

BAB II

LANDASAN TEORI

2.1 Kajian Pustaka

Penelitian mahasiswa Universitas Widyatama melakukan pengembangan aplikasi penjualan berbasis web menggunakan *Content Management System* (CMS) Joomla. Penelitian tersebut bertujuan untuk memberikan kemudahan kepada konsumen dalam mendapatkan informasi maupun dalam membeli produk yang dijual oleh Mutiara *Fashion*, membantu Mutiara *Fashion* dalam mempromosikan dan menjual produk kepada konsumen, dan memberikan kemudahan dalam pengolahan data transaksi serta pembuatan laporan penjualan. Penelitian tersebut adalah penelitian pengembangan atau *Research and Development* (R&D). instrument pengumpulan data terdiri dari lembar penilaian kualitas website menggunakan metode *blackbox* (Ginting, 2014).

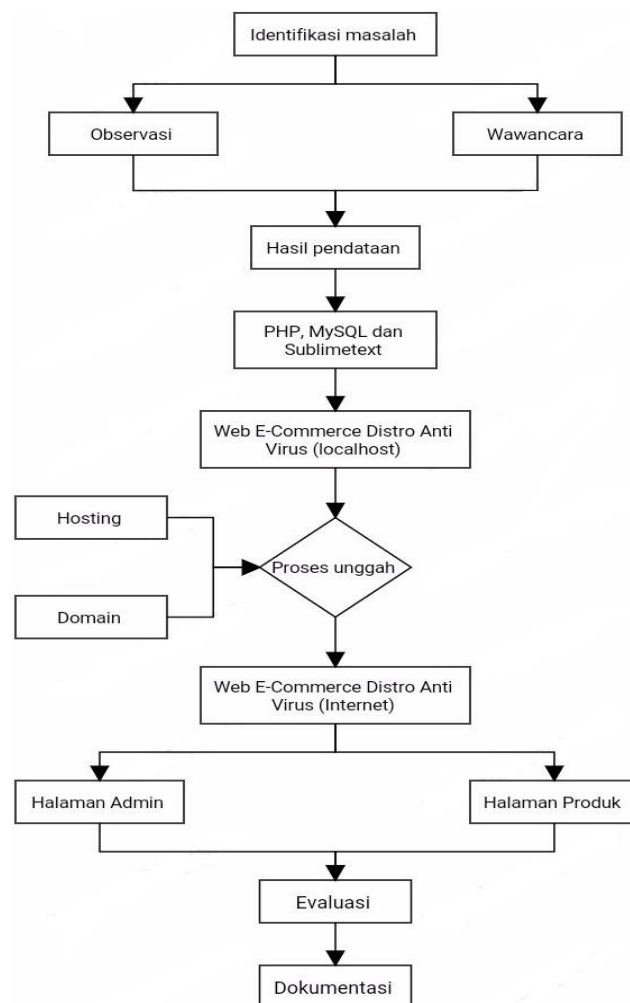
Penelitian ini merupakan penelitian mahasiswa Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta pada tahun 2013. Dalam penelitian tersebut dilatar belakangi kurang berkembangnya toko Kids Corner karena belum memiliki website toko online, sehingga dibuatlah toko online perlengkapan bayi dan anak berbasis web dengan menggunakan framework CakePHP. Sistem tersebut merupakan suatu sistem yang berbasis web dan mobile android yang dapat mengelola data penjualan dari sebuah toko perlengkapan bayi dan anak Kids Corner Yogyakarta (Nucifera, 2013).

Penelitian yang dilakukan Sukma (2012) Analisis faktor-faktor yang mempengaruhi keputusan pembelian melalui social networking websites. yang hasilnya Secara parsial faktor *trust*, *quality of service*, dan *perceived risk* berpengaruh signifikan terhadap keputusan pembelian melalui *social networking websites*, sedangkan faktor *security*, menunjukkan hasil yang tidak signifikan. Sementara secara simultan faktor *trust*, *security*, *quality of service*, dan *perceived risk* berpengaruh signifikan terhadap keputusan pembelian melalui *social networking websites*.

Penelitian yang dilakukan di Toko Distro Anti Virus Boyolali pada tahun 2019. Dalam penelitian tersebut dilatar belakangi kurang berkembangnya toko Distro Anti Virus karena belum memiliki *website* toko online, sehingga dibuatlah toko online pakaian dan sepatu berbasis *web* dengan menggunakan *framework* *CakePHP*. Sistem pembayaran melalui rekening dapat mempercepat dan mempermudah pembeli dalam melakukan pembayaran dan dapat mempermudah admin toko untuk selalu update produk terbarunya, dengan dibuatnya toko online harapan kedepannya bagi Toko Distro Anti Virus semakin mendapatkan lebih banyak pelanggan.

2.2 Kerangka Pemikiran

Kerangka Pemikiran dari penelitian tugas akhir ini terdapat pada Gambar 2.1



Gambar 2.1 Kerangka Pemikiran

Penjelasan Gambar 2.1 adalah

1. Mengidentifikasi masalah pada toko distro anti virus untuk mendapatkan data dan informasi yang dibutuhkan dalam membangun *e-commerce*.
2. Observasi dilakukan dalam mencari informasi tentang kegiatan yang berlangsung di toko distro.
3. Wawancara dilakukan untuk mengumpulkan data dan informasi yang dibutuhkan dalam pembuatan sistem dan pemanfaatan sistem. Wawancara dilakukan pada Bapak Iwan selaku admin distro, dari wawancara tersebut dapat mengetahui kelebihan dan kekurangan apa saja yang berada di distro tersebut dan upaya apa kedepannya untuk memperbaiki agar lebih baik lagi.
4. Hasil pendataan berupa data yang atau informasi yang sudah didapat melalui identifikasi masalah, observasi dan wawancara, kemudian dibuat menggunakan *tools PHP, MySQL* dan *Sublime Text*.
5. *Web e-commerce* distro anti virus merupakan hasil program dari *localhost* yang terdapat halaman admin dan pembeli.
6. Proses unggah untuk di hosting ke internet supaya mudah di akses oleh pembeli, yang terdapat halaman admin dan pembeli.
7. Evaluasi merupakan proses penilaian *e-commerce* agar rencana yang telah dibuat untuk mencapai tujuan yang ditetapkan dapat terselenggara.
8. Dokumentasi dilakukan untuk memperoleh file-file yang diperlukan dalam penyusunan Laporan Tugas Akhir seperti gambar, dan arsip data sesuai kebutuhan untuk mempercepat penyusunan Laporan Tugas Akhir.

2.3 Landasan Teori

Penyusunan tugas akhir ini diperlukan landasan teori yang memiliki relevansi dengan masalah yang akan dibahas. Landasan teori tersebut diperoleh dari beberapa literatur yang mempublikasikan pendapat beberapa ilmuwan yang dipakai sebagai penunjang pembahasan masalah.

2.3.1 *E-commerce*

Elektronik *commerce* atau biasa disingkat dengan *e-commerce* merupakan suatu bentuk dari pembelian, penjualan, pemasaran barang dan jasa yang dilakukan melalui media internet. Industri teknologi informasi melihat *e-commerce* sebagai bentuk aplikasi dan penerapan dari *e-business* yang berkaitan dengan transaksi perdagangan, seperti tranfer dana secara elektronik, pemasaran elektronik atau pemaaran online, pemrosesan transaksi *online*, pertukaran data elektronik dan lain sebagainya. Dengan kata lain, *e-commerce* sama halnya dengan *e-busimess*, dimana cakupannya dalam mendapatkan konsumen akan lebih luas.

Media sosial mempunyai peranan yang penting di dalam dunia bisnis khususnya dibidang pemasaran dan menjadi salah satu media yang dapat dengan sangat cepat menyebarkan sebuah informasi. Media sosial tidak hanya menjadi sebuah tempat untuk menunjukkan diri atau mengekspresikan diri terhadap dunia luar, namun juga sebagai tempat untuk mendapatkan banyak informasi. Berbagai macam media sosial yang muncul saat ini dimanfaatkan konsumen untuk mendapatkan informasi (Sugianto, P. C, 2016).

2.3.2 Model Transaksi *E-Commerce*

Model Transaksi *E-commerce* merupakan aktivitas pembelian dan penjualan melalui jaringan internet dimana pembeli dan penjual tidak bertemu secara langsung, melainkan berkomunikasi melalui media internet. *E-commece* memiliki berbagai macam jenis transaksi dalam menerapkan sistemnya (Ahmadi, 2013). Jenis-jenis transaksi *E-commerce* di antaranya sebagai berikut:

1. *Collaborative Commerce (C-Commerce)*

Collaborative Commerce yaitu kerjasama secara elektronik antara rekan bisnis. Kerjasama ini biasanya terjadi antara rekan bisnis yang berada pada jalur penyediaan barang (*Supply Chain*).

2. *Business to Business (B2B)*

E-Commerce tipe ini meliputi transaksi antar organisasi yang dilakukan di electronic market. *Business to Business* memiliki karakteristik:

- a. *Trading partners* yang sudah diketahui dan umumnya memiliki hubungan yang cukup lama. Informasi hanya dipertukarkan dengan partner tersebut. Dikarenakan sudah mengenal rekan komunikasi, jenis informasi yang dikirimkan dapat disusun sesuai dengan kebutuhan dan kepercayaan.
- b. Pertukaran data berlangsung berulang-ulang dan secara berkala, misalnya setiap hari dengan format data yang sudah disepakati bersama. Dengan kata lain, servis yang digunakan sudah tertentu. Hal ini memudahkan pertukaran data untuk dua *entity* yang menggunakan standar yang sama.
- c. Salah satu pelaku dapat melakukan inisiatif untuk mengirimkan data, tidak harus menunggu partnernya.
- d. Model yang umum digunakan adalah *peer to peer*, dimana *processing intelligence* dapat didistribusikan pada pelaku bisnis.

3. *Business to Consumers (B2C)*

Business to Consumers yaitu penjual adalah suatu organisasi dan pembeli adalah individu. *Business to Consumers* memiliki karakteristik sebagai berikut:

- a. Terbuka untuk umum, dimana informasi disebar ke umum.
- b. *Service* yang diberikan bersifat umum (*generic*).
- c. *Service* diberikan berdasarkan permohonan. *Consumer* melakukan inisiatif dan produsen harus memberikan respon sesuai permohonan.

4. *Consumer to Business (C2B)*

Dalam *Consumer to Business* memberitahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen.

5. *Costumer to Customer (C2C)*

Costumer to Costumer yaitu konsumen menjual secara langsung ke konsumen lain atau mengiklankan jasa pribadi di internet. Dalam *Costumer to Costumer* *costumer* menjual produk atau jasa ke *costumer* lain. Dapat juga disebut sebagai pelanggan ke pelanggan, yaitu orang yang menjual produk dan jasa ke satu sama lain.

2.3.3 Bahasa Pemrograman

2.3.3.1 HTML

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi didalam sebuah penjelajah web internet dan *formatting hypertext* sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasikan (Madcoms, 2016).

Berikut merupakan contoh program html disajikan pada Gambar 2.2.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>daftar</title>
5      <link rel="stylesheet" href="admin/assets/css/bootstrap.css">
6  </head>
7  <body>
8
9  <div class="container">
10     <div class="row">
11         <div class="col-md-8">
12             <div class="panel panel-default">
13                 <div class="panel-heading">
14                     <h3 class="panel-title">Daftar Pelanggan</h3>
15                 </div>
16                 <div class="panel-body">
17                     NUB
18                 </div>
19             </div>
20         </div>
21     </div>
22 </div>
23 </body>
24 </html>

```

Gambar 2.2 Program HTML

2.3.3.2 PHP

PHP (*HyperText Processor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs dinamis, PHP sering juga digunakan untuk membangun sebuah CMS (Madcoms, 2016).

Hubungan PHP dan *CodeIgniter* memudahkan bagi programmer untuk merancang sebuah aplikasi berbasis *website*, *tools* ini dapat meminimalkan penulisan *script* yang dilakukan secara manual karena dilengkapi *library* yang cukup lengkap untuk keperluan pembuatan *website*. Dengan demikian programmer dapat memfokuskan perhatian pada sisi perancangan.

Berikut contoh program PHP disajikan pada Gambar 2.3.

```
1 <?php
2 session_start();
3
4 //menghancurkan $_SESSION["pelanggan"]
5 session_destroy();
6
7 echo "<script>alert('anda telah logout');</script>";
8 echo "<script>location='index.php';</script>";
9 ?>
```

Gambar 2.3. Program PHP

2.3.3.3 CodeIgniter

CodeIgniter adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para programmer *web* untuk membuat atau mengembangkan aplikasi berbasis *web* (Yenda, 2016).

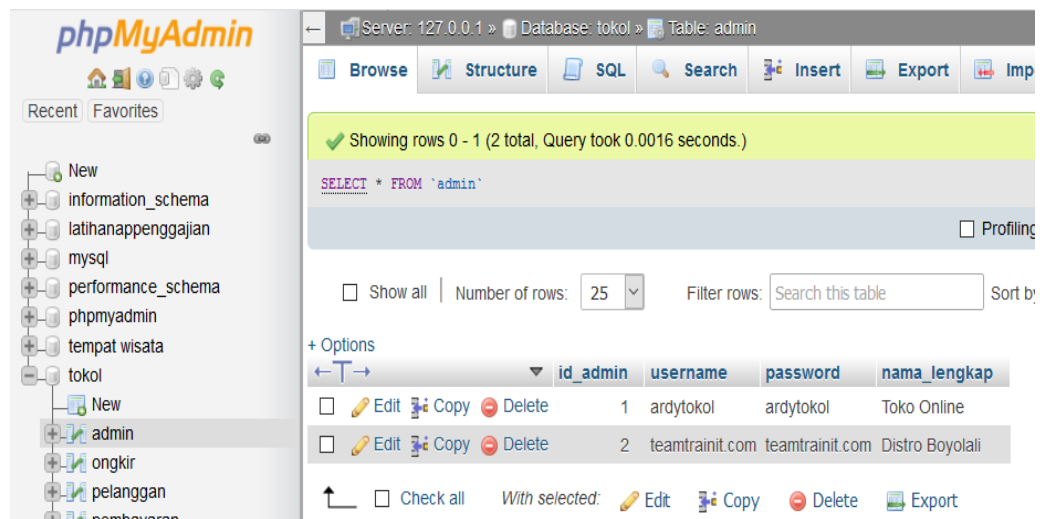
CodeIgniter adalah sebuah *framework* PHP yang kecil, sederhana, mudah digunakan dan cukup handal untuk membangun aplikasi berbasis *website* (Basuki, 2017).

2.3.4 Database

2.3.4.1 Pengertian Database

Database adalah sekumpulan file yang saling berhubungan dan terorganisasi atau kumpulan record-record yang menyimpan data dan hubungan diantaranya. (Sutarman, 2012). Database adalah sekumpulan data store (biasa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya (Ladjuman, 2016).

Berikut contoh *Database* disajikan pada Gambar 2.4.



Gambar 2.4. Database

2.3.4.2 MySQL

MySQL adalah sistem manajemen *Database SQL* yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database MySQL* mendukung beberapa fitur seperti *multithreaded*, *multi-user* dan *SQL Database management system* (DBMS). Berbeda dengan basis data konvensional seperti *.dat*, *.dbf*, *.mbf*, *MySQL* memiliki kelebihan yaitu bersifat *multithread*, dan *multi user* serta mendukung sistem jaringan. *MySQL* di distribusikan secara gratis dibawah *lisensi GNU General Public License* (GPL), namun ada juga versi komersial bagi kalangan tertentu yang menginginkan (Madcoms, 2016).

2.3.5 UML

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Utami, 2015).

UML (*Unified Modeling Language*) merupakan konsolidasi dari praktek-praktek terbaik yang telah ditetapkan selama bertahun-tahun dalam penggunaan bahasa pemodelan. UML memungkinkan kita untuk menyajikan aspek yang sangat beragam dari sistem perangkat lunak dalam kerangka tunggal dengan menggunakan konsep *object-oriented* (Seidl, et. al, 2015).

2.3.6 Sublime Text

Sublime text merupakan perangkat lunak text editor yang di gunakan untuk membuat atau meng-edit suatu aplikasi. *Sublime text* mempunyai fitur *plugin* tambahan yang memudahkan programmer (Supono, 2016).

Berikut contoh *Sublime Text* di sajikan pada Gambar 2.5.

```

1  Project Single
2  \-----*/
3
4  single-project {
5      padding-top: 700px;
6
7      @include breakpoint(m) {
8          padding-top: 550px;
9      }
10
11     @include breakpoint(s) {
12         padding-top: 0;
13     }
14 }
15
16 .single-project .main {
17     padding-top: 90px;
18
19     @include breakpoint(m-down) {
20         padding-top: 60px;
21     }
22 }
23
24 .hero {
25     position: fixed;
26     z-index: -1;
27     top: 0;
28     width: 100%;
29     height: 700px;
30 }
31
32 NORMAL scss/_modules/_single-project.scss
33 scss utf-8[unix] 2% : 5: 1
34
35 [10:52:58] Live reload server listening on: 35729
36 [10:53:00] _single-project.scss was reloaded.
37 [10:53:00] Starting 'sass'...
38 [10:53:00] Finished 'sass' after 20 ms
39 [10:53:00] _single-project.scss was reloaded.
40 [10:53:00] Starting 'sass'...
41 [10:53:00] Finished 'sass' after 2.77 ms
42
43 0:vim* 1: bash-

```

Gambar 2.5 *Sublime Text*






2.3.7 Use Case Diagram

Use case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut. Sehingga pembuatan *use case* diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem (Isa dan Hartawan, 2017).

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Hendini, 2016).

Simbol-simbol yang digunakan dalam *Use Case Diagram* ditunjukkan pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case*

Simbol	Nama	Keterangan
 Actor 1	Aktor	Merupakan Penggunaan dari sistem. Penamaan aktor menggunakan kata benda.
 UseCase 1	<i>Use Case</i>	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> dengan kata kerja.
	Sistem	Menspesifikasi paket yang menampilkan sistem secara terbatas.
	<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.
	Extends	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.

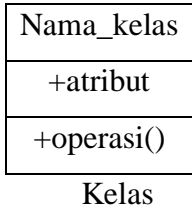
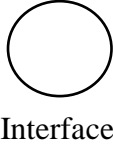
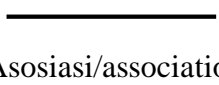
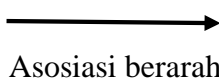
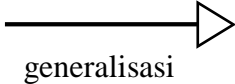


2.2.8 Class Diagram

Class Diagram Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class Diagram secara khas meliputi kelas (*Class*), *relation associations*, *generalitation* dan *aggregation*, *attributes*, *method*, dan *visibility* tingkat akses objek eksternal kepada suatu operasi atau atribut (Hendini, 2016).

Simbol-simbol yang digunakan *Class Diagram* ditunjukkan pada Tabel 2.2

Tabel 2.2 Simbol-simbol *Class Diagram*.


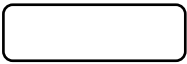
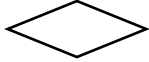
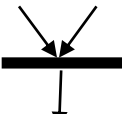

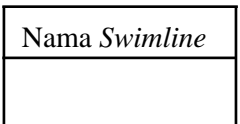
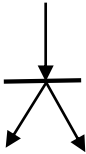
No.	Simbol	Deskripsi
1		Kelas pada struktur sistem.
2		Sama dengan konsep interface dalam pemrograman berorientasi objek.
3		Relasi antar kelas dengan makna umum, asosiasi juga disertai dengan <i>multiplicity</i> .
4		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain, disertai <i>multiplicity</i> .
5		Relasi antar kelas dengan menggunakan makna generalisasi-spesialisasi.
6		Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Relasi antar kelas dengan makna semua-bagian.

2.3.8 Activity Diagram

Activity Diagram Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses (Isa dan Hartawan, 2017).

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* ditunjukkan pada Tabel 2.3.








Tabel 2.3 Simbol-simbol *Activity Diagram* (Hendini, 2016).

N O	GAMBAR	NAMA	KETERANGAN
1		<i>Start</i>	<i>Start point</i> , diletakan pada pojok kiri atas dan merupakan awal aktivitas
2		<i>Activities</i>	Menggambarkan suatu proses kegiatan.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	<i>Join</i> atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
5.		Status akhir	Status akhir yang dilakukan sistem.
6.		<i>Swimlane</i>	Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa
7.		<i>Fork</i>	Percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel untuk menggabungkan dua kegiatan paralel menjadi satu

2.3.9 Sequence Diagram

Sequence Diagram Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence diagram* menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek (Isa dan Hartawan, 2017).

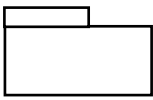



Tabel 2.4 Simbol-simbol *Sequence Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Entity Class</i>	Merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
2.		<i>Boundary Class</i>	Berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>
3.		<i>Control Class</i>	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
4.		<i>Recursive</i>	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
5.		<i>Message</i>	Simbol mengirim pesan antar <i>class</i>
6.		<i>Activation</i>	Mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
7.		<i>Lifeline</i>	Garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

2.3.10 Component Diagram

Component Diagram adalah mempresentasikan dunia rill item yaitu *component software*. *Component software* adalah bagian fisik dari sebuah sistem karena menetap di komputer. *Component diagram* mengandung *component*, *interface* dan *relationship* (Munawar, 2015).

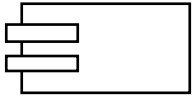
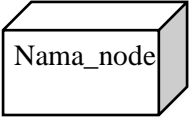

Tabel 2.5 Simbol-simbol *Component Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	Package merupakan sebuah bungkus dari satu atau lebih komponen.
2.		Komponen	Komponen sebuah sistem.
3.		<i>dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
4.		<i>Interface</i>	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antar muka komponen agar tidak mengakses langsung komponen.
5.		<i>link</i>	Relasi antar komponen.

2.3.11 Deployment Diagram

Deployment Diagram adalah salah satu model diagram dalam UML untuk mengerahkan *artifact* dalam node. digunakan untuk memvisualisasikan dan menggambarkan secara detail bagaimana komponen disusun di infrastruktur sistem. Simbol-simbol *deployment diagram* ditunjukkan pada Tabel 2.5

Tabel 2.6 Simbol-simbol *Deployment Diagram* (Hendini, 2016).

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Component</i>	Pada <i>deployment diagram</i> , komponen-komponen yang ada diletakan didalam <i>node</i> untuk memastikan keberadaan posisi mereka
2.		<i>Node</i>	<i>Node</i> menggambarkan bagian <i>hardware</i> dalam sebuah sistem. Notasi untuk <i>node</i> digambarkan sebagai sebuah kubus tiga dimensi
3.		<i>association</i>	Menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara elemen-elemen <i>hardware</i> .

2.3.12 *BlackBox testing*

BlackBox testing adalah tahap yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat dan menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program (Rosa dan Salahuddin, 2015).

BlackBox testing berfokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input dan melakukan pengetesan pada fungsional program (Mustaqbal, 2015).

2.3.13 Rancangan Kuesioner

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Kuesioner dapat berupa pertanyaan tertutup atau terbuka, dapat diberikan kepada responden secara langsung atau dikirim melalui pos atau bisa juga melalui internet. Dalam penelitian ini penulis menggunakan jenis kuesioner tertutup yaitu kuesioner yang dibagikan kepada setiap responden dengan pertanyaan yang mengharapkan jawaban singkat atau responden dapat memilih salah satu jawaban alternatif dari pertanyaan yang disediakan (Sugiyono, 2017).