BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Sistem Informasi

Sistem adalah sekumpulan subsistem, komponen ataupun elemen yang saling terhubung dan bekerjasama dengan tujuan yang sama guna menghasilkan output berdasarkan tujuan tersebut (Mulyani, 2016). Komponen-komponen tersebut tentunya tidak boleh hilang atau rusak, jika hal itu terjadi kerja sistem akan terhenti dan tujuan sistem tidak akan pernah tercapai. Wilkinson et al (2000), mengatakan ciri-ciri sistem yaitu harus mempunyai komponen-komponen, harus terintegrasi, mempunyai batasan, mempunyai tujuan yang jelas, mempunyai lingkungan, mempunyai input, output dan proses.

Informasi adalah data-data yang sudah diproses yang ditujukan pada seseorang, organisasi atau siapapun yang membutuhka (Mulyani, 2016). Kriteria sistem informasi yang baik adalah informasi harus relevan, andal, lengkap, tepat waktu, dapat dipahami, dapat diverifikasi dan dapat diakses. Sedangkan sistem informasi adalah kumpulan subsistem, komponen ataupun elemen yang saling terhubung dan berkerjasama untuk menghasilkan informasi yang bisa digunakan untuk pertimbangan dalam pengambilan dan membuat keputusan (Mulyani, 2016).

2.2 Pengelolaan

Pengelolaan adalah suatu proses yang memiliki ciri-ciri tertentu dan bersifat unik meliputi kegiatan perencanaan, pengoranisasian, pergerakan dan pengendalian untuk menentukan dan mencapai suatu tujuan tertentu dengan memanfaatkan sumberdaya yang tersedia (Terry, 2012). Pengelolaan biasanya dilakukan pada organiasi, kegiatan dan benda-benda. Contonya ialah pengelolaan pada alat-alat berat yang meliputi kegiatan perawatan, penyewaan dan penjadwalan. Penyewaan adalah kegiatan yang dilakukan untuk menggunakan suatu barang dengan persetujuan, dimana ada salah satu pihak yang menyediakan barang dan memberikannya kepada pihak lain dalam jangka waktu tertentu dengan adanya pembayaran oleh pihak yang diberikan barang tersebut (Eddy, 2010). Penjadwalan merupakan kegiatan yang digunakan untuk melakukan pengambilan keputusan kapan suatu aktivitas atau sumber daya dapat beroperasi atau digunakan, biasanya

banyak digunakan oleh perusahaan manufaktur atau jasa untuk mengoptimalkan satu atau lebih tujuan-tujuan yang ingin dicapai (Pinedo, 2012). Perawatan adalah segala kegiatan yang berkaitan dengan menjaga suatu peralatan, organisasi dan sebuah sistem dalam suatu pekerjaan (Heizer, 2010).

2.3 *Rapid Application Development* (RAD)

SDLC (*Software Development Life Cycle*) adalah sebuah pendekatan langkah demi langkah secara terstruktur untuk membangun sistem informasi (Haag dan Cumings, 2008) yaitu perencanaan, analisis, perancangan, dan implementasi. Sementara menurut Dennis, Wixom, dan Tegarden (2009) menjelaskan ada empat fase dalam SDLC yaitu perencanaan, analisis, perancangan dan implementasi. Kedua tahapan yang dijelaskan oleh masing-masing peneliti tersebut sebenarnya hampir sama, perbedaannya hanya diklasifikasi kegiatan atau fase yang ditampilkan. Dalam implementasi SDLC memerlukan metodologi pengembangan sistem yang merupakan pendekatan formal untuk menjalankan SDLC itu sendiri (Dennis, 2009). Beberapa metodologi dalam pengembangan sistem informasi yaitu structured design (contohnya waterfall), rapid aplication development (contohnya prototype), agile development (contohnya extreme programming). RAD adalah strategi siklus hidup yang ditujukan untuk menyediakan pengembangan yang jauh lebih cepat dan mendapatkan hasil dengan kulitas yang lebih baik dibandingkan dengan hasil yang dicapai melalui siklus tradisional (McLeod, 2002).

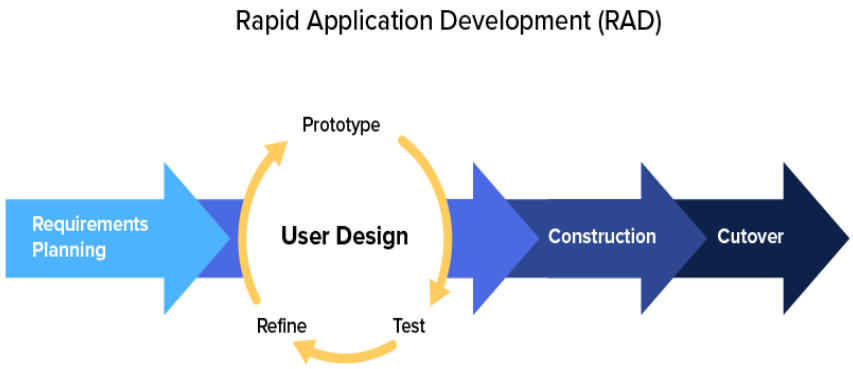
RAD digunakan pada aplikasi sistem konstruksi, maka menekankan fase-fase. Ada empat fase dalam RAD yaitu (Martin, 1991):

1. *Requirement planning*, tahap ini merupakan tahap awal, dalam tahap ini diketahui apa saja yang dibutuhkan sistem yaitu dengan mengidentifikasikan kebutuhan informasi dan masalah yang dihadapi untuk menentukan tujuan, batasan-batasan sistem, kendala dan juga alaternatif pemecahan masalah. Analisis digunakan untuk mengetahui perilaku sistem dan juga untuk mengetahui aktivitas apa saja yang ada dalam sistem tersebut.
2. *User Design*, yaitu mengidentifikasikan solusi alternatif dan yang terbaik. Kemudian membuat desain proses bisnis dan desain pemrograman untuk

data-data yang telah didapatkan dan dimodelkan dalam arsitektur sistem informasi. Pada tahap ini keaktifan pengguna sangat penting untuk mencapai tujuan karena dilakukan proses desain dan perbaikan desain secara berulang apabila masih terdapat ketidaksesuaian. Tools yang digunakaan dalam pemodelan sistem biasanya menggunakan *unified modeling language* (UML). Luaran dari tahap ini adalah spesifikasi perangkat lunak.

3. *Construction,* pada tahap ini desain sistem yang telah disepakati, selanjutnya diterjemahkan kedalam pemrograman (coding), bentuk yang dimengerti oleh mesin yang diwujudkan dalam bentuk program atau unit program.
4. *Cutover,* tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperaikan. Sebelum sistem diterapkan, terlebih dahulu dilakukan proses pengujian terhadap sistem untuk mendektesi kesalahan yang ada pada sistem yang dikembangkan.

Kelebihan RAD, sangat berguna pada kondisi pada saat pengguna tidak memahami kebutuhan apa saja yang digunakan pada proses pengembangan perangkat lunak. Metode RAD dapat digunakan dalam waktu singkat. Biaya yang dikeluarkan lebih rendah. Mudah untuk diamati sehingga pengguna lebih mengerti akan sistem yang dikembangkan. Mampu meminimalkan kesalahan-kesalahan dengan menggunakan alat-alat bantu (*case tools*). Kekurangan RAD dalam pengembangan yaitu membutuhkan SDM yang ahli untuk proyek skala besar, menuntut pengembang dan pelanggan memiliki komitmen dalam aktivitas yang diperlukan untuk melengkapi sebuah sistem dalam waktu yang singkat, jika komitmen tersebut tidak ada maka proyek RAD akan gagal.



Gambar 2. 1 Model *Rapid Application Development* (Martin,1999)

2.4 Model Prototipe

Prototipe adalah salah satu model dari metodologi RAD. Model ini dikembangkan dalam pembangunan perangkat lunak karena pada penggunaan model SDLC sebelumnya mengalami beberapa kendala terutama dalam memahami kebutuhan pelanggan secara detail, khususnya bagi *customer* yang bukan merupakan teknisi perangkat lunak. Ada kalanya *Customer* tidak dapat menjelaskan secara detail masukan, proses, dan keluaran pada perangkat lunak yang ingin dikembangkan sehingga diperlukan suatu model. Dengan menggunakan model prototipe maka pengguna dapat menjabarkan kebutuhan pelanggan secara detail. Untuk memudahkan *customer* dalam menyampaikan kebutuhan perangkat lunak maka pengembang dapat membuat sebuah *mock-up* dari perangkat lunak yang akan dikembangkan.

Adanya *mock-up* yang digunakan sebagai model desain dari perangkat lunak yang ingin dikembangkan, maka pengembang dapat mengajar dan mendemonstrasikan model tersebut sehingga customer dapat dengan mudah memberikan *feed back*. Jika prototipe yang sudah dibuat sesuai dengan keinginan *customer* maka dapat dilanjutkan implementasi prototipe kedalam kode program untuk dijadikan perangkat lunak sesungguhnya.

Tahap pengembangan prototipe menurut Pressman (2010) adalah:

1. Mendengarkan

   Tahap ini digunakan untuk mendapatkan dan mengumpulkan kebutuhan pengguna. Mengumpulkan kebutuhan tersebut dapat dilakukan dengan observasi, wawancara, ataupun kuisioner.

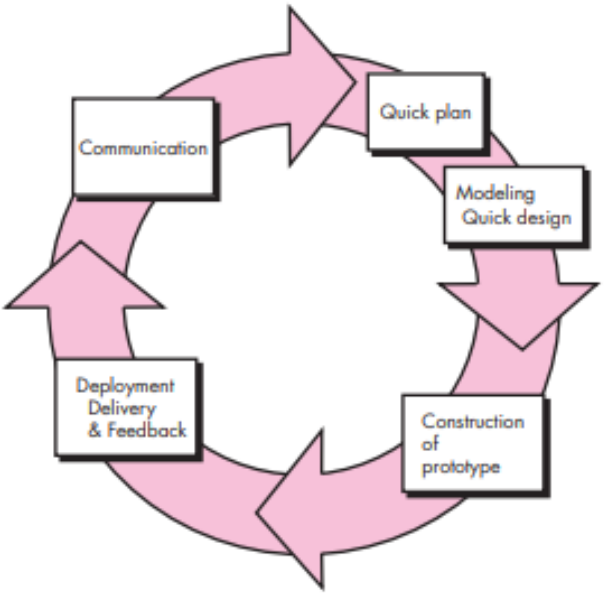
2. Merancang dan Membuat Prototipe

   Tahap ini dilakaukan dengan perancagan dan pembuatan prototipe. Prototipe dibuat berdasarkan kebutuhan yang didapat dari kegiatan pengumpulan data yang dilakukan sebelumnya.

3. Uji Coba

   Setelah prototipe selesai dibangun, maka prototipe tersebut diuji coba oleh *customer* atau *user*. Setelah pengujian *customer* dapat melakukan evaluasi dan memberi *feed back* apakah prototipe sudah sesuai dengan kebutuhan pembangunan perangkat lunak.

Kekurangan pengunaan model prototipe adalah kurang cocok untuk aplikasi dengan skala besar karena akan memakan waktu dan tenaga. Adanya prototipe pelanggan dapat sering menambah-nambah atau merubah spesifikasi kebutuhan, untuk mengatasinya maka pengembang perangkat lunak dapat membuat perjanjian dengan *customer* agar model prototipe digunakan untuk mendefinisikan kebutuhan perangkat lunak, bukan seluruh proses pengembangan perangkat lunak.



Gambar 2. 2 Model *Prototyping*. (Pressman, 2010:44-45)

2.5 Alat Bantu Perancangan Sistem

2.4.1 *Unified Modeling Language* (UML)

UML atau *Unified Modeling Laguage* adalah salah satu pemodelan yang banyak digunakan dalam pengembangan perangkat lunak. UML adalah salah satu standar bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung dan banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahuddin, 2013).

2.4.2 *Use Case Diagram*

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dengan kata lain, *use case diagram* digunakan untuk mengetahui fungs-fungsi apa saja yang terdapat di dalam sistem dan siapa saja yang berhak mengakses fungsi tersebut (Rosa dan Shalahuddin, 2013).

2.4.3 *Class Diagram*

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas pada *class diagram* memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode merupakan fungsi-fungsi yang dimiliki oleh suatu kelas. *Class diagram* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai dengan rancangan di dalam *class diagram* agar antar dokumentasi perancangan dan perangkat lunak sinkron (Rosa dan Shalahuddin, 2013).

2.4.4 *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Rosa dan Shalahuddin, 2013).

2.4.5 *Sequence Doagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima

antar objek. Untuk menggambar diagram *Sequence Diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Dalam membuat *Sequence Diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case.* Semakin banyak *use case* yang didefinisikan maka *Sequence Diagram* yang harus dibuat juga semakin banyak (Rosa dan Shalahuddin, 2013).

2.6 Basis Data

Basis Data adalah kumpulan data (elementer) yang secara logika berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basis Data adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di dalam organisasi (Hariyanto, 2004).

Basis Data adalah kumpulan data yang disimpan secara sistematis pada komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak atau program aplikasi untuk menghasilkan suatu informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur, dan juga batasan-batasan data yang akan disimpan. Basis data merupakan aspek yang sangat penting dalam pengembangan sistem informasi dimana basis data digunakan sebagai gudang penyimpanan data yang akan dikelola lebih lanjut untuk menghasilkan informasi. Basis data menjadi penting karena dapat menghindari duplikasi data, hubungan antar data yang tidak jelas, organiasi data, dan juga update yang rumit.

Proses memasukan dan mengambil data ke dan dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan sistem manajemen basis data (*Database Manajemen System/*DBMS). Tujuan utama dari DBMS adalah untuk memberikan tinjauan abstrak data kepada *user* (pengguna). Jadi sistem menyembunyikan informasi tentang bagaimana data disimpan, dipelihara, dan tetap dapat diambil (akses) secara efisien.

2.7 Sistem Berbasis *Web* (SBW)

Aplikasi *web* adalah sarana yang memperoleh informasi, struktur-struktunya, membangun informasi, dan mengirim informasi tersebut (Pressman, 2009). Sebuah *website* terdiri atas file-file *hypertext* yang memprentasikan

informasi menggunakan teks dan grafis. Untuk menerjemahkan dokumen *hypertext* kedalam bentuk dokumen yang dapat dipahami oleh manusia, maka web browser melalui web client akan membaca halaman web yang tersimpan di sebuah web server melalui protocol yang sering disebut dengan HTTP (*Hypertext Transfer Protocol*). PHP (*Hypertext Prepocessor*) merupakan sebuah bahasa scripting yang terpasang pada HTML dengan tujuan digunakan secra luas untuk penanganan pembuatan dan pengembangan sebuah web. PHP ditulis dan diperkenalkan pertama kali sekitar tahun 1994 oleh Rasmus Lerdorf melalui situsnya untuk mengetahui siapa saja yang telah mengakses ringkasan onlinenya.

Dalam dunia internet selalu terdapat dua sisi yang saling mendukung, yaitu:

1. *Server* adalah penyedia berbagai layanan termasuk *web*. Layanan *web* ditangani oleh sebuah aplikasi bernama *web server*.
2. *Client* bertugas mengakses informasi yang disediakan oleh *server*. Pada layanan *web*, *client* dapat berupa *web browser*.

Adapun kelebihan sistem informasi berbasis web sebagai berikut:

1. Dapat dijalankan dimanapun, kapanpun tanpa harus melakukan instalasi.
2. Terkait dengan isu lisensi (hak cipta), kita tidak memerlukan lisensi ketika menggunakan web-based *application*, sebab lisensi telah menjadi tanggung jawab dari web penyedia apliaksi.
3. Lintas platform seingga dapat dijalankan di semua sistem operasi, seperti *linux, windows, ios,* selama memiliki browser dan akses internet.
4. Dapat diakses diberbagai perangkat seperti: komputer dan handphone yang sudah sesuai dengan standar WAP.
5. Tidak memerlukan spesifikasi perangkat komputer yang tinggi untuk menggunakan aplikasi berbasis web, karena sebagian besar proses dilakukan di web server penyedia aplikasi berbasis web.

Selain kelebihan yang telah dipaparkan, sistem berbasis web juga memiliki kekurangan yaitu:

1. Dibutuhkan koneksi internet dan intranet yang handal dan stabil agar pada saat aplikasi dijalankan akan berjalan dengan baik dan lancar.

2. Dibutuhkan sistem keamanan yang baik dikarenakan aplikasi dijalankan secara terpusat, sehingga apabila server di pusat down maka sistem aplikasi tidak bisa berjalan.

2.8 HTML (*Hypertext Markup Language*)

HTML atau *Hypertext Markup Language* adalah bahasa yang paling banyak digunakan secara luas untuk pengembangan *website*. Seperti namanya HTML adalah bahasa *Markup*. *Markup* adalah penanda yang ditambahkan pada bagian-bagian dari sebuah dokumen untuk memberi makna khusus atau hal penting dari bagian dokumen tersebut. Dalam pengembangan sebuah web penanda ditambahkan pada struktur dokumen web dan penanda tersebut dinamakan *tag-tag* (Duckett, 2010).

Tag pada HTML ditandai dengan kurung siku "< >" untuk memulai struktur *code* dari dokumen web. Pada bagian dalam *tag* terdapat teks yang menyatakan fungsi dari *tag* tersebut. HTML bukanlah bahasa pemrograman melainkan *formatting script* yang digunakan untuk membangun fondasi/layout dari sebuah web. Untuk menulis HTML dapat menggunakan aplikasi *text editor* seperti *Notepad, Sublime Text, Visual Studio Code, Adobe Dreamweaver,* dan sebagainya.

2.9 CSS (*Casecading Style Sheet*)

CSS atau *Casecading Style Sheet* adalah kode program yang digunakan untuk menghias dan mengatur tampilan/layout halaman web agar lebih menarik dan terlihat *user friendly* (Duckett, 2010:244).

Ada beberapa *rule* yang digunakan dalam penulisan struktur kode CSS. Pengaturan atau penulisan *rule* tersebut menyesuaikan dengan keinginan *programmer.* Aturan-aturan tersebut adalah sebagai berikut:

1. *Selector*

   Menunjukan atau memilih elemen yang akan diberikan aturan CSS. Misalnya jika ingin memberi *style* pada teks paragraph maka dapat memilih elemen atau *tag "*<p>".

   Contoh penulisan *script* sebagai berikut:

   #p {pendeklarasian CSS

};

2. *Declaration*

   Aturan yang digunakan untuk menetapkan bagaimana elemen yang sebelumnya dipilih akan ditata. Penulisan *declaration* dibagi menjadi dua bagian yang dipisahkan dengan tanda titik dua ":" yaitu:

   a. *Property*

      Digunakan untuk memilih properti yang akan digunakan pada elemen atau *tag* pada dokumen HTML yang dipilih. Misalnya pemberian ukuran teks, gambar, atau video.

   b. *Value*

      Dari property yang telah dipilih maka properti tersebut dapat diberi nilai. Misalnya pemberian nilai warna hijau pada properti warna tulisan.

      Contohnya sebagai berikut:

      #p {

      Fontcolor: green;};

2.10 PHP (*PHP Hypertext Preprocessor*)

PHP atau *PHP Hypertext Preprocessor* merupakan bahasa pemrograman yang digunakan untuk membangun suatu web dinamis (Agus, 2011). PHP dirancang untuk berjalan pada sisi server sehingga diperlukan *web server*. Jika HTML digunakan untuk sisi pengguna atau *client* dan merupakan fondasi atau layout dari sebuah web, maka PHP digunakan sebagai proses sehingga adanya PHP web mudah dilakukan *maintenance*. PHP juga memungkinkan terjadinya CRUD (*create, read, update, delete*) pada web.

PHP bersifat *open source s*ehingga dapat dipakai secara cuma-cuma dan bersifat *cross platform*, yaitu dapat berjalan diberbagai sistem operasi. PHP juga dibangun sebagai modul pada *web server apache* dan sebagai *binary* yang dapat berjalan dengan CGI.

2.11 *JavaScript*

JavaScript adalah bahasa pemrograman yang digunakan dalam pemrograman *web* (Duckett, 2010: 481-482). JavaScript adalah bahasa pemrograman yang berjalan pada sisi server dan membuat web lebih interaktif serta bersifat dinamis.

Penulisan kode JavaScript ditandai dengan *tag* ‘<script type=”JavaScript”> </script> pada halaman HTML. Kode JavaScript dapat disisipkan secara internal maupun eksternal. Penulisan secara internal dapat dilakukan dengan memasukan kode di antara *tag script*. Sedangkan secara eksternal, kode berupa file yang disimpan berlainan folder maupun *link* internet dituliskan sebagai berikut: <script type = “JavaScript” src=”script/jQuery.js”> </script>.

Hal yang didapatkan jika menggunakan *JavaScript:*

1. Membaca dan menuliskan (*read & write*) elemen dan teks.
2. Memanipulasi dan memindahkan teks.
3. Menjalankan fungsi perhitungan pada data.
4. Bereaksi terhadap *events*, seperti saat pengguna menekan sebuah tombol/*button.*
5. Membaca waktu dan tanggal di komputer pengguna.
6. Menetapkan ukuran tampilan beserta resolusi layar dan web browser *version* pada komputer pengguna.
7. Melakukan pengecekan pada *input* yang diberikan oleh pengguna, seperti pada pengecekan *form validation.*

2.12 *Bootstrap*

Bootstrap adalah sebuah produk *open source* yang dikembangkan oleh Mark Otto dan Jacob Thorton. Saat merilis *Bootstrap* mereka berdua merupakan karyawan di *twitter.* Mereka membuat *Bootstrap* karena diperlukan sebuah kebutuhan standarisasi seperangkat alat untuk perancangan tampilan oleh teknisi-teknisi di *twitter.* Keunggulan dari *Bootstrap* adalah dapat menyesuaikan dengan kebutuhan website, karena dapat memilih fitur CSS dan JavaScript sesuai dengan keperluan (Spurlock, 2013). Untuk menggunakan *Bootstrap* dalam pengembangan

website dapat mendowload file *Bootstrap,* terdapat tiga folder di dalam file *Bootstrap* tersebut yaitu folder CSS, JS, dan Img.

2.13 XAMPP

XAMPP adalah paket lengkap program *web* yang dapat digunakan untuk belajar pemrograman *web,* khususnya PHP dan MySQL (Nugroho, 2013:1). XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi. XAMPP merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*) yang terdiri dari program Apache HTTP *Server*, MySQL *database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari *Cross Apache MySQL PHP and Perl.* Program ini tersedia dalam GNU (*General Public License*) dan bebas. XAMPP merupakan *web server* yang mudah digunakan yang dapat melayani tampilan web yang dinamis.

2.14 *MySQL*

MySQL secara inheren merupakan sistem dengan database jaringan, sehingga setiap klien dapat berkomunikasi dengan server yang dijalankan secara local pada mesin pengguna atau dengan server yang dijalankan pada tempat yang berlainan tanpa memandang batasan jarak (Sianipar, 2015). *MySQL* adalah salah satu sistem manajemen database yang banyak digunakan oleh pengembang perangkat lunak karena bersifat *open source.*

Server *MySQL* mengontrol akses data sehingga dapat diakses oleh banyak pengguna secara bersamaan, menyediakan akases yang cepat, dan memastikan hanya pengguna yang terotorisasi yang dapat mengakses basis data. *MySQL* menggunakan SQL (*Standard Library Language*).

MySQL dan PHP merupakan sistem yang saling terintegrasi. Maksudnya adalah pembuatan database dengan menggunakan sintak PHP dapat di buat. Sedangkan input yang dimasukan melalui aplikasi web yang menggunakan script serverside seperti PHP dapat langsung dimasukan ke database *MySQL* yang ada di server dan tentunya web tersebut berada di sebuah web server.

2.15 *PhpMyAdmin*

PhpMyAdmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL (MADCOMS, 2016: 186). Dengan menggunakan PhpMyAdmin, user dapat membuat database, membuat tabel, memasukkan, menghapus, dan mengupdate data dengan GUI dan terasa lebih mudah tanpa perlu mengetik perintah SQL secara manual.

Beberap fitur dalam PhpMyAdmin:

1. Antarmuka bebasis web
2. Impor data dari CSV dan SQL
3. Ekspor data dari berbagai format
4. Membuat grafik PDF dari tamplan dan basis data user
5. Mebuat kompleks query menggunakan *Query-by-example* (QBE)
6. Transformasi data disimpan ke dalam format yang menggunakan satu set yang telah ditetapkan seperti menampilkan data blob-data atau download-link.

2.16 *CodeIgniter*

CodeIgniter adalah *framework web* untuk bahasa pemrograman *PHP* yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab. Ellislab adalah tim kerja yang berdiri pada tahun 2002 dan bergerak dibidang pembuatan perangkat lunak dan *tool* untuk pengembangan aplikasi berbasis *web* (Raharjo, 2015:3).

2.17 Pengujian Perangkat Lunak

2.4.1 *Black Box Testing*

Black box testing atau pengujian *black box* adalah metode pengujian perangkat lunak yang memfokuskan pada keperluan fungsional apakah perangkat lunak tersebut dapat berjalan sesuai dengan tujuan awal pengembangannya. Pengujian ini memungkinkan pengembang perangkat lunak untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program.

Black box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk program (Pressman, 2010 : p495).

Black box testing berusaha untuk menemukan kesalahan dalam kategori berikut:

a. Fungsi yang tidak benar atau fungsi yang hilang
b. Kesalahan antarmuka
c. Kesalahan dalam struktur data atau akses basis data eksternal
d. Kesalahan perilaku atau kesalahan kinerja
e. Inisialisasi dan pemutusan kesalahan

Kategori error yang akan diketahui melalui *black box testing:*

a. Fungsi yang hilang atau tidak benar
b. Error dari antar muka
c. Error dari struktur data atau akses eksternal basis data
d. Error dari kinerja atau tingkah laku

2.4.2 *User Acceptance Test (UAT)*

Pengujian UAT atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwa perangkat lunak yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (*testing*) sudah bisa dianggap memenuhi kebutuhan dari pengguna.

User Aceceptence Testing merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah *staf/*karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya (Perry, 2006: 70).

Maka dari definisi tersebut, dapat dikatakan bahwa UAT merupakan pengujian yang dilakukan oleh pengguna dari sistem yang dikembangkan untuk memastikan apakah fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Proses dalam UAT adalah pemeriksaan dan pengujian terhadap hasil pekerjaan apakah item-item yang ada

dalam dokumen kebutuhan sudah atau tidak ada dalam perangkat lunak yang diuji dan apakah item yang ada tersebut telah dapat memenuhi kebutuhan penggunanya. Pengujian ini dapat dilakukan dengan berdiskusi dengan pengguna dan memberikan beberapa pertanyaan kepada pengguna mengenai sistem atau aplikasi yang telah dikembangkan untuk dilakukan evaluasi.

2.18 Kajian Terkait

Penelitian dilakukan berdasarkan penelitian yang pernah dilakukan sebelumnya. Berikut ini adalah penelitian yang serupa dengan penelitian yang akan dilakukan:

Septiani dkk (2019), melakukan penelitian tentang Perancangan Sistem Informasi Penyewaan Alat Berat yang diharapkan dapat mempermudah admin dalam pendataa dan pengelolaan data. Sistem yang dibangun menggunakan model *waterfall* dan berbasis *desktop*. Pada penelitian ini penyewa harus datang langsung untuk melakukan sewa alat berat masih dilakukan secara manual karena sistem yang dibangun berbasis *desktop* sehingga pengguna dari aplikasi ini hanya admin yang mengelola alat berat. Rachman dkk (2018), melakukan penelitian tentang Aplikasi Penyewaan Alat Berat di UPT Alat Berat Dinas Pekerjaan Umum Kota Bandung. Penelitian ini menghasilkan aplikasi yang dapat mempermudah dan mengoptimalkan proses pengelolaan alat berat dari sistem yang sudah ada sebelumnya yaitu dalam pendataan alat berat, pendataan penyewa, pendataan laporan-laporan penyewaan alat berat, pengembalian alat berat, mempercepat dalam pencarian alat berat dan mempermudah pegawai dalam mencari penyewa alat berat berat. Metode perancangan yang digunakan adalah metode *prototype*. Aplikasi yang dibangun berbasis web namun hanya digunakan pada bagian kantor saja.

Yenni dkk (2016), melakukan penelitian tentang Aplikasi Manajemen Data Alat Berat Pada PT. Swmitomo Regional Daerah Riau. Penelitian ini menghasilkan aplikasi yang digunakan untuk menunjang sistem yang sudah berjalan agar menjadi lebih optimal dalam membantu pekerjaan administrasi dalam melakukan pengelolaan alat berat. Kegiatan yang dilakukan pada aplikasi ini alah pendataan perawatan alat berat, kontarktor, dan kegiatan proyek yang menggunakan alat berat.

Berdasarkan kajian terkait maka pada maka penelitian ini bertujuan membangun sistem pengelolaan alat berat yang mengakomodir proses-proses pada pengelolaan alat berat yang meliputi kegiatan penyewaan, pendataan alat berat, penjadwalan pemakaian alat berat, perawatan alat berat, dan perekapan berkas-berkas penyewaan alat berat. Sistem juga dapat otomatis menampilkan alat-alat yang harus diservis dan merubah kondisi alat berdasarkan jumlah jam penggunaan, serta menampilkan informasi alat yang harus dikembalikan secara otomatis berdasarkan tanggal selesai pemakaian alat. Penyewa dapat menyewa alat berat dengan mudah, kapan saja, dan dimana saja. Sistem yang dibangun berbasis website menggunakan model *rapid application development* yang berfokus pada penggunaan model *protoype*.