

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Berdasarkan penelitian yang dilakukan oleh Tamsir, dkk (2021) mengenai aplikasi penjualan baju kaos berbasis *web* dan *android*, disimpulkan bahwa dengan adanya aplikasi penjualan ini dapat mempermudah penjualan baju kaos dan memberikan informasi kepada masyarakat.

Berdasarkan penelitian yang dilakukan oleh Deddy, dkk (2020) mengenai aplikasi katalog penjual media pembelajaran dan alat peraga berbasis *android*, disimpulkan bahwa penelitian ini menghasilkan aplikasi penjual katalog media pembelajaran dan alat peraga untuk membantu masyarakat yang ingin mendapatkan informasi mengenai produk tersedia dan juga pemesanan secara *online* yang dapat dilakukan menggunakan *smarthpone*.

Berdasarkan penelitian yang dilakukan oleh Lamawuran, dkk (2021) mengenai aplikasi penjualan *online* kerajinan lokal khas masyarakat nusa tenggara timur berbasis *android*, disimpulkan bahwa aplikasi yang direkayasa dapat dijadikan sebagai alternatif untuk melakukan penjualan kerajinan khas masyarakat NTT, khususnya masyarakat di desa Oemasi yang telah dibuat oleh para pengrajin.

Berdasarkan penelitian yang dilakukan oleh Malanua (2019) mengenai aplikasi penjualan obat dan peralatan pertanian berbasis *android*, disimpulkan bahwa dengan adanya aplikasi ini dapat memudahkan para petani dalam hal memilih obat-obatan yang cocok untuk tanaman mereka dan dapat memudahkan

para penjual obat untuk mempromosikan barang mereka dan menstabilkan penjualan mereka.

Berdasarkan penelitian yang dilakukan oleh Pratama, dkk (2020) mengenai penerapan aplikasi penjualan oleh-oleh secara konsinyasi berbasis *android*, disimpulkan bahwa dengan adanya aplikasi ini memudahkan proses transaksi antara penjual dengan pembeli dan mempercepat proses data pelanggan.

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

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

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

Aplikasi merupakan suatu program komputer yang dapat digunakan pada berbagai perangkat elektronik, terutama pada komputer/laptop atau telepon seluler yang berbasis *Android*. (Efendi, dkk, 2019 : 341).

Aplikasi merupakan komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti pembuatan dokumen atau pengolahan data. (Nugraha, dkk, 2020 : 138).

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi. (Tamsir, dkk, 2021 : 2).

### **II.2.2. Penjualan**

Penjualan merupakan salah satu aspek yang penting di dalam perusahaan karena penjualan merupakan salah satu kegiatan dan sumber utama bagi perusahaan. Pada umumnya perusahaan memiliki tujuan yaitu memperoleh laba.

Dalam usaha untuk memperoleh laba, kegiatan yang dapat dilakukan perusahaan dengan melakukan penjualan produk atau jasa. Penjualan dalam sebuah perusahaan memerlukan pengelolaan yang baik agar dapat mengurangi kemungkinan terjadinya penyelewengan, pemborosan, ketidakefektifan dan ketidakefisienan yang dapat merugikan perusahaan. (Pala'langan, dkk, 2020 : 825). Fungsi yang terkait dalam sistem penjualan tunai adalah :

1. Fungsi Penjualan.
2. Fungsi Kas.
3. Fungsi Gudang.
4. Fungsi Pengiriman.
5. Fungsi Akuntansi.

### **II.2.3. Roti**

Roti merupakan produk yang diperoleh dari adonan tepung terigu yang difermentasi dengan ragi roti dan dipanggang, dengan atau tanpa penambahan makanan lain dan bahan tambahan makanan yang diizinkan. Roti merupakan makanan yang bersifat praktis dan mudah didapat dengan harga terjangkau. Hal tersebut sesuai kebutuhan masyarakat Indonesia yang membutuhkan makanan cepat saji dan praktis, serta dapat memberikan energi yang cukup. (Lamanuk dan Ferrinadewi, 2020 : 1).

Roti merupakan makanan ringan yang cukup banyak penggemarnya. Selain dikenal karena harganya yang murah, juga karena ia dapat bertahan selama beberapa hari. Roti bagi masyarakat Indonesia sudah menjadi alternatif untuk

mengisi perut sebagai pengganti makanan utama atau sekedar menahan lapar sebelum mengonsumsi makanan pokok. (Nugroho, dkk, 2020 : 117).

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

*Web* merupakan kumpulan halaman *web* yang saling berkaitan. *Web* adalah sebuah sistem yang berisi beragam informasi baik berupa teks, gambar, audio maupun video dan dapat diakses melalui perangkat yang biasa disebut *web browser*. *Web* merupakan kumpulan informasi baik yang bersifat statis maupun dinamis yang terdiri dari halaman yang dibuat. (Oktasari dan Kurniadi, 2019 : 151).

*Website* dapat diartikan kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan dari semua orang di seluruh dunia. (Tamsir, dkk, 2021 : 2).

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

*Android* merupakan sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh bermacam peranti penggerak. Pemrograman *Android* membutuhkan *Android SDK (Software Development Kit)* dan juga *Java Development Kit*. Setelah itu dibutuhkan *Java IDE* yang digunakan untuk menuliskan coding *Android*. Salah satu tools yang dapat digunakan dalam pemrograman *Android* adalah *Eclipse Helios* karena *Eclipse* sangat mendukung proses penulisan *code Android* dan menjadikan proses *development Android* menjadi lebih mudah. (Efendi, dkk, 2019 : 344).

*Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan

*platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Dalam perjalanannya *Android* telah merilis banyak versi tentunya dengan berbagai macam fitur dan perbaikan di tiap versi yang baru. Berikut versi-versi *Android* yang telah dirilis hingga saat ini :

1. *Android* 1.1
2. *Android* 1.5 (*Cupcake*)
3. *Android* 1.6 (*Donut*)
4. *Android* 2.0-2.1 (*Eclair*)
5. *Android* 2.2.3 (*Froyo*)
6. *Android* 2.3-2.3.7 (*Gingerbread*)
7. *Android* 3.0-3.2.6 (*Honeycomb*)
8. *Android* 4.0-4.0.4 (*Ice Cream Sandwich*)
9. *Android* 4.1-4.3.1 (*Jelly Bean*)
10. *Android* 4.4 (*KitKat*)
11. *Android* 5.0 (*Lollipop*)
12. *Android* 6.0 (*Marshmallow*)
13. *Android* 7.0 (*Nougat*)
14. *Android* 8.0 (*Oreo*)
15. *Android* 9.0 (*Pie*). (Nugraha, dkk, 2020 : 139).

## **II.2.6. Hypertext Preprocessor (PHP)**

PHP singkatan dari *Hypertext Preprocessor* yang merupakan bahasa *server-side scripting* yang berada pada halaman *Hypertext Markup Language* (HTML) yang ditujukan untuk membantu *programmer* menulis halaman *web* secara mudah dan cepat. PHP bersifat *server-side* berarti semua pengerjaan *script* dilakukan di *server* dan hasilnya dikirim ke *browser*. PHP merupakan bahasa pemrograman yang memprogram situs *web* dinamis, yaitu mampu mengoperasikan *web* secara terus-menerus. *Hypertext Preprocessor* (PHP) merupakan perangkat lunak *open source* yang berada dibawah aturan *general purpose licences* (GPL). (Oktasari dan Kurniadi, 2019 : 150).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script/program* tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. (Sugijanto, dkk, 2020 : 2).

### **II.2.7. *Hypertext Markup Language* (HTML)**

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf,

*heading*, *link* atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. (Sugijanto, dkk, 2020 : 2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam HTML. (Rahmasari, 2019 : 415).

### **II.2.8. MySQL**

MySQL merupakan turunan dari *Structured Query Language (SQL)*, SQL merupakan operasional basis data yang dapat dikerjakan dengan mudah dan otomatis. MySQL bertugas mengatur dan mengelola data-data pada *database*. MySQL merupakan perangkat lunak relasi yang dapat digunakan secara gratis. MySQL adalah perangkat lunak sistem manajemen basis data pada *Structured Query Language (SQL)* yang *multiuser*. MySQL merupakan manajemen basis data yang paling stabil digunakan sebagai media penyimpanan data. (Oktasari dan Kurniadi, 2019 : 150).

MySQL merupakan *database engine* atau *server database* yang mendukung bahasa *database* pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License (GPL)*, tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. (Sugijanto, dkk, 2020 : 2).

## II.2.9. Unified Modelling Language (UML)



UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019 : 39).


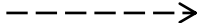



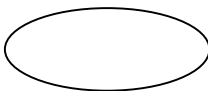
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :



### 1. Use Case Diagram

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada Tabel II.1.

**Tabel II.1. Simbol Use Case Diagram**

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi

		elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use</i> <i>case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang mnghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.


	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

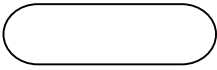



(Sumber : Andikos, 2019 : 39)

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

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

**Tabel II.2. Simbol *Activity Diagram***

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.

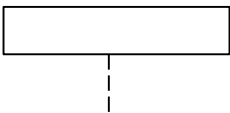
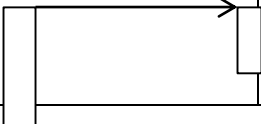
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran


(Sumber : Andikos, 2019 : 39)

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

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

**Tabel II.3. Simbol *Sequence Diagram***

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat


		informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.


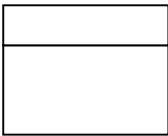

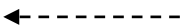
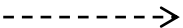
(Sumber : Andikos, 2019 : 39)

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

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

**Tabel II.4. Simbol *Class Diagram***

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek

		yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri

<hr/>	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
-------	--------------------	---

(Sumber : Andikos, 2019 : 39)