

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

Adapun teori umum yang digunakan dalam penelitian ini sebagai berikut:

##### **2.1.1 Rancang Bangun**

Menurut Bahra (2006) dalam (Prasetyo, 2015: 2) rancang bangun adalah salah satu tahapan membangun suatu sistem agar sistem tersebut bisa berjalan dengan baik. Menurut Kendall (2010) dalam (Prasetyo, 2015: 2) rancang bangun adalah bagian dari perancangan sistem informasi yang logic yaitu peralatan antarmuka pengguna. Antarmuka menghubungkan pemakai dengan sistem, jadi peranannya benar-benar penting. Jadi, dapat disimpulkan bahwa rancang bangun adalah menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut.

##### **2.1.2 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).

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 (Prahasta, 2014: 78).

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.

### **2.1.3 Pengertian Informasi**

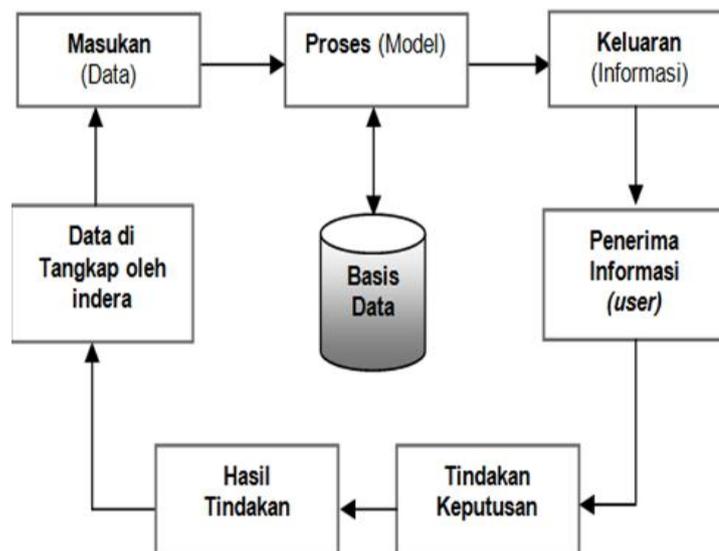
Informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian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

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 (IBISA, 2011: 4).

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

Adapun fungsi–fungsi informasi adalah sebagai berikut:

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

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

#### **2.1.4 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) .

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, 2011: 16).

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.

#### 2.1.4.1 Komponen Sistem Informasi

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

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.

### **2.1.5 Persediaan**

Persediaan 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 (Chandra & Wiradinata, 2015: 125):

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. Ketidakpastian. *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*.

#### **2.1.5.1 Pengendalian 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 & Wiradinata, 2015: 125).

#### **2.1.6 System Development Life Cycle (SDLC)**

*System Development Life Cycle* (SDLC) 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) (Rosa & Shalahuddin, 2013: 26).

*SDLC* (*System Development Life 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* (Chandra & Wiradinata, 2015: 126).

Menurut (Rosa & Shalahuddin, 2013: 26-28), 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 melakukan 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. Integrasi 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*)

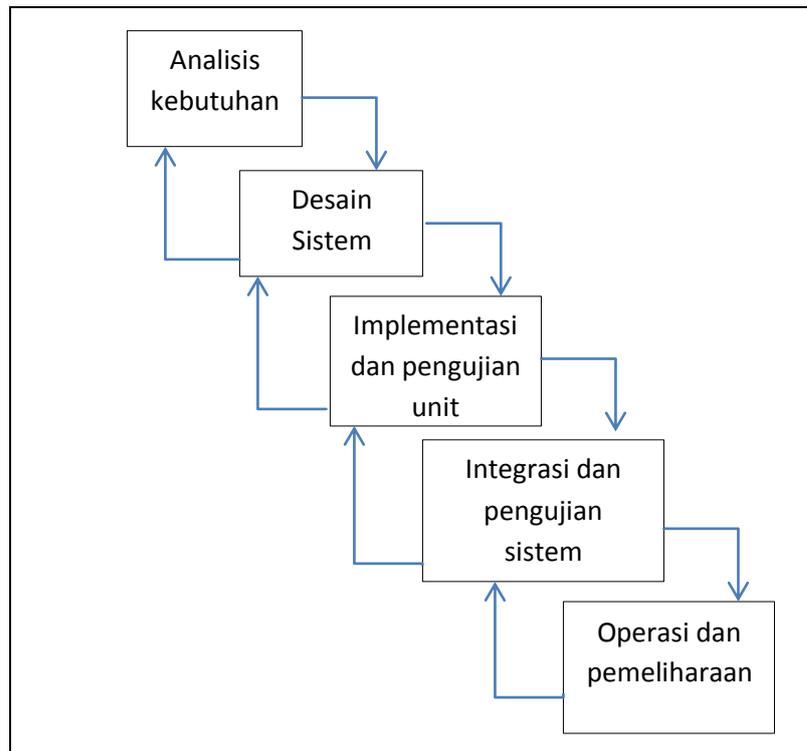
Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

8. Disposisi (*Disposisi*)

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

### **2.1.7 Model Waterfall**

SDLC (*System Development Life Cycle*) yang terkenal adalah SDLC model klasik yang biasa disebut dengan metode *waterfall*. Seperti gambar dibawah ini (Hermawan, Shinta, & Migunani, 2016: 17-18):



**Gambar 2.2** Model *Waterfall*

Berikut ini adalah fase-fase metode *Waterfall* yaitu: (Hermawan et al., 2016: 18)

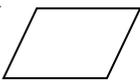
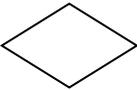
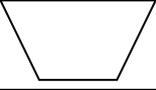
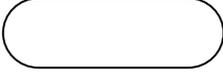
1. Pada fase analisis kebutuhan perancang sistem berkonsultasi dengan pengguna sistem butuhkan dari sistem yang akan dirancang. Hasil konsultasi ini lalu menjadi spesifikasi sistem;
2. Proses setelah proses analisa kebutuhan adalah proses desain sistem. Proses desain sistem membagi analisis kebutuhan dalam sistem perangkat keras atau perangkat lunak. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya;

3. Kemudian pada proses implementasi, desain sistem diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan dalam bentuk serangkaian program atau unit program. kemudian dilakukan pengujian terhadap unit-unit yang dihasilkan apakah sudah memenuhi spesifikasi atau belum;
4. Pada proses pengujian sistem, unit-unit tersebut disatukan dan dilakukan pengujian secara keseluruhan. Kemudian dilakukan pengoperasian sistem pada lingkungan yang sebenarnya. Setelah pengujian sistem, perangkat lunak dikirim kepada pelanggan;
5. Terakhir dilakukan perawatan atau pemeliharaan terhadap sistem tersebut. Ini merupakan fase siklus hidup paling lama. Sistem dipasang dan dipakai. Pemeliharaan mencakup koreksi dari berbagai kesalahan/*error* yang tidak ditemukan pada tahap-tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan-persyaratan baru ditambahkan (Hermawan et al., 2016: 18).

### **2.1.8 Flowchart**

Bagan aliran (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur sistem secara logika (Rismawati & Triwahyuni, 2014: 322). Adapun simbol-simbol *flowchart* adalah sebagai berikut:

**Tabel 2.1** Simbol–simbol yang ada pada *Flowchart*

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

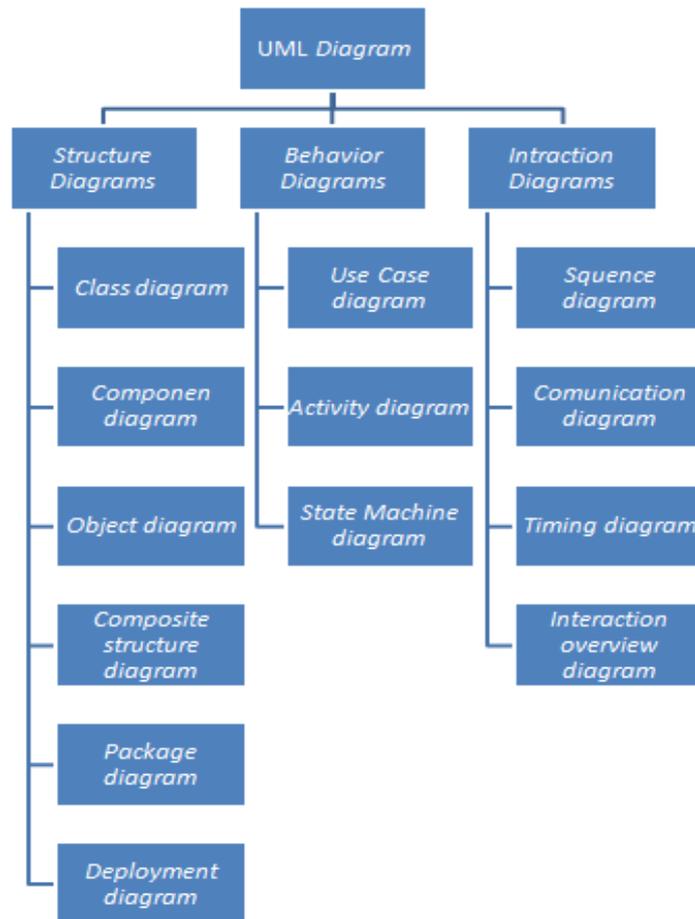
### 2.1.9 UML

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 (Rosa & Shalahuddin, 2013: 133). 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 (Rosa & Shalahuddin, 2013: 136). *Unified Modeling Language* (UML) adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG (*Object Management Group*), saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait didalamnya (Rosa & Shalahuddin, 2013: 140).



**Gambar 2.3** Diagram UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut:

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Pada penulisan skripsi ini penulis hanya akan membahas mengenai:

1. *Use Case diagram*
2. *Activity diagram*, dan
3. *Class diagram*
4. *Sequence diagram*

Berikut ini adalah penjelasan secara terperinci dari keempat diagram UML:

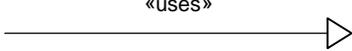
### **2.1.9.1 Use Case Diagram**

*Use case diagram* 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 (Rosa & Shalahuddin, 2013: 155).

Syarat penamaan pada *use case* adalah nama didefinisikan dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*: 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. *Use case* merupakan fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Rosa & Shalahuddin, 2013: 155).

Berikut adalah simbol-simbol yang ada pada diagram *use case*: (Rosa & Shalahuddin, 2013: 156-158)

**Tabel 2.2** Simbol–simbol yang ada pada diagram *use case*

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>&lt;&lt;extend&gt;&gt;</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 pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya :</p>
<p>Menggunakan <i>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>
<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>

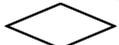
### 2.1.9.2 Activity Diagram

Diagram aktivitas atau *activity diagram* 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 (Rosa & Shalahuddin, 2013: 161). Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal sebagai berikut: (Rosa & Shalahuddin, 2013: 161-162)

1. Rancangan proses bisnis dimana setiap aktivitas 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.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol–simbol yang ada pada diagram aktivitas (Rosa & Shalahuddin, 2013: 162-163):

**Tabel 2.3** Simbol–simbol yang ada pada diagram aktivitas

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			
<i>Swimlane</i> <table border="1" data-bbox="316 1193 539 1310"> <tr> <td>Nama</td> </tr> <tr> <td><i>Swimlane</i></td> </tr> <tr> <td> </td> </tr> </table>	Nama	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
Nama				
<i>Swimlane</i>				

### 2.1.9.3 Class diagram

*Class diagram* atau klas diagram menggambarkan struktur sistem dari segi pendefinisian kelas–kelas yang akan dibuat untuk membangun sistem (Rosa & Shalahuddin, 2013: 141).

Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- a) Atribut merupakan variabel-variabel 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*

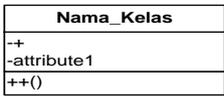
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*

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 *class diagram*: (Rosa & Shalahuddin, 2013: 141-147)

**Tabel 2.4** Simbol-simbol yang ada pada *class diagram*

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 pemrograman berorientasi objek
<p>Asosiasi (Association) </p>	Relasi antar kelas dengan aksna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Lanjutan Tabel 2.4 Simbol–simbol yang ada pada *class diagram*

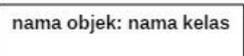
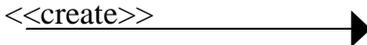
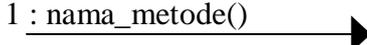
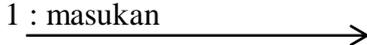
Simbol	Deskripsi
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antara kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan ( <i>dependency</i> ) 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi (Aggregation) 	Relasi antar kelas dengan makna semua bagian ( <i>Whole-part</i> )

#### 2.1.9.4 *Sequence Diagram*

Diagram Sekuen 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 (Rosa & Shalahuddin, 2013: 165).

Berikut ini adalah simbol-simbol yang ada pada diagram sekuen: (Rosa & Shalahuddin, 2013: 165-167)

**Tabel 2.5** Simbol–simbol yang ada pada diagram sekuensial

Simbol	Deskripsi
<p>Aktor/<i>actor</i></p>  <p>nama aktor</p>	<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>
<p>Garis hidup/<i>lifeline</i></p> 	<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>
<p>Objek</p>  <p>nama objek: nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe <i>create</i></p> <p>&lt;&lt;create&gt;&gt;</p> 	<p>Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat</p>
<p>pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.</p>
<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju</p>

### 2.1.10 Web

*Web* adalah sebuah penyebaran informasi melalui internet. Sebenarnya antara *www* (*world wide web*) dan *web* adalah sama karena kebanyakan orang menyingkat *www* menjadi *web* saja. *Web* merupakan hal yang tidak dapat dipisahkan dari dunia internet. Melalui *web*, setiap pemakai internet bisa mengakses informasi-informasi di situs *web* yang tidak hanya berupa *teks*, tetapi juga dapat berupa gambar, suara, film, animasi, dll. Sebenarnya, *web* merupakan kumpulan-kumpulan dokumen yang banyak tersebar di beberapa komputer *server* yang berada di seluruh penjuru dunia dan terhubung menjadi satu jaringan melalui jaringan yang disebut *internet* (Anisya & Wandyra, 2016: 14).

Pada dasarnya *web* merupakan suatu kumpulan *hyperlink* yang menuju dari alamat satu ke alamat lainnya dengan bahasa HTML (*Hypertext Markup Language*). *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebut *hyperlink*, sedangkan *teks* yang dijadikan media penghubung disebut *hypertext* (Batubara, 2012: 16-17).

### 2.1.11 Aliran Sistem Informasi

Aliran sistem informasi sangat berguna untuk mengetahui permasalahan yang ada pada suatu sistem. Dari sini dapat diketahui apakah system informasi tersebut masih layak dipakai atau tidak, masih manual atau komputerisasi. Jika sistem informasinya tidak layak lagi maka perlu adanya perubahan dalam pengolahan datanya sehingga menghasilkan informasi yang cepat dan akurat serta keputusan yang lebih baik. Berikut simbol-simbol dari Aliran Sistem Informasi (ASI) (Zefriyenni & Santoso, 2015: 26):

**Tabel 2.6** Simbol-simbol Aliran Sistem Informasi

No	Simbol	Keterangan
1		Proses komputer
2		Proses manual
3		Dokumen
4		Arsip
5		Penghubung
6		Arus data

## 2.2 Teori Khusus

Adapun teori umum yang digunakan dalam penelitian ini sebagai berikut:

### 2.2.1 *Hypertext Markup Language (HTML)*

Menurut Jayan (2012) dalam (Sagita & Sugiarto, 2016: 50) mengemukakan bahwa "hyper text markup language (HTML) merupakan bahasa". Sedangkan menurut Prasetio (2014) dalam (Sagita & Sugiarto, 2016: 50) mengemukakan bahwa "HTML (hypertext markup language) adalah bahasa pemrograman yang digunakan untuk mendesain sebuah halaman". Sebagian besar dokumen yang dilihat dalam web adalah dokumen HTML. Karena HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *world wide web consorsium* (W3C). Semua tag HTML diapit oleh < > (*bracket*) dan tidak *case sensitive* (huruf besar dan kecil dibedakan). Jadi tag, <center> adalah sama dengan penulisan <CENTER>.

*HTML* adalah sebuah bahasa *markup* yang digunakan untuk membuat halaman *web* dan menampilkan berbagai informasi di dalam sebuah *browser* internet. Berikut ini merupakan contoh ringkas sebuah *script HTML*:

```
<.html>
```

```
<.html>
```

```
<head><title>Contoh Sederhana</title></head>
```

```
</html>
```

Secara umum *script HTML* terbagi atas dua bagian utama yaitu bagian *header* dan *body*. Kedua bagian ini akan di letakkan antara dua *symbol tag* (`<html>...</html>`) untuk `<head>` dan di akhiri dengan *symbol tag* `</head>` (Anisya & Wandyra, 2016: 14).

### 2.2.2.1 PHP (*Hypertext Preprocessor*)

Menurut Sibero (2013) dalam (Sagita & Sugiarto, 2016: 50) mengemukakan bahwa “PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”. PHP merupakan singkatan dari *Hypertext Preprocessor*, dan merupakan bahasa yang Adapun *PHP* dalah singkatan dari *personal home page*, merupakan bahasa skrip yang sangat populer dan banyak dipakai oleh *programmer web* dalam pembuatan sistus yang dinamis disertakan dalam dokumen HTML sekaligus bekerja di sisi *server* (*server-side HTML embedded scripting*)”. Beberapa kelebihan dari PHP menurut Saputra dan Agustin (2012:4) dalam (Sagita & Sugiarto, 2016: 51) antara lain:

- a. Mudah dipelajari.
- b. Mampu lintas *platform*.
- c. *Free* (gratis) bersifat *open source*
- d. PHP memiliki tingkat akses yang cepat
- e. Didukung oleh beberapa macam *web server*.
- f. Mendukung *database*.

PHP merupakan sebuah bahasa *scripting* yang di bundle dengan HTML yang di jalankan jaringan disisi server. Sebagian besar perintahnya berasal dari

bahasa C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP (Anisya & Wandrya, 2016: 14).

### **2.2.3 Cascading Style Sheet (CSS)**

Menurut Saputra dan Agustin (2012:5) dalam (Sagita & Sugiarto, 2016: 51) mengemukakan bahwa “CSS atau yang memiliki kepanjangan *Cascading Style Sheet*, merupakan suatu bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* akan lebih rapi, terstruktur, dan seragam”. Sedangkan menurut Prasetio (2014:252) dalam (Sagita & Sugiarto, 2016: 51) mengemukakan bahwa “CSS adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman website (situs)”.

CSS adalah kependekan dari Cascading Style Sheet. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/*layout* halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa markup seperti HTML, XHTML, XML, SVG (Scalable Vector Graphics) dan Mozilla XUL (XML User Interface Language) (Noor, 2016: 78).

#### 2.2.4 JavaScript

Menurut Sidik dalam Prasetio (2014:292) dalam (Sagita & Sugiarto, 2016: 51) mengemukakan bahwa “*Javascript* adalah program dalam bentuk *script* yang dijalankan oleh *interpreter* yang telah ditanamkan kedalam *browser web*, sehingga *browser web* dapat mengeksekusi program *javascript*”.

#### 2.2.5 CodeIgniter

*CodeIgniter* adalah *framework* aplikasi *web* yang *open source* untuk bahasa pemrograman PHP. *CodeIgniter* memiliki banyak fitur yang membuatnya berbeda dengan *framework* lainnya. Tidak seperti beberapa *framework PHP* lainnya, dokumentasi untuk *framework* ini sangat lengkap, yang mencakup seluruh aspek dalam *framework*. *CodeIgniter* juga mampu berjalan pada lingkungan *Shared Hosting* karena memiliki ukuran yang sangat kecil, namun memiliki kinerja yang sangat luar biasa. Dari segi pemrograman, *CodeIgniter* kompatibel dengan PHP4 dan PHP5, sehingga akan berjalan dengan baik pada *web host* yang banyak dipakai saat ini. *CodeIgniter* menggunakan pola *design Model- View-Controllwe* (*MVC*), yang merupakan cara untuk mengatur aplikasi web ke dalam tiga bagian yang berbeda, yaitu *Model*- lapisan abstraksi *database*, *Views*- file-file tampilan template depan, dan *Controller*- logika bisnis dari aplikasi (Setiadi & Alfiah, 2016: 578).

Pada intinya *CodeIgniter* juga membuat penggunaan ekstensif dari pola *design Singleton*. Maksudnya adalah cara untuk me-load *class* sehingga jika *class* itu di panggil dalam beberapa kali, kejadian yang sama pada *class* tersebut akan

digunakan kembali. Hal ini sangat berguna dalam koneksi *database*, karena kita hanya ingin menggunakan satu koneksi setiap kali class itu digunakan. *CodeIgniter* dikembangkan oleh *Rick Ellis*, dengan versi awal yang dirilis pada tanggal 28 februari 2006. Dari tahun itulah hingga sekarang telah muncul banyak versi *CodeIgniter* yang terus berkembang dengan penambahan fitur baru dari sebelumnya (Setiadi & Alfiah, 2016: 578).

### **2.2.6 MVC**

*Model View Controller* (MVC) pertama kali dipublikasikan oleh peneliti *XEROX PARAC* yang bekerja dalam pembuatan bahasa pemrograman *Smalltalk* sekitar tahun 1970-1980. *MVC* adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasinya kebanyakan *framework* dalam aplikasi *website* adalah berbasis arsitektur MVC (Hanifah, Nurma, & Dewi, 2015:23).

*Model-View-Control* (MVC) pertama kali diperkenalkan peneliti *Xerox PARC* yang bekerja pada bahasa pemrograman *Smalltalk* di akhir 1970-an dan awal 1980-an. *Smalltalk* adalah bahasa pemrograman yang berorientasi objek, bertipe dinamis, dan reflektif. *Smalltalk* pertama kali digunakan dalam pembelajaran edukasi dan hal ini berbeda dari data mainframe dan struktur control dalam program *Smalltalk* yang terlibat pada *Windowed User Interfaces*, konsep pemrograman berorientasi objek, pengantar pesan antara komponen-komponen objek, dan kemampuan untuk memonitor dan memodifikasi struktur dan perilakunya sendiri. *MVC* merupakan sebuah pattern atau teknik pemrograman

yang memisahkan antar pengembang aplikasi berdasarkan komponen utama pada sebuah aplikasi, seperti manipulasi data, user interface dan bagian yang menjadi kontrol aplikasi. Secara sederhana dapat dikatakan bahwa antara desain dan proses data berada pada tempat yang terpisah. Saat ini *MVC* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi *web*, yang berawal pada bahasa *Smalltalk* (Setiadi & Alfiah, 2016: 579).

### **2.2.7 Business Intelligence Reporting Tool (BIRT)**

*Business Intelligence Reporting Tool (BIRT)* adalah sebuah perangkat lunak *open source* dilengkapi dengan *platform* teknologi *BIRT* untuk membuat visualisasi data dan laporan yang bisa digabungkan ke *client* dan *web application*, terutama yang berbasis pada *java* dan *java enterprise edition*. *BIRT* memiliki dua komponen utama yaitu sebuah *visual report designer* untuk membuat *design BIRT*, dan *runtime component* digunakan untuk *men-generate design* yang dapat digunakan oleh banyak *java environment*. Pada *BIRT project* telah memiliki *charting engine* yang telah terintegrasi pada *BIRT design* dan dapat digunakan *standalone* untuk mengintegrasikan *charts* ke aplikasi. *Design BIRT* yang merupakan *XML* dan dapat diakses oleh berbagai sumber data yang berbeda termasuk *JDO datastores*, *JFire Scripting Objects*, *POJOs*, *database SQL*, *webservice* dan *XML* (Hanifah et al., 2015:24).

### 2.2.8 JQuery

Menurut Saputra dan Agustin (2012:7) dalam (Sagita & Sugiarto, 2016: 51) mengemukakan bahwa “*JQuery* merupakan salah satu teknik atau kumpulan *library javascript* yang sangat terkenal dengan animasinya. Dengan sedikit sentuhan, animasi dalam *website* mudah kita ciptakan. *JQuery* dapat dianggap sebagai *frameworknya javascript*”.

*JQuery* adalah *framework php* yang memudahkan penerapan *JavaScript*. Dengan *JQuery*, anda tidak perlu melakukan banyak pengetikan kode secara manual. Hanya dengan sedikit pengetikan, anda bisa melakukan banyak hal. Itulah kegunaan dari *framework Javascript JQuery* telah lama menjadi *library JavaScript* yang populer untuk menciptakan website interaktif yang kaya serta bisa dipakai juga untuk aplikasi *web*. Namun, karena dirancang untuk browser desktop, *JQuery* tidak memiliki banyak fitur yang khusus dirancang untuk membangun aplikasi *web mobile* (Noor, 2016: 78).

*JQuery* adalah sebuah *framework* berbasis *Javascript*. *JQuery* sama dengan *Javascript Library* yaitu kumpulan kode atau fungsi *Javascript* siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *Javascript*. Hal yang menarik dari *JQuery* adalah penekanan interaksi antara *javascript* dan *HTML* (Warman & Zahni, 2013: 33).

Beberapa kemampuan yang dimiliki oleh JQuery sebagai berikut:

1. Memanipulasi elemen HTML

2. Memanipulasi CSS
3. Penanganan event HTML
4. Efek-efek *javascript* dan animasi
5. Modifikasi HTML DOM
6. AJAX

Kelebihan dari *Jquery* yakni *Writeless, do more* yaitu menyederhanakan penggunaan *javascript* yang ada, karena kita cukup menggunakan fungsi library *javascript* yang ada, juga mempercepat *coding javascript* dalam sebuah *website*, dibandingkan kita harus memulai sebuah *script javascript* satu persatu. *JQuery Selector* adalah fungsi utama pada *jQuery*. Semua fungsi di *jQuery* dapat diakses melalui *selector*. Penggunaan paling dasarnya adalah mempassingkan sebuah ekspresi yang kemudian selanjutnya *jQuery* akan mencari elemen yang cocok. Pada intinya ini adalah fungsi untuk memilih elemen-elemen pada halaman *web* (Warman & Zahni, 2013: 33).

### **2.2.9 Notepad++**

Notepad++ adalah program aplikasi pengembang yang berguna untuk mengedit *teks* dan skrip kode pemrograman. Versi terbaru program ini adalah *Notepad++ v5.9*, yang dirilis pada tanggal 06 April 2012. *Software Notepad++* dibuat dan dikembangkan oleh Tim *Notepad++*. Perangkat lunak komputer ini memiliki kelebihan pada peningkatan kemampuan sebuah program text editor, lebih dari sekedar program *Notepad* bawaan *Windows*. *Notepad++* bisa mengenal tag dan kode dalam berbagai bahasa pemrograman. Fitur pencarian tingkat lanjut

dan pengeditan teks yang tersedia juga cukup ampuh, sangat membantu tugas seorang *programmer* atau *developer* dalam menyelesaikan skrip kode programnya (Noor, 2016: 78).

Program *Notepad++* banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman aplikasi desktop dan web. *Notepad++* merupakan software gratis (*opensource*). *Notepad++* dapat dijalankan di sistem operasi *Win2K*, *Windows XP*, *Vista*, dan *Windows 7*. Untuk menginstal versi terbaru program ini, komputer *Windows* Anda cukup memiliki kapasitas kosong harddisk minimal 12 MB (Noor, 2016: 78).

#### **2.2.10 XAMPP**

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* (Dadan, 2015: 27).

*XAMPP* adalah buatan dari *team apache friands*. *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)*

### **2.2.11 Apache Tomcat**

*Apache Tomcat* merupakan salah satu *servlet/web container* paling populer dilingkungan pemrograman *web java* yang bernaung dibawah *apache software foundation* yang terdapat *project-project open source* lainnya (Khadafi, Away, & Nasaruddin, 2017:22).

### **2.2.12 PostgreSQL**

*PostgreSQL* merupakan sebuah *Object-Relational Database Management System (ORDBMS)* yang *open source*. *PostgreSQL* merupakan DBMS yang dapat menjalankan bahasa SQL dan mempunyai kelebihan fitur – fitur modern seperti, *Complex Queries, Foreign Keys, Triggers, View, Transactional Integrity, Multiversion Concurrency Control*. Untuk memperoleh *postgresql* dapat mendownload pada alamat berikut <http://www.postgresql.org/download/>. Pada saat anda menginstall *postgresql* anda akan diminta untuk mengisi

*password* yang nantinya digunakan seterusnya pada saat menggunakan *postgresql* (Hakim, 2017: 19).

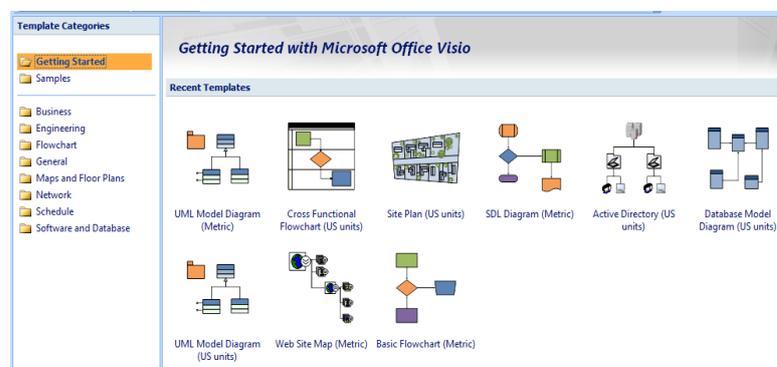
### 2.2.13 Ms. Visio 2007

*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 (Sutomo, 2009: 1).



**Gambar 2.4** Menu bar pada *Ms.Visio 2007*

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 (Sutomo, 2009: 2).



**Gambar 2.5** Get Started pada visio 2007

### 2.2.14 *Black box*

*Black box* adalah suatu metode ujicoba pendekatan yang melengkapi untuk menemukan kesalahan lainnya yang memfokuskan pada keperluan fungsional dari sistem, karena ujicoba *black box* memungkinkan pengembang sistem atau aplikasi untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya: fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *database* eksternal (jika ada), kesalahan performa, serta kesalahan inisialisasi dan terminasi (Pare, 2013: 226).

## 2.3 Penelitian terdahulu

Berikut ini adalah penelitian terdahulu yang digunakan sebagai acuan peneliti:

**Tabel 2.7** Penelitian terdahulu

No	Nama Penulis	Judul Jurnal	Kesimpulan
1	Lilis maryani, Asep deddy supriatna, Erwin Gunadhi, (2012, ISSN: 2302- 7339)	Perancangan sistem informasi persediaan barang masuk dan barang keluar studi kasus di PD Sumber Sayur	1. Dapat melihat persediaan barang tanpa batas waktu. 2. Mempermudah dan mempercepat kinerja petugas bagian admin. 3. Dapat menginformasikan

Lanjutan dari tabel 2.7 penelitian terdahulu

No	Nama Penulis	Judul Jurnal	Kesimpulan
			data stok, data supplier dengan waktu yang tidak ditentukan (Maryani, Supriatna, & Gunadhi, 2012: 6)
2	Khabbazi M.R, Hasan M.K, Shapi'I A, Sulaiman R, Taei-Zadeh A, (2013, ISSN: 1992-8645)	Inventory system and functionality evaluation for production logistics	1. Sistem informasi terpadu. 2. Dapat melacak data dari pembelian sampai pengiriman. 3. Meningkatkan visibilitas pengambilan data (Khabbazi, Hasan, Shapi'i, Sulaiman, & Taei-Zadeh, 2013:386)
3	Adysta Rahadi, Mochammad Al Musadieq, Heru Susilo, (2014, VOL 8 NO 2)	Analisis dan desain sistem informasi persediaan barang berbasis komputer (studi kasus pada toko Arta Boga)	1. Dapat mengelola data informasi barang dengan cepat. 2. Memberikasn informasi barang yang akurat. 3. Mengurangi biaya operasional penyimpanan arsip

Lanjutan dari tabel 2.7 penelitian terdahulu

No	Nama Penulis	Judul Jurnal	Kesimpulan
			4. Pemenuhan laporan tentang barang bisa tersedia dengan cepat (Rahadi et al., 2014: 7)
4	Novianti Madhona Faizah dan Nina Amelia, (2016, ISSN: 1978- 001X)	Perancangan aplikasi sistem persediaan sembako pada toko harapan baru	Aplikasi inventory dapat membantu proses pendataan barang, masuk maupun keluar berbasis komputer yang sistematis dan terarah, sehingga data yang diperoleh lebih tepat dan akurat (Faizah & Amelia, 2016: 100).
5	Ahamd Yusuf Bakhtiar dan Bunyamin, (2016, ISSN: 2302-7339)	Pengembangan sistem informasi aplikasi persediaan barang masuk dan barang keluar dengan menggunakan PHP Framework di Nafisa production	Perancangan sistem informasi persediaan barang dapat membantu kinerja di nafisa production dalam pengolahan data dan dapat membuat laporan persediaan barang, dan dapat mempermudah dalam pengolahan data persediaan barang sehingga kinerja lebih cepat

Lanjutan dari tabel 2.7 penelitian terdahulu

No	Nama Penulis	Judul Jurnal	Kesimpulan
			(Bakhtiar & Bunyamin, 2015)
6	Samuel indra hermawan, Qorinta shinta, dan migunani, (2016, ISSN: 2087-0868)	Rancang bangun sistem informasi persediaan barang pada toko cahaya baru semarang	Sistem informasi persediaan barang memudahkan pencatatan stok dan memudahkan pembuatan laporan persediaan (Hermawan et al., 2016: 23)