

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Anthony, dkk (2017) mengenai analisis dan perancangan sistem informasi penjualan berdasarkan stok gudang berbasis *client server* (studi kasus toko grosir “restu anda”), Anthony, dkk menyimpulkan bahwa dengan adanya sistem penjualan barang ini maka toko restu anda mendapatkan kemudahan dalam melakukan penjualan.

Berdasarkan penelitian yang dilakukan oleh As'ary dan Fitriyadi (2018) mengenai sistem informasi penjualan pada toko berkat saudara elektronik berbasis *android*, As'ary dan Fitriyadi menyimpulkan bahwa sistem informasi penjualan pada toko berkat saudara elektronik mampu memberikan pelayanan yang lebih baik kepada pelanggan dengan adanya media *online* yang telah dibuat dan dapat memperluas area cakupan pemasaran toko berkat saudara elektronik.

Berdasarkan penelitian yang dilakukan oleh Hidayat (2018) mengenai pengaruh sistem informasi akuntansi penjualan dan persediaan barang dagang berbasis *client server* terhadap kinerja karyawan cipto jaya sadel lamongan, Hidayat menyimpulkan bahwa bahwa sistem informasi akuntansi penjualan dan persediaan barang dagang berbasis *client server* tidak berpengaruh signifikan terhadap kinerja karyawan pada Cipto Jaya Sadel Lamongan ditunjukkan dari hasil uji t dimana angka Sig. $0,312 > 0,05$. Hal ini disebabkan karena karyawan-karyawan

masih belum memiliki keahlian yang cukup untuk menjalankan sistem informasi tersebut.

Berdasarkan penelitian yang dilakukan oleh Ma'mur (2017) mengenai sistem informasi penjualan barang (studi kasus : minimarket gemart bandar lampung), Ma'mur menyimpulkan bahwa sistem informasi penjualan dapat mempermudah dan membantu kinerja minimarket gemart bandar lampung dalam mengelola penjualan barang.

Berdasarkan penelitian yang dilakukan oleh Sofyan dan Usman (2019) mengenai sistem informasi penjualan sepatu berbasis *web* pada toko stephen sports, Sofyan dan Usman menyimpulkan bahwa sistem informasi penjualan sepatu berbasis web pada toko stephen mempermudah pelanggan dalam pembelian.

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. Client Server

Client Server adalah pada suatu jaringan 1 atau beberapa komputer *server* dan komputer *client*. Komputer yang akan menjadi komputer *server* maupun menjadi komputer *client* dan diubah-ubah melalui *software* jaringan pada protokolnya. Komputer *client* sebagai perantara untuk dapat mengakses data pada komputer *server* sedangkan komputer *server* menyediakan informasi yang diperlukan oleh komputer *client*. (Sriwijaya dan Ma'mur, 2019 : 24).

II.2.3. Penjualan

Penjualan merupakan pembelian sesuatu (barang atau jasa) dari suatu pihak kepada pihak lainnya dengan mendapatkan ganti uang dari pihak tersebut. Penjualan juga merupakan suatu sumber pendapatan perusahaan, semakin besar penjualan maka semakin besar pula pendapatan yang diperoleh. (As'ary dan Fitriyadi, 2018 : 99).

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. Penjualan

adalah pendapatan lazim dalam perusahaan dan merupakan jumlah kotor yang dibebankan kepada pelanggan atas barang dan jasa. Dapat disimpulkan penjualan adalah pemindahan hak kepemilikan atas barang atau jasa yang dilakukan perusahaan sebagai usaha pokok. Ada beberapa teori yang disampaikan yaitu :

1. Strategi penjualan Periklanan, semua bentuk penyajian non personal promosi ide-ide barang produk atau jasa yang dilakukan oleh sponsor tertentu yang dibayar.
2. Promosi penjualan, insentif jangka pendek untuk merangsang pembeli atau penjualan suatu produk.
3. Publisitas, suatu stimulasi non personal terhadap permintaan suatu produk, jasa atau unit dagang dengan penyebaran kebutuhan akan produk-produk tertentu disuatu media yang disebarluaskan atau menghasilkan suatu sosok kehadiran yang menarik mengenai produk di radio, ditelivisi atau panggung yang tidak dibayar oleh sponsor.
4. Penjualan pribadi, penyajian lisan dalam suatu pembicaraan dengan satu atau beberapa pembeli potensial dengan tujuan untuk melakukan penjualan.
5. Pemasaran langsung, penggunaan surat, telepon penghubung non personal lainnya untuk berkomunikasi dengan atau mendapatkan respon dari pelanggan dan calon pelanggan tertentu. (Wulandari dan Aprilia, 2015 : 43).

II.2.4. Pelabuhan Perikanan Samudera Belawan

Pelabuhan Perikanan Samudera Belawan (PPS) Belawan merupakan salah satu pelabuhan perikanan terbesar di Sumatera Utara. PPS Belawan mempunyai peran

penting dalam kegiatan perikanan tangkap dan pemasarannya. PPS Belawan terletak pada posisi yang cukup strategis, yakni terletak diantara Perairan Pantai Timur Sumatera (Selat Malaka), Perairan Zona Ekonomi Eksklusif Indonesia (ZEEI) dan Laut Cina Selatan, serta merupakan pintu masuk bagi kegiatan ekonomi beberapa negara di Asia. (Siahaan, dkk, 2016 : 55).

II.2.5. Web

Website adalah keseluruhan halaman-halaman web yang terdapat dari sebuah domain yang mengandung informasi. *Web server* adalah *server web* merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemaka (Andrianof, 2017 : 271).

Web service adalah sistem *software* yang dirancang untuk mendukung interoperabilitas mesin ke mesin yang dapat berinteraksi melalui jaringan. *Web service* memiliki antarmuka yang dijelaskan dalam format mesin-*processable*. *Web service* merupakan suatu penghubung antara *server* ke *client* dengan menggunakan jaringan. Dibagian klien tidak hanya dapat berupa *web*, aplikasi juga bisa menerima atau berbagi informasi ke bagian *server*. Dengan menggunakan *web service* dapat memberikan pelayanan yang dibutuhkan oleh permintaan klien. (Inayah, dkk, 2015 : 3).

II.2.6. *Android*

Android adalah sistem operasi untuk perangkat selular yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. Pada saat perilis perdana *Android*, 5 November 2007, *Android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi. (Sanjaya, 2015 : 9).

Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi inti yang dirilis oleh Google. Sedangkan *Android SDK (Software Development Kit)* menyediakan *tools* dan API yang diperlukan untuk mengembangkan aplikasi pada *platform android* dengan menggunakan bahasa pemrograman *java*. *Android* disebut sebagai *platform mobile* pertama yang lengkap, terbuka, memecakan hambatan dan bebas. (Sede, 2015 : 1). Penjelasannya sebagai berikut :

1. Lengkap (*complete platform*) : *android* merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun *software* dan memungkinkan untuk pengembangan aplikasi.
2. Terbuka (*Open Source Platform*) : pengembang dapat dengan bebas untuk mengembangkan aplikasi.
3. Memecahkan hambatan pada aplikasi : *Android* memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari *web* dengan data pada ponsel seseorang seperti kontak pengguna, kalender, atau lokasi geografis.
4. Bebas (*free Platform*) : *android* adalah *platform* yang bebas untuk *develop*, tidak ada lesensi atau biaya royalti untuk dikembangkan pada *platform android*. (Sede, 2015 : 1).

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 background-  
color.</h1>  
  
</body>  
  
</html>. (Wulandari dan Siska, 2015 : 43).
```

II.2.8. *Hypertext Preprocessor (PHP)*

PHP adalah bahasa scripting yang menyatu dengan HTML dan dijalankan pada serverside. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada server sedang yang dikirim ke browser hanya hasilnya saja. Ketika seorang pengguna internet membuka situs yang menggunakan fasilitas serverside scripting PHP, maka terlebih dahulu server yang bersangkutan akan memproses semua perintah PHP diserver lalu mengirimkan hasilnya dalam format TML ke web server pengguna internet tadi. Sehingga kode asli yang ditulis dengan PHP tidak terlihat di browser pengguna. PHP berfungsi mengambil informasi dari form berbasis web dan menggunakannya untuk berbagai macam fungsi, sebagai bahasa untuk mengidentifikasi seberapa banyak pengunjung menggunakan bahasa PHP, Pengaturan layout dalam berbagai macam browser seperti Firefox, Bahasa Pemrograman Php dalam web sangat luas, jadi semua tergantung pada anda, kalau anda sangat teliti anda akan tau fungsinya. Contoh sekilas PHP :

<? print (“Nama saya nurmanto”); ?>. (Wulandari dan Aprilia, 2015 : 43).

II.2.9. Basis Data (*Database*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. (Wulandari dan Aprilia, 2015 : 42).

II.2.10. MySQL

MySQL merupakan RDBMS (*Relational Database Management System*) atau *server database* yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat diakses oleh banyak *user*. (Novianti, 2017 : 17).

MySQL merupakan Database Management System (DBMS) *tool open source* yang mendukung *multiuser*, *multireaded popular* dan *free*. Fungsi Mysql dapat dikarakan sebagai *interpreter query* karena setiap kata menggunakan *query SQL* (perintah SQL) kita harus meletakkannya didalam fungsi ini. (Andrianof, 2017 : 271).

II.2.11. Java

Bahasa pemrograman *Java* (awalnya bernama pemrograman Oak), menurut berbagai *literature* diperkenalkan ke hadapan komunitas pemrograman di seluruh dunia oleh James Gosling dari Sun Microsystem Inc. pada sekitar tahun 1990-an.

Java berawal dari sebuah proyek pada tahun 1991 hingga musim panas 1992 yang diberi nama *The Green Project*, yang berjalan selama 18 bulan. Proyek ini dipimpin oleh James Gosling, Mike Sheridan, Patrick Naughton dan Billy Joy, beserta Sembilan pemrogram lainnya dari Sun Microsystem. Tujuan dari proyek ini adalah untuk merancang bahasa computer untuk perangkat konsumen seperti kabel TV box. (Novianti, 2017 : 16).

Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya.

1. *Java.lang* : Peruntukan kelas elemen-elemen dasar.
2. *Java.io* : Peruntukan kelas input dan output, termasuk penggunaan berkas.
3. *Java.util* : Peruntukan kelas pelengkap seperti kelas struktur data dan kelas-kelas penanggalan.
4. *Java.net* : Peruntukan kelas TCP/IP yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
5. *Java.awt* : Kelas dasar untuk aplikasi antarmuka dengan pengguan (GUI).
6. *Java.applet* : Kelas dasar aplikasi antarmuka untuk diterapkan pada penjelajah *web*. (Novianti, 2017 : 16).

II.2.12. UML

UML adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun aplikasi. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan

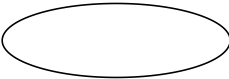
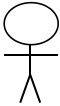
alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2015 : 95).


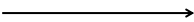
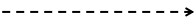
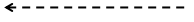
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

II.2.12.1. *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 case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. 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</p>



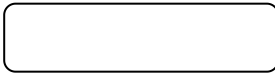
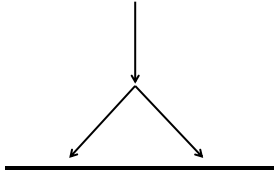
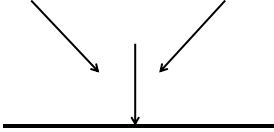
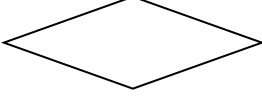

	<p>pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan 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)

II.2.12.2. 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.2 dibawah ini :

Tabel II.2. Simbol *Activity Diagram*

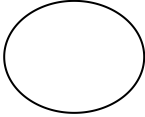
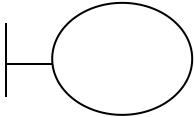
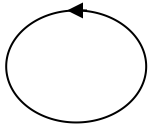

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, 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.

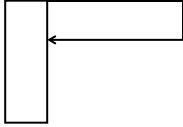


(Sumber : Urva dan Siregar, 2015 : 94)

II.2.12.3. 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.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><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.</p>
	<p><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.</p>
	<p><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.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>

	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>
	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Urva dan Siregar, 2015 : 95)

II.2.12.4. *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.4 dibawah ini :

Tabel II.4. Multiplicity Class Diagram

Multiplicity	Penjelasan
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)