

I. PENDAHULUAN

1.1 Latar Belakang

Seiring kemajuan dunia perbankan di kehidupan manusia yang sangat mobile dan menuntut gaya hidup yang praktis dan cepat maka terbentuklah suatu teknologi yang disebut dengan ATM (*Automated Teller Machine*) atau dalam Bahasa Indonesia sering disebut dengan Anjungan Tunai Mandiri. ATM adalah mesin atau alat yang berfungsi untuk melayani nasabah bank secara otomatis, dapat melakukan penarikan uang pembayaran tagihan transfer dan lainnya (Yunita, 2016).

Mesin ATM terdiri dari beberapa bagian suku cadang atau yang disebut *sparepart*, maka dari itu pihak bank bekerja sama dengan beberapa perusahaan untuk mengakomodir kebutuhan *sparepart* mesin ATM dan mesin hitung uang . Perusahaan yang bekerja sama dengan bank tersebut harus bersedia memperbaiki melakukan perawatan dan menyediakan suku cadang atau *sparepart* untuk mesin ATM. Salah satu perusahaan tersebut adalah PT. PQR. Perusahaan yang bergerak dibidang perawatan dan perbaikan mesin ATM dan mesin hitung uang. PT. PQR memiliki kantor pusat yang berada di Jakarta dan memiliki kantor cabang di seluruh daerah di Indonesia salah satunya Lampung. PT.PQR wilayah Lampung sendiri memiliki karyawan yang disebut teknisi. PT.PQR memiliki kantor sekaligus menjadi gudang tempat penyimpanan *sparepart* dan mesin hitung uang yang masih manual untuk mengakomodir kebutuhan *sparepart* mesin ATM dan mesin hitung uang di wilayah Lampung.

Semakin banyaknya *sparepart* yang ada maka akan membutuhkan pendataan *sparepart* yang terstruktur untuk meminimalisir terjadinya redudansi data, kehilangan data *sparepart* maupun mesin hitung uang. Pendataan *sparepart* mesin ATM sangat penting dalam untuk mendukung kelancaran aktivitas perusahaan tersebut, pendataan tersebut berfungsi untuk mengetahui jumlah dan keadaan dari *sparepart* yang disimpan. Saat ini proses pendataan *sparepart* masuk dan keluar masih dengan cara ditulis secara manual menggunakan *form* lalu

dimasukkan kedalam *Microsoft excel* kemudian hasilnya dikirim ke kantor pusat dalam jangka waktu satu tahun sekali. Pengolahan data informasi pendataan *sparepart* tersebut harus di perbarui untuk dapat mempermudah kantor pusat, kepala cabang, maupun teknisi dalam mengetahui ketersediaan *sparepart* dan kondisinya didalam gudang tersebut.

Pemanfaatan teknologi informasi adalah sebagai sarana untuk meningkatkan efisiensi dan efektivitas dalam bekerja serta menghemat waktu dan tenaga juga pikiran (Anshori, 2017.). Pemanfaatan teknologi yang terus berkembang saat ini, teknologi informasi sangat dibutuhkan diperusahaan ini, baik dari sistem pembagian teknisi maupun dalam gudang sekalipun. Teknologi informasi dapat mempermudah dalam hal pengolahan, penyimpanan, serta pendataan *sparepart* masuk dan keluar untuk memberikan informasi yang akurat sehingga mengurangi terjadinya kesalahan yang tidak diinginkan, dapat meningkatkan kinerja yang lebih cepat dan efisien serta dapat mengurangi penggunaan kertas sehingga dapat mengurangi penebangan pohon sebagai bahan baku pembuatan kertas.

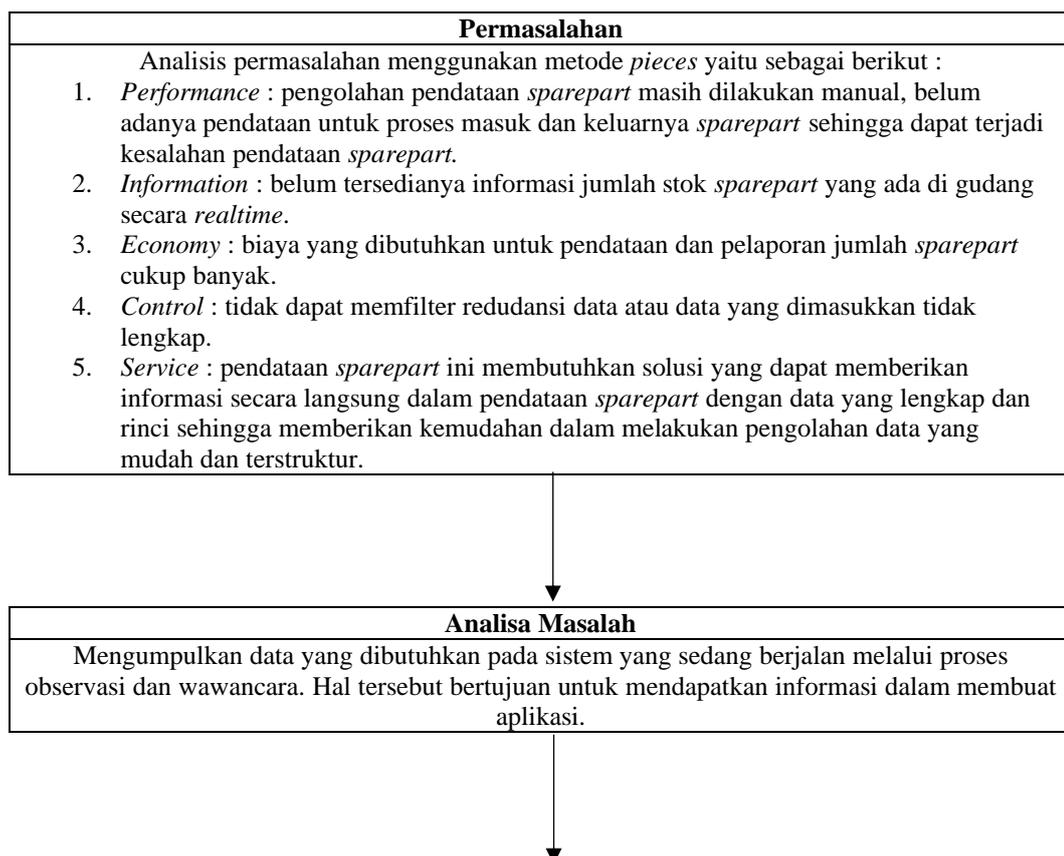
Berdasarkan permasalahan diatas, maka dibuatkanlah aplikasi berbasis *web* untuk menggantikan pendataan *sparepart* secara manual menjadi milenial, sehingga diharapkan pemanfaatan teknologi informasi ini dapat mempermudah dalam memonitoring data *sparepart* yang masuk maupun keluar menggunakan *barcode*, meminimalisir terjadinya kesalahan pendataan maupun kehilangan *sparepart*. Aplikasi ini juga dapat membantu teknisi untuk mengetahui stok *sparepart* sekaligus kondisi *sparepart* tersebut dan kantor pusat juga dapat memantau secara langsung *sparepart* apa saja yang tersedia di gudang penyimpanan.

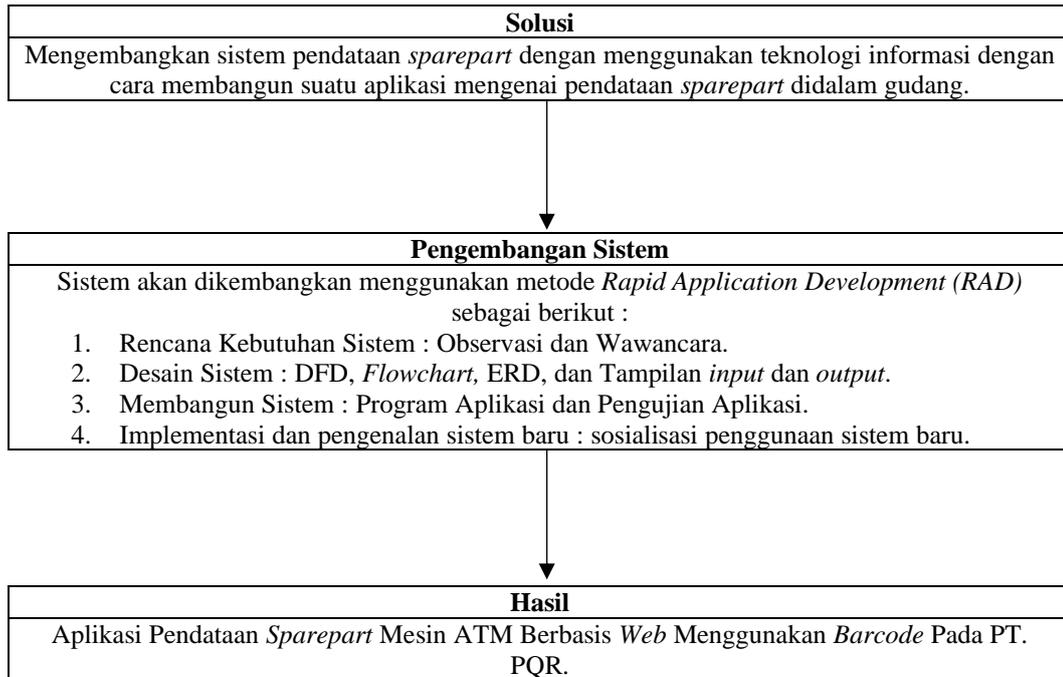
1.2 Tujuan

Tujuan yang ingin dicapai pada penulisan Tugas Akhir ini adalah untuk menghasilkan “Aplikasi Pendataan *Sparepart* Mesin ATM Berbasis *Web* Menggunakan *Barcode* Pada PT. PQR”, yaitu untuk menghasilkan aplikasi untuk pendataan *sparepart* berbasis *web* menggunakan *barcode* serta memberikan kemudahan dalam pengolahan maupun monitoring data *sparepart* yang masuk maupun keluar.

1.3 Kerangka Pemikiran

Berdasarkan permasalahan diatas maka dibuatlah aplikasi untuk mempermudah proses pendataan *sparepart*. Aplikasi yang dibuat adalah Aplikasi Pendataan *Sparepart* Mesin ATM Berbasis *Web* Menggunakan *Barcode* Pada PT. PQR. Aplikasi tersebut akan dikembangkan menggunakan metode *Rapid Application Development* (RAD). Kerangka pemikiran pada aplikasi tersebut disajikan pada Gambar 1.





Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Manfaat yang diharapkan pada Aplikasi Pendataan *Sparepart* Mesin ATM Berbasis *Web* Menggunakan *Barcode* Pada PT. PQR ini dapat memberikan kontribusi pada pihak yang menggunakannya antara lain :

1. Kantor pusat
 - a. Mempermudah dalam melihat stok *Sparepart* yang ada digudang.
 - b. Dapat melihat semua riwayat dan informasi yang ada.
2. Kepala Cabang
 - a. Mempermudah dalam pendataan *Sparepart* masuk maupun keluar.
 - b. Penyimpanan *Sparepart* menjadi terstruktur.
 - c. Memiliki rekap apabila terjadi kerusakan data.
 - d. Mempermudah dalam membuat laporan.
3. Teknisi ATM
 - a. Dapat melihat stok data *Sparepart* yang ada di gudang saat ini.
 - b. Pengambilan *Sparepart* menjadi terstruktur.

II. TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi adalah instruksi (*instruction*) atau pernyataan (*statement*) dalam sebuah komputer yang terstruktur sehingga dapat mengendalikan aktivitas pengguna (*user*) dalam memproses sebuah *input* menjadi *output* (Setiadi dkk., 2017). Aplikasi ini berasal dari kata Bahasa Inggris “*application*” yang artinya mengolah, diambil dari kata kerja *to apply*. Aplikasi ini bersumber dari *aljabar booelan*, yang menggunakan kode *binary* digit (bit) yang terdiri dari 2 angka. *On* berarti benar dan *off* berarti salah. Kode *binary* ini juga disusun dalam beberapa kelompok bit, yaitu : 4 bit (*nible*), 8 bit (*byte*), 2 *byte* (*word*) dan 32 bit (*double bit*). Pengelompokan ini berfungsi untuk menyederhanakan perangkat lunak komputer dalam melakukan aktivitas. Sedangkan menurut (Solichin, 2016). Aplikasi dibagi sesuai dengan lingkungan perkembangan yang digambarkan sebagai berikut :

2.1.1 Aplikasi Berbasis *Desktop*

Aplikasi berbasis *desktop* adalah aplikasi yang membutuhkan proses *instalasi* pada setiap komputer *desktop* yang akan digunakan. Contoh dari aplikasi berbasis *desktop* yaitu *Microsoft office*, *coreldraw*, *adobe photoshop* dan masih banyak lagi (Solichin, 2016).

2.1.2 Aplikasi Berbasis *Mobile*

Aplikasi berbasis *mobile* yaitu aplikasi yang hanya dapat digunakan atau dijalankan pada perangkat seluler seperti ponsel, *smartphone* dan *tablet*. Contoh aplikasi berbasis *mobile* yaitu *whatsapp*, *Instagram*, *facebook* dan lain-lain (Solichin, 2016).

2.1.3 Aplikasi berbasis *Web*

Aplikasi berbasis *web* adalah aplikasi yang ada didalam *server* sehingga tidak perlu menginstal setiap komputernya. Situs *web* adalah salah satu contoh aplikasi berbasis *web* (Solichin, 2016). Aplikasi berbasis *web* memiliki beberapa keunggulan dibandingkan dengan aplikasi berbasis *desktop*, yaitu :

1. Proses instalasi aplikasi tidak diperlukan *client*.

2. Dapat diakses melalui jaringan darimana saja.
3. *Server* merupakan tempat penyimpanan data.
4. Aplikasi bersifat *cross-platform* yang dapat diakses melalui komputer dengan sistem operasi berbeda-beda.

2.2 Pendataan

Data adalah deskripsi objek, peristiwa, aktivitas dan transaksi yang tidak berarti atau tidak memiliki efek langsung pada pengguna. Data juga sering disebut sebagai informasi bahan baku. Pendataan adalah proses mengubah peristiwa tanpa nilai apapun menjadi suatu kegiatan agar peristiwa tersebut dapat membuat keputusan yang berarti (Julianti dkk., 2019).

2.3 Sparepart

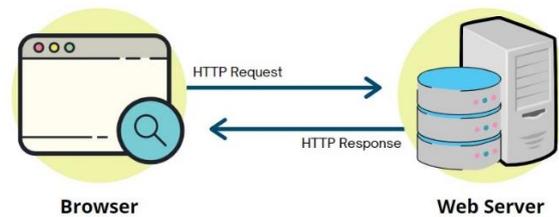
Sparepart adalah alat yang mendukung pengadaan barang untuk peralatan yang dibutuhkan digunakan dalam proses produksi atau *sparepart* adalah suku cadang dari data yang mendukung pengadaan barang untuk peralatan yang dibutuhkan digunakan dalam proses produksi (Wijaya & Nababan, 2020). Berdasarkan definisi diatas, suku cadang atau *sparepart* merupakan faktor penentu utama jalannya proses produksi di perusahaan, sehingga bisa dikatakan *sparepart* ini memiliki peran yang cukup besar dalam serangkaian kegiatan perusahaan.

2.4 Web Server

Web server adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirim melalui *browser* yang merespon permintaan dalam halaman situs *web* dalam bentuk dokumen HTML menurut (Yudhanto, 2018) pada bukunya yang berjudul “Panduan Mudah Belajar *Framework Laravel*”.

Pada *web server* terdapat dua perangkat yaitu perangkat keras (*hardware*) yang digunakan untuk menyimpan semua data seperti HTML, dokumen, gambar, *file stylesheet* CSS, dan *file JavaScript*. Perangkat lunak (*software*) memiliki fungsi sebagai pusat kendali dalam memproses permintaan yang diterima dari *browser*. Jadi dapat disimpulkan bahwa *web server* mengacu pada perangkat keras dan perangkat lunak yang memiliki tugas mengelola semua komunikasi yang terjadi

antara *browser* dan *server* untuk memproses situs *web*. Cara kerja *webserver* dapat dilihat pada Gambar 2.



Gambar 2. Cara Kerja *Webserver* (Yudhanto, 2018)

Cara kerja *web server* adalah pada saat mengambil halaman *web*, *browser* akan mengirimkan permintaan ke *server* yang kemudian akan diproses oleh *web server*, kemudian permintaan HTTP akan dikirim ke *web server* dan akan memeriksa keamanan terlebih dahulu. Permintaan HTTP terbantu dengan *server* HTTP untuk mengakses alamat situs *web* (terjemahkan URL). Kemudian *web server* akan mengirimkan *response* dan diolah menjadi halaman *web*.

2.5 Website

Website menurut Ibrahim dan Ambarita (2018), merupakan kumpulan dari halaman *web* yang telah dipublikasikan di internet dan memiliki *domain* dan URL (*uniform resource locator*) yang dapat diakses melalui alamat oleh semua pengguna internet. Hal ini dimungkinkan dengan *World Wide Web* (WWW).

Halaman *web* biasanya berisi dokumen dalam bentuk : menulis *Hyper Text Markup Language* (HTML) yang dapat diakses menggunakan HTTP atau HTTPS untuk berkomunikasi antara komputer dalam suatu jaringan, sehingga dapat menyampaikan berbagai informasi dari situs *web* untuk ditampilkan kepada pengguna.

2.5.1 Elemen *Website*

Website dibuat dari berbagai elemen yang terkait satu sama lain dan saling mendukung, unsur-unsur tersebut dapat digambarkan sebagai berikut :

2.5.1.1 Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) adalah *file* teks atau *file* ASCII yang berisi instruksi atau skrip untuk digunakan oleh *web browser* menampilkan tampilan grafis dari halaman *web* (Sari & Suryana, 2019).

Menurut Sovia & Febio, (2017) HTML merupakan dasar pembentukan *web*. Dan kode-kode HTML bersifat *universal*, akan diterjemahkan oleh komputer pengguna dengan bentuk tampilan yang sama baik itu teks, grafik, atau bahkan multimedia.

2.5.1.2 Hypertext Preprocessor (PHP)

PHP adalah singkatan dari *Hypertext Preprocessor* yang dipakai sebagai Bahasa *script server side* untuk pengembangan *web* dan disisipkan dalam dokumen HTML, artinya dokumen HTML yang dihasilkan dari aplikasi bukan dokumen HTML yang dibuat menggunakan editor teks atau editor HTML. (Muslihudin & Larasati, 2018).

Menurut (Ibrahim & Ambarita, 2018) PHP adalah sebuah proses penerjemahan baris sumber kedalam kode mesin yang dapat dimengerti komputer ketika kode baris tersebut dieksekusi.

2.5.1.3 Cascading Style Sheet (CSS)

CSS memiliki arti gaya halaman halaman bersarang, yang berarti setiap elemen yang telah diformat dan memiliki anak dan telah diformat. Maka anak elemen secara otomatis mengikuti format elemen induk (Hidayat, 2017).

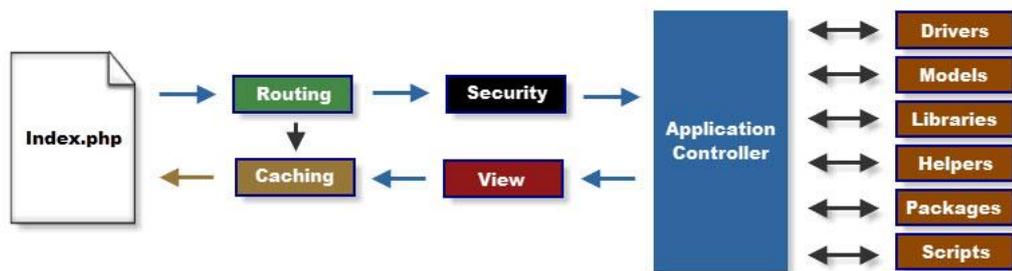
2.5.1.4 JavaScript

JavaScript menurut (Siahaan & Sianipar, 2018) pada bukunya yang berjudul “*JavaScript: Dari A Sampai Z*” adalah Bahasa skrip populer yang digunakan untuk membuat halaman *web* yang dapat berinteraksi dengan pengguna dan merespon peristiwa yang terjadi pada halaman tersebut.

2.6 Framework Codeigniter

Framework adalah kumpulan komponen yang mengandung (fungsi, prosedur, dan kelas) yang dapat langsung digunakan sehingga membantu meringankan pekerjaan *developer* atau *programmer* dalam membuat perangkat lunak. *CodeIgniter* adalah aplikasi *open source* berupa *framework* yang menggunakan MVC untuk membangun *website* dinamis dengan Bahasa pemrograman PHP (Supardi & Hermawan, 2018).

Menurut (Widodo, 2015a) dalam bukunya yang berjudul “Membangun Web Super Cepat dengan *Codeigniter GroceryCURD* dan *TnakAuth*” menerangkan alur kerja dari *Framework* yang dapat dilihat pada Gambar 3.



Gambar 3. Alur Kerja Framwork (Widodo, 2015b)

Dari gambar diatas dapat dijelaskan sebagai berikut :

1. Index.php adalah pengontrol awal yang menginisiasi perlu menjalankan *Codeigniter*.
2. *Router / Routing* adalah bagian yang menentukan aktivitas yang harus dilakukan Ketika ada *request* atau permintaan dari *client* atau *browser*.
3. *Catching* adalah bagian yang mengecek apakah data sudah diminta atau tidak, jika *cache* aktif, itu akan segera dikirim ke *client* atau *browser* untuk mengabaikan alur kerja normal.
4. Keamanan, sebelum aplikasi dikirim, data akan disaring terlebih dahulu sebagai keamanan.
5. Pengontrol adalah pengontrol berjalannya aplikasi dan akan segera memproses sesuai *request* atau permintaan yang diminta yaitu model, perpustakaan, pembantu, *plugin*, dan skrip.

2.6.1 Konsep MVC (*Model, View, Controller*)

Menurut (Airputih, 2014) dalam buku “Modul Panduan *Framework Codeigniter (CI)*” menjelaskan bahwa MVC (*Model, View, Controller*) hanyalah sebuah paten atau Teknik dalam *framework* yang memisahkan desain, data, dan proses. Berikut ini adalah penjelasan konsep MVC dapat dilihat pada Gambar 4.



Gambar 4. Konsep MVC (Airputih, 2014)

1. *Model* adalah sesuatu yang berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*) dan menangani data validasi dari bagian *controller*, tetapi tidak dapat berhubungan langsung dengan bagian *view*.
2. *View* adalah bagian yang menangani *presentation logic* yang menerima dan menyajikan data (file template HTML, yang dikelola oleh 12 pengontrol) kepada pengguna. *View* juga tidak memiliki akses langsung ke bagian model.
3. *Controller* adalah jembatan yang mengatur antara bagian-bagian model yang berbeda terhubung ke *database* dengan membuat *class* dan fungsi di *controller* dan terhubung ke bagian *view* untuk ditampilkan aplikasi. *Controller* juga berfungsi untuk menerima *request* dan data dari *user*, kemudian menentukan aplikasi apa yang akan di proses.

Ada beberapa keuntungan menggunakan *framework codeigniter* dalam membangun aplikasi berbasis *web* yaitu:

1. Mudah dipelajari dan dipahami karena turunan dari PHP.
2. Menghemat waktu produksi karena memiliki struktur MVC dan *library* yang sudah disediakan sehingga *programmer* tidak perlu membangun dari awal.
3. URL *friendly* hanya dengan menggunakan *routes.php* kita bisa membuat url dengan meminimalkan penggunaan `&_GET`

2.7 MySQL

MySQL adalah *database* yang awalnya hanya berjalan di sistem *Unix* dan *linux*. Seiring waktu dan banyaknya peminat yang menggunakan *database* ini, MySQL merilis versi yang dapat diinstal di hampir semua *platform*, termasuk *Windows* (Sovia & Febio, 2017).

Menurut (Komputer, 2010) dalam buku yang berjudul “Panduan Belajar MySQL *Database Server*” MySQL adalah *database server* yang dapat mengirim dan menerima data dengan sangat cepat dan *multi-user*. MySQL memiliki keunggulan sebagai berikut:

1. MySQL adalah *server* tercepat dan merupakan sistem manajemen sebuah *opensource* dan *database* kinerja tinggi dan sederhana.
2. *Database* MySQL memahami Bahasa SQL (*Structured Query Language*).
3. MySQL dapat diakses oleh semua *server*, selain itu semua *client* juga dapat mengakses *server* pada satu waktu.
4. MySQL dapat berjalan diberbagai sistem operasi seperti *Linux*, *Windows*, *Solaris*, dan lain-lain.

2.8 Metode dan Teknik Pengembangan Sistem

Metode adalah Langkah-langkah atau aturan untuk melakukan sesuatu. Metode dan Teknik pengembangan sistem yang digunakan adalah:

2.8.1 *Rapid Application Development (RAD)*

RAD adalah metode pengembangan perangkat lunak dengan pendekatan berorientasi objek (*object oriented approach*) untuk pengembangan sistem. Cara ini bertujuan untuk mempersingkat waktu dalam perencanaan, desain, dan implementasi sistem seketika dibandingkan dengan metode tradisional (Setiawan Putra & Fauziah, 2018). Penggunaan metode ini memungkinkan organisasi untuk mengurangi biaya, waktu pengembangan dan meningkatkan keberhasilan.

2.8.2 Tujuan RAD

Tujuan utama penerapan metode pengembangan perangkat lunak ini adalah jika terjadi perubahan spesifikasi atau desain dapat ditangani secara langsung dan cepat (Wicaksono, 2017).

2.8.3 Tahapan-tahapan RAD

Menurut (Trimahardhika & Sutinah, 2018) tahapan dalam metode pengembangan perangkat lunak RAD adalah sebagai berikut:

1. *Requirement Planning*

Tahap ini adalah tahap pertemuan antara penganalisis dan pengguna untuk mengidentifikasi tujuan dari sistem yang akan dibangun dan diidentifikasi kebutuhan informasi yang akan muncul untuk mencapai tujuan tersebut dan menganalisis semua sistem yang diperlukan oleh pengguna.

2. *User Design*

Setelah melalui tahapan perencanaan, penganalisis dan pemrogram saling bekerja sama untuk membangun sistem dan tujuan representasinya berupa desain visual dan pola kerja kepada pengguna sistem.

3. *Construction*

Tahapan ini adalah eksekusi dalam bentuk *scripting* program dan merupakan kelanjutan dari tahapan kedua, tahapan ini juga menunjukkan *platform, hardware, dan software* yang digunakan.

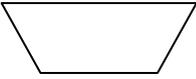
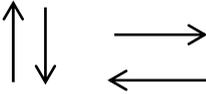
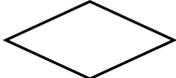
4. *Cutover*

Tahapan terakhir ini penganalisa bekerja dengan pengguna secara intensif selama merancang beberapa aspek teknis dan non-teknis yang dibutuhkan.

2.9 Bagan Alir Dokumen

Bagan alir dokumen merupakan bagan yang menjelaskan aliran dokumen beserta tembusannya (Fitriyana & Susianto, 2018). Adapun simbol bagan alir dokumen disajikan pada Tabel 1.

Tabel 1. Simbol-simbol bagan alir dokumen

Simbol	Nama	Keterangan
	Terminal	Memulai atau mengakhiri program.
	Dokumen	Masukan dan keluaran yang berasal dari dokumen.
	Kegiatan Manual	Kegiatan secara manual.
	Proses	Pengolahan yang akan dilakukan dalam komputer.
	Connecting Line	Arus dari proses.
	Decession	Proses yang akan dilakukan berdasarkan kondisi tertentu
	Input/output	Input dan output tanpa melihat jenisnya.
	Display	Keluaran melalui layar monitor.

(Sumber : Kusbianto, 2010)

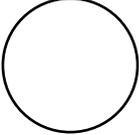
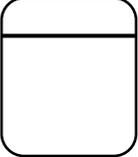
2.10 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah diagram notasi yang menggambarkan aliran data dalam suatu sistem sehingga mudah untuk dipahami oleh sistem secara logis dan terstruktur (Rusmawan, 2019).

Menurut (Muslihudin & Oktafianto, 2016) *Data Flow Diagram* (DFD) adalah aliran-aliran data logis yang mampu menggambarkan sistem lama atau sistem baru tanpa melibatkan lingkungan di luar sistem. DFD juga mampu memberikan

pemahaman bagi pengguna sistem yang tidak mengerti bidang komputer untuk memahami sistem yang berjalan. Simbol *Data Flow Diagram* dapat dilihat pada Tabel 2.

Tabel 2. Simbol-simbol *Data Flow Diagram*

Simbol		Nama
Gane dan Trish Sarson	Edward Yourdon dan Tom De Marco	
		Entitas (<i>Entity</i>)
		Proses (<i>Process</i>)
		Arus Data (<i>Data Flow</i>)
		Penyimpanan Data (<i>Data Storage</i>)

(Sumber : Rusmawan, 2019).

Proses DFD dibagi menjadi 3 *level*, yaitu:

1. Diagram konteks

Mewakili semua proses yang berjalan pada sistem atau aplikasi yang direpresentasikan dalam bentuk gambar dan lingkaran besar. Diagram konteks adalah *level* tertinggi dari *level* DFD (*level-0*) yang menunjukkan semua entitas eksternal dan termasuk aliran data utama ke sistem dan berasal dari sistem.

2. Diagram 0 (diagram *level-1*)

Diagram nol adalah Sebagian kecil dari diagram konteks yang berisi penyimpanan data. Satu lingkaran besar pada diagram nol mewakili lingkaran kecil didalamnya.

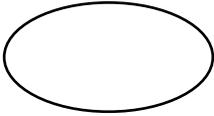
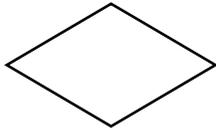
3. Diagram terperinci

Diagram terperinci akan menguraikan proses yang berasal dari diagram nol.

2.11 *Entity Relationship Diagram (ERD)*

Diagram relasi entitas menurut (Rusmawan, 2019) merupakan model data yang memuat detail entitas, relasi, dan kendala dalam memenuhi perkembangan suatu sistem dalam bentuk gambar grafik. Simbol *Entity Relationship Diagram* dapat dilihat pada Tabel 3.

Tabel 3. *Simbol-simbol Entity Relationship Diagram*

Simbol	Nama
	Himpunan entitas (objek yang diidentifikasi secara unik).
	Atribut (penjelasan detail tentang entitas).
	Menggambarkan hubungan (<i>Relasi</i>) antara satu atau lebih entitas.
	Hubungan antara entitas dengan atribut

(Sumber : Rusmawan, 2019)

2.12 *Database*

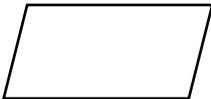
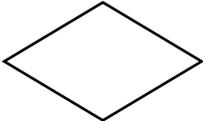
Database menurut Sovia & Febio, (2017) adalah kumpulan *file* yang saling terkait dan berinteraksi, relasi Ketika ditunjukkan oleh kunci dari setiap *file* yang ada. Satu *database* menunjukkan kumpulan data yang digunakan dalam suatu perusahaan atau instansi. Perprosesan *database* adalah suatu metode yang dilakukan pada *file-file* yang berbeda dalam suatu instansi yang *file* dapat dikompilasi, diurutkan, diambil kapan saja dan dapat ditampilkan dalam bentuk *file* laporan sehingga dapat memproses *file* yang berisi informasi dengan rapih.

2.13 Flowchart

Flowchart dapat digunakan untuk menyajikan kegiatan manual, pemrosesan komputer atau keduanya. *Flowchart* adalah serangkaian simbol yang digunakan untuk membangun (Diaraya, 2017).

Sedangkan menurut (Ridlo, 2017) *Flowchart* adalah deskripsi prosedur atau Langkah-langkah diurutkan dari program yang mampu menganalisis dalam memecahkan masalah. Simbol *flowchart* dapat dilihat pada Tabel 4.

Tabel 4. Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
	<i>Terminal</i>	Memulai atau mengakhiri suatu program
	<i>Input/Output</i>	Suatu masukan atau keluaran
	<i>Process</i>	Proses yang dilakukan komputer
	<i>Decision</i>	Kondisi yang memiliki dua kemungkinan jawaban ya atau tidak
	<i>Predefined Process</i>	Proses menjalankan sub program

(Sumber : Ridlo, 2017).

2.14 Barcode

Barcode adalah kode yang berbentuk jumlah baris vertikal. Dalam Bahasa Indonesia sering disebut sebagai kode baris atau kode batang, ada juga yang hanya menulis barkod (Mustafa, 2018). Ada beberapa tipe *barcode* yang sering digunakan tetapi *code 39* adalah tipe yang cocok untuk digunakan pada pendataan kode *sparepart* karena *code 39* dapat mewakili abjad (A-Z) dan angka (0-9) dan memiliki jumlah digit maksimal 16.

2.15 *Black Box Testing*

Metode pengujian perangkat lunak yang digunakan dalam aplikasi ini adalah metode pengujian *black box*. Metode ini menurut (Wicaksono, 2017) berfokus pada pengujian fungsionalitasnya tanpa perlu mengetahui kinerja didalamnya. Pengujian yang dilakukan memperlakukan perangkat lunak seperti kotak hitam, yaitu dengan pengujian diluar tidak perlu melihat ini atau dalamnya.

2.16 **Artikel Terkait**

Artikel terkait adalah teori yang diperoleh dari karya penelitian ilmiah atau penelitian sebelumnya, yang dapat digunakan sebagai referensi dan data pendukung dalam studi kasus yang diangkat oleh penulis. Adapun referensi karya ilmiah yang penulis peroleh adalah sebagai berikut:

1. Mahmud, (2018), dalam karya ilmiahnya yang berjudul “Rancang Bangun Sistem Perpustakaan menggunakan *QR Code* pada SD Negeri Muktiharjo Lor Berbasis *Web* dengan *Framework Codeigniter*” yang bertujuan untuk membuat aplikasi pengolahan data perpustakaan berbasis *web* pada SD Negeri Muktiharjo Lor untuk mempermudah pengguna dalam melakukan peminjaman buku dengan cepat.
2. Purwanto & Wahyudi, (2018), dalam karya ilmiahnya yang berjudul “Rancang Bangun Aplikasi Penjualan Berbasis *Web* Terintegrasi *Barcode*” yang bertujuan untuk membuat aplikasi penjualan berbasis *web* menggunakan *barcode* sehingga dapat mempercepat pembuatan laporan penjualan barang setiap periode tertentu.
3. Sudarma & Mertasana, (2017), dalam karya ilmiahnya yang berjudul “Perancangan Aplikasi Sistem Inventori Barang Menggunakan *Barcode Scanner* Berbasis *Android*” yang bertujuan untuk membuat aplikasi berbasis *android* menggunakan *barcode scanner* sehingga dapat memudahkan pegawai Krisna Oleh-oleh Khas Bali dalam meningkatkan pelayanan kepada pelanggan dalam hal penginformasian data stok barang.