

BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

Adapun teori umum yang digunakan dalam penelitian ini yang di lakukan pada PT Panji Jaya adalah sebagai berikut ini:

2.1.1 Pengertian Sistem

Kata “Sistem” banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan diantara mereka (Husda, 2012:111).

Sedangkan menurut Prahasta (2014: 78), Sistem dapat didefinisikan sebagai sekumpulan objek, ide, berikut keterkaitannya didalam mencapai tujuan. Dengan kata lain sistem adalah sekumpulan komponen (sub-sistem fisik dan non fisik atau logika) yang saling berhubungan satu sama lainnya dan bekerja sama untuk mencapai suatu tujuan.

Dan dari pengertian kedua pakar di atas, peneliti dapat menyimpulkan sistem adalah subsistem yang saling berhubungan, berkumpul bersama-sama yang

membentuk satu kesatuan, saling berinteraksi, sehingga dapat menyelesaikan sasaran tertentu.

2.1.2 Pengertian Informasi

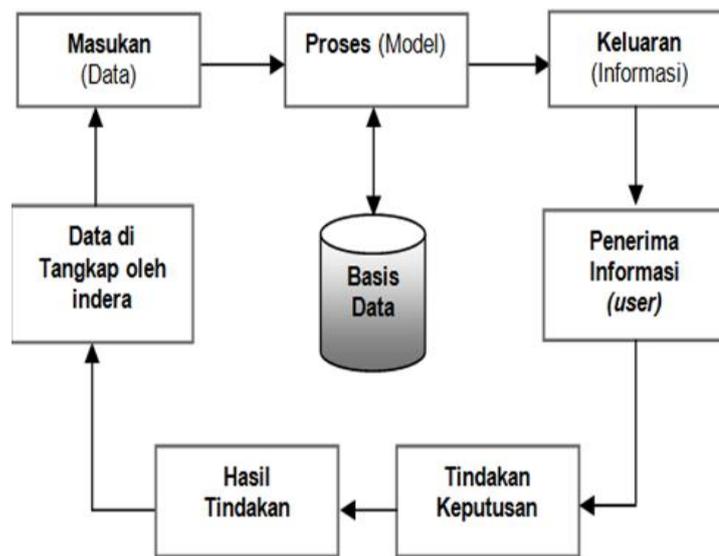
Informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata (Husda, 2012:117).

Sedangkan menurut IBISA (2011: 4), Informasi adalah merupakan aset perusahaan yang harus dilindungi dari ancaman penyalahgunaan. Informasi dalam bentuk *hardcopy* yang dihasilkan dengan jerih payah perusahaan merupakan investasi yang memakan biaya banyak demi menunjang dan memajukan perusahaan.

Peneliti memberikan kesimpulan informasi berdasarkan referensi dari kedua sumber di atas yaitu, hasil pengolahan data dalam bentuk *softcopy* atau *hardcopy* yang merupakan nilai terpenting atau aset pada sebuah perusahaan berguna untuk memajukan perusahaan dalam dunia persaingan bisnis karena bisa menyediakan pengolahan informasi yang dibutuhkan oleh perusahaan maupun oleh pelanggan.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan

tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali.



Gambar 2. 1 Siklus Informasi (Husda, 2012:118)

Adapun fungsi–fungsi informasi adalah sebagai berikut :

- a. Untuk meningkatkan pengetahuan bagi si pemakai.
- b. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
- c. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

Informasi yang berkualitas harus akurat, tepat pada waktunya dan relevan.

2.1.3 Sistem Informasi

Sistem informasi adalah sebuah sekumpulan komponen-komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi terkait untuk mendukung proses pengambilan keputusan, koordinasi dan pengendalian (Prahasta, 2014:78).

Sedangkan Menurut Sutanta (2011: 16), Dalam arti yang luas sistem informasi dapat dipahami sebagai sekumpulan subsistem yang saling berhubungan, berkumpul bersama-sama dan membentuk satu kesatuan, saling berinteraksi dan bekerja sama antara bagian satu dengan yang lainnya dengan cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (*input*) berupa data-data, kemudian mengolahnya (*processing*) dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar bagi pengambilan keputusan yang berguna dan mempunyai nilai nyata yang dapat dirasakan akibatnya baik pada saat itu juga maupun dimasa mendatang, mendukung kegiatan operasional, manajerial, dan strategis organisasi dengan memanfaatkan berbagai sumber daya yang ada dan tersedia bagi fungsi tersebut guna mencapai tujuan (Sutanta, 2003).

Sehingga peneliti dapat menyimpulkan bahwa sistem informasi adalah Sekumpulan komponen-komponen yang saling berhubungan dan bekerja sama satu dengan yang lainnya dengan cara tertentu untuk melakukan fungsi pengolahan data untuk mencapai tujuan yang mempunyai nilai.

2.1.3.1 Komponen Sistem Informasi

Menurut Husda (2012: 120), Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*), yaitu:

1. Komponen input atau komponen masukan
2. Komponen model
3. Komponen output atau komponen keluaran
4. Komponen teknologi
5. Komponen basis data
6. Komponen kontrol atau komponen pengendalian

Keenam komponen ini harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya, yaitu pengolahan informasi yang relevan, tepat waktu dan akurat. Komponen-komponen dari sistem informasi ini dapat digambarkan sebagai berikut ini :

1. Blok Masukan (*Input Block*)

Input merupakan data yang masuk ke dalam sistem informasi

2. Blok Model (*Model Block*)

Kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan yang tersimpan dibasis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan kotak alat dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem serta menyeluruh.

5. Blok Basis Data (*Database Block*)

Merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*)

Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.

2.1.4 Inventory (Persediaan)

Menurut Chandra (2015: 125), Inventory merupakan stok bahan baku, pasokan, komponen, barang yang masih dalam pengerjaan dan barang jadi yang

muncul sepanjang proses produksi dan logistik dalam suatu perusahaan. Adapun lima alasan untuk memiliki persediaan:

1. Spesialisasi untuk memaksimalkan keuntungan, setiap pabrik dapat memproduksi satu varian produk ketimbang memproduksi berbagai macam variasi produk.
2. Menjaga keseimbangan permintaan dan penawaran. Permintaan barang bervariasi pada waktu tertentu sedangkan kapasitas produksi adalah tetap.
3. Ketidakpastiaan. Inventory dimiliki sebagai *buffer* / cadangan untuk mengantisipasi ketidakpastian jumlah permintaan
4. *Economies of scale*. Pembelian, perpindahan, penyimpanan dalam jumlah banyak dapat mengurangi ongkos produksi sehingga meningkatkan keuntungan.
5. *Buffer interface*. Inventory sebagai unit persediaan untuk melindungi kemungkinan *stock out*.

2.1.4.1 Pengendalian Inventory (Persediaan)

Pengendalian persediaan merupakan penentuan kebijakan kapan memesan suatu produk dan dalam jumlah berapa untuk dapat memenuhi permintaan sehingga meminimalkan biaya persediaan. Masalah persediaan merupakan hal yang penting bagi perusahaan karena jika jumlah persediaan lebih banyak ketimbang permintaan, maka beban penyimpanan dan pemeliharaan akan semakin

tinggi. Sebaliknya jika persediaan kurang, maka perusahaan tidak dapat memenuhi permintaan pelanggan.

Beberapa tujuan pengendalian persediaan adalah:

1. Pemasaran ingin melayani konsumen secepat mungkin sehingga menuntut persediaan dalam jumlah besar
2. Efisiensi produksi sehingga memproduksi dalam jumlah besar untuk mengurangi biaya. Hal ini menyebabkan persediaan yang besar dan menuntut komponen atau bahan baku yang cukup agar produksi tidak terganggu.

Personalia (*personel and industrial relationship*) dimana persediaan dibutuhkan untuk mengantisipasi fluktuasi kebutuhan tenaga kerja sehingga PHK tidak perlu dilakukan (Chandra, 2015: 125).

2.1.5 SDLC (*Software Development Life Cycle*)

Menurut Shalahuddin (2011: 24), *SDLC* atau *Software Development Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model–model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara–cara yang sudah teruji dengan baik).

Sedangkan menurut Chandra (2015: 126), *SDLC* (*Software Development Cycle*) merupakan metodologi dalam pembuatan sistem informasi yang terdiri dari beberapa tahapan. *SDLC* disusun sesuai dengan kebutuhan manajemen atau

organisasi. *SDLC* memiliki beberapa model seperti: *fountain*, *spiral*, *RAD*, *Waterfall*, *Incremental* dan *prototyping*.

Menurut Shalahuddin (2011: 24), Tahapan-tahapan yang ada pada *SDLC* secara global adalah sebagai berikut:

1. Inisiasi (*Initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak

2. Pengembangan Konsep sistem (*System Concept Development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*Planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resource*) yang dibutuhkan untuk memperoleh solusi.

4. Pengembangan (*Development*)

Mengonversi disain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakuakn instalasi lingkungan sistem yang dibutuhkan; membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program peninjauan pengujian.

5. Integrasasi dan pengujian (*Integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas dan *user*.

6. Implementasi (*Implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

7. Operasi dan pemeliharaan (*operational and Maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

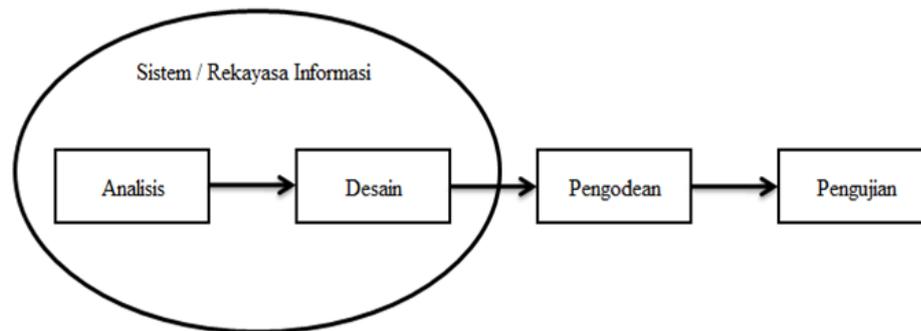
8. Disposisi (*Disposisi*)

Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

2.1.6 Waterfall

Menurut Shalahuddin (2011: 27), model *SDLC* air terjun (*waterfall*) sering juga disebut model sekuensial linier atau alur hidup klasik (*classic life sicle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara

sekuensial atau terurut dimulai dari analisis, disain, pengodean, pengujian dan tahap pendukung (*support*).



Gambar 2. 2 Ilustrasi Model Waterfall (Shalahuddin, 2011: 27)

Keterangan Gambar:

1. Analisi kebutuhan perangkat lunak
 Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*
2. Disain perangkat lunak adalah proses multilangkah yang fokus pada disain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean
3. Pembuatan kode program
 Disain harus di translasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan disain yang telah dibuat pada tahap disain.

4. Pengujian

Fokus pada perangkat lunak dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji.

5. Pendukung (*support*) atau pemeliharaan (*Maintenance*)

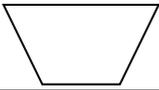
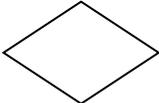
Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.7 Flowchart

Menurut Jogiyanto (2005: 795) dalam Rismawati (2014: 322), menyatakan bagian aliran (flowchart) adalah bagan (chart) yang menunjukkan alir (flow) didalam program atau prosedur system secara logika. Adapun simbol-simbol flowchart adalah sebagai berikut:

Tabel 2. 1 simbol–simbol yang ada pada Flowchart Rismawati (2014: 322)

Simbol	Deskripsi
Proses 	Simbol proses ini digunakan untuk mewakili data <i>input / output</i>
<i>Input / Output</i> 	Simbol ini digunakan untuk mewakili data <i>input / output</i>
Simbol Dokumen 	Menunjukkan dokumen <i>input / output</i> baik untuk proses manual, mekanik atau computer.

Simbol	Deskripsi
Kegiatan Manual 	Menunjukkan pekerjaan Manual
Terminal 	Simbol ini digunakan untuk menunjukkan awal dan akhir dari suatu proses
Garis Alir 	Untuk Menunjukkan alur dari proses
<i>Decision Symbol</i> 	Digunakan Untuk suatu penyeleksian kondisi didalam program
Penyimpanan <i>Harddisk</i> 	<i>Menunjukkan input / output menggunakan harddisk</i>

2.1.8 UML

UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan disain, serta menggambarkan arsitektur dalam pemograman berorientasi objek (Shalahuddin, 2011: 113). Pemodelan perangkat lunak memiliki beberapa abstraksi, misalnya sebagai berikut:

1. Petunjuk yang terfokus pada proses yang dimiliki oleh sistem
2. Spesifikasi struktur secara abstrak dari sebuah sistem (belum detail)
3. Spesifikasi lengkap dari sebuah sistem yang sudah final
4. Spesifikasi umum atau khusus sistem

5. Bagian penuh atau parsial dari sebuah sistem

Perangkat pemodelan adalah suatu model yang digunakan untuk menguraikan sistem menjadi bagian-bagian yang dapat diatur dan mengkomunikasikan ciri konseptual dan fungsional kepada pengamat. (Shalahuddin, 2011: 116).

2.1.8.1 *Class Diagrams*

Menurut Shalahuddin (2011: 122), *Class diagrams* atau klas diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- a) Atribut merupakan variable-variable yang dimiliki oleh suatu kelas
- b) Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas Main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai

3. Kelas yang diambil dari pendefinisian *use case*

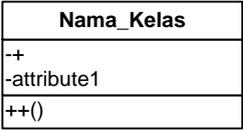
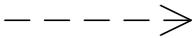
Kelas yang menangani fungsi–fungsi yang harus ada diambil dari pendefinisian *use case*

4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol–simbol yang ada pada diagram kelas :

Tabel 2. 2 simbol–simbol yang ada pada diagram kelas (Shalahuddin, 2011: 123)

Simbol	Deskripsi
 <p>Kelas</p>	Kelas pada terstruktur sistem
<p>Antarmuka (<i>interface</i>)</p>  <p><i>Nama_interface</i></p>	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek
<p>Asosiasi (Association)</p> 	Relasi antar kelas dengan aksna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antara kelas dengan makna generalisasi-spesialisasi (umum khusus)
Simbol	Deskripsi
<p>Kebergantungan (<i>dependency</i>)</p> 	Relasi antar kelas dengan makna kebergantungan antar kelas
<p>Agregasi (Aggregation)</p> 	Relasi antar kelas dengan makna semua bagian (<i>Whole-part</i>)

2.1.8.2 *Use Case Diagrams*

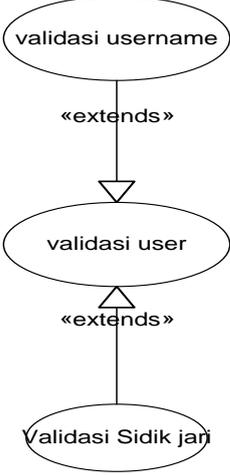
Use case diagrams merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi–fungsi itu (Shalahuddin, 2011: 130).

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

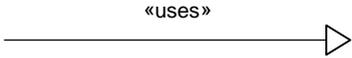
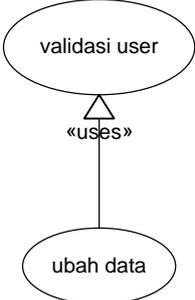
1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol- simbol yang ada pada diagram *use case*:

Tabel 2. 3 Tabel simbol–simbol yang ada pada diagram *use case* (Shalahuddin, 2011: 130)

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Actor (<i>actor</i>)</p> 	<p>Orang, proses atau sistem lainnya yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Asosiasi (<i>association</i>)</p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau memiliki interaksi dengan aktor</p>
<p>Ekstensi (<i>extend</i>)</p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek; biasanya <i>use cse</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya :</p>
	
<p>Generalisasi (<i>Generalixation</i>)</p> 	<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>

Tabel 2.3 Lanjutan (Shalahuddin, 2011: 130)

Simbol	Deskripsi
<p><i>Menggunakan uses</i></p> 	<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 sebagai syarat dijalankan <i>use case</i> ini.</p>
	

2.1.8.3 Activity Diagrams

Diagram aktivitas atau *activity diagrams* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Shalahuddin, 2011:134).

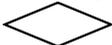
Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal- hal sebagai berikut:

1. Rancangan proses bisnis dimana setiap aktiviatas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah simbol–simbol yang ada pada diagram aktiviatas:

Tabel 2. 4 Tabel simbol–simbol yang ada pada diagram aktifitas (Shalahuddin, 2011:134)

Simbol	Deskripsi
Status Awal 	Status awal aktivitas sistem. Sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan seistem, aktivitas biasanya diawali dengan kata kerja
Percabangan (<i>Decision</i>) 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan (<i>Join</i>) 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir dilakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir

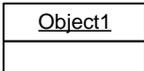
2.1.8.4 *Object Diagrams*

Menurut Shalahuddin (2011: 124), diagram objek menggambarkan struktur sistem dari segi dan jalannya objek dalam sistem. Pada diagram objek harus

dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan. Untuk apa mendefinisikan sebuah kelas sedangkan pada jalannya sistem, objeknya tidak pernah dipakai. Hubungan *link* pada diagram objek merupakan hubungan memakai dan dipakai dimana dua buah objek akan dihubungkan oleh *link* jika ada objek yang dipakai oleh objek lainnya.

Berikut adalah simbol-simbol yang ada pada diagram objek:

Tabel 2. 5 Tabel simbol–simbol yang ada pada diagram objek (Shalahuddin, 2011: 130)

Simbol	Deskripsi
Objek 	Objek dari kelas yang berjalan saat sistem dijalankan
Link 	Relasi antar Objek

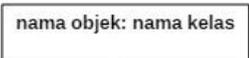
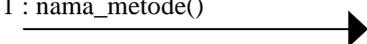
2.1.8.5 Diagram Sekuensial

Menurut Shalahuddin (2011: 137), Diagram Sekuensial menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat

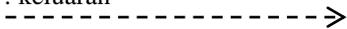
dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu.

Berikut ini adalah simbol-simbol yang ada pada diagram sekuen:

Tabel 2.6 Tabel simbol–simbol yang ada pada diagram sekuensial (Shalahuddin, 2011: 138)

Simbol	Deskripsi
<p>Aktor/<i>actor</i></p>  <p>nama aktor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
<p>Garis hidup/<i>lifeline</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
<p>Objek</p>  <p>nama objek: nama kelas</p>	Menyatakan objek yang berinteraksi pesan
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
<p>Pesan tipe <i>create</i></p> <p><<create>></p> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat
<p>pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.

Tabel 2.6 Lanjutan (Shalahuddin, 2011: 138)

Simbol	Deskripsi
Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju
pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek penerima
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri

2.1.9 Sekilas Mengenai Web

Web merupakan media informasi berbasis jaringan computer yang dapat diakses di mana saja dengan biaya relative murah. *Web* merupakan bentuk implementasi dari bahasa pemograman *web*. Sejarah perkembangan bahasa pemograman *web* diawali dengan munculnya *HTML (Hypertext Markup Language)*, yang kemudian dikembangkan dengan munculnya *CSS (Cascading Style Sheet)* yang bertujuan untuk memperindah tampilan *website* (Hermawan, 2009:2).

World Wide Web (WWW), lebih dikenal dengan *web*, merupakan salah satu layanan yang didapat oleh pemakai computer yang terhubung ke internet. *Web* pada awalnya adalah ruang informasi dalam internet, dengan menggunakan

teknologi *hypertext*, pemakai dituntun untuk menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen *web* yang ditampilkan dalam *browser web* (Sidik dan Pohan, 2009: 1).

Web memudahkan pengguna komputer untuk berinteraksi dengan pelaku internet lainnya dan menelusuri (informasi) di internet.

Menurut Sidik dan Pohan (2009: 2), Selain itu *web* telah diadopsi oleh perusahaan sebagai sebagian dari strategi teknologi informasinya, karena beberapa alasan:

1. akses informasi mudah.
2. *Setup server* lebih mudah.
3. Informasi mudah distribusikan, dan
4. Bebas platform; informasi dapat disajikan oleh *browser web* pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan.

2.2 Teori Khusus

Penelitian ini yang dilakukan pada PT Panji Jaya mengemukakan teori khusus yang peneliti pakai dan dibahas adalah seperti di bawah ini:

2.2.1 HTML5 (*Hypertext Markup Language*)

Menurut Saputra (2012: 11), Hal yang paling mencolok yang peneliti lihat adalah tersedianya fitur baru seperti elemen multimedia, misalnya <audio> dan

<video>, yang tak lain adalah fungsi untuk memutar audio dan juga video. Pada versi HTML sebelumnya, jika Anda ingin memutar perangkat multimedia, haruslah menggunakan perintah <embed>. Dengan adanya penambahan fitur dan tag khusus yang ada di dalam HTML 5, akan membuat semuanya menjadi sangat mudah.

Menurut Hidayat (2009: 6), *HTML (HyperText Markup Language)* merupakan sebuah bahasa *markup* (tanda) yang digunakan dalam membuat sebuah halaman *web*, serta menampilkan berbagai informasi di dalam sebuah *web browser HTML* adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. *HTML* saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (W3C)*.

Prinsip kerja pengaksesan dokumen *web* yang berbasis *HTML* adalah sebagai berikut:

- a. Browser meminta sebuah halaman ke suatu situs *web* melalui protocol *HTTP*
- b. Permintaan diterima oleh *Web Server*
- c. *Web* segera mengitinkan dokumen *HTML* yang diminta ke klien
- d. *Browser* pada klien segera menampilkan dokumen yang diterima berdasarkan kode –kode performat yang terdapat pada dokumen *HTML*.

Selain itu kita harus mengenal sebuah *database*. *Database* merupakan kumpulan informasi yang disimpan di dalam komputer secara sistematis,

sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

HTML (Hypertext Markup Language) merupakan suatu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman *web*. Dokumen ini dikenal sebagai *web page*. Dokumen *HTML* merupakan dokumen yang disajikan pada *web browser*.

Fungsi dari *HTML* adalah digunakan untuk membuat sebuah halaman *website* dan juga menampilkan berbagai informasi yang di inputkan ke dalam html tersebut seperti gambar, teks, video, dan lainnya sehingga bisa di tempilkan melalui *web browser*. Selain itu *HTML* juga bisa digunakan untuk membuat sebuah link yang nantinya menuju ke sebuah halaman *website* lainnya dengan kode tertentu.

2.2.2 PHP (Hypertext Preprocessor)

Adapun *PHP* dalah singkatan dari *personal home page*, merupakan bahasa skrip yang sangat populer dan banyak dipakai oelh programmer *web* dalam pembuatan sistus yang dinamis. Dalam aplikasi CMS tersebut, hampir seluruhnya terbuat dari bahasa pemograman skrip *PHP*. *PHP* adalah bahasa *open source* yang dapat digunakan diberbagai *runtime* melalui *console* serta dapat menjalankan perintah – perintah sistem (Hidayat, 2009: 9).

PHP adalah bahasa *server side scripting* yang menyatu dengan *HTML* untuk membuat halaman *web* yang dinamis. Karena *PHP* merupakan *server-side*

scripting maka *sintaks* dan perintah-perintah *PHP* akan dieksekusi ke *server* yang kemudian hasilnya dikirimkan ke *browser*. *PHP* adalah skrip yang dijalankan di *server*. Jadi konsepnya berbeda dengan *JavaScript*, yang dijalankan di klien. Keuntungan penggunaan *PHP*, kode yang menyusun program tidak perlu diedarkan ke pemakai sehingga kerahasiaan kode dapat dilindungi. Hal menarik yang didukung oleh *PHP* tetapi tidak didukung oleh *JavaScript* adalah kenyataan bahwa *PHP* bisa digunakan untuk mengakses berbagai macam *database* seperti *Access*, *Oracle*, *MySQL* dll (Hidayat, 2009: 9).

2.2.3 CSS (*Cascading Style Sheet*)

Menurut Sidik dan Pohan (2009: 132), *CSS* merupakan fitur baru dari *HTML 4.0*. Hal ini diperlukan setelah melihat perkembangan *HTML* menjadi kurang praktis karena *web pages* terlalu banyak dibebani hal-hal yang berkaitan dengan faktor tampilan seperti *font*, dan lain-lain. Bentuk penggunaan *CSS* dapat dimodelkan sebagai berikut :

Terdapat tiga cara pendefinisian dalam menggunakan *CSS*, yaitu (Sidik dan Pohan, 2009:134):

1. *Style sheet external*

Pada teknik ini, *style sheet* didefinisikan di luar dokumen *HTML* dan disimpan dalam *file* berekstensi *css* (*.css). Dalam pendefinisian *external* tidak perlu lagi menggunakan *taghtml* diawal dan akhir dokumen.

2. *Style sheet internal*

Style sheet didefinisikan secara *internal* biasanya karena *web page* tertentu bersifat sangat unik sehingga membutuhkan definisi terpisah dibandingkan dengan *web page* lainnya.

3. *Inline style sheet*

Style sheet inline hanya bisa digunakan pada lokasi yang sangat spesifik dimana *style sheet* ditempatkan. Kekurangan dari teknik ini adalah dokumen menjadi lebih besar karena *style* didefinisikan satu per satu.

2.2.4 *JavaScript*

Javascript merupakan salah satu bahasa *script* yang dijalankan pada *client* yaitu *browser*. *JavaScript* memungkinkan anda untuk membuat aplikasi *web* lebih menarik dan juga menambah kemampuan lebih pada *website*. *JavaScript* dapat memanipulasi *DOM* pada sisi *client* sehingga banyak *developer* yang menggunakan *JavaScript* menjadi bagian dari *website* mereka. Fungsi-fungsi *JavaScript* siap pakai yang memudahkan mereka untuk memanipulasi *DOM* (*Document Object Model*). Kumpulan dari fungsi-fungsi *JavaScript* tersebut disebut dengan *framework*.

Framework JavaScript yang ada sekarang ini. Salah satu diantaranya adalah *JQuery*. *JQuery* merupakan *framework* yang paling banyak digunakan dan sangat populer dalam dunia *web* karena lebih sederhana serta mudah digunakan dan dipelajari. Selain itu terdapat pula ribuan *plugin* yang telah dikembangkan dari

JQuery sehingga menambah kemudahan *developer* untuk membangun *website*. (Westriningsih, 2012: iii).

2.2.5 *JQuery*

Menurut Westriningsih (2012: 2), *JQuery* merupakan salah satu dari sekian banyak *framework* yang digunakan dan memiliki pengguna yang paling banyak. *JQuery* merupakan pustaka *JavaScript* yang dibangun untuk mempercepat dan memperingkas serta menyederhanakan manipulasi dokumen *HTML*, penanganan *event*, animasi dan interaksi *Ajax* untuk mempercepat pengembangan *web*. Dengan *JQuery* *developer* akan dimanjakan dengan suatu program *JavaScript* yang sangat sederhana jika dibandingkan dengan *native JavaScript*.

Adapun kemampuan yang dimiliki library *JQuery* antara lain:

1. Mampu mengakses bagian halaman tertentu dengan mudah.
2. Mampu mengubah tampilan bagian halaman tertentu.
3. Mampu mengubah isi dari halaman.
4. Mampu merespon interaksi *user* dalam halaman.
5. Mampu menambahkan animasi ke halaman.
6. Mampu mengambil informasi dari *server* tanpa *me-fresh* seluruh halaman.

7. Mampu menyederhanakan kode *JavaScript* lainnya.

Untuk dapat memulai mempelajari *JQuery*, harus mendownload *JQuery.js* dari situs <http://www.JQuery.com>.



Gambar 2. 3 Logo *JQuery* (Westriningsih, 2012: 4)

2.2.6 *Notepad++*

Menurut Dadan (2015: 33), *software* editor untuk keperluan coding, yaitu *notepad++*. Boleh dibidang ini adalah *tools* utama kita untuk melakukan aktivitas *coding*. *Notepad++* adalah *software* yang memiliki banyak fitur, namun tetap ringan.



Gambar 2. 4 Gambar *Notepad++* (Westriningsih, 2012: 33)

2.2.7 XAMPP (XApache MySQL PHP Perl)

Menurut Dadan (2015: 27), Semua komponen-komponen pendukung *web programming*. XAMPP adalah salah satu aplikasi *web server apache* yang terintegrasi dengan *MySQL* dan *phpMyAdmin*. XAMPP adalah singkatan dari X, *Apache server*, *MySQL*, *phpMyAdmin* dan *python*. Huruf X didepan menandakan XAMPP bisa diinstal diberbagai operating sistem. XAMPP diinstal pada *windows*, *Linux*, *MacOS* dan *Solaris*.

XAMPP adalah buatan dari *team apache friends*. XAMPP dapat didownload langsung dari situs resmi <http://www.apache-friends.org>. Panduan ini khusus membahas cara instal XAMPP versi Windows. Beberapa fitur yang terdapat dalam XAMPP 18.3 diantaranya:

1. *Apache 2.4.7*
2. *MYSQL 5.6.14*
3. *PHP 5.5.6*
4. *phpMyadmin 4.0.9*
5. *FileZilla FTP Server 0.9.41*
6. *Tomcat 7.0.42 (with mod proxy_ajp as connector)*
7. *Stawberry Perl 5.16.1 Portable*
8. *XAMPP Control Panel 3.2.1 (From hackattack142)*

2.2.8 MySQL

Menurut Arief (2011: 152), *MySQL* adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya. *MySQL* merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk *internet* (*PHP* dan *Perl*). *MySQL* dan *PHP* dianggap sebagai pasangan *software* pembangun aplikasi *web* yang ideal. *MySQL* lebih sering digunakan untuk membangun aplikasi berbasis *web*, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script PHP*.

Di dalam Arief (2011: 152), menurut Alexander F.K Sibero (2011: 97) *MySQL* adalah suatu *RDBMS* (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. *MySQL* pertama dikembangkan oleh *MySQL AB* yang kemudian diakui oleh Sun Microsystem dan terakhir dikelola oleh *Oracle Corporation*.

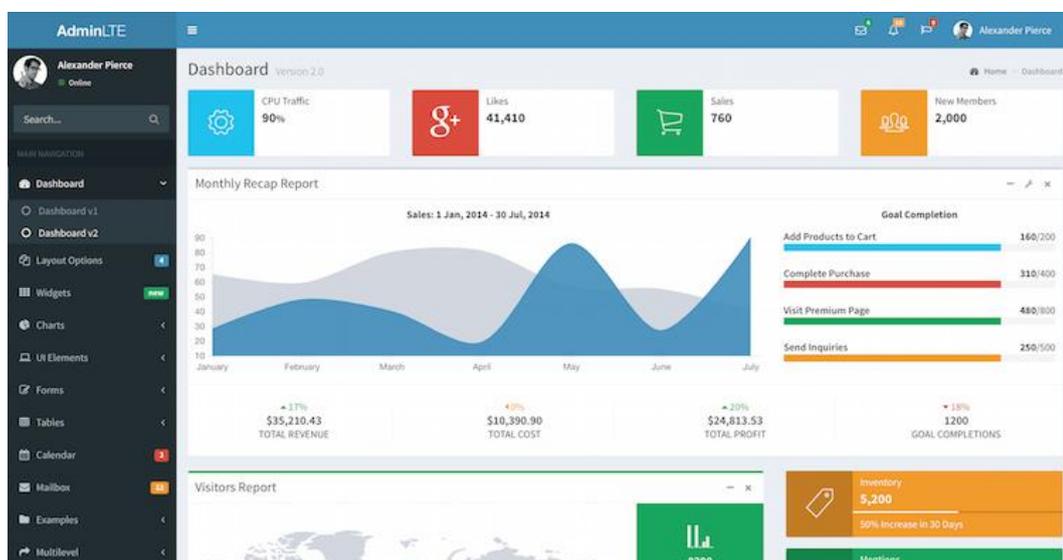
MySQL merupakan salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman *PHP*. *MySQL* bekerja menggunakan bahasa *SQL* (*Structure Query Language*) yang merupakan bahasa standar yang digunakan untuk manipulasi *database*. Pada umumnya perintah yang paling sering digunakan dalam *MySQL* adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu *SQL* juga menyediakan perintah untuk membuat *database*, *field* ataupun *index* untuk

menambah atau menghapus data. Ada beberapa alasan yang menjadikan database *MySQL* sangat diminati oleh para programmer, diantaranya :

1. Bersifat open source.
2. Menggunakan bahasa *SQL*, yang merupakan bahasa standar dalam pengolahan data.
3. *Performance* dan *reliable*, pemrosesan databasenya sangat cepat dan stabil.
4. Sangat mudah dipelajari.
5. Memiliki dukungan (*group*) pengguna *MySQL*.
6. Lintas *Platform*, dapat digunakan pada berbagai sistem operasi berbeda.
7. *Multiuser*, dimana *MySQL* dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.
8. *MySQL* merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran gigabyte sekalipun.
9. *MySQL* adalah *database* yang menggunakan enkripsi *password*.
10. *MySQL* dapat menciptakan lebih dari 16 kunci per tabel dan satu kunci memungkinkan belasan fields.
11. *MySQL* merupakan *server* tercepat.
12. *MySQL* dapat diakses melalui *protocol ODBC (Open Database Connectivity)* buatan *Microsoft*. Ini menyebabkan *MySQL* dapat diakses oleh banyak *software*.

2.2.9 AdminLTE 2.1

AdminLTE terbesar *open source webapp* Template populer untuk *dashboards* admin dan panel kontrol. itu adalah template *HTML* responsif yang didasarkan pada *CSS framework Bootstrap3*. Dimanfaatkan semua komponen dengan metode *bootstrap* adalah adalah disain dan re-gaya banyak plugin yang biasa digunakan untuk membuat disain yang konsisten yang dapat digunakan sebagai *user interface* untuk aplikasi *backend*. *AdminLTE* didasarkan pada disain modular, yang memungkinkan untuk menjadi mudah disesuaikan dan bukti pada dokumentasi ini akan memandu Anda melalui menginstal *template* dan menjelajahi berbagai komponen yang bundel dengan *template* (<https://github.com/almsaeed2010/AdminLTE>, 2015).



Gambar 2. 5 *AdminLTE2.1* (<https://almsaeedstudio.com/>, 5 feb 2017)

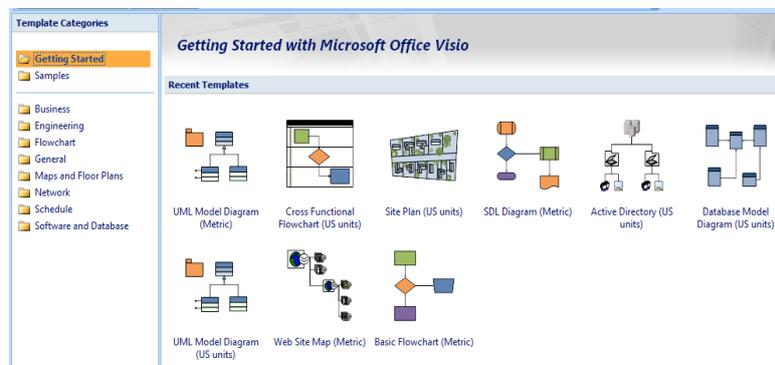
2.2.10 Ms. Visio 2007

Menurut Sutomo (2009: 1), *Ms. Visio 2007* mengalami beberapa perbaikan dari pendahuluannya, *Ms. Visio 2003*. Untuk tampilan menubar tidak banyak berubah seperti produk *Ms. Visio* yang menggunakan Ribbon. *Visio 2007* tetap menggunakan menubar untuk “menampung” perintah-perintah yang digunakan.



Gambar 2. 6 Menu bar pada *Ms.Visio 2007* (Sutomo, 2009: 1)

Yang terlihat berbeda diantara keduanya adalah tampilan awal pada bagian [*Get Started*]. Kategori yang ditampilkan visio 2003 terlihat lebih banyak dibandingkan dengan *Ms.visio 2007*. Terdapat enam belas kategori pada *Ms.visio 2003* sedangkan pada *Ms.visio 2007* disederhanakan “hanya” menjadi separuhnya saja.



Gambar 2. 7 *Get Started* pada *visio 2007* (Sutomo, 2009: 2)