



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Jogiyanto (2005:1), komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas; menerima *input*, memproses *input* sesuai dengan program, menyimpan perintah-perintah dan hasil dari pengolahan, menyediakan *output* dalam bentuk informasi.

Wahyudi (2012:3), komputer adalah peralatan (device) yang menerima data (input) dan menyimpan (storage) kemudian di proses (process) untuk menghasilkan data dalam bentuk lain (output).

2.1.2. Pengertian Sistem

Ladjamudin (2013:3), sistem adalah bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud.

Sukamto dan Shalahuddin (2013:2), sistem adalah kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

2.1.3. Karakteristik Sistem

Al Fatta (2007:5), untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut ini karakteristik sistem yang dapat membedakan sistem dengan sistem lainnya:

1. Batasan (*boundary*): Penggambaran dari suatu elemen atau unsur mana yang termasuk didalam sistem dan mana yang diluar sistem.
2. Lingkungan (*environment*): Segala sesuatu diluar sistem, lingkungan yang menyediakan asumsi, kendala dan input terhadap suatu sistem.
3. Masukan (*input*): Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu system



-
4. Keluaran (*output*): Sumber daya atau produk (informasi, laporan, dokumen, tampilan layer komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
 5. Komponen (*component*): Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*).
 6. Penghubung (*interface*): Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.

Penyimpanan (*storage*): Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga diantara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

2.1.4. Pengertian Klasifikasi Sistem

Ladjamudin (2013:6-7), sistem dapat diklasifikasikan ke dalam beberapa sudut pandang antara lain:

a. Sistem Abstrak dan Sistem Fisik

Sistem Abstrak merupakan sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem Fisik merupakan sistem yang ada secara fisik. Misalnya sistem computer, sistem operasi, sistem penjualan, dan lain sebagainya.

b. Sistem Alamiah dan Sistem Buatan

Sistem alamiah merupakan sistem yang terjadi karena proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada sang pencipta alam). Misalnya sistem perputaran bumi, sistem pergantian siang dan malam, sistem yang dirancang oleh manusia. Sistem buatan merupakan sistem yang dirancang oleh manusia.

c. Sistem Tertentu dan Sistem Tak Tentu

Sistem Tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Sistem tertentu relative stabil atau konstan dalam jangka waktu yang lama.



Contohnya adalah sistem computer. Sistem Tak Tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsure probalitas. Contohnya adalah sistem social, sistem politik dan sistem demokrasi.

d. Sistem Tertutup dan Sistem Terbuka

Sistem Tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan sekitarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Sedangkan Sistem Terbuka adalah sistem yang berhubungan dan terpengaruh dengan bagian luar sistem. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lain.

2.1.5. Tahap Pengembangan Sistem

Al-Fatta (2007:27), metodologi pengembangan sistem mengacu pada proses seperti berikut :

a. Tahapan Analisis

Tahapan analisis adalah dimana sistem yang sedang berjalan dipelajari dan sistem pengganti diusulkan. Ada lima aktivitas utama dalam tahapan ini yaitu :

1. Pengumpulan informasi

Langkah awal pada tahapan analisis ini adalah mengumpulkan informasi tentang bagaimana proses-proses bisnis yang ada pada sistem lama berjalan.

2. Mendefinisikan sistem *requirement*

Berdasarkan informasi kelemahan sistem yang didapat, analisis sistem kemudian mendefinisikan apa saja yang sebenarnya yang dibutuhkan oleh sistem lama untuk mengatasi masalahnya.

3. Memprioritaskan kebutuhan

Dalam beberapa kasus, kebutuhan yang diperoleh sangat lengkap dan rumit. Ketersediaan waktu dan sumber daya lain untuk menyelesaikan keseluruhan *requirement* bisa saja tidak mencukupi.



4. Menyusun dan mengevaluasi alternatif

Hal yang tidak boleh dilupakan analisis adalah rencana kedua. Setelah menyusun dan memprioritaskan kebutuhan, analisis harus menyiapkan alternatif jika seandainya susunan kebutuhan nantinya akan ditolak oleh *klien*.

5. Mengulas kebutuhan dengan pihak manajemen

Langkah terakhir adalah mengulas kebutuhan yang sudah ada dengan pihak *klien*, karena pihak *klien* yang tahu kebutuhan sistem mereka.

b. Tahapan Desain

Tahapan desain adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang *rill*. Tahapan desain sistem dapat dibagi menjadi dua tahap, yaitu :

1. Desain Logis (*Logical Design*)

Tahapan desain logis biasanya menghasilkan beberapa dokumen, diantaranya dokumen model data, dokumen model proses, rancangan table, hierarki antar modul, sampai dengan antar muka dari sistem yang akan dibuat.

2. Desain Fisik (*Physical Design*)

Bagian ini spesifikasi logis diubah ke dalam detail teknologi dimana pemrograman dan pengembangan sistem bisa diselesaikan.

Pada tahapan desain ada beberapa aktivitas utama yang dilakukan, yaitu :

- a. Merancang dan mengintegrasikan jaringan.
 - b. Merancang arsitektur aplikasi.
 - c. Mendesain antarmuka program.
 - d. Mendesain sistem antarmuka.
 - e. Mendesain dan mengintegrasikan database.
 - f. Membuat prototype untuk detail dari sistem.
 - g. Mendesain dan mengintegrasikan kendali sistem.
-



c. Implementasi

Tahapan implementasi ini terdapat beberapa hal yang perlu dilakukan, yaitu :

1. Testing yaitu menguji hasil kode program yang telah dihasilkan dari tahapan desain fisik. Tujuan pengujian ada dua. Dari sisi pengembang sistem, harus dijamin kode program yang dibuat bebas dari kesalahan *sintaks* maupun logika. Dari sisi pengguna, program yang dihasilkan harus mampu menyelesaikan masalah yang ada pada *klien* dan sistem baru harus mudah dijalankan dan dipahami oleh pengguna akhir.
2. Instalasi. Setelah program lulus ujicoba, maka perangkat lunak dan perangkat keras akan diinstal pada organisasi atau perusahaan *klien* dan secara resmi mulai digunakan untuk menggantikan sistem lama.

d. Pemeliharaan

Hasil dari tahapan ini adalah versi baru dari perangkat lunak yang telah dibuat. Perbaikan yang dilakukan tingkatannya bisa sangat variatif, mulai dari memperbaiki program yang *crash* hingga berfungsi kembali sampai pada penambahan modul-modul program yang baru sebagai jawaban atas perubahan kebutuhan pengguna.

2.1.6. Pengertian Database

Jogiyanto (2005:711), basisdata atau *database* adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan untuk memanipulasinya.

Sukanto dan Shalahuddin (2013:43), basisdata adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.

2.1.7. Pengertian Table

Table adalah objek utama dalam *database* yang digunakan untuk menyimpan sekumpulan data sejenis dalam sebuah objek.



2.1.8. Pengertian Field

Jogiyanto (2005:714), *field* adalah suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya.

2.1.9. Pengertian Record

Jogiyanto (2005:714), record adalah kumpulan dari *Field*. Record menggambarkan suatu unit data individu yang tertentu.

2.1.10. Pengertian Web

Setiawan (2004:15), *web* adalah suatu ruangan yang dapat menampung informasi dalam jaringan internet pada sebuah browser, dengan menambahkan kemampuan untuk mengolah kode-kode tertentu yang secara umum dinamakan tag-tag (delimiter) dan kemampuan untuk dapat meloncat (link) dari halaman ke halaman yang lainnya.

2.2. Teori Khusus

2.2.1. Pengertian Kamus Data

Kristanto (2008:72), kamus data adalah kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap *field* atau *file* di dalam sistem.

Sukanto dan Shalahuddin (2013: 73), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan dan keluaran dapat dipahami secara umum. Berikut akan dijelaskan simbol-simbol pada kamus data.

Tabel 2.1 Simbol-simbol dalam Kamus Data

No	Simbol	Arti
1.	=	Disusun atau terdiri atas
2.	+	AND atau dan
3.	[]	Baik..atau..



4.	{ } _n	N kali diulang / bernilai banyak
5.	()	Data Opsional
6.	*...*	Batas komentar

Sumber : Sukamto dan Shalahuddin (2013:73)

2.2.2. Pengertian UML (Unified Modelling Language)

Munawar (2005:17), UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek”.

Ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

2.2.3. Tipe Diagram UML (Unified Modelling Language)

Diagram	Tujuan	Keterangan
Activity	Perilaku prosedural dan paralel	Sudah ada di UML 1
Class	Class, fitur, dan relasinya	Sudah ada di UML 1
Communication	Interaksi diantara obyek, lebih menekankan ke link	Di UML 1 disebut collaboration
Component	Struktur dan koneksi dari komponen	Sudah ada di UML 1
Composite Sturcture	Dekomposisi sebuah class saat runtime	Baru untuk UML 2
Deployment	Penyebaran / instalasi ke klien	Sudah ada di UML 1
Interaction Overview	Gabungan antara activity dan sequence diagram	Baru untuk UML 2
Object	Contoh konfigurasi	Tidak resmi ada di UML



	instance	1
Package	Struktur hierarki saat kompilasi	Tidak resmi ada di UML 1
Sequence	Interaksi antar obyek. Lebih menekankan pada urutan	Sudah ada di UML 1
State Machine	Bagaimana event mengubah sebuah obyek	Sudah ada di UML 1
Timing	Interaksi antar obyek. Lebih menekankan pada waktu.	Baru untuk UML 2
Use case	Bagaimana user berinteraksi dengan sebuah sistem	Sudah ada di UML 1

Sumber : Munawar (2005 : 23)

2.2.4. Diagram UML (Unified Modelling Language)

Widodo dan Herlawati (2011:10), beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram. Demikian jenis-jenis itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis, jenis diagram itu antara lain :

2.2.4.1. Diagram Kelas (Class Diagram).

Bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek, meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. Berikut akan dijelaskan simbol-simbol dalam Diagram Kelas :

**Tabel 2.2 Simbol-simbol Diagram Kelas**

Simbol	Deskripsi
<p>kelas</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>nama_kelas</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>+atribut</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p>+operasi()</p> </div>	Kelas pada struktur sistem
<p>Antarmuka/interface</p> <div style="text-align: center; margin: 10px 0;">  </div>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
<p>Asosiasi/association</p> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
<p>Asosiasi berarah</p> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
<p>Generalisasi</p> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
<p>Kebergantungan/dependency</p> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna kebergantungan antarkelas
<p>Agregasi/aggregation</p> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna semua-bagian (whole-part).

Sumber : Sukamto dan Shalahuddin (2013 : 146)



2.2.4.2. Diagram Paket (Package Diagram).

Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen. Berikut akan dijelaskan simbol-simbol dari diagram Paket :

Tabel 2.3 Simbol-simbol Diagram Paket

Simbol	Deskripsi
	Package merupakan sebuah bungkusan dari suatu atau lebih kelas elemen.

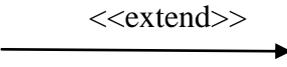
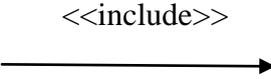
2.2.4.3. Diagram Use-Case.

Bersifat statis, diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. Berikut akan dijelaskan simbol-simbol dalam diagram Use-Case :

Tabel 2.4 Simbol-simbol Diagram Use-Case

Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
<p><i>Aktor/actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar



	orang, tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
<p>Ekstensi/<i>extend</i></p> <p style="text-align: center;">  </p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
<p>Generalisasi/<i>generalization</i></p> <p style="text-align: center;">  </p>	Hubungan generalisasi dan spesialisasi (umum–khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p>Menggunakan/<i>include/uses</i></p> <p style="text-align: center;">  </p> <p style="text-align: center;">  </p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
<p>Asosiasi/<i>association</i></p> <p style="text-align: center;">  </p>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

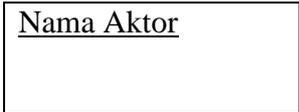
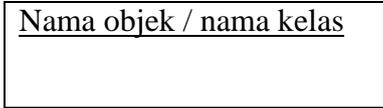
Sumber : Sukamto dan Shalahuddin (2013 : 155)



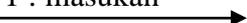
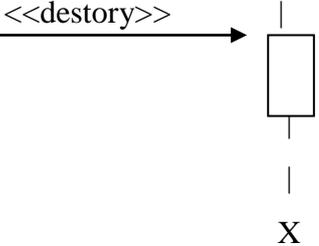
2.2.4.4. Diagram Interaksi dan Sequence (Urutan).

Bersifat dinamis, diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

Tabel 2.5 Simbol-simbol Diagram Sequence

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama Aktor</p> <p>atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem itu sendiri.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p>  <p><u>Nama objek / nama kelas</u></p>	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.</p>



Pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe send 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya.
Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.
Pesan tipe destroy <<destory>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain.

Sumber : Sukamto dan Shalahuddin (2013 : 165)

2.2.4.5. Diagram Komunikasi (Communication Diagram).

Bersifat dinamis, diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

2.2.4.6. Diagram Statechart (Statechart Diagram).

Bersifat dinamis, diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktifitas. Diagram ini

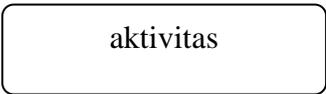
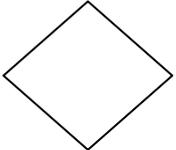
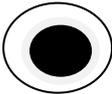


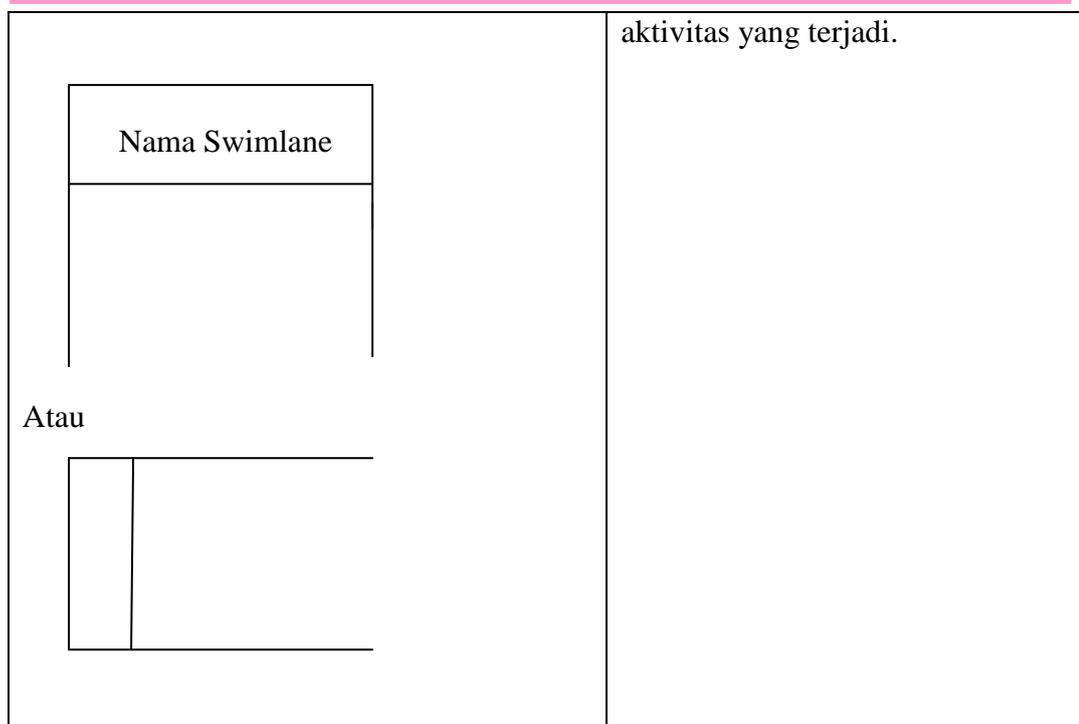
terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (interface), kelas, kolaborasi, dan terutama penting pada pemodelan sistem-sistem yang reaktif.

2.2.4.7. Diagram Aktivitas (Activity Diagram).

Bersifat dinamis, diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek. Berikut akan dijelaskan simbol-simbol dari Diagram aktifitas :

Tabel 2.6 Simbol-simbol Diagram Aktivitas

Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem, sebuah sistem diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/join 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap



Sumber : Sukamto dan Shalahuddin (2013 : 162)

2.2.4.8. Diagram Komponen (Component Diagram).

Bersifat statis, diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.

2.2.4.9. Diagram Deployment (Deployment Diagram).

Bersifat statis, diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram deployment berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (distributed computing).



2.3. Pengertian Judul

2.3.1. Pengertian Sistem

Sukamto dan Shalahuddin (2013:2), sistem adalah kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

2.3.2. Pengertian Informasi

Jogiyanto (1999: 692), informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan.

2.3.3. Pengertian Trayek

Kamus Besar Bahasa Indonesia, trayek adalah jalan yang dilalui atau jarak perjalanan yang ditempuh.

2.3.4. Pengertian Transportasi

Salim (2000), transportasi adalah kegiatan pemindahan barang (muatan) dan penumpang dari suatu tempat ke tempat lain.

2.3.5. Pengertian Angkutan Perkotaan

Sebuah moda transportasi perkotaan yang merujuk kepada kendaraan umum dengan rute yang sudah ditentukan.

2.3.6. Pengertian Sistem Informasi Trayek Transportasi Angkutan Umum Perkotaan di Kota Palembang

Sistem informasi yang mengolah tentang informasi angkutan umum perkotaan yang meliputi informasi tarif, rute, trayek, dan kode kendaraan di Kota Palembang



2.4. Metode Pengembangan Sistem

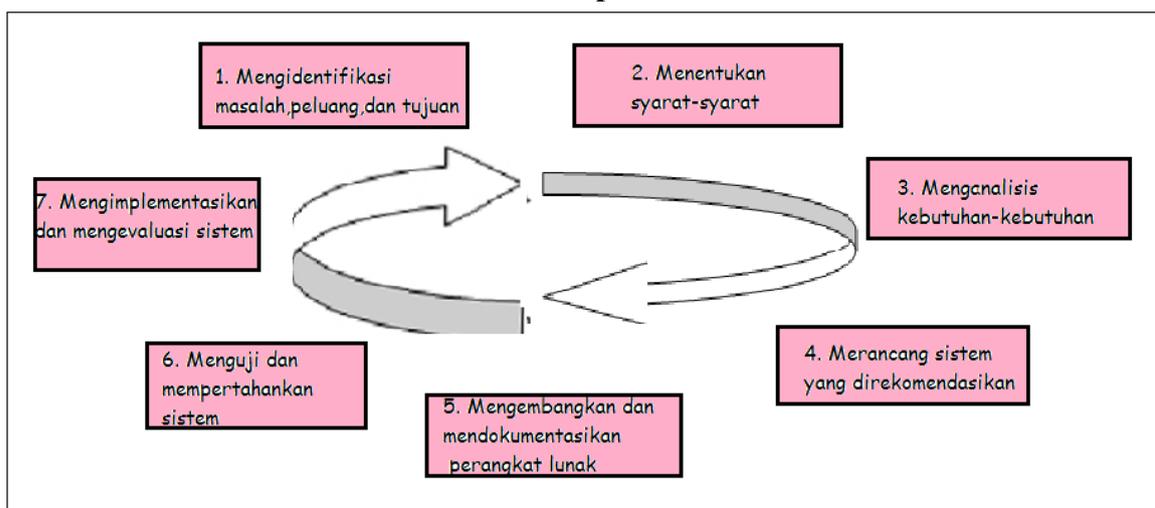
2.4.1. Pengertian SDLC (System Development Life Cycle)

Kendall&Kendall (2006), Metode SDLC adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan sangat baik melalui penggunaan siklus kegiatan penganalisis dan pemakai secara spesifik.

2.4.2. Tahapan – Tahapan SDLC (System Development Life Cycle)

Metode SDLC terdiri dari tujuh tahapan, yang akan digambarkan seperti berikut.

Gambar 2.1 Tahapan Metode SDLC



Sumber : Kendall&Kendall (2006)

Berikut penjelasan tahapan SDLC (System Development Life Cycle) :

1. Mengidentifikasi Masalah, Peluang dan Tujuan.

Tahap ini, dibagi menjadi beberapa bagian yaitu identifikasi masalah, identifikasi peluang, dan tujuan.

- a. Identifikasi masalah, dilakukan dengan melihat kenyataan yang terjadi dalam suatu perusahaan.
- b. Identifikasi peluang, dilakukan untuk mengetahui proses apa saja yang dapat diubah menjadi lebih baik dengan adanya sistem terkomputerisasi.



-
- c. Identifikasi tujuan, dilakukan untuk mengetahui proses apa saja yang dicapai perusahaan.
2. Menentukan Syarat-syarat informasi.

Tahap ini, penganalisis memasukkan apa saja yang menentukan syarat-syarat informasi untuk para pemakai yang terlibat. Diantara perangkat-perangkat yang dipergunakan untuk menetapkan syarat-syarat informasi di dalam bisnis diantaranya ialah menentukan sampel, dan memeriksa data mentah, wawancara mengamati perilaku pembuat keputusan dan lingkungan kantor dan prototyping.
 3. Menganalisis Kebutuhan Sistem.

Tahap ini, perangkat dan teknik-teknik tertentu akan membantu penganalisis menentukan kebutuhan. Perangkat yang dimaksud adalah penggunaan diagram aliran data, untuk menyusun daftar input, proses, dan output fungsi bisnis dalam bentuk grafik terstruktur. Pada poin ini, penganalisis sistem menyiapkan suatu proposal sistem yang berisikan ringkasan apa saja yang ditemukan, analisis biaya atau keuntungan alternatif yang tersedia serta rekomendasi atas apa saja (bila ada) yang harus dilakukan.
 4. Merancang Sistem yang Direkomendasikan.

Tahap ini, penganalisis merancang prosedur data entry sedemikian rupa, sehingga data yang dimasukkan ke dalam sistem informasi benar-benar akurat. Tahap perancangan juga mencakup perancangan file-file atau basis data yang bisa menyimpan data-data yang diperlakukan oleh pembuat keputusan. Dalam tahap ini penganalisis juga bekerja sama dengan pemakai untuk merancang output.
 5. Mengembangkan dan Mendokumentasikan Perangkat Lunak.

Tahap ini, penganalisis bekerja sama dengan pemrogram untuk mengembangkan suatu perangkat lunak awal yang dieprlakukan.
 6. Menguji dan Mempertahankan Sistem.

Tahap ini, sebelum sistem informasi dapat digunakan, maka harus dilakukan pengujian terlebih dahulu.
-



7. Mengimplementasikan dan Mengevaluasi Sistem.

Tahap terakhir ini, penganalisis membantu untuk mengimplementasikan sistem informasi. Tahap ini melibatkan pelatihan bagi pemakai untuk mengendalikan system.

2.5. Teori Program

2.5.1. Pengertian *PHP* (*Hypertext Preprocessor*)

Master.com (2012:5), PHP merupakan singkatan dari *PHP Hypertext Preprocessor* yang digunakan sebagai bahasa *script serverside* dalam pengembangan Web yang disisipkan pada dokumen HTML.

2.5.2. Fungsi *PHP*

Master.com (2012:5) PHP bisa berinteraksi dengan *database*, file, dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website*.

Master.com (2012:41) PHP juga digunakan dalam pembuatan web dinamis (menggunakan database, manajemen web dilakukan dari halaman administrator) yang berbeda dengan web statis (tidak menggunakan database, mengganti isi halaman dengan mengedit kode/bahasa HTML).

2.5.3. Skrip *PHP*

Master.com (2012:7), skrip *PHP* ditulis dalam apitan tanda khusus PHP. Ada 3 macam pasangan tag PHP yang tepat digunakan untuk menandai blok script PHP, yakni :

1. `<?PHP...?>`
2. `<script language="PHP">...</script>`
3. `<?...?>`

Cara 1 dan 2 merupakan cara yang paling umum digunakan sekalipun cara 3 tampak lebih praktis karena cara 3 tidak selalu diaktifkan pada konfigurasi file PHP.



2.5.4. Pengertian *MySql*

Kadir (2013:15), MySQL adalah nama database server, dengan menggunakan MySQL kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat.

2.5.5. Tipe Data Dalam *MySQL*

Secara garis besar, *Mysql* menyediakan beberapa kategori tipe data, yaitu :

a. Tipe Numeris

Kadir (2008:48), tipe numeris adalah tipe yang menyatakan bilangan. Secara garis besar, numeris dikelompokkan menjadi tipe bilangan bulat dan tipe bilangan pecahan.”

Kadir (2008:49), juga menjelaskan beberapa tipe data numeris, yaitu :

Tabel 2.7 Tipe Numeris

No	Tipe Data	Keterangan	Kebutuhan Memori
1.	BIT	Menyatakan tipe bit. Satu bit dapat berupa bilangan 0 atau 1.	8 byte
2.	TINYINT[(M)]	Menyatakan bilangan bulat yang berkisar antara -128 sampai 127.	1 byte
3.	BOOLEAN	Untuk menyatakan nilai benar dan salah.	1 byte
4.	SMALLINT[(M)]	Menyatakan data bilangan bulat yang berkisar antara -32768 sampai 32767.	2 byte
5.	MEDIUMINT[(M)]	Menyatakan data bilangan bulat yang berkisar antara -8388608 sampai 8388607.	3 byte
6.	INT[(M)] atau INTEGER[(M)]	Menyatakan data bilangan bulat yang berkisar antara -2147683648 sampai 2147683647.	4 byte
7.	BIGINT[(M)]	Menyatakan data bilangan bulat yang berkisar antara -9223372036854775808 sampai 9223372036854775807.	6 byte
8.	FLOAT[(M,D)]	Tipe bilangan pecahan yang dapat menampung bilangan antara - 3.4028234466E+38 sampai dengan 1.175494351E-38, 0, dan 1.175494351E-38	4 byte



		sampai 3.4028234466E+38.	
9.	DOUBLE[(M,D)]	Tipe bilangan pecahan yang dapat menampung bilangan antara -1.79769313348623157E+308 sampai dengan 2.2250738585072014E-308, 0, dan 2.2250738585072014E-308 sampai 1.79769313348623157E+308.	8 byte
10.	FLOAT(<i>p</i>)	Tipe bilangan pecahan. Dalam hal ini, <i>p</i> menyatakan jumlah presisi dalam bit.	4 byte jika $\leq p \leq 24$, 8 byte jika $25 \leq p \leq 35$
11.	DEC[(M[,D])] atau DECIMAL[(M[,D])]	Menyatakan bilangan dengan sejumlah digit di belakang titik desimal.	

Sumber : Kadir (2008:49)

b. Tipe Tanggal dan Waktu

Kadir (2008:51), menjelaskan bahwa tipe tanggal dan waktu untuk menangani waktu dan tanggal.

Table 2.8 Tipe Tanggal dan Waktu

No	Tipe Data	Keterangan	Kebutuhan Memori
1.	DATE	Menyatakan tipe untuk data tanggal. Tanggal yang didukung berkisar antara '1000-01-01' sampai dengan '9999-12-31'. Format : 'tahun-bulan-tanggal'.	3 byte
2.	TIME	Menyatakan tipe untuk data waktu yang berformat : 'jam:menit:detik'.	3 byte
3.	DATETIME	Menyatakan tipe data yang menggabungkan tanggal dan waktu. Format : 'tahun-bulan-tanggal jam:menit:detik'.	8 byte
4.	TIMESTAMP	Tipe yang nilainya akan diisi secara otomatis dengan tanggal dan jam saat ada operasi INSERT atau UPDATE.	4 byte



5.	YEAR	Tipe untuk menyatakan tahun.	1 byte
----	------	------------------------------	--------

Sumber: Kadir (2008:51)

c. Tipe String

Kadir (2008:51), string berarti deretan karakter. Sebuah string dapat tersusun atas nol, satu, atau banyak karakter. String yang tidak mengandung satu karakter pun dinamakan string kosong.

Table 2.9 Tipe String

No	Tipe Data	Keterangan	Kebutuhan Memori
1.	CHAR(M)	String dengan panjang tetap yaitu M karakter. Nilai terbesar untuk M adalah 255.	M byte, $0 \leq M \leq 255$
2.	CHAR	Sinonim dengan CHAR(1)	1 byte
3.	VARCHAR(M)	String dengan panjang variabel. M menyatakan jumlah maksimum karakter yang bisa disimpan. Nilai terbesar untuk M yaitu 65535.	L+1 byte, $L \leq M$ dan $0 \leq M \leq 65535$
4.	BINARY(M)	Serupa dengan VARCHAR(M), tetapi disimpan dalam bentuk biner.	M byte, $0 \leq M \leq 255$
5.	VARBINARY(M)	Serupa dengan VARCHAR(M), tetapi disimpan dalam bentuk biner.	L+1 byte, $L \leq M$ byte, $0 \leq M \leq 255$
6.	TINYBLOB	BLOB dengan ukuran paling kecil	L+1 byte, $L < 2^8$
7.	BLOB[(M)]	BLOB dengan ukuran lebih besar dari pada TINYBLOB	L+2 byte, $L < 2^{16}$
8.	MEDIUMBLOB	BLOB berukuran medium	L+3 byte, $L < 2^{24}$
9.	LOB	BLOB berukuran paling besar	L+4 byte, $L < 2^{32}$
10.	TINYTEXT	String teks dengan ukuran paling kecil	L+1 byte, $L < 2^8$
11.	TEXT[(M)]	String teks dengan ukuran lebih besar dari pada TINYTEXT	L+2 byte, $L < 2^{16}$



12	MEDIUMTEXT	String teks berukuran medium	L+3 byte, L<2 ²⁴
13	LONGTEXT	String teks berukuran paling besar	L+4 byte, L<2 ³²
14	ENUM('nilai1', 'nilai2', ...)	Tipe data yang nilai kolom berupa salah satu string yang disebutkan pada ENUM. Jumlah elemen ENUM dapat mencapai 65535.	1 atau 2 byte, bergantung pada jumlah nilai enumerisasi
15	SET('nilai1', 'nilai2', ...)	Merupakan suatu objek string yang berisi nol, satu, atau beberapa nilai. Sebuah kolom bertipe SET maksimum berisi 64 elemen yang berbeda. Nilai antar-elemen ditulis dengan pemisah koma.	1,2,3,4, atau 8 byte, bergantung pada jumlah anggota himpunan

Sumber : Kadir (2008:51)

2.5.6. Pengertian *Dreamweaver*

Elcom (2013:1), *dreamweaver* adalah sebuah editor profesional yang menggunakan *HTML* untuk mendesain *web* secara visual dan mengelola situs atau halaman *web*. *Dreamweaver* juga merupakan *software* utama yang digunakan oleh *desainer web* dan *programmer web* untuk mengembangkan suatu situs *web*.

Master.com (2012:37), *dreamweaver* merupakan sebuah *HTML authoring (tool* pembuat halaman *web* berbasis *HTML* untuk membuat / mengedit berbagai kode pemrograman *web* secara cepat) dan dapat digunakan dalam manajemen sebuah *website* (dari sisi *developing*).

2.5.7. Komponen-komponen yang terdapat di dalam ruang kerja *Dreamweaver*

Ruang kerja *Dreamweaver* memiliki komponen-komponen yang memberikan fasilitas untuk menuangkan kreasi saat bekerja, berikut komponen-komponen dalam ruang kerja *dreamweaver* :

1. *Insert Bar*

Berisi tombol-tombol untuk menyisipkan berbagai macam objek seperti, *image*, *table* dan *layer* ke dalam dokumen.



2. *Document Toolbar*

Berisi tombol-tombol dan menu *pop-up* yang menyediakan tampilan berbeda dari jendela dokumen.

3. *Coding Window*

Berisi kode-kode *HTML* dan tempat untuk menuliskan kode-kode pemrograman.

4. *Panel Group*

Berisi kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan di bawah satu judul.

5. *Property Inspector*

Digunakan untuk melihat dan mengubah berbagai *property* objek atau teks.

6. Jendela Dokumen

Digunakan untuk menampilkan dokumen saat bekerja.

7. *Ruler*

Mempermudah ukuran dalam mendesain halaman *web*.

8. *Site Panel*

Digunakan untuk mengatur file-file dan folder-folder yang membentuk situs *web*.

2.5.8. Pengertian XAMPP

Nugroho (2013:1), XAMPP adalah paket program web lengkap yang dapat dipakai untuk belajar pemrograman web, khususnya PHP dan MySQL.

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

2.5.9. Pengertian CSS

Elcom (2013:145), CSS singkatan dari *Cascading Style Sheet* yang berfungsi untuk mengatur keseluruhan tampilan halaman pada web. CSS dapat mengatur posisi *layout*, jenis dan tipe huruf dengan mudah dan *fleksibel*, bahkan CC juga dapat digunakan untuk mengatur *property* yang tidak dapat diatur jika hanya menggunakan *HTML*.
