

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Dasar**

##### **2.1.1 Sistem**

Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen, atau variable yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu ( Tata Sutabri, 2012 ).

Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu, sebuah sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai sesuatu sistem.

Adapun karakteristik yang dimaksud adalah sebagai berikut:

##### 1. Komponen sistem (*component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar atau sering disebut “supra sistem”

##### 2. Batasan sistem (*boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

### 3. Lingkungan luar sistem (*envirotment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang memepengaruhi oprasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

### 4. Penghubung sistem (*interface*)

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

### 5. Masukan sistem (*input*)

Energi yang dimasukkan kedalam sistem tersebut masukkan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh, di dalam suatu unit sistem komputer, “program” adalah maintenance input yang digunakan untuk mengoprasikan komputernya dan “data” adalah signal input untuk diolah menjadi informasi.

### 6. Keluaran sistem (*output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

## 7. Pengolahan sistem (proses)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

## 8. Sasaran sistem (*objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministic. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan yang telah direncanakan.

### **2.1.2 Informasi**

Informasi adalah data yang telah diklasifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari tak berguna menjadi berguna bagi penerimanya. Nilai informasi berhubungan dengan keputusan maka informasi menjadi tidak diperlukan keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang. Nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan.( Tata Sutabri, 2012 )

### **2.1.3 Sistem Informasi**

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.( Tata Sutabri, 2012 )

#### **2.1.4 Website**

Website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan. Hubungan antara satu halaman web dengan halaman web yang lainnya disebut dengan hyperlink, sedangkan teks yang dijadikan media penghubung disebut *hypertext* (Yuhfizar, 2009).

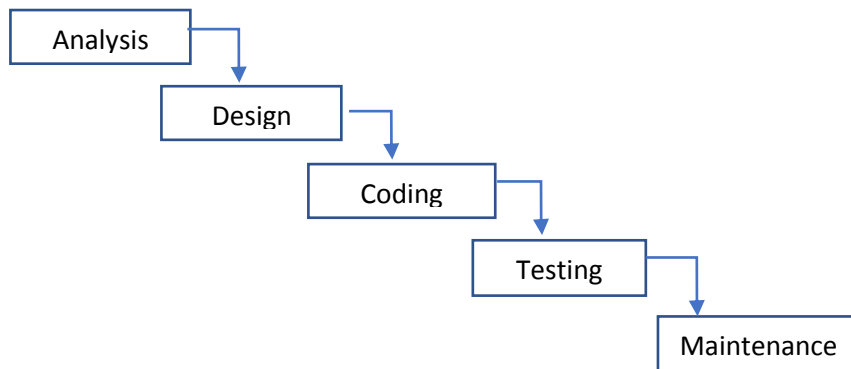
#### **2.1.5 Promosi**

Secara etimologi promosi berasal dari bahasa latin *promovere*, *pro* berarti *forward* (ke depan) dan *movere* berarti *move* (bergerak). Jadi dapat dikemukakan bahwa promosi “*to move forward*” atau bergerak ke arah depan. Para ahli ekonomi dalam mengemukakan pendapat tentang pengertian promosi mempunyai konsep yang berbeda-beda antara satu dengan yang lainnya. Namun, meski konsep yang dikemukakan berbeda tetapi tujuan akhirnya tetap sama. Menurut Mursid (2008:95): “Promosi adalah komunikasi yang persuasif, mengajak, mendesak, membujuk, dan meyakinkan”. Menurut Djaslim Saladin (2003:123): “Promosi yaitu suatu komunikasi informasi penjual dan pembeli yang bertujuan untuk merubah sikap dan tingkah laku pembeli, yang tadinya tidak mengenal menjadi mengenal sehingga menjadi pembeli dan tetap mengingat produk tersebut”.

### **2.2 Siklus Hidup Sistem**

Pembuatan program sistem berbasis web ini menggunakan model sekuensial linier sering disebut juga dengan siklus kehidupan klasik atau model air terjun (*waterfall*). Model ini mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dalam tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan.

Model *waterfall* merupakan metode yang paling banyak digunakan dalam *software engineering*, karena pemodelan sistem terbagi menjadi tahapan-tahapan yang mengikuti pola teratur, seperti layaknya air terjun. Tahapan-tahapan pada model *waterfall* dapat dilihat pada gambar sebagai berikut :



Gambar 2.1 Metode Waterfall

Tahapan utama siklus hidup dalam membangun sistem terdiri dari :

*a. Analysis*

Tahapan *analysis* adalah tahapan yang dilakukan pendefinisian, pemahaman atas masalah-masalah dan hambatan-hambatan pada sistem yang ada serta dengan kebutuhan-kebutuhan pemakai pada sistem baru.

*b. Design*

Tahapan design adalah tahap yang dilakukan untuk memberikan gambaran secara umum kepada user tentang sistem yang baru. Desain sistem adalah persiapan dari sistem secara rinci. Desain sistem ini bertujuan untuk mengidentifikasi komponen-komponen sistem yang akan didesain secara rinci meliputi desain model, output, input, database, teknologi dan kendali.

*c. Coding*

Tahap coding ini adalah tahapan pembuatan program yang telah dirancang sebelumnya.

*d. Testing*

Tahap testing ini adalah tahapan pengujian program yang telah dibangun.

*e. Maintenance*

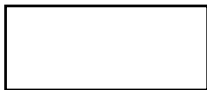
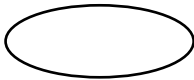
Tahapan maintenance adalah tahapan yang dilakukan untuk mengetahui tingkat kebutuhan program setelah program tersebut di implementasi.

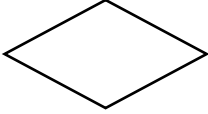

### 2.2.1 Alat bantu yang digunakan dalam perancangan sistem

Alat bantu yang digunakan dalam perancangan sistem memaparkan relasi antar entitas dalam basis data dengan *Entity Relationship Diagram (ERD)* dan bagaimana data mengalir ke sebuah sistem dengan *Data Flow Diagram (DFD)*.

#### 2.2.1.1 ERD (*Entity Relationship Diagram*)

ERD adalah model konseptual yang mendeskripsikan hubungan setiap penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antardata, karena hal ini relatif kompleks. ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Berikut adalah notasi-notasi yang digunakan dalam ERD yang ditunjukkan pada tabel 2.1 :

Simbol	Keterangan
	<p>Himpunan Entitas</p> <p>Himpunan entitas menunjukan objek-objek dasar yang terkait di dalam sistem.</p>
	<p>Atribut</p> <p>Bertugas untuk mengekspresikan karakteristik entitas, missal : sebuah entitas pegawai yang mempunyai atribut nama, alamat, email. Atribut diklasifikasikan sebagai entity ley atau entity descriptor.</p>

	<p>Relasi</p> <p>Kerelasian antar entitas mendefinisikan hubungan antar dua buah entitas.</p>
	<p>Data Flow</p> <p>Berfungsi menggambarkan aliran data atau paket informasi dari satu bagian sistem ke bagian lain. Arah panah menggambarkan aliran data.</p>

T  
abel 2.1  
Simbol  
ERD

Relasi antar entitas dapat dikelompokkan dalam empat jenis yaitu:

1. *One to one relationship*

Satu entitas pada himpunan entitas A dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas B, dan entitas pada himpunan entitas B dihubungkan dengan paling banyak satu entitas dalam himpunan entitas A.

2. *One to many relationship*





Satu entitas pada himpunan entitas A dapat berhubungan dengan lebih dari satu entitas pada himpunan entitas B, dan entitas pada himpunan entitas B hanya dapat berhubungan dengan paling banyak satu entitas dalam himpunan entitas A.

3. *Many to one relationship*

Satu entitas pada himpunan entitas A dapat berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan entitas pada himpunan entitas B dapat berhubungan dengan lebih dari satu entitas dalam himpunan entitas A.

#### 4. Many to many relationship

Satu entitas pada himpunan entitas A dapat berhubungan dengan lebih dari satu entitas pada himpunan entitas B, dan entitas pada himpunan entitas B dapat berhubungan dengan lebih dari satu entitas dalam himpunan entitas A yang ditunjukkan pada tabel 2.2

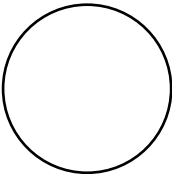

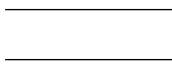
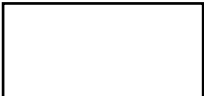
Jenis Relasi	Simbol
1-ke-1	
1-ke-n	
n-ke-1	
n-ke-n	

Tabel 2.2 Simbol jenis relasi



### 2.2.1.2 DFD (*Data Flow Diagram*)

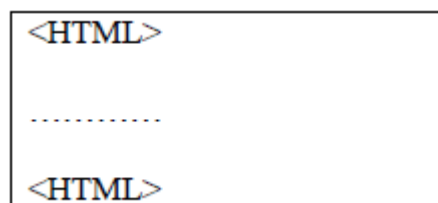
DFD merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan menggunakan DFD adalah memudahkan user yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan. Simbol yang akan digunakan dalam DFD ditunjukkan pada tabel 2.3

Simbol	Keterangan
	<p>Proses</p> <p>Proses sering juga disebut dengan bubble. Berfungsi menggambarkan aliran data masuk menjadi aliran data keluar.</p>
	<p>Data Flow</p> <p>Berfungsi menggambarkan aliran data atau paket informasi dari satu bagian lain. Arah panah menggambarkan aliran data.</p>
	<p>Data Store</p> <p>Berfungsi menggambarkan model dari kumpulan paket data yang tersimpan.</p>
	<p>Eksternal Entity</p> <p>Sering juga disebut terminator. Berfungsi menggambarkan kesatuan luar yang berhubungan dengan sistem.</p>

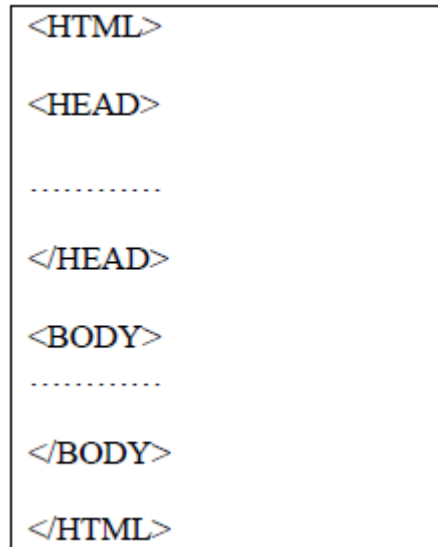
Tabel 2.3 Simbol DFD

### 2.3 HTML (*Hyper Text Markup Language*)

HTML adalah bahasa yang digunakan untuk menulis halaman web. Sekalipun banyak yang menyebutnya sebagai suatu bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman, HTML merupakan pengembangan dari standar pemformatan dokumen teks, yaitu *Standard Generalized Markup Language* (SGML). HTML sering juga disebut sebagai bahasa penandaan (*markup*) terhadap sebuah dokumen teks. Tanda tersebut digunakan untuk menentukan format atau style dari teks yang ditandai. Simbol markup yang digunakan oleh HTML ditandai dengan lebih kecil (<) dan tanda lebih besar (>) dan disebut tag. Sebuah file HTML merupakan file teks biasa yang mengandung tag-tag HTML. Karena merupakan file teks, maka HTML dapat dibuat dengan menggunakan teks editor yang sederhana, misalnya NotePad . Tanda bahwa sebuah file teks merupakan HTML terdapat pada eksistensi filenya, yaitu .htm atau .html. Untuk menandai suatu file teks berupa HTML, terlebih dahulu harus terkandung struktur sebagai berikut :



Tag <HTML> tersebut harus diletakkan pada paling awal dan tag </HTML> harus diletakkan pada bagian akhir. Strukturnya sebagai berikut :



Bagian head umumnya berisi informasi mengenai dokumen tersebut, misalnya judul dokumen, sedangkan bagian body berisi layout atau desain halaman web.

#### 2.4 PHP (*Hypertext Preprocessor*)

PHP adalah sebuah bahasa pemrograman yang berbentuk scripting, sistem kerja dari program ini adalah sebagai interpreter bukan sebagai compiler. Sebagian besar sintaks PHP mirip dengan bahasa C dan Java, tetapi ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat. Menurut cara penulisannya, script PHP dibagi menjadi dua, yaitu *Embedded Script* dan *NonEmbedded Script*.

##### 1. Embedded Script

*Embedded Script* adalah script PHP yang disisipkan di antara tag-tag dokumen HTML.

##### 2. NonEmbedded Script

*NonEmbedded Script* adalah script atau program PHP murni. Termasuk tag HTML yang disisipkan dalam script PHP.

#### 2.5 MySQL (*My Structure Query Language*)

MySQL merupakan database yang paling digemari dikalangan programmer web, program ini merupakan database yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Sebagai database server yang mampu mememanajemen database dengan baik, MySQL terhitung merupakan database yang paling digemari dan paling banyak digunakan dibanding database lainnya . Di dalam dunia Internet, MySQL dijadikan sebagai sebuah database yang paling banyak digunakan selain database yang bersifat share ware seperti *Ms Access*, penggunaan MySQL ini biasanya dipadukan dengan menggunakan program aplikasi PHP, karena dengan menggunakan kedua program tersebut diatas telah terbukti akan kehandalannya dalam menangani permintaan data. Kemampuan lain yang dimiliki MySQL adalah mampu mendukung *Relasional Database Manajemen System* (RDBMS), sehingga dengan kemampuan ini MySQL akan mampu menangani data-data sebuah perusahaan yang berukuran sangat besar hingga berukuran giga byte.

## **2.6 PhpMyAdmin**

PhpMyAdmin merupakan sebuah program bebas yang berbasis web yang dibuat menggunakan aplikasi PHP, yang bertujuan untuk mengakses database MySQL yang digunakan untuk menjadi administrator dari server MySQL. Dengan kelebihan-kelebihan yang ada mengakibatkan para pengguna awal tidak harus mampu untuk mengetahui sintaks-sintaks SQL dalam pembuatan database dan table.

## **2.7 Teori Pengembangan Sistem**

### **2.7.1 *Unified Modeling Language* (UML)**

Menurut Verdi Yasin, mendefinisikan *Unified Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan

Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. tipe-tipe Diagram UML adalah sebagai berikut :

a. Use Case Diagram

*Use Case Diagram* adalah gambar dari beberapa atau seluruh aktor dan use case dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

Sebuah use case mempresentasikan sebuah interaksi antara actor dan sistem.

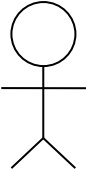
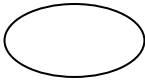

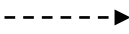
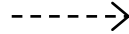


Komponen-komponen relasi dalam use case antara lain :

1. *Association*, Menghubungkan link antar element.
2. *Generalization*, Disebut juga inheritance (pewaris), sebuah elemen dapat merupakan spesialis dari elemen lainnya.
3. *Dependency*, Sebuah element bergantung dalam beberapa cara ke element lainnya.
4. *Aggregation*, Bentuk association dimana sebuah elemen berisi elemen lainnya.

Tipe relasi/ stereotype yang mungkin terjadi pada use case diagram:

- a. *<<include>>*, Yaitu kelakuan yang harus terpenuhi agar sebuah event dapat terjadi, dimana pada kondisi ini sebuah use case adalah bagian dari use case lainnya.
- b. *<<extends>>*, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.
- c. *<<communicates>>*, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah *communicates association*. Ini merupakan pilihan selama asosiasi hanya tipe relationship yang

dibolehkan antara actor dan use case. Dalam use case diagram terdapat istilah seperti aktor, *use case* dan *case relationship*. Penjelasan simbol pada tabel 2.4

Simbol	Keterangan
	<b>Aktor</b> : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<b>Use case</b> : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.
	<b>Association</b> : adalah relasi antara actor dan <i>use case</i> .
	<b>Generalisasi</b> : untuk memperlihatkan struktur pewaris yang terjadi.
	<b>Include</b> : menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i>
	<b>Collaboration</b> : interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi)
	<b>Extend</b> : menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan

Tabel 2.4 Simbol Use Case

## 1. Activity Diagram

*Activity diagram* menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi. *Activity Diagram* berupa flowchart yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity* diagram adalah sebagai berikut :

a. *Activity*

Notasi yang menggambarkan pelaksanaan dari beberapa proses dalam aliran pekerjaan.

b. *Transition*

Notasi yang digunakan untuk memperlihatkan jalan aliran kontrol dari *activity* ke *activity*.

c. *Decision*




Notasi yang menandakan kontrol cabang aliran berdasarkan *decision point*.

d. *Synchronization bars*

Aliran kerja notasi ini menandakan bahwa beberapa aktivitas dapat diselesaikan secara bersamaan.

Komponen-komponen dari *activity* diagram dapat dilihat pada tabel 2.5 berikut :

No	Gambar	Nama	Keterangan
1		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi

3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Tabel 2.5 Simbol activity diagram

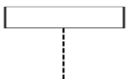
## 2. Sequence Diagram

*Sequence diagram* menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

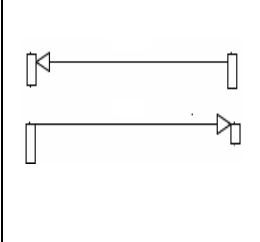
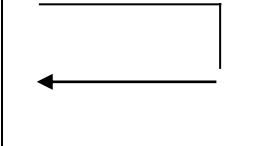
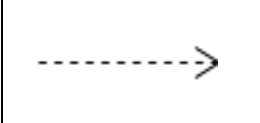
*Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Dalam sequence diagram terdapat 2 simbol yaitu :

- a. *Actor*, untuk menggambarkan pengguna sistem.
- b. *Lifeline*, untuk menggambarkan kelas dan objek.

Komponen-komponen dari Sequence Diagram dapat dilihat pada tabel 2.6 berikut:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.



2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message to self</i>	Menggambarkan pesan yang menuju dirinya sendiri
4		<i>Return Message</i>	Menggambarkan pengembalian dari pemanggilan prosedur

Tabel 2.6 Sequence Diagram

### 3. Class Diagram


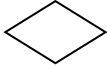
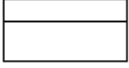


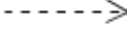

*Class diagram* menggambarkan struktur data dan deskripsi class, package, dan objek beserta hubungan satu sama lain. *Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain.

Class memiliki 3 area pokok antara lain :

- a. Nama
- b. Atribut
- c. Metode

*Class* menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut. Komponen-komponen dari *Class Diagram* dapat dilihat pada tabel 2.7 berikut:

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.
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 mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2.7 Class Diagram

## 2.8 Kamus Data

Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD ( Menurut rossa.a.s dan M.Shalahuddin, 2013) Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi :

- A. Nama-nama dari data.
- B. Gunakan pada - merupakan proses-proses yang terkait dengan data.
- C. Deskripsi - merupakan deskripsi data.
- D. Informasi tambahan – seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk data.

kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

SIMBOL	KETERANGAN
=	Disusun atau terdiri dari
+	Dan
[]	Baik...atau.....
{ }n	N kali diulang atau bernilai banyak

( )	Data optional
*...*	Batas komentar

Tabel 2.8 Simbol Kamus Data

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data dengan perancangan dengan DFD masih belum sesuai.

## 2.9 Perangkat Lunak Pendukung

### 2.9.1 *Adobe dreamweaver*

( Menurut Sigit, 2010 ) *Adobe Dreamweaver* adalah sebuah HTML editor profesional untuk mendesain web secara visual dan mengelola situs atau halaman web. Saat ini terdapat *software* dari kelompok *Adobe* yang belakangan banyak digunakan untuk mendesain suatu *web*. Versi terbaru dari *Adobe Dreamweaver* memiliki beberapa kemampuan bukan hanya sebagai *software* untuk desain *web* saja, tetapi juga menyunting kode serta pembuatan aplikasi *web*. Antara lain: JSP, PHP, ASP, XML, dan *ColdFusion*.

*Adobe Dreamweaver* merupakan *software* utama yang digunakan oleh *Web Designer* dan *Web Programmer* dalam mengembangkan suatu situs *web*. Hal ini disebabkan oleh ruang kerja, fasilitas, dan kemampuan *Dreamweaver* yang mampu meningkatkan produktivitas dan efektifitas, baik dalam desain maupun membangun suatu situs *web*.

### 2.9.2 *Notepad ++*

Selain menggunakan aplikasi *Dreamweaver*, peneliti juga menggunakan aplikasi notepad ++ dikarenakan tampilan interface aplikasi yang lebih mudah dipahami dalam melihat source program. Notepad++ merupakan sebuah penyunting teks dan penyunting kode sumber yang tersedia dalam beberapa sistem operasi seperti Windows, Linux, dan Mac OS X. Notepad ++ menggunakan komponen Scintilla

untuk dapat menampilkan dan menyunting teks dan berkas kode sumber berbagai bahasa pemrograman. Notepad ++ didistribusikan sebagai perangkat lunak bebas. Proyek ini dilayani oleh *Sourcefoge.net* dengan telah diunduh lebih dari 27 juta kali dan dua kali memenangkan penghargaan *SorceForge Community Choice Award for Best Developer Tools*.