin (2013 : 20), “*Software* merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melakukan tugas tertentu”.

2.1.3 Pengertian Pengolahan Data

Menurut Ladjamudin (2013 : 9) . “Pengolahan Data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan”.

2.1.4 Pengertian Data

Menurut Ladjamudin (2013 : 8), “Data adalah deskripsi dari sesuatu dan kejadian-kejadian yang nyata”.

Menurut Ladjamudin (2013 : 20), “Data merupakan komponen dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi”.

2.1.5 Pengertian Dokumentasi

Menurut Purwono (2009 : 362), “Dokumentasi adalah semua tulisan yang dikumpulkan dan disimpan yang dapat digunakan bila diperlukan”.

2.2 Teori Khusus

2.2.1 UML (*Unified Modelling Language*)

Menurut Pratama (2013 : 48), “*Unified Modelling Language (UML)* adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemrograman berorientasi objek”.



Gambar 2.1 Logo *UML*

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

2.2.2 Jenis-Jenis Diagram *UML*



Tabel 2.1 Tipe Diagram *UML*

No	Diagram	Tujuan
1	<i>Class</i>	Memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi
2	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen
3	<i>Use Case</i>	Diagram ini memperlihatkan himpunan <i>use case</i> dan ctor-aktor (suatu jenis khusus dari kelas)
4	<i>Sequence</i>	Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu
5	<i>Communication</i>	Sebagai pengganti diagram kolaborasi <i>UML</i> 1.4 yang menekankan organisasi ctoral dari obyek-obyek yang menerima serta mengirim pesan
6	<i>Statechart</i>	Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (<i>state</i>), transisi, kejadian serta aktivitas
7	<i>Activity</i>	Tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem
8	<i>Component</i>	Memperlihatkan organisasi serta kebergantungan sistem / perangkat lunak pada komponen-komponen yang telah ada sebelumnya

Lanjutan Tabel 2.1 Tipe Diagram *UML*

No	Diagram	Tujuan
9	<i>Deployment</i>	Memperlihatkan konfigurasi saat aplikasi dijalankan (<i>run-time</i>)

Sumber : elib.unikom.ac.id

2.2.3 Use Case Diagram

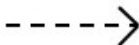
Menurut pratama (2013 : 48), “*Use case diagram* menggambarkan aliran kegiatan dan proses bisnis yang dilakukan oleh pengguna (aktor)”.

Komponen pembentuk diagram use case adalah:

1. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
2. *Use case*, aktivitas/sarana yang disiapkan oleh bisnis / sistem.
3. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

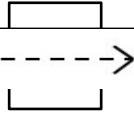
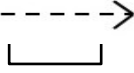
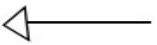
Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.2 Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung yang tidak mandiri.

Lanjutan Tabel 2.2 Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada

			di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan dan elemen lain yang bekerja sama untuk agar perilaku yang lebih besar dari jumlah dan elemen-elemennya

Lanjutan Tabel 2.2 Simbol *Use Case* Diagram

No	Gambar	Nama	Keterangan
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber : elib.unikom.ac.id

2.2.4 Class Diagram

BAB II Tinjauan

Menurut pratama (2013 : 48), “*Class diagram* menggambarkan struktur kelas yang terdapat pada perangkat lunak yang akan dibangun”.

Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

Tabel 2.3 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

Lanjutan Tabel 2.3 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen



			mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>)
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber : elib.unikom.ac.id

2.2.5 Activity Diagram

Menurut pratama (2013 : 50), “*Activity diagram* menggambarkan rangkaian aliran dari aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksinya”.

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut :

Tabel 2.4 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Start Point</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2		<i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih



			dari satu.
4	<i>Join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	<i>End Point</i>		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

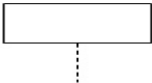
Title
Function

2.2.6 Sequence Diagram

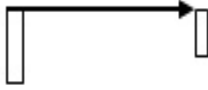
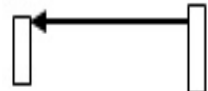
Menurut Pratama (2013 : 48), “*Sequence* diagram menggambarkan *sequence* (aliran) pengiriman pesan (*message*) yang terjadi di aplikasi, sebagai bentuk interaksi dengan pengguna (*user*)”.

Adapun simbol-simbol yang digunakan dalam *sequence* diagram adalah sebagai berikut :

Tabel 2.5 Simbol *Sequence* Diagram

No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	Objek <i>entity</i> , antar muka yang saling berinteraksi.



2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : elib.unikom.ac.id

2.3 Teori Program

2.3.1 Pengertian *PHP*

Menurut Pratama (2013 : 50), “*PHP* adalah singkatan dari *Hypertext Processor*, sebuah bahasa *scripting* server-side yang dirancang untuk pengembangan web tetapi juga digunakan sebagai bahasa pemrograman tujuan umum.”

Menurut Winarno,et.al (2014 : 49), “ *PHP* adalah bahasa *script* yang sangat cocok untuk pengembangan *Web* dan dapat dimasukkan kedalam *HTML*.”

2.3.1.1 Sintaks Dasar *PHP*

Dengan adanya kode *PHP*, sebuah halaman web bisa melakukan banyak hal yang dinamis, seperti mengakses database, membuat gambar, membaca dan menulisi file, dan sebagainya.

Adapun 4 jenis tag yang terdapat dalam *PHP* yaitu:

Tabel 2.6 Sintaks Dasar *PHP*

No	Jenis Tag	Tag pembuka	Tag penutup
1	Tag standar	<?php	?>



2	Tag pendek	<?	?>
3	Tag ASP	<%	%>
4	Tag Script	<script language="php">	</script>

Adapun contoh sintaks dasar dalam *PHP* yaitu:

```
<?php /*Pembuka PHP*/  
/* Isi Program */  
/* Penutup */ ?>
```

2.3.2 Pengertian *Notepad++*

Menurut Pratama (2013 : 53) , “*Notepad ++* adalah editor teks dan sumber editor kode untuk digunakan dengan Microsoft Windows. Tidak seperti Notepad, built-in Windows editor teks, mendukung pengeditan tab, yang memungkinkan bekerja dengan beberapa file yang terbuka dalam satu jendela.”

2.3.3 Pengertian *Database*

Menurut Pratama (2013 : 52) , “Basis data (*Database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis, sehingga data digunakan oleh suatu program komputer untuk memperoleh informasi dari basis data tersebut.”

Menurut Kristanto (2004 : 2) , adapun definisi-definisi yang terdapat pada *Database*, yaitu:

a. Entity

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang Administrasi Siswa misalnya entity adalah siswa, buku, pembayaran, nilai test. Pada bidang kesehatan, entity adalah pasien, dokter, obat, kamar, diet.

b. Attribute



Setiap *entity* mempunyai *atribute* atau sebutan untuk mewakili suatu *entity*. Seorang siswa dapat dilihat dari atributenya, misalnya nama, nomor siswa, alamat, nama orang tua, hobby. *Atribute* juga disebut sebagai data elemen, *data field*, *data item*.

c. Data Value

Data value adalah data aktual atau informasi yang disimpan pada tiap data elemen atau *atribute*. *Atribute* nama karyawan menunjukkan tempat dimana informasi nama karyawan disimpan, sedang *data value* adalah Sutrisno, Budiman, merupakan isi data nama karyawan tersebut.

d. Record

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap. Satu *record* mewakili satu data atau informasi tentang seseorang misalnya, nomor karyawan, nama karyawan, alamat, tanggal masuk.

e. File

Kumpulan *record-record* sejenis yang mempunyai panjang elemen yang sama, *atribute* yang sama, namun berbeda data valuenya.

f. Database Management System(DBMS)

Kumpulan file yang saling berkaitan bersama dengan program untuk pengelolaannya disebut sebagai DBMS. *Database* adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam satu paket program yang komersial untuk membaca data, mengisi data, menghapus data, melaporkan data dalam *database*.

2.3.4 Pengertian MySQL

Menurut Asropudin (2013 : 54), “*MySQL* adalah salah satu jenis *database server* yang sangat terkenal disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses databasenya. Selain itu, *MySQL* bersifat *Open Source* dan *MySQL* termasuk kedalam jenis *RDBMS (Relational Database*



Management System), itulah sebabnya MySQL menggunakan istilah seperti tabel, baris, dan kolom.

2.3.5 Xampp

Menurut Komputer Wahana (2014 : 72), “XAMPP merupakan singkatan dari X(empat sistem operasi apapun), Apache, MySql, PHP, Dan Perl. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam Paket XAMPP sudah terdapat Apache (*Web Server*), MySql (*Database*), PHP (*Server Side Scripting*), Perl, FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya”.

2.3.6 Pengertian HTML

Menurut Asropudin. “*Hyper Text Markup Language*, adalah bahasa computer yang digunakan untuk membuat sebuah halaman *webpage*.”