

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

2.1.1 Definisi Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 2005). Definisi ini lebih menekankan kepada elemen atau komponen daripada sistem tersebut.

Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memperoleh masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan (Kristanto, 2008).

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama (Pratama, 2014).

2.1.2 Definisi Informasi

Informasi sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan (Jogiyanto, 2005).

Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberi nilai, arti, dan manfaat (Pratama, 2014).

2.1.3 Sistem Informasi

Sistem informasi merupakan gabungan dari empat bagian utama mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur dan Sumber daya Manusia (SDM) (Pratama, 2014).

Contoh sistem informasi diantaranya adalah sistem informasi keuangan, sistem informasi manajemen, sistem informasi akuntansi, sistem informasi manufaktur. Jadi dapat disimpulkan bahwa sistem informasi merupakan kerangka

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

kerja yang terdiri dari perangkat keras, perangkat lunak, prosedur, orang, dan data yang berinteraksi satu sama lain untuk menghasilkan informasi yang berguna bagi pemakainya.

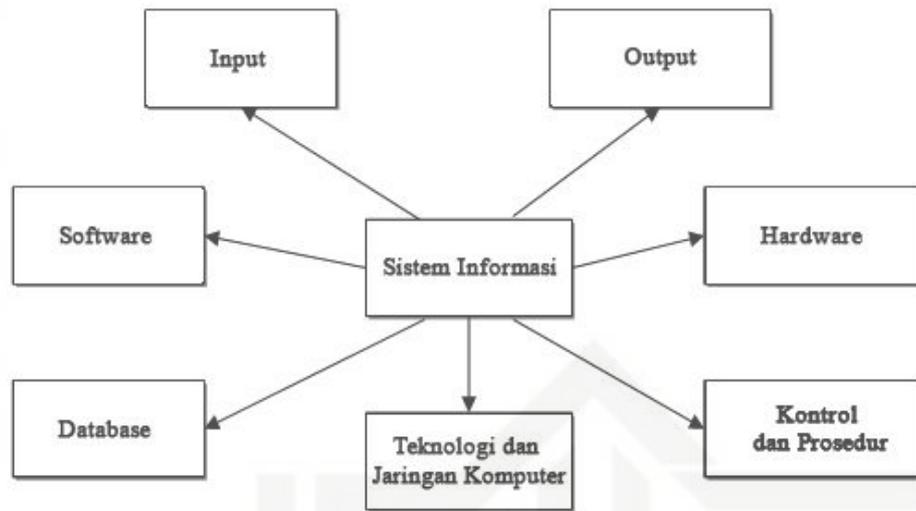
2.1.4 Fungsi Sistem Informasi

Menurut O'Brien yang diterjemahkan oleh (Fitriasari, 2005) fungsi-fungsi sistem informasi ada lima, dijelaskan sebagai berikut:

1. Area fungsional utama dari bisnis yang penting dalam keberhasilan bisnis, seperti fungsi akuntansi, keuangan, manajemen operasional, pemasaran, dan manajemen sumber daya manusia.
2. Kontributor penting dalam efisiensi operasional, produktivitas dan moral pegawai, serta layanan dan kepuasan pelanggan.
3. Sumber utama informasi dan dukungan yang dibutuhkan untuk menyebarluaskan pengambilan keputusan yang efektif oleh para *manager* dan praktisi bisnis.
4. Bahan yang sangat penting dalam mengembangkan produk dan jasa yang kompetitif, yang memberikan organisasi kelebihan strategis dalam pasar global.
5. Peluang berkarir yang dinamis, memuaskan, serta menantang bagi jutaan pria dan wanita.
6. Komponen penting dari sumber daya, infrastruktur, dan kemampuan perusahaan bisnis yang membentuk jaringan.

2.1.5 Komponen-Komponen Sistem Informasi

Sebuah sistem informasi memiliki sejumlah komponen di dalamnya. Komponen-komponen ini memiliki fungsi dan tugas masing-masing yang saling berkaitan satu sama lain. Keterkaitan antar kelompok ini membentuk suatu kesatuan kerja, yang menjadikan sistem informasi dapat mencapai tujuan dan fungsi yang ingin dicapai oleh pengguna dan pengembang sistem informasi bersangkutan. Komponen-komponen yang terdapat di dalam semua jenis sistem informasi mencakup tujuh poin. Komponen sistem informasi dapat dilihat pada Gambar 2.1.



Gambar 2.1 Komponen Sistem Informasi

(Sumber: Pratama, 2014)

Berikut penjelasan masing-masing tujuh komponen Sistem Informasi (Pratama, 2014):

1. *Input*

Komponen *input* berfungsi untuk menerima semua *input* (masukan) dari pengguna. *Input* yang diterima dalam bentuk data. Data ini berasal dari satu maupun beberapa sumber.

2. *Output*

Komponen *output* berfungsi untuk menyajikan hasil akhir ke pengguna sistem informasi. Informasi yang disajikan ini merupakan hasil dari pengolahan data yang telah diinputkan sebelumnya.

3. *Software* (Perangkat Lunak)

Komponen *software* mencakup semua perangkat lunak yang digunakan di dalam sistem informasi. Adanya komponen *software* ini membantu sistem informasi di dalam menjalankan tugasnya dan untuk dapat dijalankan sebagaimana mestinya. Komponen ini melakukan proses pengolahan data, penyajian informasi, penghitungan data, dan lain-lain.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4. *Hardware*
Komponen *hardware* mencakup semua perangkat keras komputer yang digunakan secara fisik di dalam sistem informasi, baik di komputer *server* maupun komputer *client*.
5. *Database* (Basis Data)
Komponen basis data berfungsi untuk menyimpan semua data dan informasi ke dalam satu atau beberapa tabel. Dengan adanya komponen basis data, maka akan memudahkan di dalam penyimpanan dan pengolahan data.
6. Kontrol dan Prosedur
Komponen kontrol berfungsi untuk mencegah terjadinya beragam gangguan dan ancaman terhadap data dan informasi yang ada di dalam sistem informasi, termasuk juga sistem informasi itu sendiri beserta komponen fisiknya. Kemungkinan gangguan dan ancaman dapat berupa kejahatan di dunia komputer (*cyber crime, cracker*), bencana alam, pencurian data, listrik yang tidak stabil, dan lainnya.
7. Komponen prosedur mencakup semua prosedur dan aturan yang harus dilakukan dan wajib ditaati bersama, guna mencapai tujuan yang diinginkan. Komponen prosedur berkaitan dengan komponen kontrol dalam hal mencegah terhadap kemungkinan ancaman dan gangguan yang terjadi pada sistem informasi. Yang berpengaruh terhadap layanan yang diberikan, informasi yang disajikan, dan tingkat kepuasan pengguna.
8. Teknologi dan Jaringan Komputer
Komponen teknologi mengatur *software, hardware, database*, kontrol dan prosedur, *input* dan *output*, sehingga sistem dapat berjalan dan terkendali dengan baik. Komponen jaringan komputer berperan di dalam menghubungkan sistem informasi dengan sebanyak mungkin pengguna, baik melalui kabel jaringan (*wired*) maupun tanpa kabel (*wireless*). Jaringan komputer dapat berupa jaringan lokal (*private*) hingga jaringan internet (publik).

2.1.6 Perancangan Sistem Informasi

Menurut (O'Brien, 2008), perancangan sistem memutuskan bagaimana sebuah sistem informasi yang diusulkan akan memenuhi kebutuhan informasi bagi pengguna akhir. Termasuk kegiatan desain logis dan fisik, dan antarmuka, data dan kegiatan proses desain yang menghasilkan spesifikasi sistem yang memenuhi persyaratan sistem yang dikembangkan pada tahap analisis sistem ada dua hal:

1. Sasaran perancangan sistem informasi ada tiga, yaitu:
 - a. Evaluasi atas alternatif perancangan sistem.
 - b. Penyiapan spesifikasi perancangan yang berorientasi pada pemakai tertentu.
 - c. Pengajuan perancangan kepada manajemen puncak.
2. Aspek-aspek dalam perancangan sistem ada tiga, yaitu:
 - a. Rancangan Masukan (*Input*)
Digunakan dalam proses data pada tahap analisa sistem. Rancangan meliputi pembuatan format pembuatan seluruh bukti transaksi.
 - b. Rancangan Prosedur
Meliputi seluruh bagian sistem yang ada di dalam sistem informasi yang dapat dibuat dengan dua gambar yang berbeda. Gambar pertama berupa *flowchart* (langkah-langkah berkaitan dengan pelaksanaan dengan prosedur dalam suatu bagian sistem). Gambar kedua berupa langkah-langkah pemrosesan data dalam bagian pengolahan data elektronik.
 - c. Rancangan Keluaran (*output*)
Output berupa laporan-laporan yang dibutuhkan oleh pihak manajemen perusahaan atau pihak intern perusahaan sebagai bahan pertimbangan dalam keputusan.

Dalam menciptakan sebuah aplikasi, terdapat beberapa hal yang perlu diperhatikan guna perolehan hasil yang maksimal (Whitten dkk, 2004), antara lain sebagai berikut:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

1. Produktivitas

Saat ini hampir segala bidang memerlukan aplikasi yang dapat digunakan sesuai dengan keperluan dalam bidangnya. Hal ini menyebabkan permintaan terhadap pengadaan aplikasi lebih banyak. Tuntutan terhadap kualitas aplikasi yang lebih bagus dan handal, tentunya hal ini membutuhkan lebih banyak *programmer* dan penganalisa sistem yang berkualitas, kondisi kerja ekstra, kemampuan pemakai untuk mengembangkan sendiri, bahasa pemrograman yang lebih baik, perawatan sistem yang lebih baik, disiplin teknis pemakaian perangkat lunak dan perangkat pengembangan sistem yang terotomasi.

2. Reliabilitas

Reliabilitas suatu perangkat lunak tidak seperti faktor kualitas lain yang dapat diukur, diarahkan dan diestimasi dengan menggunakan data pengembangan historis. Reliabilitas perangkat lunak didefinisikan dalam bentuk statistik sebagai kemungkinan operasi program komputer bebas kegagalan didalam suatu lingkungan dalam kurun waktu tertentu.

3. Maintabilitas

Maintabilitas mencakup perawatan aplikasi, seperti:

- a. Koreksi jika ditemukan kesalahan pada program.
- b. Pengadaptasian jika lingkungan berubah.
- c. Modifikasi jika pengguna membutuhkan perubahan kebutuhan.

4. Integritas

Integritas adalah mengukur kemampuan sistem suatu aplikasi untuk menahan serangan terhadap sekuritasnya. Dalam hal ini kekuatan sistem akan diuji terhadap serangan dari tipe tertentu yang dapat terjadi suatu waktu.

5. Usabilitas

Usabilitas merupakan ukuran terhadap kualitas interaksi yang terjadi antara aplikasi dengan pengguna. Ukuran usabilitas dapat diketahui melalui tampilan fisik suatu aplikasi (*user friendly*), penggunaan waktu yang efisien dan lain sebagainya.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Proses perancangan perangkat lunak merupakan serangkaian kegiatan dan hasil yang berhubungan dengan perangkat lunak, yang bertujuan untuk dihasilkannya suatu produk perangkat lunak. Walaupun ada banyak proses dalam perancangan suatu perangkat lunak, ada kegiatan-kegiatan mendasar yang umum bagi semua proses perancangan perangkat lunak (Sommerville, 2003) antara lain:

1. Spesifikasi Perangkat Lunak

Fungsionalitas Perangkat Lunak dan batasan operasinya harus didefinisikan.

2. Perancangan dan Implementasi Perangkat Lunak

Perangkat Lunak yang memenuhi persyaratan harus dibuat.

3. Validasi Perangkat Lunak

Perangkat lunak tersebut harus divalidasi untuk menjamin bahwa perangkat lunak bekerja sesuai dengan apa yang diinginkan.

4. Pengevolusian Perangkat Lunak

Perangkat Lunak harus dapat berkembang untuk menghadapi kebutuhan yang dapat berubah sewaktu-waktu.

2.2 *Object Oriented Analysis and Design (OOAD)*

2.2.1 *Pengertian Object*

Menurut Britton (2005), objek adalah paket *software* yang didalamnya terdapat dua metode yang digunakan untuk memanipulasi data tersebut.

2.2.2 *Pengertian Object Oriented*

Menurut Britton (2005), orientasi objek adalah sebuah pendekatan untuk mengembangkan sistem perangkat lunak berdasarkan hal data, perlengkapan, dan operasi yang akan mendefinisikan perangkat lunak tersebut.

Menurut Britton (2005) *Object Oriented* memiliki empat kelebihan dalam hal-hal sebagai berikut:

1. *Maintainable*

Pemeliharaan *software* dimulai saat sebuah sistem *software* diserahkan pada *client*. Selama siklus hidupnya sebuah aplikasi mungkin memerlukan perubahan dalam rangka memenuhi kebutuhan.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. Testable

Pendekatan *object oriented* dapat menyerderhanakan pengetesan *software*. *Software* yang dibangun dengan menggunakan pendekatan *object oriented* mampu mencukupi kebutuhan sendiri dan independen dengan *interface* yang terdefinisi dengan jelas. Hal ini menyebabkan setiap unit dapat secara hati-hati di tes sebelum sistem terintegrasi menjadi satu kesatuan.

3. Reusable

Jika *system developer* ingin mengembangkan sistem atau membangun suatu sistem untuk kasus yang serupa, maka pendekatan *object oriented* memungkinkan program yang telah dibangun untuk digunakan kembali.

4. Able to cope with large and complex system

Sistem *software* pada saat ini semakin besar dan kompleks. Pendekatan struktur desain *top down* dirasa sudah tidak lagi memadai untuk menangani skala dan kompleksitas dari sistem tertentu. Dimana sebelumnya *user* hanya puas dengan tampilan sederhana, namun pada saat ini *graphical user interface* diperlukan bagi *user*. Oleh karena itu, untuk memenuhi kebutuhan tersebut diperlukan pendekatan *object oriented* dalam mengembangkan sistem *software*.

Secara garis besar, *object oriented* adalah sebuah pendekatan untuk pengembangan suatu *software* dimana dalam struktur *software* tersebut didasarkan kepada interaksi objek dalam penyelesaian suatu proses.

2.3 Unified Modelling Language (UML)

UML adalah bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk mencetak biru atas visual mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Nugroho, 2005).

UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML ini akan mencakup lebih luas dari pada OOAD. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan *Object*

Management Group (OMG) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang dintegrasikan lewat *extensible markup language* (XML). Standar UML dikelola oleh OMG.

Tujuan utama perancangan UML ada enam, (Nugroho, 2005) yaitu:

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep ini.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework*, dan *pattern*.

2.3.1 Gambaran UML

UML adalah sebuah alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini di sebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah di mengerti serta dilengkapi dengan mekanisme efektif untuk berbagi dan mengkomunikasikan rancangan mereka yang lain (Nugroho, 2005).

UML adalah bahasa untuk memvisualisasi, menspesifikasikan dan mendokumentasikan artifak-artifak sistem perangkat lunak. UML merupakan sistem notasi yang membantu pemodelan sistem menggunakan konsep berorientasi objek.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Sekeloa Islamic University of Sultan Saifuddin Kasim Riau

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Dalam proyek pengembangan sistem apapun, fokus utama dalam analisis dan perancangan adalah model. Hal ini berlaku umum tidak hanya untuk perangkat lunak. Menurut Nugroho (2005) dengan model kita dapat mempresentasikan sesuatu karena lima hal, yaitu:

1. Model mudah dan cepat dibuat.
2. Model bisa digunakan sebagai simulasi untuk mempelajari lebih detail tentang sesuatu.
3. Model bisa dikembangkan sejalan dengan pemahaman kita tentang sesuatu.
4. Konsep bisa mewakili sesuatu yang nyata maupun tidak nyata.
5. Konsep-konsep yang diterapkan UML adalah suatu model berisikan informasi mengenai sistem (*domain*) model-model berisi elemen-elemen model seperti kelas, simpul-simpul, paket-paket, dan sebagainya. Satu diagram menunjukkan satu pandangan tertentu dari model.

Ada enam tujuan utama perancangan UML (Nugroho, 2005), dijelaskan sebagai berikut:

1. Menyediakan bahasa pemodelan Visual yang Ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern*.

2.3.2 Use Case Diagram

Use case diagram digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. *Use case diagram* menyediakan cara mendeskripsikan pandangan eksternal terhadap sistem dan interaksi-interaksinya dengan dunia luar. Pemodelan ini biasa dilakukan lewat proses berulang interaksi antara pengembang

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dan pemakai untuk memperoleh spesifikasi kebutuhan yang sama-sama disepakati.

Diagram *use case* berguna dalam tiga hal (Nugroho, 2005), yaitu:

1. Menjelaskan fasilitas yang ada (*requirements*).
2. Komunikasi dengan klien.
3. Membuat test dari kasus-kasus secara umum.

Tujuan utama pemodelan *use case* ada empat (Nugroho, 2005), seperti:

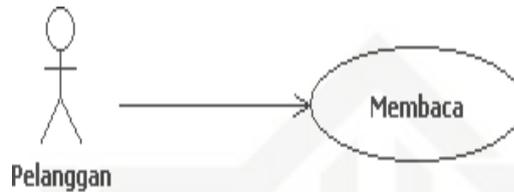
1. Memutuskan dan mendiskripsikan kebutuhan-kebutuhan fungsional sistem.
2. Memberikan deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan, sehingga model *use case* digunakan diseluruh proses pengembangan untuk komunikasi dan menyediakan basis untuk pemodelan berikutnya yang mengecu sistem harus memberikan fungsionalitas yang dimodelkan para *use case*.
3. Menyediakan basis untuk melakukan pengujian sistem yang memverifikasi sistem. Menguji apakah sistem telah memberikan fungsionalitas yang diminta.
4. Menyediakan kemampuan melacak kebutuhan fungsionalitas menjadi kelas-kelas dan operasi-operasi *actual* di sistem. Untuk menyederhanakan perubahan dan ekstensi ke sistem dengan mengubah model *use case* dan kemudian melacak *use case* yang dipengaruhi ke perancangan dan implementasi sistem.

Syarat penamaan *use case* adalah nama didefinisikan sesederhana 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 di buat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang tapi aktor belum tentu orang
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Dalam *use case* juga dikenal dengan duahubungan antar *use case* yang merupakan generalisasi antara *use case* (Nugroho, 2005) yaitu:

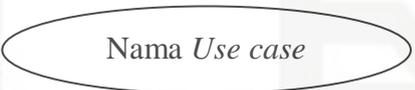
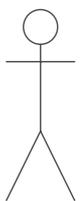
1. **Include**, yaitu perilaku *use case* merupakan bagian dari *use case* yang lain.
2. **Extend**, yaitu perilaku *use case* memperluas perilaku *use case* yang lain.



Gambar 2. 2 *Use Case Diagram*
(Sumber: Nugroho, 2005)

Simbol-simbol pada *Use Case Diagram* dapat dilihat pada Tabel 2. 1.

Tabel 2. 1 Simbol-Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit atau <i>Actor</i>, biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama <i>use case</i>.</p>
<p>Aktor/<i>Actor</i></p>  <p>Nama aktor</p>	<p>Orang, yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>Actor</i> adalah gambar dari orang, tapi <i>Actor</i> belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p>Asosiasi/ <i>association</i></p> 	<p>Komunikasi antar <i>Actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use</i></p>

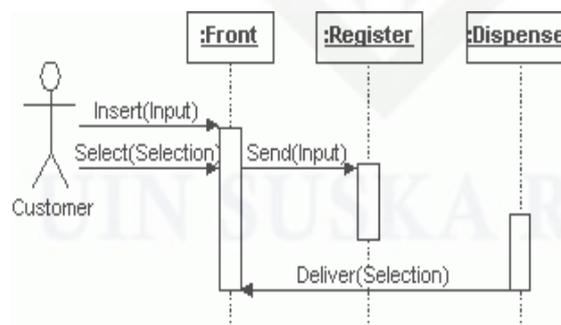
Tabel 2.1 Simbol-Simbol *Use Case Diagram* (Lanjutan)

Simbol	Deskripsi
	<i>case</i> memiliki interaksi dengan aktor.
Ekstensi/ <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>include/ users</i> 	<i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan.

(Sumber: Rosa, 2014)

2. 3. 2 *Sequence Diagram*

Sequence diagram merupakan salah satu diagram interaksi yang menjelaskan bagaimana suatu operasi itu dilakukan, pesan apa yang dikirim dan kapan pelaksanaannya. Diagram ini disusun berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



Gambar 2. 3 *Sequence Diagram*

(Sumber: Nugroho, 2005)

Simbol-simbol *Sequence Diagram* dapat dilihat pada Tabel 2. 2.

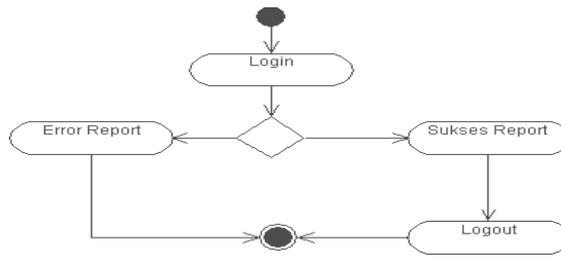
Tabel 2. 2 Simbol-simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama aktor</p>	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu menggunakan orang, biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor
<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan hidup suatu objek
<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 pesan
<p>Pesan tipe <i>create</i></p> 	Objek yang lain, arah panah mengarah pada objek yang di buat

(Sumber: Rosa, 2014)

2.3.3 *Activity Diagram*

Pada dasarnya *activity diagram* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *statechart*. Diagram *statechart* berfokus pada objek yang dalam suatu proses (atau proses menjadi suatu objek), diagram *activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. *activity diagram* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Activity Diagram

(Sumber: Nugroho, 2005)

Simbol-simbol *Activity Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Activity Diagram*

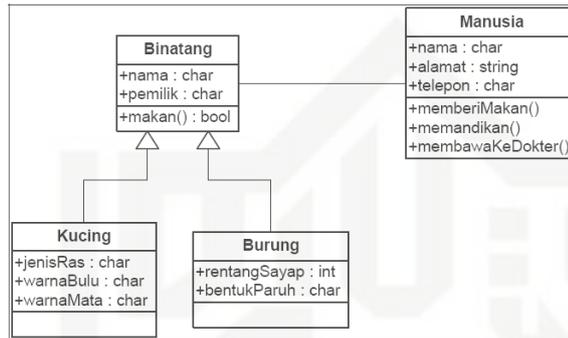
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

(Sumber: Rosa, 2014)

2.3.4 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan keadaan (atribut/properti)

suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* juga menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Bentuk *class diagram* seperti pada Gambar 2.5 dan keterangan simbol-simbol *class diagram* dapat dilihat pada tabel 2.4.



Gambar 2.5 *Class Diagram*
 (Sumber: Nugroho, 2005)

Tabel 2.4 Simbol-simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> <pre> Nama-kelas + atribut + operasi () </pre>	Kelas pada struktur sistem
<p>Antarmuka/<i>interface</i></p> <p>○ Nama_<i>interface</i></p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Asosiasi/<i>association</i></p> <p>—</p>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan <i>multiplicity</i>
<p>Generalisasi</p> <p>→</p>	Relasi antar kelas dengan makna generalisasi-spesifikasi (umum-khusus)

Tabel 2.4 Simbol-simbol *Class Diagram* (Lanjutan)

Simbol	Deskripsi
 ketergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna ketergantungan antar kelas
 Agregasi/ <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian

(Sumber: Rosa, 2014)

2.4 *Hypertext Preprocessor* (PHP)

PHP merupakan singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen html. Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga maintenance situs *web* tersebut menjadi lebih mudah dan efisien (Peranginan, 2006).

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* sejenis. PHP difokuskan pada pembuatan *script server-side*, yang bisa melakukan apa saja yang dapat dilakukan oleh *common gateway interface* (CGI), seperti mengumpulkan data dari *form*, menghasilkan isi halaman *web* dinamis, dan kemampuan mengirim serta menerima *cookies* bahkan lebih daripada kemampuan CGI.

PHP dapat digunakan pada semua sistem operasi, antara lain *Linux*, *Unix* (termasuk variannya HP-UX, *Solaris* dan *OpenBSD*), *Microsoft Windows*, *Mac OS X*, dan masih banyak lagi lainnya, bahkan PHP dapat bekerja sebagai suatu CGI *processor*. PHP tidak terbatas pada hasil keluaran *HyperText Markup Languages* (HTML). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, file PDF, dan *movies Flash*. PHP juga dapat menghasilkan teks seperti XHTML dan file XML lainnya.

Menurut Hidayatullah (2014) dalam bukunya yang berjudul *Pemrograman Web* menyatakan ada sepuluh kelebihan PHP, diantaranya:

1. PHP berbasis *Server Side Scripting*
2. *Command Line Scripting* pada PHP
3. PHP dapat membuat aplikasi *Desktop*

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4. Digunakan untuk berbagai macam *Platform OS*
5. Mendukung berbagai macam *Web Server*
6. *Object Oriented Programming* atau *Procedural*
7. *Output file* PHP pada XHTML, HTML dan XML
8. Mendukung banyak RDMS (*Database*)
9. Mendukung banyak komunikasi
10. Pengolahan teks yang sangat baik

2.5 MySQL

MySQL merupakan salah satu *database server* yang berkembang di lingkungan *open source* dan didistribusikan secara *free* (gratis) di bawah lisensi GPL. MySQL merupakan *Relational Database Management Sistem* (RDBMS) server. RDBMS adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada *database* memiliki relasi antara satu tabel dengan tabel lainnya (Prasetyo, 2004).

Menurut Prasetyo (2004) ada empat keunggulan dari MySQL yaitu:

1. Cepat, handal dan mudah dalam penggunaannya

MySQL lebih cepat tiga sampai empat kali dari pada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan MySQL.

2. Didukung oleh berbagai bahasa

Database server MySQL dapat memberikan pesan error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.

3. Mampu membuat tabel berukuran sangat besar

Ukuran maksimal dari setiap tabel yang dapat dibuat dengan MySQL adalah 4 *Gigabyte* (GB) sampai dengan ukuran *file* yang dapat ditangani oleh sistem operasi yang dipakai.

4. Lebih murah

MySQL bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk UNIX *platform* dan Windows *platform*.

2.6 Framework CodeIgniter (CI)

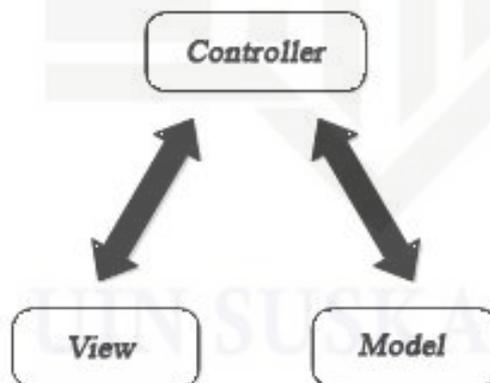
Buku Pemrograman *Web* yang ditulis oleh Hidayatullah (2014) menjelaskan CI adalah salah satu *framework* PHP yang paling *powerful* saat ini karena di dalamnya terdapat fitur lengkap aplikasi *web* dimana fitur-fitur tersebut sudah dikemas menjadi satu. Selain itu CI juga saat ini banyak digunakan khususnya bagi *developer web* untuk mengembangkan aplikasi berbasis *web* tersebut.

Menurut Hidayatullah (2014) ada empat keuntungan yang didapat dari penggunaan *framework* adalah:

1. Menghemat waktu pengembangan
2. Penggunaan ulang program/kode
3. Bantuan komunitas
4. Kumpulan program terbaik

Model View Controller (MVC)

Model View Controller atau yang disebut dengan MVC adalah suatu metode yang memisahkan *data logic (Model)* dari *presentation logic (View)* dan *process logic (Controller)* atau secara sederhana adalah memisahkan antara desain *interface*, data dan proses (Hidayatullah, 2014). Metode MVC dapat dilihat pada Gambar 2.6.



Gambar 2.6 Metode MVC

(Sumber: Hidayatullah, 2014)

Terdapat tiga komponen dalam metode MVC menurut Hidayatullah (2014) yaitu:

1. *Model*

Model mengelola basis data (RDBMS) seperti MySQL ataupun RDMS. *Model* berhubungan dengan *database* sehingga biasanya dalam *model* akan berisi *class* ataupun fungsi untuk membuat (*create*), melakukan pembaruan (*update*), menghapus data (*delete*), mencari data (*search*), dan mengambil data (*select*) pada *database*.

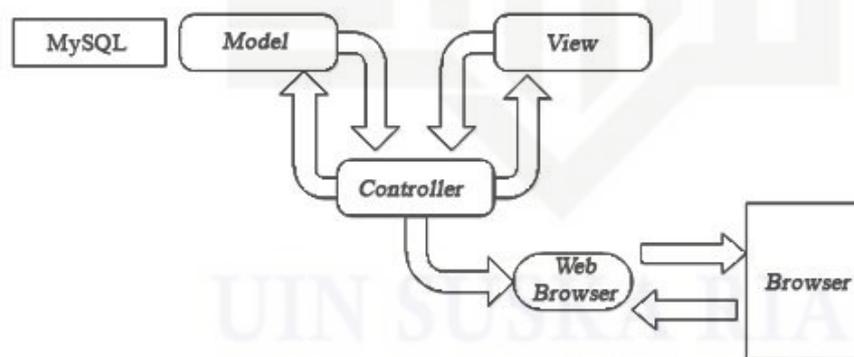
2. *View*

View adalah bagian *user interface* atau bagian yang nantinya merupakan tampilan untuk *end-user*. *View* bisa berupa halaman HTML, CSS, RSS, *javascript*, *jquery*, *ajax*.

3. *Controller*

Merupakan penghubung antara *view* dan *model*, maksudnya ialah karena *model* tidak dapat berhubungan langsung dengan *view* ataupun sebaliknya. Sehingga tugas *controller* ialah sebagai jembatan pemrosesan data atau alur *logic* program, menyediakan variabel yang akan ditampilkan di *view*, pemanggilan *model* sehingga *model* dapat mengakses *database*.

Untuk mengetahui bagaimana alur MVC pada CI dapat dilihat pada Gambar 2.7 dalam hal ini *user* mengakses *browser* untuk menjalankan *web* yang berbasis CI.



Gambar 2.7 *Flow Diagram CodeIgniter*

(Sumber: Hidayatullah, 2014)

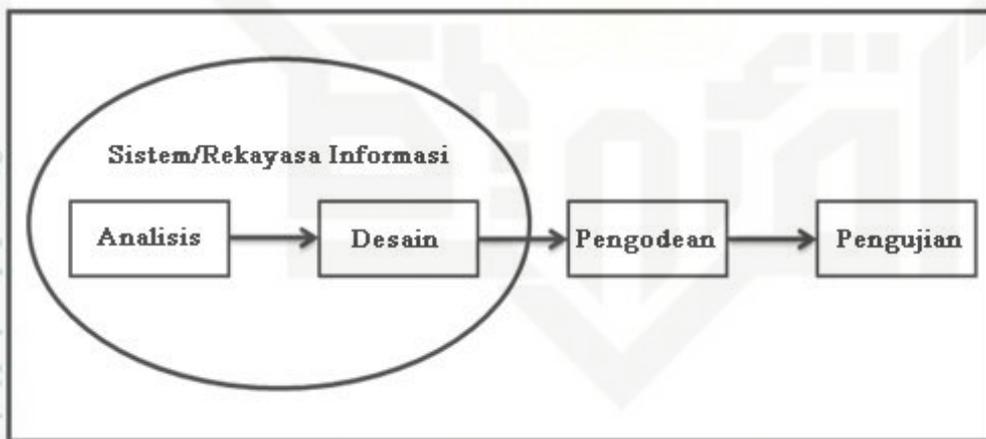
2.7 System Development Life Cycle (SDLC)

SDLC dimulai dari tahun 1960-an, untuk mengembangkan sistem skala usaha besar secara fungsional untuk para konglomerat pada zaman itu. Sistem-sistem yang akan dibangun mengelola informasi kegiatan dan rutinitas dari perusahaan-perusahaan yang berpotensi memiliki data yang besar dalam perkembangannya.

SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metode yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. SDLC memiliki beberapa model dalam penerapan tahapan prosesnya.

2.7.1 Model Waterfall atau Air Terjun

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensi linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakann pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Ilustrasi *waterfall* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Ilustrasi Model *Waterfall*

(Sumber: Rosa, 2014)

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi perangkat lunak, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

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

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. 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.

Model air terjun sangat cocok digunakan untuk kebutuhan pelanggan yang sudah sangat dipahami dan kemungkinan terjadinya perubahan kebutuhan selama pengembangan perangkat lunak kecil. Hal positif dari metode air terjun adalah

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahapan sebelumnya selesai dijalankan (Rosa, 2014).

2.7.2 Pengujian Perangkat Lunak

Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian pada perangkat lunak ada empat bagian yaitu sebagai berikut (Rosa, 2014).

1. Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (*independent test group* (ITG)).
4. Pengujian dan penirkutuan (*debugging*) merupakan aktifitas yang berbeda, tapi *debugging* harus diakomodasi pada berbagai strategi pengujian. Pengujian lebih fokus untuk mencari adanya kesalahan (*error*) baik dari sudut pandang orang secara umum atau dari sudut pandang pengembang tanpa harus menemukan lokasi kesalahan pada kode program. *Debugging* adalah proses mencari lokasi kesalahan (*error*) pada kode program sehingga dapat segera diperbaiki oleh pembuat program (*programmer*). Pengujian memiliki beberapa pendekatan sebagai berikut (Rosa, 2014):

- a. *Black-box Testing* (pengujian kotak hitam)

Pengujian ini dilakukan terhadap perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi,

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka harus uji yang dibuat adalah:

- 1) Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- 2) Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar atau kata sandi salah, atau sebaliknya, atau keduanya salah.

2. 8 Konsep Dasar Akuntansi

2.8.1 Pengertian Akuntansi

Akuntansi dapat didefinisikan sebagai sistem informasi yang menghasilkan laporan kepada pihak-pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi perusahaan (Niswonger, 1999).

Akuntansi atau ada juga yang menyebutnya akunting merupakan bahasa bisnis yang dapat memberikan informasi tentang kondisi bisnis dan hasil usahanya pada suatu waktu atau periode tertentu. Bahasa ini terbentuk melalui laporan keuangan yang dihasilkan dari sistem akuntansi (Harahap, 2008).

2.8.2 Transaksi

Transaksi merupakan kejadian atau kondisi ekonomi yang secara langsung mempengaruhi kondisi keuangan atau hasil operasi suatu entitas (Niswonger, 1999). Transaksi adalah kegiatan yang mempengaruhi atau kepentingan dari perusahaan serta diproses oleh sistem informasinya sebagai unti pekerjaan (Mahatmyo, 2014).

Menurut Zulkifli (2003), Secara umum transaksi dapat diartikan sebagai kejadian ekonomi/keuangan yang melibatkan paling tidak 2 pihak (seseorang dengan seseorang atau beberapa oranglainnya) yang saling melakukan

pertukaran, melibatkan diri dalam perserikatan usaha, pinjam meminjam atas dasar sama-sama suka ataupun atas dasar suatu ketetapan hukum atau syariah yang berlaku. Dalam *system* ekonomi yang paradigma Islami, transaksi harus dilandasi oleh aturan hukum-hukum Islam (syariah) karena transaksi adalah manifestasi amal manusia yang bernilai ibadah dihadapan Allah SWT, yang dapat dikategorikan menjadi 2 yaitu transaksi halal dan haram.

2.8.3 Daftar Akun (*Chart of Account*)

Akun adalah suatu alat untuk mencatat transaksi-transaksi keuangan yang bersangkutan dengan aktiva, kewajiban, modal, pendapatan, dan biaya. Tujuan pemakaian akun adalah untuk mencatat data yang akan menjadi dasar penyusunan laporan keuangan (Yusuf, 2007).

Menurut Niswonger (1999), klasifikasi akun utama terbagi atas:

1. Aktiva

Aktiva (*asset*) adalah sumber daya yang dimiliki oleh entitas bisnis atau usaha. Sumber daya ini dapat berbentuk fisik ataupun hak yang mempunyai nilai ekonomis. Contoh aktiva adalah kas (uang tunai), piutang usaha, perlengkapan, beban dibayar muka (seperti asuransi), bangunan, peralatan, tanah, dan hak paten.

2. Kewajiban

Kewajiban (*liabilities*) adalah utang kepada pihak luar (kreditor). Kewajiban biasanya diidentifikasi dalam neraca sebagai jumlah terutang (*payable*). Contoh kewajiban adalah utang usaha, wesel bayar, dan utang upah. Pembayaran kas yang diterima sebelum jasa diberikan kepada pembeli. Komitmen pelayanan yang akan diberikan kepada pembeli tersebut sering disebut pendapatan diterima di muka (*unearned revenue*).

3. Ekuitas pemilik

Ekuitas atau disebut juga modal pemilik (*owner equity*) adalah hak pemilik terhadap aktiva bisnis. Untuk perusahaan perorangan, ekuitas pemilik didalam neraca terlihat dalam saldo akun modal pemilik. Akun penarikan (*drawing*) menunjukkan penarikan modal yang dilakukan oleh pemilik.

4. Pendapatan

Pendapatan (*revenue*) adalah peningkatan ekuitas pemilik yang diakibatkan oleh proses penjualan barang atau jasa kepada para pembeli.

Jumlah kenaikan yang dicatat dalam akun biasanya sama atau lebih besar dari pada jumlah penurunan yang tercatat dalam akun tersebut. Karena itu, saldo normal dari semua akun biasanya akan bernilai positif.

Kaidah debit-kredit dan saldo normal dari berbagai akun dapat dilihat pada Tabel 2.5.

Tabel 2.5 Saldo Normal Akun

akun	Kenaikan (Saldo Normal)	Penurunan
Akun neraca:		
Aktiva	Debit	Kredit
Kewajiban	Kredit	Debit
Ekuitas Pemilik:		
Modal	Kredit	Debit
Penarikan	Debit	Kredit
Akun laporan laba rugi:		
Pendapatan	Kredit	Debit
Beban	Debit	Kredit

(Sumber: Niswonger, 1999)

Jika suatu akun yang biasanya mempunyai saldo debit mempunyai saldo kredit atau sebaliknya, maka telah terjadi kesalahan atau kondisi yang tidak normal.

Untuk memudahkan dalam mencari serta menggolongkan dari sekian banyak akun yang digunakan maka setiap akun dalam perusahaan diberi kode akun dan disusun dalam suatu bagan perkiraan (*chart of account*). Dalam bagan akun tersebut dimuat akun-akun yang sesuai dengan akun riil dan akun nominal (Muliawati dkk, 2007).

2.8.4 Jurnal Umum

Jurnal umum, atau yang disebut juga buku jurnal harian menggolongkan transaksi mana yang dikelompokkan mempengaruhi perkiraan debit, dan mana yang digolongkan sebagai yang akan mempengaruhi perkiraan sebelah kredit, kegiatan ini merupakan awal dari proses akuntansi (Harahap, 2008).

Jurnal umum adalah aktivitas meringkas dan mencatat transaksi koperasi berdasarkan dokumen dasar secara kronologis beserta penjelasan yang diperlukan dalam buku harian. (Rudianto, 2010).

Jurnal merupakan catatan pertama yang bersifat kronologis tentang transaksi maupun kejadian yang menunjukkan akun-akun apa yang harus di debit dan dikredit beserta jumlah dan keterangan yang berkaitan dengan transaksi maupun kejadian tersebut (Muliawati dkk, 2007).

Ada beberapa fungsi jurnal sebagai berikut:

1. Fungsi mencatat

Jurnal wajib mencatat setiap peristiwa ekonomis suatu perusahaan. Setiap ada perubahan harta, utang, modal, pendapatan maupun beban harus dicatat didalam kemudian baru diposting ke buku besar.

2. Fungsi historis

Jurnal mencatat peristiwa ekonomis menurut urutan waktu terjadinya (kronologis) dari hari ke hari.

3. Fungsi analisis

Jurnal merupakan hasil analisis peristiwa ekonomi dari petugas akuntansi untuk menentukan akun-akun apa saja yang harus dikredit atau didebit.

4. Fungsi instruktif

Jurnal memberi perintah untuk mencatat ke dalam akun tertentu dengan jumlah rupiah tertentu pada sebelah debit atau kredit.

5. Fungsi informatif

Jurnal memberikan informasi tentang uraian dari setiap peristiwa keuangan, seperti tanggal terjadinya, jumlah, nama orang, nama barang, tempat, dan sebagainya.

Bentuk jurnal umum dapat dilihat pada Tabel 2.6.

Tabel 2.6 Bentuk Jurnal Umum

Tanggal	Keterangan	P/R	Debit	Kredit
20.	Akun yang didebit	...	Rp. xxx	Rp. –
	Akun yang dikredit	...	Rp. -	Rp. Xxx

(Sumber: Muliawati dkk, 2007)

2.8.5 Buku Besar

Buku besar merupakan tempat menampung seluruh transaksi yang telah diklasifikasikan melalui jurnal. Jadi seluruh jurnal dimasukkan sesuai kelasnya kedalam buku besar dengan cara memindah bukukan semua jurnal (posting) ke buku besar tadi (Harahap, 2008).

Akun-akun buku besar berfungsi untuk mengikhtisarkan akibat-akibat transaksi secara lengkap atas harta, utang, dan modal. Buku besar juga digunakan sebagai dasar untuk menyusun laporan keuangan (Muliawati dkk, 2007). Bentuk buku besar dapat dilihat pada Tabel 2.7.

Tabel 2.7 Bentuk Buku Besar

Tanggal	Uraian	Ref	Debit	Kredit	Saldo	
					Debit	Kredit
20.			Rp. -	Rp. -	Rp. -	Rp. -
..						

(Sumber: Muliawati dkk, 2007)

2.8.6 Trial Balance

Trial balance atau neraca saldo adalah kumpulan dari saldo-saldo yang ada pada setiap perkiraan di buku besar. pada kasus ini jumlah dari kolom debit dan kredit harus sama. Jika tidak, maka telah terjadi kesalahan pencatatan, mungkin dari jurnal umum atau buku besar (Harahap, 2007).

Neraca saldo adalah suatu daftar yang memuat nama-nama buku-buku besar beserta saldonya yang digunakan oleh suatu perusahaan dan berfungsi sebagai alat pemeriksa terhadap kebenaran pancatatan dalam buku besar.

Tujuan dari penyusunan neraca saldo adalah untuk memeriksa keseimbangan antara jumlah saldo baik debit maupun kredit dari masing-masing akun, untuk segera dapat mengetahui adanya kesalahan, dan sebagai awal pengikhtisaran dalam menyusun perhitungan laba-rugi dan neraca (Muliawati dkk, 2007). Bentuk *trial balance* dapat dilihat pada Tabel 2.8.

Tabel 2.8 Bentuk *Trial Balance*

No. Akun	Nama Akun	Debit	Kredit
11	Kas (<i>cash</i>)	Rp. -	-
21	Utang Usaha	-	Rp. -
31	Modal	-	Rp. -
32	Prive	Rp. -	-
41	Pendapatan	-	Rp. -
51	Beban	Rp. -	-
Total		Rp. -	Rp. -

(Sumber: Muliawati dkk, 2007)

2.8.7 Laba Rugi

Laporan laba rugi adalah laporan yang menyajikan ukuran keberhasilan operasi perusahaan selama periode tertentu. Lewat laporan laba rugi (Hery, 2016).

Laporan laba rugi adalah laporan yang disusun sistematis, isinya penghasilan yang diperoleh perusahaan dikurangi dengan beban-beban yang terjadi dalam perusahaan selama periode tertentu. Dalam laporan laba rugi menjabarkan elemen-elemen penghasilan dan beban perusahaan sehingga menghasilkan suatu laba atau rugi (Sujarweni, 2017).

Laporan laba rugi bersumber dari 2 hal yaitu laba dan biaya, karena itu dalam penyusunan laporan ini seorang akuntan harus menyadari dengan baik yang mana yang termasuk dalam kategori laba dan begitu pula sebaliknya, yang mana masuk kedalam kategori biaya. Jika terlalu besar biaya, maka memperlihatkan bahwa laporan tersebut lebih besar kerugiannya dibandingkan laba, begitu pula sebaliknya (Fahmi, 2014). Bentuk laporan laba rugi dapat dilihat pada Tabel 2.9.

Tabel 2.9 Bentuk Laporan Laba Rugi

Nama Perusahaan		
Laporan Laba Rugi		
periode		
Pendapatan		Rp. –
Beban	Rp. –	
Jumlah beban operasi		<u>Rp. –</u>
Laba		Rp. –

(Sumber: Muliawati dkk, 2007)

2.8.8 Laporan Ekuitas Pemilik (Laporan Perubahan Modal)

Laporan ekuitas pemilik melaporkan perubahan ekuitas pemilik selama jangka waktu tertentu. Laporan ekuitas dibuat setelah laporan laba rugi, karena didalam laporan ekuitas dimasukkan keterangan mengenai laba bersih ataupun rugi bersih. Begitu juga laporan ekuitas ini dibuat sebelum pembuatan laporan neraca, karena didalam laporan neraca akan dimasukkan keterangan mengenai ekuitas. Oleh karena itu, laporan ekuitas sering disebut penghubung antara laporan laba rugi dengan neraca. (Niswonger, 1999). Bentuk laporan perubahan modal dapat dilihat pada Tabel 2.10.

Tabel 2.10 Bentuk Laporan Perubahan Modal

Nama Perusahaan		
Laporan Perubahan Modal		
Periode		
Modal		Rp. –
Laba		Rp. –
Prive	RP. -	
Perubahan modal		Rp. –

(Sumber: Muliawati dkk, 2007)

2.8.9 Neraca

Laporan neraca atau daftar neraca disebut juga laporan posisi keuangan perusahaan. Laporan ini menggambarkan posisi aktiva, kewajiban dan modal pada saat tertentu. Laporan ini bisa disusun setiap saat dan merupakan opname situasi posisi keuangan pada saat itu. (Harahap, 2008).

Menurut Niswonger (1999), Neraca melaporkan jumlah aktiva, kewajiban, dan ekuitas perusahaan. Seksi aktiva dalam neraca disusun berdasarkan urutan dimana aktiva tersebut akan dikonversikan menjadi kas atau digunakan dalam operasi. Kas diurutkan pertama, diikuti oleh piutang, dan aktiva lainnya. Kemudian disajikan aktiva yang sifatnya tetap seperti tanah, bangunan, dan peralatan. Pada seksi kewajiban, utang usaha merupakan satu-satunya kewajiban.

Menurut Muliawati (2007), unsur-unsur neraca terdiri atas akun-akun riil yang dikelompokkan atas kelompok harta, kelompok utang, dan kelompok modal.

1. Harta/aktiva (*asset*), meliputi:
 - a. Aktiva/harta lancar
 - b. Investasi/penyertaan jangka panjang
 - c. Harta tetap
 - d. Harta tidak berwujud
 - e. Harta lain-lain
2. Utang/kewajiban (*liabilities*), meliputi:
 - a. Utang lancar
 - b. Utang jangka panjang
 - c. Utang lain-lain
3. Modal

Bentuk laporan neraca dapat dilihat pada Tabel 2.11.

Tabel 2.11 Bentuk Laporan Neraca

Nama Perusahaan		
Laporan Neraca		
periode		
Harta		
Harta lancar		
Kas	RP. –	
Harta tetap		
Gedung		Rp. –
Jumlah Harta		<u>RP. ----</u>
Utang		
Utang Usaha		
	RP. –	
Modal		
Modal		
	RP. –	
Jumlah utang dan modal		<u>RP. ----</u>

(Sumber: Muliawati dkk, 2007)

2.9 Sistem Informasi Akuntansi

Menurut Niswonger (1999), sistem akuntansi adalah metode dan prosedur untuk mengumpulkan, mengklasifikasikan, mengikhtisarkan, dan melaporkan informasi mengenai keuangan dan operasi usaha.

Perbedaan antara subsistem Sistem Informasi Akuntansi (SIA) dengan Sistem Informasi Manajemen (SIM) berpusat pada proses transaksi. Sistem informasi menerima input, yang disebut transaksi, yang akan dikonversikan melalui berbagai proses menjadi informasi *output*, yang akan diberikan kepada pengguna (Mahatmyo, 2014).

2.10 Profil Perusahaan

2.10.1 PT. Kalber Reksa Abadi

PT. Kalber Reksa Abadi berdiri pada tanggal 26 Oktober 2001 oleh Ir. Hary Tri Prasetyo. Melalui keputusan pemerintah dengan nomor Surat Izin Usaha Perusahaan (SIUP) NOMOR: 43/M. 04. 01/BPTPM/I/2015 Tanggal 21 Januari

2015. Surat Ijin Tempat Usaha (SITU) NOMOR:0160/03. 01/BPTPM/I/2015 tanggal 21 Januari 2015, Sertifikat Badan Usaha (SBU) NOMOR: 0178271, Surat Izin Usaha Jasa Konstruksi (SIUJK) NOMOR: 0160/03. 01/BPTPM/I/2015 tanggal 21 Januari 2015.

PT. Kalber Reksa Abadi adalah perusahaan yang bergerak dalam bidang konstruksi jalan, jembatan, dan bangunan. Perusahaan ini beralamat di Jl. Cipta Sari No. 2A Pekanbaru.

2.10.2 Visi dan Misi

Visi: Menjadikan perusahaan yang memiliki kompetensi handal dan terkemuka di bidang jasa konstruksi.

Misi:

1. Mengutamakan keselamatan kesehatan kerja (K3) dalam setiap aktifitas perusahaan.
2. Mengutamakan kualitas terbaik pada setiap aktifitas perusahaan.
3. Prioritas nomor satu untuk konsumen dan pangsa pasar.
4. Menjadi perusahaan yang memiliki kredibilitas handal

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.