I. PENDAHULUAN

1.1. Latar Belakang

Anjungan Tunai Mandiri (*Autommated Teller Machine*) atau lebih sering dikenal dengan sebutan ATM adalah alat telekomunikasi berbasis komputer yang digunakan dalam melakukan transaksi keuangan tanpa melibatkan seorang *teller* bank (Syamsir & Sarvia, 2018). Mesin ATM berfungsi sebagai alat untuk melakukan penarikan uang tunai. Mesin ATM dilengkapi dengan berbagai fitur yang memudahkan nasabah dalam melakukan transaksi. Mesin ATM juga memiliki beberapa fungsi layanan seperti pengubahan PIN, informasi terkait saldo, setor tunai, serta membayar berbagai jenis tagihan.

Mesin ATM dapat mengalami berbagai kerusakan seperti kartu ATM yang tidak dapat dikeluarkan kembali, kartu ATM terblokir, struk tidak keluar, *keypad* yang rusak dan gangguan pada *software*. PT. PQR merupakan salah satu perusahaan penerima jasa yang bergerak dibidang perbaikan mesin ATM dan berbagai fitur yang ada pada mesin ATM. PT. PQR memiliki kantor pusat di wilayah Jakarta dan memiliki kantor cabang diseluruh Provinsi di Indonesia. PT. PQR ialah perusahaan penerima jasa yang dipimpin oleh seorang Kepala Cabang dan dibantu oleh 7 teknisi. Kantor yang digunakan oleh PT. PQR sekaligus menjadi pusat barang yang dibutuhkan dalam proses perbaikan.

PT. PQR melakukan perbaikan kepada Bank yang ada di Indonesia. Bank yang melakukan kerjasama disebut sebagai *customer*. Mesin ATM yang mengalami permasalahan akan dilaporkan oleh *customer* kepada kepala cabang melalui telepon.Kepala cabang PT.PQR akan menugaskan teknisi untuk memperbaiki kerusakan yang terjadi. Kepala cabang harus menghubungi satu per satu teknisi yang dapat ditugaskan untuk melakukan perbaikan pada mesin ATM.

Teknisi membawa alat yang dibutuhkan sesuai dengan kerusakan mesin ATM. Jika teknisi memerlukan *sparepart* tambahan, maka teknisi harus mengambil ke kantor PT. PQR. Teknisi akan mengecek kerusakan pada mesin ATM dan mengisi *form* secara tertulis. Teknisi memperbaiki kerusakan dan mengisi kembali *form* secara tertulis. Teknisi akan menghubungi kepala cabang

terkait perbaikan yang telah selesai dilakukan dan kepala cabang menginformasikan kepada *customer* melalui via telepon.

Jasa perbaikan mesin ATM selalu dibutuhkan saat terjadi kerusakan, sehingga teknisi harus bergerak cepat dan paham akan keluhan *customer* yang menghubungi pihak perusahaan. Hal ini menyebabkan kepala cabang dari perusahaan PT. PQR harus bisa membagi jadwal teknisi yang bertugas untuk memperbaiki permasalahan yang ada.

Berdasarkan hal tersebut, maka dibutuhkan sebuah aplikasi khusus yang dapat membantu *customer* dalam menyampaikan keluhan kerusakan mesin ATM, serta memudahkan kepala cabang untuk memonitoring atau menugaskan teknisinya. Aplikasi yang dibutuhkan dilengkapi dengan *database* yang berfungsi sebagai tempat penyimpanan data. *Database* dibangun oleh kepala cabang pada aplikasi agar dapat merangkum mengenai riwayat kerusakan yang sering dialami pada mesin ATM. *Database* akan menjadi bahan pertimbangan antara pihak *customer* ataupun kepala cabang serta teknisi. Pertimbangan yang dilakukan yaitu mencari permasalahan yang ada baik pada sisi mesin ataupun sisi kinerja teknisi.

Aplikasi ini akan memudahkan proses bisnis yang dilakukan baik oleh pihak *customer* bank ataupun kepala cabang perusahaan serta teknisi pada PT. PQR. Penulis dalam Tugas Akhir ini mengambil judul "Aplikasi Monitoring Perbaikan Mesin ATM Berbasis *WEB* pada PT. PQR".

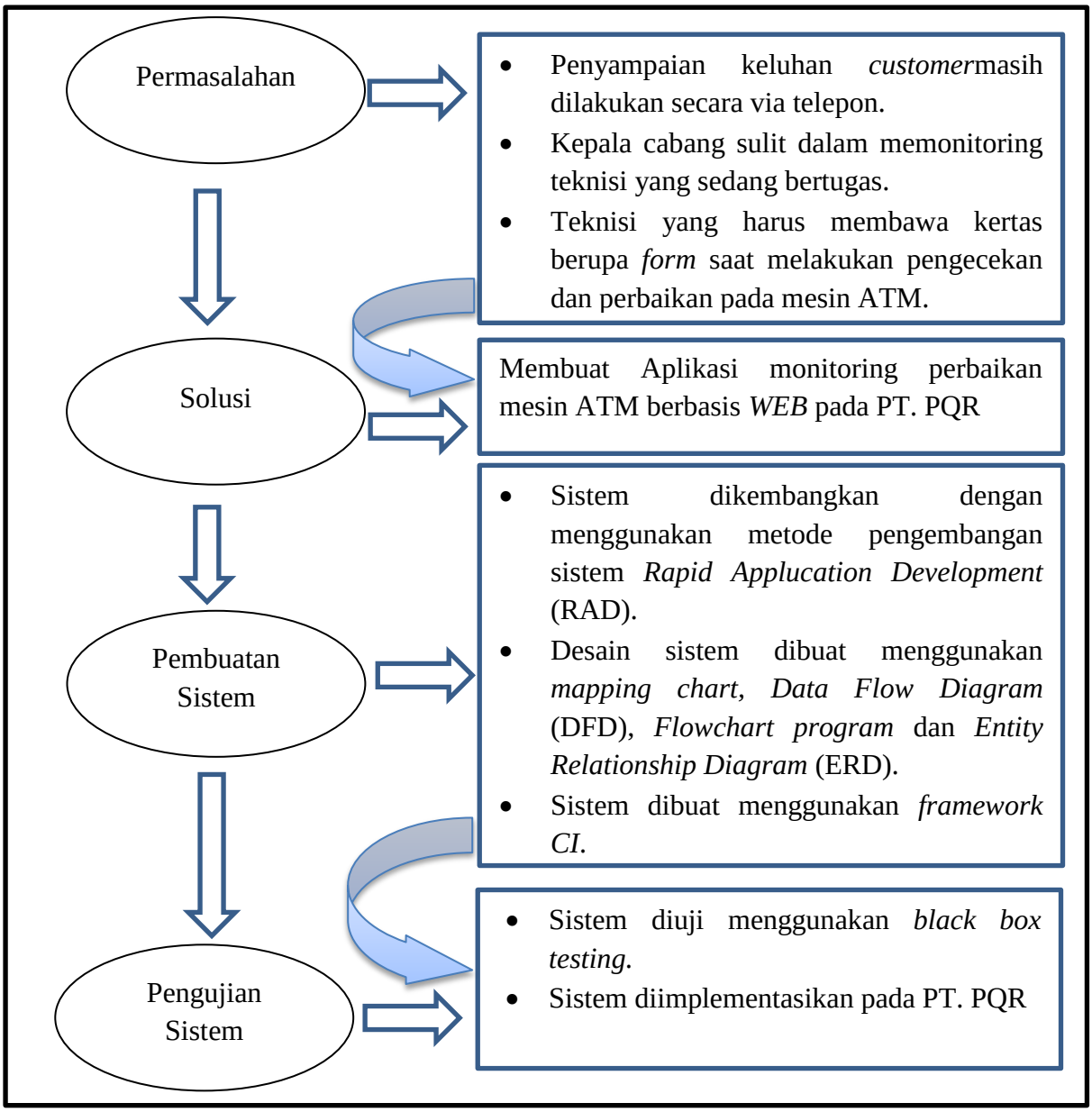
1.2. Tujuan

Tujuan dari disusunnya tugas akhir ini adalah menghasilkan Aplikasi Monitoring Perbaikan Mesin ATM Berbasis *WEB* pada PT. PQR yang akan memudahkan proses bisnis yang dilakukan.

1.3. Kerangka Pemikiran

Berdasarkan masalah yang ada, dapat ditarik kesimpulan untuk membuat sebuah aplikasi monitoring perbaikan mesin ATM berbasis *web*. Aplikasi ini akan dikembangkan dengan menggunakan metode *Rapid Application Development* (RAD) yang memiliki 4 tahapan yaitu *requirements planning, user design, contruction* dan *cutover*. *Requirements planning* ialah proses menganalisis sistem apa yang dibutuhkan oleh PT.PQR, berdasarkan hasil wawancara dan pengamatan

disimpulkan bahwa dibutuhkannya sebuah aplikasiuntuk memonitoring perbaikan mesin ATM. Kemudian pada tahap *user design* yaitu membuat desain aplikasi yang akan dibuat dengan menggunakan bagan alir dokumen , *Data Flow Diagram* (DFD), *fFlowchart* program dan *Entity Relationship Diagram* (ERD). Setelah itu, tahap *contruction* yaitu tahapan dimana hasil dari tahap *user design* diimplementasikan ke dalam bahasa pemrograman PHP, HTML dan MySQL. Tahapan terakhir yaitu tahap *cutover* yaitu proses pengujian aplikasi dengan *Black Box Testing.* Berdasarkan uraian yang telah dipaparkan, maka kerangka pemikiran disajikan pada Gambar 1.



Gambar 1.Kerangka Pemikiran

1.4. Kontribusi

Kontribusi yang diharapkan dengan sistem aplikasi monitoring yang dibangun adalah diantaranya :

1. *Customer*

   Kontribusi yang didapatkan oleh *customer* pada sistem aplikasi yaitu :

   a. Memudahkan *customer* dalam menyampaikan keluhan terkait permasalahan yang ada pada mesin ATM kepada kepala cabang PT. PQR.
   b. Memudahkan *customer* untuk mengetahui hasil akhir perbaikan pada mesin ATM.

2. Kepala cabang PT. PQR

   Kontribusi yang didapatkan oleh Kepala cabang pada sistem aplikasi yaitu :

   a. Memudahkan kepala cabang dalam mendapatkan informasi keluhan dari *customer*.
   b. Memudahkan kepala cabang dalam memonitoring teknisi dalam melakukan perbaikan.

3. Teknisi PT. PQR

   Kontribusi yang didapatkan oleh teknisi pada sistem aplikasi yaitu :

   a. Memudahkan teknisi dalam mendapatkan informasi yang disampaikan oleh kepala cabang.
   b. Memudahkan teknisi dalam mempersiapkan alat dan bahan yang dibutuhkan dalam permasalahan yang dihadapi *customer*.
   c. Memudahkan teknisi dalam mengimputkan tugas yang telah dilakukan.

II. TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi adalah sebuah subkelas perangkat lunak (*software*) komputer yang memanfaatkan kemampuan komputer langsung untuk menjalankan suatu tugas yang diinginkan oler pengguna (*user*) (Hidayatulloh dkk., 2020). Aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk tugas-tugas atau pekerjaan tertentu seperti penerapan, penggunaan dan penambahan data (Alfarisi, 2019).

Dilihat dari lingkungan pengembangan, aplikasi dapat dibagi menjadi aplikasi berbasis *web*, aplikasi berbasis *desktop* dan aplikasi berbasis *mobile*. Aplikasi *web* adalah sebuah aplikasi yang ditempatkan pada komputer *server* yang dapat dijalankan hanya dengan *browser* dari perangkat *client* yang terhubung melalui jaringan ke komputer *server*. Aplikasi *web* dapat diakses dari mana saja melalui jaringan yang terhubung ke *server*.

Aplikasi berbasis *desktop* merupakan sebuah aplikasi yang perlu diinstalasi terlebih dahulu pada setiap komputer yang akan menggunakannya. Aplikasi berbasis *mobile* merupakan sebuah aplikasi yang hanya dapat diakses atau digunakan dalam perangkat *mobile* seperti *smartphone*, *handphone* dan sebagainya.

2.2 *Web*site

Menurut (Rochman dkk., 2020), *website* merupakan sebuah sistem yang memberikan informasi berbentuk teks, gambar, suara dan lainnya yang tersimpan dalam sebuah *server web* internet yang ditampilkan dalam bentuk *hypertext* (Rabani & Firmansyah, 2020). *Website* adalah halaman informasi yang disediakan melalui jalur internet sehingga dapat diakses dimanapun dengan menggunakan jaringan internet.

Website dapat digolongkan menjadi 3 golongan yaitu *website* statis, *website* dinamis dan *website* interaktif. *Website* statis merupakan *website* yang mempunyai halaman yang tidak berubah. Jika dalam suatu halaman ingin diubah,

maka peng*edit*an dapat dilakukan manual dengan cara mengubah kode yang menjadi struktur *website* tersebut. *Website* dinamis yaitu *website* yang informasinya dapat di *update* melalui *website* tersebut. *Website* interaktif, dimana *user* dapat berinteraksi dan beradu argumen mengenai apa yang menjadi pemikiran mereka (Rochman dkk., 2020) .

2.2.1 *Uniform Resource Locator*

Uniform Resource Locator (URL) merupakan kumpulan dari halaman-halaman situs atau *link* yang biasanya terangkum dalam sebuah domain atau subdomain yang tepatnya berada dalam *World Wide Web* (WWW) di internet (Mawarni dkk., 2019). Suatu halaman *web* merupakan dokumen yang ditulis dalam format *Hyper Text Markup Langguage* (HTML).

2.2.2 *Homepage*

Homepage merupakan sebuah halaman utama dari suatu *website* yang akan dilihat pertama kali oleh pengunjung *website. Homepage* berfungsi sebagai *landing page* saat *user* akan mencari sesuatu dari situs *web*. Desain *homepage* harus mampu memberikan gambaran terkait hal apa yang tersedia di situs *web* tersebut (Febriani & Suhartono, 2019).

2.2.3 *Web browser*

Web browser merupakan suatu program yang dirancang untuk mengambil informasi-informasi dari suatu *server* komputer pada jaringan internet (Hutahaean, 2018). *Web browser* bertugas menerjemahkan *tag-tag* HTML yang ditampilkan pada jendela *browser*.

2.2.4 *Web server*

Menurut (Riswandi dkk., 2020) *Web server* merupakan sebuah komputer yang difungsikan sebagai host dari aplikasi *web*, baik dalam jaringan internet maupun intranet. *Web server* adalah perangkat lunak (*software*) yang menjadi tulang belakang dari *world wide web* (www), atau perangkat yang mengatur permintaan pengguna dari protokol HTTP atau HTTPS melalui *web browser* yang hasilnya dikirim kembali ke *web browser*.

2.2.5 *Apache*

Apache merupakan perangkat lunak (*software*) berbasis *web server* yang bersifat *open source*. *Apache* adalah *web server* yang paling sering digunakan. *Apache* dibuat dan diatur oleh *Apache Software Foundation* dan sudah cukup lama beroperasi. *Apache* memiliki nama asli yaitu *Apache HTTP Server* yang berfungsi sebagai developer sebuah situs *web* untuk menciptakan konten-konten pada situs *web* tersebut (Minokaura dkk., 2020).

2.3 Internet

Internet merupakan suatu jaringan komputer yang sangat luas di seluruh dunia yang menghubungkan jutaan jaringan yang lebih kecil seperti jaringan pendidikan, komersial, nirlaba, militer, bahkan individual. Internet menghubungkan begitu banyak komputer sebagai media komunikasi dan informasi modern yang dapat dimanfaatkan secara global oleh pengguna dimanapun dan kapanpun (Laksana dkk., 2020).

2.4 *Rapid Application Development*

Rapid Application Development (RAD) merupakan sebuah model proses perangkat lunak (*software*) yang berfokus pada daur pengembangan hidup yang singkat (Putri dkk., 2018). RAD merupakan versi adaptasi cepat dari model *waterfall* dengan menggunakan pendekatan konstruksi komponen, dan gabungan dari bermacam-macam teknik terstruktur dengan teknik *prototyping* dan teknik pengembangan *joint application* untuk mempercepat pengembangan sistem atau aplikasi.

Pengembangan aplikasi dengan metode *Rapid Application Development* (RAD) dapat dilakukan dalam waktu yang lebih cepat. Metode pengembangan aplikasi perangkat lunak ini memiliki 4 tahapan yaitu *requirement planning*, *user design*, *construction* dan *cotuver* (Purnia, 2018).

Tahap *requirement planning* merupakan kombinasi dari tahapan *sitems planning* dan *system analisis* dari metode *Sistems Development Life Cycle* (SDLC). Pada tahap ini penelitian yang dilakukan yaitu pengumpulan data yang sesuai dengan penelitian yang akan dibuat serta membuat analisa kebutuhan dalam sisi *user* ataupun *admin*.

Tahap *user design* merupakan tahap dimana pengguna berinteraksi dengan sistem analis untuk membangun model pengembnagan yang menggambarkan semua proses, input dan output pada sistem. Pada tahap penelitian ini, dilakukan dengan merancang desain aplikasi sesuai kebutuhan yang dimulai dari alur proses yang berjalan dan alur proses yang diusulkan serta perancangan untuk *database*nya.

Tahap *construction* merupakan tahap yang berfokus untuk membuat sebuah aplikasi dan program dimana *user* tetap dapat berpartisipasi dengan memberikan saran untuk meningkatkan sistem yang akan dibuat. Pada tahap penelitian ini, yang dilakukan adalah membangun aplikasi dengan mengimplementasikan hasil dari tahapan sebelumnya yaitu *user design* ke dalam bahasa pemrograman yang akan dipakai.

Tahap *cotuver* merupakan tahapan yang paling akhir dimana sistem diimplementasikan dengan melakukan konversi data, pengujian dan pergantian ke sistem yang baru. Pada tahap ini dilakukan pengujian aplikasi dengan menggunakan metode *blackbox testing* untuk memastikan semua berfungsi dengan baik.

2.5 *Database*

Database merupakan kumpulan data yang terstruktur sebagai sarana yang berfungsi untuk penyimpanan data, penambahan data baru, mengubah data dan menghapus data serta menghubungkan antara data-data yang disimpan, sehingga dapat dilakukan dengan cepat saat data dibutuhkan (Irawan & Simargolang, 2018). Menurut (Harianja, 2018) *database* merupakan kumpulan file-file yang mempunyai kaitan antara satu file dengan file lainnya sehingga membentuk data untuk menginformasikan suatu hal tertentu.

2.5.1 MySQL

MySQL merupakan *software* berbasis *structure query language* (SQL) tergolong sebagai DBMS (*Database Management System*) yang bersifat *open source*. *MySQL* adalah sebuah implementasi dari sistem manajemen *database* relasional yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*) (Harumy dkk., 2018). Pengguna dapat menggunakan *MySQL* secara

bebas, namun dengan batasan *software* tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. SQL adalah sebuah konsep pengoperasian basis data yaitu untuk proses seleksi, pemasukan, pengubahan dan penghapusan data yang dimungkinkan dapat dikerjakan secara otomatis.

2.5.2 *PhpMyAdmin*

PhpMyAdmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen SQL (Martadinata & Zaliman, 2020). *PhpMyAdmin* dapat dijalankan di berbagai OS, selama dapat menjalankan *web server* dan *MySQL*. Menjalankan *PhpMyAdmin* dapat dibuka dengan mengakses alamat *https://localhost/phpMyAdmin* pada *browser*.

2.6 Bahasa Pemrograman

Bahasa pemrograman merupakan suatu kumpulan instruksi berdasarkan kalimat yang dapat dianalogikan dengan suatu program. Bahasa pemrograman adalah langkah untuk menyelesaikan suatu masalah dengan menggunakan perangkat lunka (*software*) untuk menuliskan algoritma dalam bentuk teks perintah yang dapat dimengerti oleh komputer (Rahayu dkk., 2019).

2.6.1 HTML

Hypertext Merkup Languange (HTML) merupakan bahasan standar pemrograman yang digunakan untuk menampilkan halaman *web*site yang diakses melalui jaringan internet (Ayu & Permatasari, 2018). HTML dapat mengatur tampilan dari halaman *web* dan isinya, membuat tabel, mempublikasikan halamaan *web* secara *online*.

2.6.2 PHP

Menurut (Rubiati & Harahap, 2019) PHP (*Hypertext Prepocessor*) merupakan bahasa pemrograman pada sisi *server* akan dieksekusi sebelum perintah dikirim oleh halaman ke *browser* yang memintanya. Sesuai dengan fungsinya yang berjalan pada sisi *server*, maka PHP adalah bahasa pemrograman yang digunakan untuk membangun teknologi *web application Cascading Style Sheet*.

Casscading style sheet (CSS) merupakan dokumen yang berisi aturan untuk mengatur sebuah tampilan halaman *web* dengan cara mengganti atribut dari *style* CSS yang diinginkan. CSS adalah sebuah fitur yang diperkenalkan sejak HTML versi 4.0 dan berfungsi untuk menangani masalah tampilan pada HTML seperti jenis, ukuran dan warna *font*, posisi teks, margin *font* dan sebagainya (Hasan & Muhammad, 2020).

2.7 *Framework CodeIgniter*

Framework merupakan perangkat lunak (*software*) yang memiliki beberapa fungsi yaitu plugin dan konsep yang biasanya digunakan oleh para *developer website* untuk memudahkan proses pengembangan *website* tersebut karena dapat membuat struktur dan susunan sebuah sistem lebih rapi (Minokaura dkk., 2020). *Framework* juga diartikan sebagai kumpulan perintah atau fungsi dasar yang membentuk sebuah aturan tertentu yang siap digunakan. Menurut (Supriyono & Muslimah, 2018), *codeigniter* merupakan *framework* pengembangan program aplikasi dengan menggunakan bahasa pemrograman PHP.

2.7.1 *Model-View-Controller*

Model-View-Controller (MVC) merupakan sebuah arsitektur untuk membuat sebuah program. Tujuan dari pembagian program ke dalam tiga bagian ini adalah untuk memisahkan fokus perhatian, tanggung jawab, dan logika ke dalam bagian masing-masing (Kelen, 2018). Menurut pendapat ahli lainnya, MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan antara data (model), dari tampilan (*view*), dan cara bagaimana memprosesnya (*controller*) (Yesputra & Marpaung, 2018). Model merupakan struktur data, *view* merupakan informasi yang akan disampaikan ke pengguna dan *controller* merupakan sebuah perantara antara model dan *view* serta semua sumber yang dibutuhkan untuk memproses permintaan HTTP dalam sebuah halaman *web*.

2.7.2 *Bootstrap*

Bootstrap merupakan sebuah *framework front-end* yang intutif dan *powerful* untuk pengembangan aplikasi *web* yang lebih cepat dan mudah. *Bootstrap* menggunakan HTML, CSS, dan *javascript* (Hanafri & Luthfiudin,

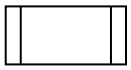
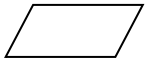
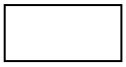
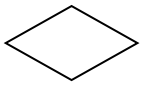
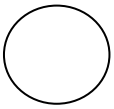
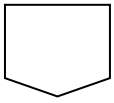
2018). Menurut pendapat ahli lainnya, *bootstrap* adalah *framework* yang digunakan untuk membangun desain *web* secara responsif. Artinya tampilan *web* yang dibuat oleh *Bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita pada *device* apapun (Suprayogi & Rahmanesa, 2019).

2.8 Rancangan Bagan dan Data Sistem

2.8.1 *Flowchart*

Flowchart merupakan bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika (Rubiati & Harahap, 2019). *Flowchart* adalah diagram alir yang menggambarkan sebuah alur logika atau langkah-langkah sistematis pada sebuah program untuk menyelesaikan masalah tertentu. Simbol *flowchart* tersaji pada Tabel 1.

Tabel 1. Simbol-simbol *flowchart*

Simbol	Nama	Keterangan
(1)	(2)	(3)
	*Predefined process*	Permulaan sub program.
	*Simbol input output*	Proses *input* atau *output* data.
	*Processing symbol*	Proses penghitung atau proses pengolahan data.
	*Decision symbol*	Perbandingan, pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
	*Connector symbol*	Penghubung bagian-bagian *flowchart* yang berada satu halaman.
	Simbol dokumen	*Output* dokumen cetak.
	*Connector symbol*	Penghubung bagian-bagian *flowchart* yang berada pada halaman berbeda.
	Simbol punch card	Simbol yang menyatakan *input* berasal dari kartu atau *output* ditulis pada kartu.

Tabel 1. (Lanjutan)

(1)	(2)	(3)
	Simbol punch card	Simbol yang menyatakan *input* berasal dari kartu atau *output* ditulis pada kartu.
	Preparation	Proses inisialisasi atau pemberian harga awal.
	Flow direction symbol	Arah aliran program.

Sumber : Nafiudin (2019)

2.8.2 *Bagan Alir Dokumen*

Bagan alir dokumen merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya (Putra dkk., 2018). Simbol bagan alir dokumen tersaji pada Tabel 2.

Tabel 2. Simbol bagan alir dokumen

Simbol	Nama	Keterangan
(1)	(2)	(3)
	Dokumen	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
	Kegiatan manual	Menunjukan pekerjaan yang dilakukan tidak dengan program komputer.
	Simpanan / arsip	Menunjukan dokumen yang diarsipkan (arsip manual).
	Proses	Menunjukan kegiatan proses yang dilakukan dengan program komputer.
	Hard disk	Menunjukan input / output menggunakan *hard disk*.
	Terminasi	Menunjukan awal atau akhir.

Tabel 2. (Lanjutan)

(1)	(2)	(3)
	Decession	Menunjukan penyeleksian kondisi di dalam program.
	Display	Menunjukkan output yang ditampilkan di monitor.
	Pemasukan	Menunjukan input data secara manual melalui keyboard.
	Garis alir	Menunjukkan arus dari data.
	Penjelasan	Menunjukan penjelasan dari suatu proses.
	Penghubung	Menunjukkan penghubung kehalaman yang masih sama atau ke halaman lain.

Sumber : (Putra dkk., 2018)

2.8.3 *Data Flow Diagram*

Data *Flow* Diagram (DFD) merupakan salah satu *network* yang menggambarkan aliran data sampai sebuah sistem selesai dan sebuah kerja atau proses dilakukan dalam sistem tersebut (Wadisman, 2018). Beberapa simbol yang digunakan pada DFD tersaji Tabel 3.

Tabel 3. Simbol-simbol Data *Flow* Diagram

Simbol	Nama	Keterangan
	External entity	Menggambarkan sumber asli suatu transaksi serta penerimaan akhir dari sistem.
	Data *store*	Menyimpan data dan proses.
	Data *flow*	Menggambarkan arus data yang mengalir sebagai input atau output.
	proses	Suatu proses yang menunjukan adanya tindakan yang dilakukan dan digambarkan.

Sumber : (Solikhin dkk., 2018)

2.8.4 *Entity Relationship Diagram*

Entity Relation Diagram (ERD) merupakan diagram yang menggambarkan hubungan antara objek data yang disimpan dalam satu sistem (Hasan & Muhammad, 2020). ERD berisi komponen-komponen dari suatu himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau, shingga dapat diketahui hubungan antara *entity* yang ada dengan atributnya. Simbol ERD tersaji pada Tabel 4.

Tabel 4. Simbol-simbol *Entity Relation* Diagram

Simbol	Nama	Keterangan
	Entitas/ *entity*	Merupakan data inti yang akan disimpan atau akan menjadi tabel pada basis data.
	Atribut	*Field* atau kolom data yang harus disimpan dalam suatu entitas.
	Relasi	Relasi yang menghubungkan antar entitas dan diawali dengan kata kerja.
	Asosiasi / *association*	Garis penghubung antara relasi dan entitas dimana pada kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakai.

Sumber : (Wadisman, 2018)

Menurut (Anwar & J., 2019) dalam ERD terdapat 4 hubungan antar entitas, yaitu :

a. *One to one* : menjelaskan sebuah relasi antara 1 entitas dengan 1 entitas lainnya.
b. *One to many* : menjelaskan sebuah relasi antara 1 entitas terhadap banyak entitas.
c. *Many to one* : menjelaskan sebuah relasi antara banyak entitas terhadap 1 entitas.
d. *Many to many* : menjelaskan sebuah relasi antara banyak entitas dengan banyak entitas.

2.9 *XAMPP*

XAMPP merupakan perangkat lunak (*software*) bebas yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program diantaranya *Apache, MySQL,PHP* dan Perl. Keempatnya merupakan bagian dari kepanjangan XAMPP dimana penjabarannya yaitu X sebagai sistem operasi, A sebagai *Apache* (*web server*), M sebagai *MySQ* (*database*), P sebagai *PHP* (*server side scripting*) dan Perl (Novendri dkk., 2019).

2.10 *Black Box Testing*

Black box testing merupakan teknik pengujian perangkat lunak (*software*) yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Black box testing* memungkinkan pengembangan *software* untuk membuat pengujian hanya pada bagian luarnya saja. Keuntungan menggunakan metode ini yaitu penguji dilakukan dari sudut pandang pengguna tidak perlu mengetahui bahasa pemrograman tertentu, sehingga *programmer* dan *tester* saling bergantung satu sama lain. Kelemahannya yaitu uji kasus akan sulit disalin tanpa spesifikasi yang jelas, terjadi kemungkinan pengulangan tes yang sudah dilakukan *programmer* (Jaya, 2018).

2.11 Monitoring

Monitoring adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program dalam memantau perubahan yang berfokus pada proses dan keluaran (*output*). Proses monitoring dilakukan untuk memeriksa suatu proses suatu objek, mengevaluasi kondisi menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis tindakan untuk mempertahankan manajemen yang sedang berjalan (Gunawan dkk., 2020)

2.12 Automatic Teller Machine (ATM)

Automatic Teller Machine (ATM) merupakan sistem pelayanan yang diberikan kepada nasabah secara elektronik dengan menggunakan komputer untuk mengupayakan penyelesaian secara otomatis dari sebagian fungsi yang biasanya dilakukan oleh *teller*. ATM dapat menggantikan fungsi *teller* dalam melayani beberapa jenis transaksi perbankan (Syamsir & Sarvia, 2018). Beberapa jenis transaksi yang dapat dilakukan dengan menggunakan mesin ATM antara lain :

a. Penarikan tunai.
b. Transfer antar rekening dalam bank yang sama atau bank yang berbeda.
c. Pembayaran berbagai macam tagihan dan berbagai jenis transaksi perbankan lainnya.

2.13 *Sparepart*

Sparepart merupakan suatu barang yang terdiri dari beberapa komponen yang membentuk satu kesatuan dan mempunyai fungsi tertentu (Jp dkk., 2019). *Sparepart* atau suku cadang merupakan faktor penentu dalam perusahaan sehingga perannya sangat penting.

2.14 Artikel Ilmiah Terkait

Artikel ilmiah terkait terdiri dari beberapa penelitian terdahulu yang digunakan sebagai acuan dan data pendukung, sehingga memperbanyak teori dalam pembuatan tugas akhir ini. Berikut beberapa penelitian berupa artikel yang penulis jadikan sebagai referensi dalam pembuatan tugas akhir :

1. (Endra dkk., 2020) dalam penelitiannya yang berjudul "Perancangan Aplikasi Berbasis *Web* Pada *System Aeroponik* untuk Monitoring Nutrisi Menggunakan *Framework CodeIgniter*". Penelitian ini merancang pemeliharaan tanaman aeroponik yang dilakukan secara otomatisasi berbasis *web* dengan menerapkan *Internet of Things* (IOT).
2. (Hamdala & Esabella, 2020) dalam penelitiannya yang berjudul "Rancang Bangun Aplikasi Monitoring Dan Evaluasi Penyaluran Dana Sosial Di Dinas Sosial Kabupaten Sumbawa Berbasis *Web*". Penelitian ini berbasis *web* agar dapat membantu Dinas Sosial di bidang Fasilitasi Perlindungan dan Jaminan Sosial dalam melaporkan atau mengevaluasi.
3. (Salim dkk., 2021) dalam penelitiannya yang berjudul "Penerapan Metode RAD Pada Sistem Informasi Layanan Umroh Di PT. Galang Saudi Tourism Jakarta Berbasis *Web*site". Penelitian ini menggunakan metode *Rapid Application Development* dengan tahap pemodelan bisnis, pemodelan data, pemodelan proses, pembuatan aplikasi, pengujian dan pergantian. Aplikasi ini berbasis *web* sehingga memudahkan staff pelayanan damam pengolahan data.

4. (Putra dkk., 2018) dalam penelitiannya yang berjudul “Implementasi Genetic *Fuzzy* system untuk mengidentifikasi hasil curian kendaraan bermotor di Polda Lampung”. Penelitian ini diselesaikan oleh genetik *fuzzy* sistem, terutama pengembangan strukyur genetik *fuzzy* metodologi.