

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Fanart Shoes merupakan usaha mandiri kreatif berupa layanan *custom* sepatu kekinian yang dibentuk oleh mahasiswi DKV Universitas Sahid Surakarta sejak tahun 2018, Fanart shoes merupakan jasa *custom* sepatu dan pewarnaan ulang sepatu yang dilakukan dengan manual menggunakan cat khusus. Fanart shoes sendiri dibentuk awalnya karena kejenuhan dikala tidak ada kuliah, serta inisiatif pemilik ketika melihat teman-teman mengeluhkan sepatu kesayangannya dengan harga mahal sudah terlihat kusam dan berubah warna, oleh karena itu pemilik *custom* memberikan solusi kepada teman-teman agar tidak membeli sepatu baru ketika sepatu yang lama masih bisa dipakai, hal ini digunakan untuk menginspirasi yaitu “ Dari pada sibuk mencari yang baru, lebih baik yang lama diperbaharui”. Berawal dari inisiatif dan banyak teman-teman yang memesan, akhirnya pemilik *custom* memutuskan untuk membuka jasa *repaint* dan *custom* sepatu bernama Fanart Shoes. Usaha ini sendiri masih bersifat mandiri, untuk pengelolaannya di kerjakan oleh dua orang, satu sebagai marketing, satu lagi sebagai konsep sepatu. Lokasinya di Surakarta, tepatnya di Kos Putri Omah Rejeki.

Adapun dalam pembuatan rancang bangun *online shop* pada toko sepatu Fanart Shoes dengan fitur kustomisasi desain, ada beberapa tinjauan pustaka sebagai dasar dari penelitian ini adalah sebagai berikut:

Penelitian pertama dari Rozak (2015) yang berjudul “Aplikasi Penjualan Sepatu *Online* Toko Sport Bandung Berbasis *Web*”. Permasalahan yang dibahas pada penelitian tersebut adalah tanpa adanya perhitungan biaya kirim, pembayaran melalui paypal. Tool yang digunakan pemrograman *PHP*, *MYSQL*, *XAMPP*, *Macromedia Dreamweaver MX*. Dengan adanya sistem ini dapat membantu proses penjualan *online* pada Toko Sport Bandung dengan menggunakan metode pembayaran *online* Paypal dan dapat pula dilakukan langsung di toko sehingga catatan transaksi tercatat pada *database*.

Penelitian kedua dari Imaniawan dan Elsa (2017) yang berjudul “Sistem Informasi Penjualan Sepatu Berbasis *Web* Pada *Vegas Hyper* Purwokerto”. Toko *Vegas Hyper* merupakan sebuah perusahaan yang bergerak di bidang penjualan sepatu. Dalam melakukan kegiatan penjualan perusahaan ini masih menggunakan proses yang kurang efektif, yaitu konsumen harus datang langsung ke toko jika ingin membeli sepatu. Permasalahan yang dibahas pada penelitian tersebut adalah belum tersedianya media pemasaran *online*.

Penelitian ketiga dari Kirlyana dan Rosyida (2016) yang berjudul “Sistem Informasi Penjualan Sepatu *Handmade* Berbasis *Web*”. Permasalahan yang dibahas pada penelitian tersebut adalah sulitnya dalam mengelola produk pemesanan sepatu dan pengelolaan transaksi pemesanan sepatu.

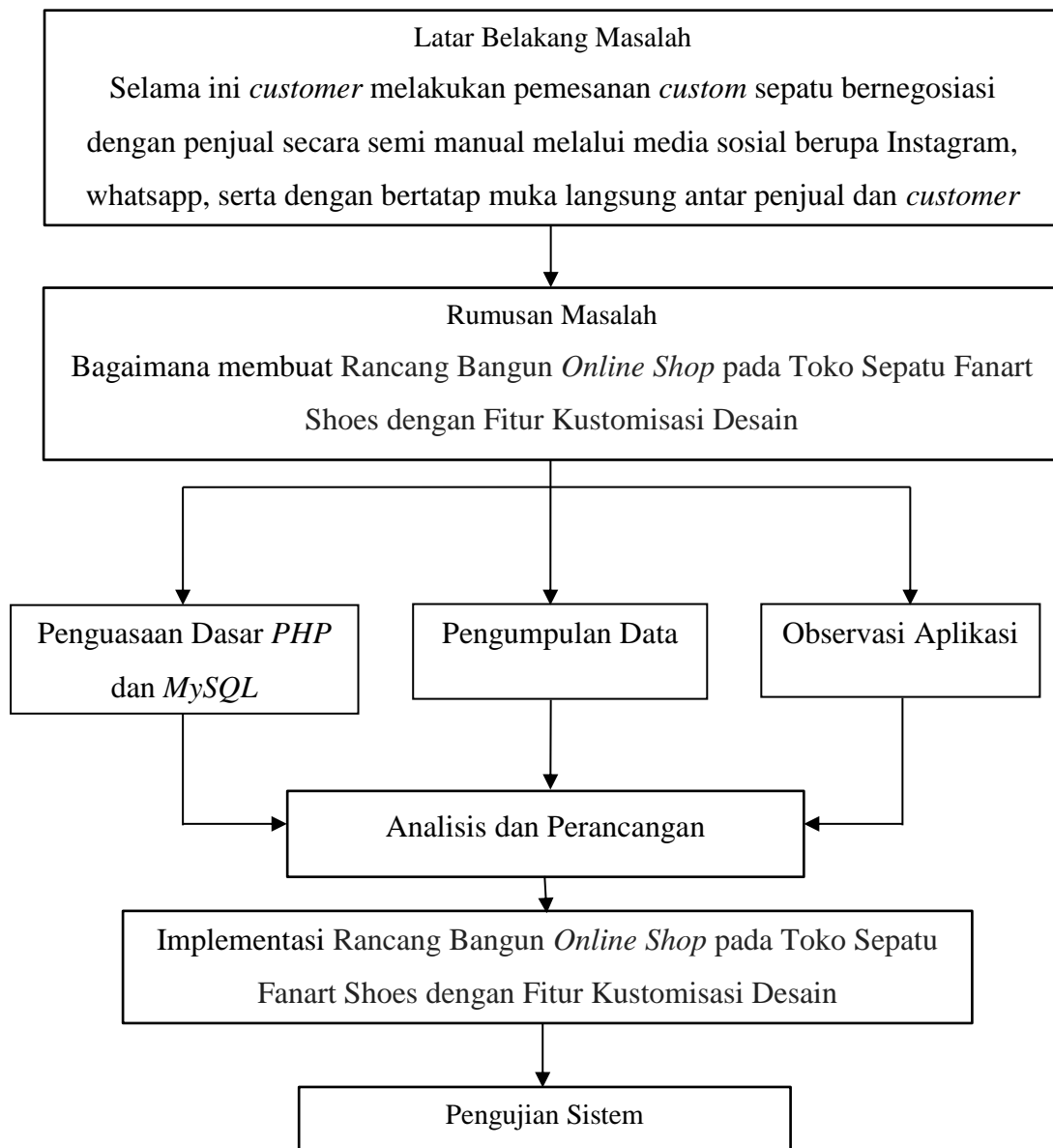
Penelitian keempat dari Lestari (2019) yang berjudul “*Rancang Bangun Toko Online Buket Bunga Wisuda “Craftty Solo” Berbasis Website Menggunakan Metode Waterfall* “. Berdasarkan observasi dengan melakukan pengamatan langsung pada sistem dengan melihat proses yang terjadi dan wawancara penelitian yang dilakukan pada *Craftty Solo*, sistem yang berjalan saat ini masih bersifat semi manual yaitu dengan pemasaran menggunakan media sosial Instagram dan Whatsapp. Maka dapat dibangun sistem penjualan yang berfungsi untuk memperluas pemasaran dengan berbasis *website*.

Penelitian kelima dari Laksono (2019) yang berjudul “Sistem Pemesanan Dan Penjualan Roti Pada *Amelia Cake*”. Pemasaran yang dilakukan *Amelia Cake* belum optimal karena belum memiliki *website* sendiri dilihat dari permasalahan yang ada, maka diperlukan suatu sistem yang baru pada *Amelia Cake* yaitu dengan membuat *website* yang bertujuan untuk memperluas pemasaran dan juga mempermudah kontrol bisnis secara penuh yaitu pemilik bisnis bisa menentukan tema desain situsnya sendiri yang bagus untuk lebih menarik minat konsumen.

Berdasarkan kesimpulan penelitian sebelumnya yaitu penjualan sepatu secara *online* dengan media berbasis *web* saja. Maka *sistem* yang dibangun saat ini menyediakan fitur *custom* sepatu agar dapat membantu *customer* dalam *custom* sepatu sesuai yang diinginkan *customer*.

## 2.2 Kerangka Pemikiran

Kerangka pemikiran penelitian tentang rancang bangun *online shop* pada toko sepatu Fanart Shoes dengan fitur kustomisasi desain, maka dapat digambarkan diagram alur seperti yang ditunjukkan pada Gambar 2.1.



Gambar 2.1 Kerangka Pemikiran Penelitian

Penjelasan dari kerangka pemikiran yang dijalankan dalam penelitian ini adalah sebagai berikut :

1. Latar Belakang Masalah

Promosi *custom* sepatu Fanart Shoes melalui media sosial berupa Instagram, whatsapp, serta hanya dari mulut ke mulut.

2. Rumusan Masalah

Bagaimana membuat Rancang Bangun *Online Shop* Pada Toko Sepatu Fanart Shoes dengan *Fitur* Kustomisasi Desain.

3. Penguasaan PHP dan MySQL

Tahap untuk mempelajari dasar-dasar *PHP* dan *MySql* agar lebih menguasai program-program yang akan digunakan untuk membangun sistem.

4. Pengumpulan Data

Tahap pengumpulan data pada penelitian ini melalui observasi, wawancara dan studi literatur. Pengumpulan data bertujuan untuk mengetahui permasalahan dan kebutuhan toko sepatu Fanart Shoes.

5. Observasi Aplikasi

Observasi aplikasi bertujuan untuk mencari gambaran sistem yang akan dibuat dengan melihat beberapa aplikasi yang sudah ada.

6. Analisis dan Perancangan Aplikasi

Analisis dan perancangan sistem rancang bangun *online shop* pada toko sepatu Fanart Shoes dengan fitur kustomisasi desain bertujuan untuk mengetahui kelemahan serta kekurangan dari sistem yang telah berjalan saat ini, serta membuat perancangan sistem baru yang diharapkan dapat memberikan suatu informasi dengan efektif. Perancangan sistem pada penelitian ini menggunakan UML (*Unified Modeling Language*).

7. Implementasi Rancang Bangun *Online Shop* pada Toko Sepatu Fanart Shoes dengan *Fitur* Kustomisasi Desain. Implementasi sistem adalah langkah-langkah atau prosedur-prosedur yang dilakukan dalam menyelesaikan desain sistem yang telah disetujui, untuk menginstal, menguji dan memulai sistem baru atau sistem yang diperbaiki. Lingkungan implementasi *website* ini meliputi kebutuhan perangkat lunak, perangkat keras, *form* program yang

sesuai, *query* yang digunakan, pemrograman, pengujian program dan pengujian Sistem Informasi yang telah dirancang sesuai dengan kebutuhan dalam rancang bangun *online shop* pada toko sepatu Fanart Shoes dengan fitur kustomisasi desain.

## 8. Pengujian Sistem

Pengujian sistem adalah prosedur yang dilakukan untuk menyelesaikan rancangan sistem yang telah disetujui, menguji sistem, menginstal serta memulai penggunaan sistem baru atau sistem yang telah diperbaiki. Dalam pengujian sistem ini menggunakan metode *black box testing* dan kuesioner.

### 2.3 Landasan Teori

Penjelasan dari beberapa istilah yang berhubungan dengan pembuatan rancang bangun *online shop* pada toko sepatu Fanart Shoes dengan fitur kustomisasi desain.

#### 2.3.1. Rancang Bangun

Rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan (Fajriyah, 2017).

#### 2.3.2. Toko Online

Toko *online* atau *Online Shop* adalah tempat pembelian barang dan jasa melalui media Internet, merupakan salah satu bentuk perdagangan elektronik (*e-commerce*) yang digunakan untuk kegiatan transaksi penjual ke penjual ataupun penjual ke konsumen. Toko *Online* di Indonesia semakin hari semakin menunjukkan perkembangan yang signifikan, belanja secara *online* tidak hanya dimonopoli belanja barang, namun juga layanan jasa seperti perbankan yang memperkenalkan teknik *e-banking*. Melalui teknik *e-banking* pelanggan dapat melakukan kegiatan seperti transfer uang, membayar tagihan listrik, air, telepon, Internet, pembelian pulsa, pembayaran uang kuliah dan lain sebagainya. Toko *Online* di Indonesia untuk pembelian suatu barang mengalami perkembangan yang cukup pesat. Mulai dari situs jualan handphone, gitar, butik, toko buku, makanan,

fashion bahkan hingga ke alat elektronik pun mulai dirambah oleh layanan belanja *online* (Harahap & Amanah, 2018).



Gambar 2.2 Tampilan Sistem Informasi Penjualan Sepatu  
Sumber: (Lesmono, 2018)

### 2.3.3 Website

*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 itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Awalnya suatu *web* hanya dapat menyajikan informasi, saat ini suatu *web* telah dapat berinteraksi dengan pengguna melalui pengisian form validasi input atau transaksi *online* (Mayadewi & Juanda, 2016).

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

*Hypertext Markup Language* adalah merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web

(Kristiawan & Wardati, 2015). Oleh karena itu agar dapat membuat program aplikasi diatas halaman web ,anda terlebih dahulu harus mengenal dan menguasai HTML .

### **2.3.5 PHP (*Perl Hypertext Preprocessor*)**

PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya akan dijalankan diserver tetapi disertakan pada dokumen *HTML*. *PHP* bisa berinteraksi dengan *database*, *file* dan *folder*, contohnya *Blog*, *Toko Online*, *CMS*, *Forum*, dan *Website Social Networking*. *PHP* adalah bahasa *scripting*, bukan bahasa *tag-based* seperti *HTML*. *PHP* termasuk bahasa *cross-platform*, ini artinya *PHP* bisa berjalan di sistem operasi yang berbeda-beda (*Windows*, *Linux*, *ataupun MAC*) “Untuk dapat berjalan, *PHP* membutuhkan *web server*, yang bertugas untuk memproses *file php* dan mengirimkan hasil pemrosesan yang akan ditampilkan di *browser client* (Kristiawan & Wardati, 2015). Oleh karena itu, *PHP* termasuk *server-side scripting* (*script* yang diproses di *server*). *Web server* sendiri adalah *software* yang diinstal di komputer lokal ataupun komputer lain yang berada di jaringan intranet/internet yang berfungsi untuk melayani permintaan-permintaan *web* dari *client*. *Web server* yang paling digunakan saat ini untuk *PHP* adalah “*Apache*”. Untuk media penyimpanan datanya (*database server*), *PHP* biasa menggunakan *MySQL*”. “Untuk menginstall dan mengkonfigurasi ketiga *software* tersebut (*Apache*, *PHP*, *MySQL*) agar dapat berjalan dan selalu terhubung, memang cukup sulit. Maka dari itu dibuatlah paket *software LAMP*, *XAMPP*, *MAMP*, *WAMP* yang tinggal kita install dalam satu kali instalasi. Pembuatan *web* ini merupakan kombinasi antara *PHP* sendiri sebagai bahasa pemrograman dan *HTML* sebagai pembangun halaman *web*.

### **2.3.6 Database**

Basis data (*Database*) merupakan salah satu bagian dalam rekayasa perangkat lunak yang terkomputerisasi dan bertujuan utama memelihara data yang

sudah diolah atau media penyimpanan informasi agar dapat diakses dengan mudah dan cepat (Sukamto & Shalahuddin, 2016).

### 2.3.7 MySQL

*MySQL* adalah *multiuser* database yang menggunakan bahasa *Structured Query Language (SQL)*. *MySQL* dalam operasi *client server* melibatkan *server daemon MySQL* disisi server dan berbagai macam program serta library yang berjalan disisi client. *MySQL* mampu mengangani data yang cukup besar. Perusahaan yang mengembangkan *MySQL* yaitu TEX, mengaku mampu menyimpan data lebih dari 40 database, 10.000 tabel, dan sekitar 7.000.000 baris totalnya kurang lebih 100 *Gigabyte* data.

Pada bulan Mei 1996, *MySQL* versi 1.0 berhasil dirilis namun penggunaanya terbatas 4 orang saja. Namun di bulan Oktober di tahun yang sama versi 3.11.0 dilepaskan ke publik tapi belum bersifat *open source*. Bulan Juni 2000, *MySQL AB* mengumumkan bahwa sejak versi 3.23.19, *MySQL* merupakan *software database* yang bebas berlisensi GPL atau *General Public License* yang *open source*. Mulanya *MySQL* hanya berjalan di sistem operasi *linux* namun pada saat *MySQL* versi 3.22 tahun 1998-1999 sudah tersedia diberbagai *platform* termasuk *windows*. Ini terjadi karena *MySQL* menjadi semakin populer dan dilirik banyak orang karena kestabilan dan kecepatan yang meningkat (Kristiawan & Wardati, 2015).

### 2.3.8 Xampp

Xampp merupakan sebuah perangkat lunak gratis sehingga bebas digunakan. Xampp berfungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari *Apache*, *HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan PHP dan *Perl*. Xampp dikembangkan oleh perusahaan *Apache Friends* yang biasanya digunakan untuk simulasi pengembangan (Supono, 2016).

### 2.3.9 Sublime Text

*Sublime text* adalah teks *editor* berbasis *Python*, sebuah teks *editor* yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal di



kalangan *developer* (pengembang), penulis dan *desainer* (Faridl, 2015). Para *programmer* biasanya menggunakan *sublime text* untuk menyunting *source code* yang sedang ia kerjakan. Sampai saat ini *sublime text* sudah mencapai versi 3. *Sublime text* mempunyai beberapa keunggulan-keunggulan yang dapat membantu pengguna dalam membuat sebuah *web development*.

### **2.3.10 UML (*Unified Modeling Language*)**

*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. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

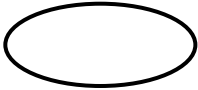
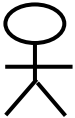

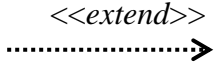
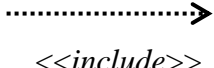



*UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

*UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek (Sukamto & Shalahuddin, 2016).

#### **2.3.10.1 Use case Diagram**

*Use case* atau diagram *Use case* merupakan pemodelan untuk kelakuan (*behaviour*) 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 (Sukamto & Shalahuddin, 2016). Berikut merupakan penjelasan mengenai simbol *use case* diagram yang ditunjukkan pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
1.	 <i>Use case</i>	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> .
2.	 <i>Aktor/Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	 <i>Association</i>	Komunikasi antara aktor dan <i>Use case</i> yang berpartisipasi pada <i>Use case</i> atau <i>Use case</i> memiliki interaksi dengan aktor.
4.	 <i>&lt;&lt;extend&gt;&gt;</i>	Relasi <i>Use case</i> tambahan ke sebuah <i>Use case</i> dimana <i>Use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>Use case</i> tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek.
5.	 <i>&lt;&lt;include&gt;&gt;</i>	Relasi <i>Use case</i> tambahan ke sebuah <i>Use case</i> dimana <i>Use case</i> yang ditambahkan memerlukan <i>Use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use case</i> ini.
6.	 <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.
7.	 <i>Realization</i>	Penjelasan use case dari sisi physical yaitu basis datanya.
8.	<i>Note</i> 	Memberikan informasi method apa yang ada didalamnya dan nama database.



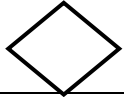
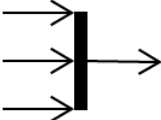
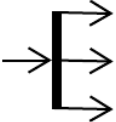

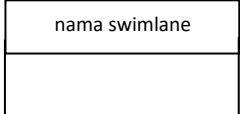
### 2.3.10.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas juga banyak digunakan untuk mendefinisikan rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan

merupakan proses bisnis sistem yang didefinisikan, urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan, rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya, rancangan menu yang ditampilkan pada perangkat lunak (Sukamto & Shalahuddin, 2016).

Berikut merupakan penjelasan mengenai simbol *activity* diagram ditunjukkan pada Tabel 2.2.

Tabel 2.2 Simbol-simbol *Activity* Diagram

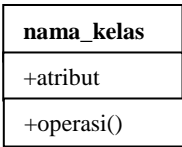
No.	Simbol	Deskripsi
1.	status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Fork Node 	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
6.	status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
7.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

### 2.3.10.3 Class Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Sukanto & Shalahuddin, 2016).

Berikut ini merupakan tabel penjelasan mengenai simbol *class* diagram ditunjukkan pada Tabel 2.3.

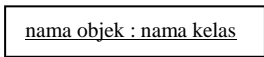






Tabel 2.3 Simbol-simbol *Class* Diagram

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem.
2.	antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	generalisasi 	Relasi antar kelas dengan menggunakan makna generalisasi-spesialisasi (umum-khusus).
6.	kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )
8.	<i>multiplicity</i> 1...*	Satu atau lebih objek sebuah kelas yang berelasi dengan sebuah objek lain dari kelas lain yang berasosiasi dengan kelas tersebut.

### 2.3.10.4 Sequence Diagram

*Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case* (Sukamto & Shalahuddin, 2016). Simbol *sequence* diagram ditunjukkan pada Tabel 2.4.

Tabel 2.4 Tabel *Sequence* Diagram

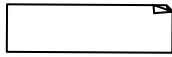
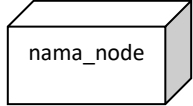
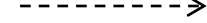

No.	Simbol	Deskripsi
1.	objek 	Menyatakan objek yang berinteraksi pesan.
2.	garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
3.	waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi.
4.	pesan tipe create <code>&lt;&lt;create&gt;&gt;</code> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
5.	pesan tipe call 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
6.	pesan tipe send 1 : masukan 	Menyatakan suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
7.	pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

### 2.3.10.5 Deployment Diagram

Diagram *deployment* atau *deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut : (1) Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*, (2) Sistem *client/server*, (3) Sistem terdistribusi murni, (4) Rekayasa ulang aplikasi (Sukamto

& Shalahuddin, 2016). Berikut merupakan penjelasan mengenai Simbol *deployment* diagram ditunjukkan pada Tabel 2.5.

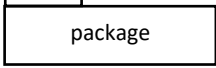




Tabel 2.5 Simbol-simbol Deployment

No.	Simbol	Deskripsi
1.	<i>Constraint</i> 	<i>Constraint</i> adalah mekanisme perpanjangan yang memungkinkan untuk menyempurnakan semantik elemen model UML.
2.	<i>Node</i> 	Biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
3.	<i>dependency</i> 	Kebergantungan / <i>depedency</i> antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.	<i>link</i> 	Relasi antar <i>node</i> .

#### 2.3.10.6 Component Diagram

Diagram komponen atau *component* diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah komponen *user interface* yang menangani tampilan, komponen *bussiness processing* yang menangani fungsi-fungsi proses bisnis, komponen *data* yang menangani manipulasi data, komponen *security* yang menangani keamanan sistem (Sukamto & Shalahuddin, 2016). Simbol *component* diagram ditunjukkan pada Tabel 2.6.

Tabel 2.6 Simbol *Component Diagram*.

No.	Simbol	Deskripsi
1.	<i>Package</i>  package	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen.
2.	komponen 	Komponen sistem.
3.	kebergantungan / <i>dependency</i> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
4.	antar muka / <i>interface</i>  nama_interface	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
5.	<i>link</i> 	Relasi antar komponen.

### 2.3.11 *Black-box Testing*

*Black-box testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, 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 *black-box testing* harus dibuat dengan kasus benar dan kasus salah (Sukanto & Shalahuddin, 2016).

### 2.3.12 Pengujian Betha

Kuesioner adalah teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Kuesioner merupakan teknik pengumpulan data yang efisien bila peneliti tahu dengan pasti variabel yang akan diukur dan tahu apa yang bisa diharapkan dari responden. Selain itu, kuesioner juga cocok digunakan bila jumlah responden cukup besar dan tersebar di wilayah yang luas (Sugiyono, 2017).