

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Berdasarkan penelitian yang dilakukan oleh Tanjung (2019) mengenai dampak *personal selling* terhadap pencapaian penjualan komputer pada CV. Shaheen Global Production, Tanjung menyimpulkan bahwa hambatan-hambatan yang dialami oleh CV. Shaheen Global Production yaitu pemahaman tentang produk yang ditawarkan oleh karyawan masih sangat lemah dikarenakan hampir kebanyakan status pendidikan karyawan hanya sebatas SMU, sedangkan produk yang ditawarkan oleh perusahaan banyak yang menggunakan istilah-istilah IT layaknya perusahaan teknologi, mengenai pengaturan waktu, dimana para karyawan selain melakukan *proces prospecting* atau mencari pelanggan baru untuk mencapai target penjualan disisi lain karyawan harus melakukan hubungan kepada pelanggan untuk memastikan kepuasan pelanggan. Saran yang dapat peneliti kemukakan yaitu, perusahaan sebaiknya dalam merekrut para karyawan yang berperan penting dalam kegiatan *personalselling* setidaknya yang memiliki pengalaman dibidang pemasaran produk khususnya pengalaman dibidang Teknologi, sehingga communication skill sudah dikuasai dan perusahaan hanya memberikan pengarahan saja, perusahaan harus memperluas area penjualan, selain itu perusahaan harus menambah jumlah personil agar perolehan laba dapat meningkat setiap tahunnya.

Berdasarkan penelitian yang dilakukan oleh Jhosua, dkk (2019) mengenai sistem inventori dan penjualan berbasis *Web* dan *Android* pada Toko Cahaya Arloji menggunakan konsep 7c framework, Jhosua, dkk menyimpulkan bahwa aplikasi ini bertujuan untuk membantu proses penjualan dan catatan produk. Aplikasi ini digunakan oleh pemilik dan staf untuk mengelola Toko Cahaya Arloji online, sedangkan bagi pelanggan yang terbiasa melihat produk apapun yang dijual di Toko Cahaya Arloji, dan membantu membuat pesanan online. Aplikasi yang dirancang menggunakan bahasa pemrograman *PHP* dan *Android* Metode aplikasi yang dirancang menggunakan *System Development Life Cycle* dan dimana menggunakan *Waterfall*. Hasil tes diuji menggunakan Metode Penerimaan Pengguna dan kepuasan diri pengguna menggunakan kuesioner. Dengan aplikasi ini, dapat membantu proses pencatatan pembelian dan penjualan di Toko Cahaya Arloji.

Berdasarkan penelitian yang dilakukan oleh Kusuma dan Prasetya (2017) mengenai perancangan dan implementasi *e-commerce* untuk penjualan baju *online* berbasis *Android*, Kusuma, dkk menyimpulkan penelitian ini dilakukan untuk perancangan dan pembangunan aplikasi berbasis *mobile* serta pengujian dengan menggunakan metode *white box*. Metode ini diimplementasikan *e-commerce* ke dalam sebuah aplikasi penjualan baju berbasis *mobile*. Pengimplementasian aplikasi berbasis *mobile* ini dirancang menggunakan bahasa pemrograman *Java* dengan aplikasi pendukung *android studio* versi 1.51. Hasil pengujian menggunakan metode *white box* untuk menguji *software* untuk mengetahui cara kerja suatu perangkat lunak secara *internal*.

Berdasarkan penelitian yang dilakukan oleh Rosmitalia, dkk (2016) mengenai sistem pemesanan makanan di Rumah Makan Palapa Indah berbasis *Web service* menggunakan

*mobile Android*, Rosmitalia, dkk menyimpulkan bahwa dengan adanya sistem pemesanan makanan di Rumah Makan Palapa Indah Berbasis *web service* menggunakan *mobile android* sebagai peningkatan atau penyempurnaan sarana dan prasarana yang telah ada sehingga dapat meningkatkan kinerja karyawan dalam melayani pelanggan dan secara tidak langsung dapat meningkatkan pemasukan di Rumah Makan Palapa Indah.

## **II.2. Landasan Teori**

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang di ambil dari beberapa jurnal dan buku :

### **II.2.1. Aplikasi**

Istilah aplikasi berasal dari bahasa Inggris *application* yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara umum, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta jasa pengguna aplikasi lain yang dapat digunakan oleh pengguna yang akan dituju. Menurut kamus komputer eksekutif, pengertian aplikasi merupakan pemecahan masalah yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Aplikasi biasanya berupa perangkat lunak yang berbentuk *software* yang berisi kesatuan perintah atau program yang dibuat untuk melaksanakan sebuah pekerjaan yang diinginkan.(Inayah, 2015 : 3).

## **II.2.2. Penjualan**

Penjualan merupakan salah satu kegiatan pemasaran yang cukup penting dalam rangka pencapaian tujuan perusahaan. Tujuan umum setiap perusahaan sama, yaitu memperoleh laba yang sebesar-besarnya. Selain itu perusahaan juga mempunyai tujuan khusus, yaitu untuk mencapai kelangsungan hidup dan perkembangan perusahaan. Salah satu kategori dimana suatu perusahaan dikatakan berhasil, apabila dapat memproduksi dan melakukan proses penjualan secara terus-menerus dengan tingkat penjualan yang semakin meningkat. Dalam kaitannya dengan usaha perusahaan untuk meningkatkan penjualan, yaitu melalui salah satu kegiatan pemasaran berupa promosi, dimana promosi memegang peranan yang penting dan tidak boleh diabaikan, karena dengan promosi yang baik maka pemasaran dapat berhasil. (Tanjung, 2019 : 22).

Penjualan merupakan kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli.

(Munaawar dan Rozi, 2019 : 74).

## **II.2.3. Sparepart Komputer**

Suku cadang atau biasanya disebut *sparepart*. Ketersediaan suku cadang ini terkadang tidak selalu tersedia dan terbatas jumlahnya. Hal ini disebabkan karena besarnya biaya investasi yang harus dikeluarkan untuk pembelian dan penyimpanan beberapa suku cadang. Suku cadang adalah alat penunjang mesin-mesin produksi dimana suku cadang

tersebut memegang peranan yang penting bagi kelangsungan produksi suatu perusahaan. Suku cadang adalah suatu alat yang mendukung pengadaan barang untuk keperluan peralatan yang digunakan dalam proses produksi. Jadi, *sparepart* komputer adalah alat pada komputer yang mendukung pengadaan komputer untuk keperluan peralatan yang digunakan dalam proses kinerja komputer. (Juliawati, dkk, 2019 : 7296).

#### **II.2.4. Web Server**

*Web server* merupakan *software* yang memberikan layanan data yang berfungsi menerima HTTP atau HTTPS yang dikenal dengan *browser Web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *Web* yang umumnya berbentuk dokumen HTML. *Web* adalah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar, animasi, suara, yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman yang biasa kita sebut *link*. (Wulandari dan Siska, 2015 : 43).

*Web Service* merupakan komputasi yang dapat diakses melalui jaringan internet maupun *intranet* dengan standar *protocol* tertentu dalam *platform* dan antarmuka bahasa pemrograman yang independen. *Web Service* teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan *transactional Web*, yaitu kemampuan *Web* untuk saling berkomunikasi dengan pola *program to program*. (Rosmitalia, 2016 : 46).

#### **II.2.5. Android**

*Android* (sistem operasi) – OS *Android* – Merupakan sebuah sistem operasi yang berbasis Linux untuk telepon *seluler* seperti telepon pintar dan komputer tablet. *Android*

menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. (Sede, 2015 : 1).

*Android* merupakan sistem operasi *mobile* yang tumbuh di tengah sistem operasi lainnya yang sedang berkembang dewasa ini. Sistem operasi lainnya seperti windows *Mobile*, IOS, Symbian, dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat keras (*hardware*) yang ada. Akan tetapi, sistem operasi yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. (Kusuma dan Prasetya, 2017 : 2).

*Android 4.1. Jelly Bean* diluncurkan pertama kali pada Juli 2012, dengan berbasis *Linux Kernel 3.0.31*. Terdiri dari *Android 4.1 API level 16*, *Android 4.2 API level 17*, *Android 4.3 API level 18*. Penamaan *Jelly Bean* mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

(Lengkong, dkk, 2015 : 21).

## **II.2.6. Versi *Android***

*Android* adalah sistem operasi *seluler* yang didasarkan pada versi modifikasi linux. Hampir semua *smartphone* memiliki sistem operasi *android*.

Dalam pengembangannya *android* telah mengalami cukup banyak pembaruan sejak awal dirilis yang akan ditunjukkan pada Tabel II.1. (Pramadana ; 2018 : 14)

**Tabel II.1. Versi-versi Android**

Versi	Nama	Tanggal Rilis
1.5	<i>Cupcake</i>	30 April 2009
1.6	<i>Donut</i>	15 September 2009
2.0-2.1	<i>Éclair</i>	26 Oktober 2009
2.2	<i>Froyo</i>	20 Mei 2010
2.3-2.3.2	<i>Gingerbread</i>	6 Desember 2010
2.3.3-2.3.7	<i>Gingerbread</i>	9 Februari 2011
3.1	<i>Honeycomb</i>	10 Mei 2011
3.2	<i>Honeycomb</i>	15 Juli 2011
4.0.3-4.0.4	<i>Ice Cream Sandwich</i>	16 Desember 2011
4.1.x	<i>Jelly Bean</i>	9 Juli 2012
4.2.x	<i>Jelly Bean</i>	13 November 2012
4.3.x	<i>Jelly Bean</i>	24 Juli 2013
4.4.x	<i>Kitkat</i>	31 Oktober 2013
5.0	<i>Lollipop</i>	15 Oktober 2014
6.0	<i>Marshmallow</i>	5 Oktober 2015
7.1	<i>Nougat</i>	4 Oktober 2016
7.4	<i>Nougat</i>	5 Desember 2016
8.0	<i>Oreo</i>	21 Agustus 2017
9.0	<i>Pie</i>	6 Agustus 2018
10.0	<i>Android Q</i>	7 Agustus 2019

### **II.2.7. Hyper Text Markup Language (HTML)**

*Hypertext Markup Language* adalah bahasa yang digunakan untuk membuat suatu situs *Web* atau *homepage*. Setiap dokumen dalam *Web* ditulis dalam format HTML. Semua format dokumen, *hyperlink* yang dapat diklik, gambar, dokumen multimedia *form* yang dapat diisi dan sebagainya didasarkan atas HTML. HTML lebih menekankan pada penggambaran komponen-komponen struktur dan *formatting* di dalam halaman *Web* dari pada menentukan

penampilannya, HTML tidak didesain untuk *desktop publishing*, tetapi didesain sebagai bahasa pengkodean untuk *World Wide Web*. (Wulandari dan Siska, 2015 :

43). Contoh dari HTML :

```
<html>

<head></head>

<body style="background-color:yellow">

<h1>Contoh membuat background warna dengan atribut style backgroundcolor.</h1>

</body>

</html>. (Wulandari dan Siska, 2015 : 43).
```

### **II.2.8. Hypertext Preprocessor (PHP)**

Terdapat dua jenis bahasa pemrograman, yaitu *client side* dan *server side*. Saat ini sistem informasi yang dibangun banyak menggunakan pemrograman yang bersifat *server side*. Salah satu yang digunakan adalah PHP. PHP adalah bahasa pemrograman yang digunakan untuk membuat *Website* yang dinamis yang mana bahasa pemrograman ini adalah bahasa pemrograman yang bersifat *server side*. (Yunita dan Sunardi, 2017 : 60).

### **II.2.9. MySQL**

MySQL merupakan *Relational Database Management System* (RDBMS) yang menggunakan bahasa SQL (*Structured Query Language*). MySQL ini adalah *database* yang bersifat, artinya bebas digunakan oleh siapa saja tanpa harus wajib mendapatkan lisensinya. MySQL juga merupakan *database* ini dapat diakses dari jalur manapun, baik dari *client* ataupun dari *server*. (Yunita dan Sunardi, 2017: 60).

### **II.2.10. Normalisasi**

Normalisasi merupakan parameter digunakan untuk menghindari duplikasi terhadap tabel dalam basis data dan juga merupakan proses mendekomposisikan sebuah tabel yang masih memiliki beberapa anomali atau ketidakwajaran sehingga menghasilkan tabel yang lebih sederhana dan struktur yang bagus, yaitu sebuah tabel yang tidak memiliki data *redundancy* dan memungkinkan *user* untuk melakukan *insert*, *delete*, dan *update* pada baris (*record*) tanpa menyebabkan inkonsistensi data. (Triyono, 2015 : 20).

### **II.2.11. Java**

*Java* merupakan bahasa pemrograman berorientasi objek dan bebas *platform*, dikembangkan oleh SUN *Micro System* dengan sejumlah keunggulan yang memungkinkan *java* dijadikan sebagai bahasa pengembangan *enterprise*.

(Ikhlas, 2018 : 49). Keunggulan dari bahasa pemrograman *Java*, yaitu :

1. *Multiplatform applications.*
2. *Secure applications.*
3. *Mission-critical applications.*
4. *Multithreaded applications.*
5. *GUI-based applications.*
6. *Web applications.* (Ikhlas, 2018 : 49).

### **II.2.12. UML**

*Unified Modelling Language* (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software*

yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi *diagram*. (Omni Alfina dan Fitriana Harahap, 2019 :144).


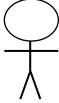

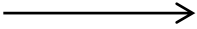

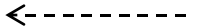
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

#### 1. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use casediagram* dapat dilihat pada tabel

II.2 dibawah ini:

**Tabel II.2. Simbol Use Case**



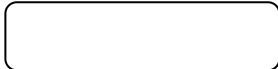
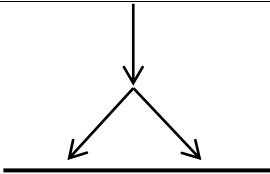
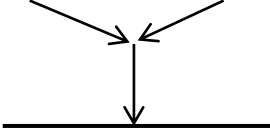
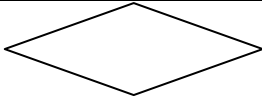

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber:Urva dan Siregar, 2015 : 94)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini:

**Tabel II.3. Simbol *Activity Diagram***

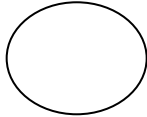
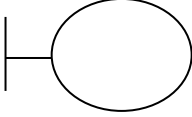
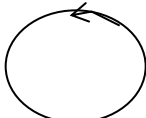

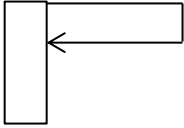

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara <i>parallel</i> atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

(Sumber : Urva dan Siregar, 2015 : 94)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

**Tabel II.4. Simbol *Sequence Diagram***

Gambar	Keterangan
	<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.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i>
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.

     	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .
----------------	---

(Sumber: Urva dan Siregar, 2015 : 95)

#### 4. *Class Diagram* (Diagram Kelas)

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*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.5 dibawah ini:

**Tabel II.5. *Multiplicity Class Diagram***

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva dan Siregar, 2015 : 95)