

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Berdasarkan penelitian yang dilakukan oleh Firmansyah dan Mulyani (2017) mengenai Rancang Bangun Aplikasi Penjualan *Online* Berbasis *Web* Pada Toko Spiccato Bandung. Penelitian Firmansyah dan Mulyani menggunakan aplikasi penjualan untuk toko Spiccato Bandung dan dapat disimpulkan bahwa aplikasi ini dapat membantu masyarakat dalam melakukan transaksi dan penjualan dimana saja dan kapan saja dengan cara memanfaatkan internet sebagai media aksesnya dan membantu pihak toko dalam melakukan penjualan.

Berdasarkan penelitian yang dilakukan oleh Arfianto dan Nugrahanti (2018) mengenai Rancang Bangun Aplikasi Penjualan Perumahan Berbasis *Web* Pada CV. Grand Permata Residence Magetan. Penelitian Arfianto dan Nugrahanti menggunakan aplikasi penjualan perumahan dan dapat disimpulkan bahwa aplikasi ini akan sangat mempermudah dan memperhemat waktu dan juga tenaga para pegawai menawarkan barang mereka dengan aplikasi ini dan aplikasi ini akan menampilkan letak dan model serta daftar harga dari perumahan tersebut. Aplikasi ini juga dapat menginputkan data para customer yang telah melakukan transaksi, sehingga pegawai tidak perlu mencatat lagi secara manual.

Berdasarkan penelitian yang dilakukan oleh Nurudin, dkk (2019) mengenai Pengujian Black Box pada Aplikasi Penjualan Berbasis *Web* Menggunakan Teknik

Boundary Value Analysis. Penelitian Nurudin, dkk menggunakan aplikasi penjualan teknik boundary value analysis dapat disimpulkan bahwa dengan adanya aplikasi tersebut sangat memungkinkan pengguna melakukan transaksi penjualan, pembelian, memantau stok yang ada, dan dapat menarik laporan penjualan yang ada kapan pun. Tiap pengguna akan diberikan akses tertentu dan fungsi tanggung jawab masing-masing sehingga kasir dan manger dapat mengakses aplikasi dari sisi yang berbeda.

Berdasarkan penelitian yang dilakukan oleh Mawarni, dkk (2020) mengenai Analisa Dan Perancangan Sistem Informasi Penjualan Produk Fashion Berbasis Web Pada Toko Lidya Boutique. Penelitian Mawarni, dkk menggunakan aplikasi penjualan produk fashion dapat disimpulkan bahwa dengan adanya aplikasi ini konsumen tidak perlu harus datang ke toko untuk membeli atau menanyakan ukuran dan stok barang yang dijual, hanya dengan mengakses aplikasi yang dibuat konsumen dapat melihat langsung data barang yang dijual secara langsung. Aplikasi yang dibuat ini menggunakan data informasi penjualan dan produk fashion sehingga dapat melakukan input data informasi produk dengan baik.

Berdasarkan penelitian yang dilakukan oleh Tamsir, dkk (2021) mengenai Aplikasi Penjualan Baju Kaos Berbasis Web dan Android. Penelitian Tamsir, dkk menggunakan aplikasi penjualan baju kaos dapat menyimpulkan bahwa dengan adanya aplikasi ini dapat mempermudah karyawan dan mengelola dalam hasil pelaporan penjualan dan dapat menampilkan hasil informasi. Aplikasi ini menggunakan data pelaporan penjualan sehingga karyawan dapat dengan mudah dalam mengelola data penjualan

II.2. Landasan Teori

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

II.2.1. Rancang Bangun

Rancang bangun adalah menciptakan dan membuat suatu aplikasi ataupun sistem yang belum ada pada suatu instansi atau objek tersebut. Sistem adalah suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan. (Kurnianto dan Lukman, 2021 : 717).

Pengertian perancangan adalah tahapan perancangan (*design*) memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. (Arfianto dan Nugrahanti, 2018 : 174).

II.2.2. Aplikasi

Aplikasi merupakan sebuah subkelas dari perangkat lunak komputer dengan memanfaatkan kemampuan komputer sebagai penerapannya, sebagai penyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau bahkan ke dalam sebuah wadah yang dapat dipakai sebagai penerapan dan pengimplementasian terkait hal atau persoalan yang ada sampai beralih menjadi suatu bentuk yang baru tanpa menghapus poin dasar dari berbagai hal diatas, seperti data, persoalan dan keharusan itu sendiri. Dan aplikasi merupakan suatu perangkat

komputer siap pakai bagi user, seperti mengerjakan suatu tugas yang di input kan oleh pengguna. (Faris dan Wisaksono, 2021 : 62).

Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan kedalam suatu media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan yang ada sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal data, permasalahan, dan pekerjaan itu sendiri. (Sani, 2021 : 401).

Aplikasi Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. (Turaina, dkk, 2021 : 15).

II.2.3. Penjualan

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.

(Munawar dan Rozi, 2020 : 74).

Penjualan adalah menciptakan hubungan jangka panjang bersama pelanggan dengan produk atau jasa perusahaan. Seling merupakan taktik yang menghubungkan perusahaan pelanggan dan relasi antara keduanya. Jenis penjualan yang umum dikenal masyarakat yaitu:

1. *Trade Selling* (Penjualan Perdagangan).
2. *Missionary Selling* (Penjualan Misionaris).
3. *Technical Selling* (Penjualan Teknis).
4. *New Business Selling* (Penjualan Bisnis Baru).
5. *Responsive Selling* (Penjualan Responsif). (Setiawan, dkk, 2021 : 6).

II.2.4. Sparepart

Pengertian dari *Sparepart* adalah suatu barang yang memiliki fungsi tertentu dan memiliki beberapa komponen yang membentuk satu kesatuan. Setiap alat berat memiliki banyak komponen, *sparepart* merupakan komponen pendukung dari mesin utama, setiap mesin yang mengalami kerusakan maka ketersediaan *sparepart* adalah hal yang penting. (Mahdiansyah, dkk, 2021 : 1430).

II.2.5. Mesin

Mesin merupakan kesatuan dari berbagai komponen yang selalu berkaitan dengan elemen-elemen mesin yang bekerja sama satu dengan yang lainnya secara kompak sehingga menghasilkan suatu rangkaian gerakan yang sesuai dengan apa yang sudah direncanakan. Dalam merencanakan sebuah mesin harus

memperhatikan faktor keamanan baik untuk mesin maupun bagi operatornya. Dalam pemilihan elemen-elemen dari mesin juga harus memperhatikan kekuatan bahan, *safety factor*, dan ketahanan dari berbagai komponen tersebut. (Wijaya dan Rodiah, 2020 : 29).

II.2.6. **Mesin *Decanter***

Mesin *Decanter* adalah sebuah mesin pemisah yang di dasarkan pada sedimentasi dengan cara pemusingan (pemutaran) dari *solid* yang terkandung di dalam satu atau dua *liquid*, umpan/bahan yang digunakan adalah sludge dengan komposisi *moisture* (kandungan air) minyak, nos (non oil solid) ialah kotoran yang berupa oil tetapi bukan minyak. Mesin ini di produksi oleh Negara SWEDIA di benua EROPA. Manfaat dari mesin *Decanter* yaitu: Berperan penting untuk hasil akhir pengolahan minyak buah kelapa sawit mentah (CPO) yang siap dijual ke perusahaan- perusahaan yang akan mengolah selanjutnya. (Maulana, 2018 : 23).

II.2.7. **Android**

Android merupakan sistem operasi untuk telepon seluler yang berbasiskan *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, 2020 : 344).

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 : 138).

II.2.8. PHP

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). PHP merupakan singkatan dari “*Hypertext Preprocessor*”. PHP adalah sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang Spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019 : 415).

II.2.9. MySQL

MySQL merupakan salah satu sistem *database* yang sangat handal karena menggunakan sistem SQL. Pada awalnya SQL berfungsi sebagai Bahasa penghubung antara program *database* dengan Bahasa pemrograman yang kita gunakan. Dengan adanya SQL maka para pemrogram jaringan dan aplikasi tidak

mengalami kesulitan sama sekali dalam menghubungkan aplikasi yang mereka buat. (Kuncoro, dkk :1679).

II.2.10. Unified Modelling Language (UML)

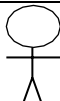
Unified Modelling Language (UML) yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. *Unified Modelling Language* (UML) merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan system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

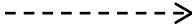
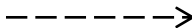
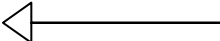
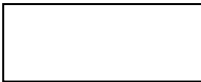
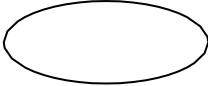
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

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

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

	<i>Depedency</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 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 actor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-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*

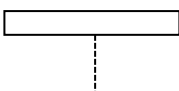
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana 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*

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.

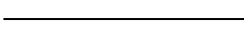
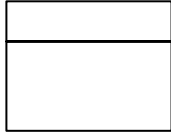
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

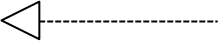
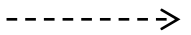
(Sumber : Andikos, 2019 : 39)

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. *Classdiagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

Tabel II.4. *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 (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Assocation</i>	Apa yang menghubungkan ant objek satu dengan objek lainny

(Sumber : Andikos, 2019 : 39)

;