

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Addyna, dkk (2020) mengenai Pengolahan Data Penjualan Baju *Online* Pada Grosir Hasibuan Busana Pematangsiantar Berbasis *Web*, Addyna, dkk menyimpulkan bahwa sistem informasi penyajian penjualan *online* ini meminimaliskan kesalahan dalam input data juga sebagai acuan menyempurnakan sistem kerja yang sedang berjalan.

Berdasarkan penelitian yang dilakukan oleh Andi, dkk (2021) mengenai Perancangan Sistem Pengolahan Data Pembelian Tbs Berbasis *Web* Pada PT. Palma Jaya Sejahtera Jambi, Andi, dkk menyimpulkan bahwa Sistem pengolahan data pembelian TBS berbasis *web* yang telah dirancang menggunakan bahasa pemrograman PHP dan *database* MySQL dalam penelitian ini dapat membantu memecah permasalahan yang terjadi sehingga memberikan kemudahan kepada PT. Palma Jaya Sejahtera Jambi dalam pengolahan data pembelian TBS menjadi lebih cepat dan akurat.

Berdasarkan penelitian yang dilakukan oleh Indriyani, dkk (2019) mengenai Perancangan Sistem Informasi Pengolahan Data *Service* Mobil Pada *Dealer Chevrolet* Kota Jambi Berbasis *Web*, Indriyani, dkk menyimpulkan bahwa aplikasi ini dapat membantu bagian Admin dalam pengolahan data dan memperoleh informasi Pengolahan Data *Service* Mobil dalam bentuk Laporan *Customer*, Mobil, *Service*, Pembayaran.

Berdasarkan penelitian yang dilakukan oleh Kurniawan dan Nugroho (2021) mengenai Aplikasi Pengolahan Data Barang Pada PT.Andal Hasa Prima, Indriyani, dkk menyimpulkan bahwa aplikasi Pengolahan Data Barang Pada PT Andall Hasa Prima, diharapkan dapat digunakan juga untuk memberikan informasi tentang data barang.

Berdasarkan penelitian yang dilakukan oleh Sudibyo dan Asril (2021) mengenai Rancang Bangun Aplikasi Pengelolaan Data Barang Dan *Sparepart* Pada Yamaha Tjahaja Baru Berbasis *Web*, Sudibyo dan Asril menyimpulkan bahwa Sistem Informasi ini dapat memudahkan Pengelolaan Data Barang dan *Sparepart* di Yamaha Tjahaja Baru Muara Bungo.

II.2. Landasan Teori

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

II.2.1. Perancangan

Perancangan merupakan dasar utama dalam membuat aplikasi, dengan tujuan memberikan gambaran lengkap dengan jelas kepada *programmer* tentang aplikasi yang akan dibuat. perancangan atau rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan. (Rivanthio, 2020: 188).

Perancangan merupakan sebuah alur atau proses aplikasi macam-macam teknik prinsip dengan tujuan pendefinisian suatu sistem, perangkat, atau proses detail yang mencukupi untuk memungkinkan pengaktualan fisiknya. (Saputri, 2020: 42).

II.2.2. Sistem

Sistem merupakan serangkain dua atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai tujuan. Sedangkan informasi merupakan data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sebuah organisasi tergantung pada sistem informasi yang baik untuk menjadi daya saing perusahaan, sebagai faktor penting untuk mempertahankan daya saing perusahaan, sistem informasi harus ditingkatkan menjadi lebih baik untuk mempermudah karyawan menyelesaikan pekerjaannya. (Setiawansyah, dkk, 2021: 2).

Sistem merupakan sekelompok unsur yang erat hubungannya satu dengan lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu Tujuan tertentu di hubungkan dengan ruang lingkup yang luas. Bila merupakan sistem yang utama, misalnya sistem bisnis maka istilah goal lebih diterapkan. Jadi tergantung dari ruang lingkup dari mana memandang sistem tersebut. (Permatasari dan Safitri, 2021: 2).

II.2.3. Pengolahan Data

Pengelolaan data merupakan masa atau waktu yang digunakan untuk

mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan. Ada beberapa operasi yang dilakukan dalam pengolahan data, antara lain sebagai berikut:

1. Data Masukan

Kumpulan data transaksi kesebuah pengolahan data medium (contoh, *punchingnumber* kedalam kalkulator) merupakan data masukan. Contoh lain dari data masukan adalah pengkodean data transaksi kedalam bentuk lain (contoh: *converting* atribut kelamin *female* kehuruf F), dan penyortiran data atau informasi untuk pengambilan keputusan (*potential information for future*).

2. Data Transformasi

Beberapa bentuk data transformasi diantaranya sebagai berikut:

- a. Kalkulasi operasi aritmatika terhadap data *field*.
- b. Menyimpulkan proses akumulasi beberapa *data*.
- c. Melakukan klasifikasi terhadap data *group-group* tertentu, seperti *categorizing* (mengelompokan) data kedalam *group* berdasarkan kateristik tertentu.

3. Informasi Keluaran

Informasi keluaran merupakan kegiatan untuk menampilkan informasi yang dibutuhkan pemakai melalui monitor atau cetakan, sedangkan *reproducing* (memproduksi ulang) merupakan kegiatan penyimpanan data yang digunakan untuk pemakai lain yang membutuhkan. *Telecommunicating* (Telekomunikasi) adalah kegiatan penyimpanan data secara elektronik

melalui saluran komunikasi. (Permatasari dan Safitri, 2021: 4).

II.2.4. Distributor

Distributor adalah pihak yang membeli produk secara langsung dari produsen dan menjualnya kembali ke *retailer*/ pengecer, atau bisa juga menjual langsung ke konsumen akhir (*end user*). Pendapat lain mengatakan, pengertian distributor adalah suatu badan usaha atau perorangan yang bertanggungjawab untuk mendistribusikan atau menyalurkan produk perdagangan, baik itu barang maupun jasa, ke *retailer* atau konsumen akhir. Dalam hal ini, distributor hanya mengambil produk yang sudah jadi dan siap digunakan tanpa perlu memodifikasinya. Dalam perdagangan, distributor adalah rantai pertama setelah produsen. Distributor bisa dalam bentuk perorangan atau perusahaan yang membeli produk secara langsung dari produsen dalam jumlah yang sangat besar. Distributor mendapatkan keuntungan dari potongan harga pembelian produk dari produsen. Semakin banyak produk yang dibeli dari produsen, maka potongan harga produk biasanya akan semakin besar. (Suhendri, dkk, 2020: 25).

II.2.5. Web

Web adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya. *Web* merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung dengan fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, suara, animasi dan multimedia lainnya. (Muhaimin, dkk, 2020: 72).

Web server merupakan inti dari suatu *website*. *Webserver* berfungsi sebagai pusat kontrol dari pengolahan data dari *website* sehingga setiap instruksi yang diberikan oleh pemakai internet akan diolah dan dikembalikan lagi kepada *user*. Ada beberapa macam *webserver* yang bisa digunakan, di antaranya milik Windows seperti Apache, Tomcat, IIS, dan lain sebagainya. Yang paling sering digunakan adalah Apache karena lebih banyak mendukung format *file server* tanpa perlu tambahan komponen aplikasi lagi. (Nahlah, dkk, 2018: 12).

II.2.6. PHP

PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya dikirimkan ke *browser* dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*. (Arfianto dan Nugrahanti, 2018: 175).

II.2.7. MySQL

Salah satu contoh *database management system* adalah MySQL. MySQL adalah *database open source* terpopuler di dunia. Dengan kinerja, kehandalan dan kemudahan penggunaan yang terbukti, MySQL telah menjadi pilihan *database*

terdepan untuk aplikasi berbasis *web*, yang digunakan oleh properti *web* profil tinggi termasuk Facebook, Twitter, YouTube, Yahoo! dan banyak lagi. Kata “SQL” dari “MySQL” adalah singkatan dari “*Structured Query Language*”. SQL adalah bahasa standar yang paling umum digunakan untuk mengakses *database*. Bergantung pada lingkungan pemrograman yang kita pakai, kita bisa memasukkan SQL secara langsung (misalnya, untuk men-*generate* laporan), memasukkan pernyataan SQL ke dalam kode yang ditulis dalam bahasa lain, atau menggunakan API khusus yang dapat menyembunyikan sintaks SQL. (Arfianto dan Nugrahanti, 2018: 175).

II.2.8. *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 berbasiskan UML adalah sebagai berikut:

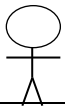
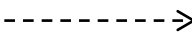
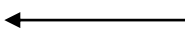
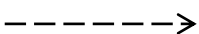
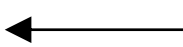
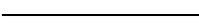
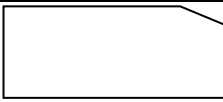
1. *Use Case Diagram*

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

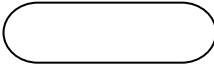
Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Depedency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang 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 <i>actor</i> .
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<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*

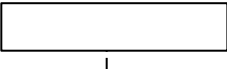
Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
	<i>Activity Final</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

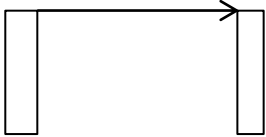
(Sumber: Andikos, 2019: 39)

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.

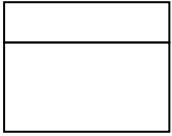
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

(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. *Class diagram* 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 antara objek satu dengan objek lainnya

(Sumber: Andikos, 2019: 39)