

BAB II

TINJAUAN PUSTAKA

2.1 Studi Literatur

Adapun penelitian tentang aplikasi pengelolaan aset ini menggunakan beberapa referensi dari penelitian terdahulu yang terkait dengan judul pada penelitian ini, yaitu:

1. Penelitian yang dilakukan oleh Heri Sudibyo (2014) berjudul “Sistem Informasi Pengelolaan Aset Desa Berbasis *Web* Pada Desa Purwosari”. Penelitian tersebut dilakukan dengan tujuan untuk menghasilkan sebuah sistem informasi pengelolaan aset desa yang lebih cepat, tepat guna, efektif dan efisien pada Kantor Desa Purwosari, untuk mengatasi permasalahan dari sistem pengelolaan aset secara konvensional seperti kesalahan dalam pencatatan serta resiko kehilangan data. Penelitian ini juga bertujuan agar data aset desa tersebut dapat tersimpan secara digital dengan memanfaatkan *database* sehingga akan memudahkan proses pencarian kembali data apabila sewaktu-waktu akan diperlukan. Keterkaitan antara penelitian tersebut dengan penelitian ini adalah dalam hal bagaimana membuat atau menyediakan sebuah sistem informasi pengelolaan aset desa yang baik, memiliki fungsi pengelolaan aset desa yang sesuai dengan kebutuhan data dan informasi saat ini.
2. Penelitian yang dilakukan oleh Khusnul Hotimah, Dedy Kurnia Sunaryo dan Jasmani (2019) berjudul “Sistem Informasi Geografis Berbasis *Web* Untuk Pemerintah Desa (Studi Kasus: Desa Dukuwaluh, Kecamatan Kembaran, Kabupaten Banyumas). Adapun hasil dari penelitian tersebut adalah sistem informasi geografis (SIG) desa yang dapat menyajikan informasi berupa demografi, tanah milik desa, dan informasi pembangunan Desa Dukuwaluh. Adapun manfaat dari pembuatan SIG tersebut adalah untuk membantu pemerintahan desa dalam menyajikan data desa dan informasi pembangunan serta membantu masyarakat untuk mengetahui informasi mengenai data desa dan pembangunannya. Keterkaitan antara penelitian tersebut dengan penelitian ini adalah sama-sama membahas tentang bagaimana menampilkan data desa (dalam hal ini aset) sesuai dengan kebutuhan desa dalam bentuk sistem informasi geografis berbasis *web*.

3. Penelitian yang dilakukan oleh Amelia Fadhila (2017) berjudul “Pembuatan Sistem Informasi Geografis (SIG) Berbasis *Web* Untuk Pemetaan Pemetaan Usaha Mikro Kecil Menengah (UMKM) Di Kabupaten Blitar”. Penelitian ini dilakukan untuk memecahkan permasalahan yang terjadi seperti kurangnya informasi mengenai wilayah UMKM, belum adanya sistem yang dapat membantu dalam kegiatan *monitoring* dan evaluasi terhadap UMKM wilayah Blitar, memudahkan pemerintah daerah setempat dalam pengelompokkan jenis industri. Hasil dari penelitian ini berupa *web* SIG yang memiliki kemampuan untuk menampilkan informasi Pemetaan UMKM wilayah Blitar dengan dilengkapi fitur pencarian berdasarkan nama UMKM, kategori UMKM dan dengan penelitian ini yaitu dalam hal hasil penelitian yang berupa sistem informasi yang pendataan dan pengelolaan yang dilengkapi dengan informasi pemetaan objek pada penelitian dan dalam hal penggunaan *Google Map API*.

2.2 Landasan Teori

2.2.1 Aplikasi

Menurut Karnovi & Habibi (2020), aplikasi adalah sebuah program yang siap pakai yang bisa dipakai untuk menjalankan sebuah perintah dari pengguna aplikasi itu sendiri. Karnovi & Habibi (2020) juga menyatakan bahwa aplikasi adalah pemecah masalah yang memakai salah satu teknik pemrosesan data aplikasi yang mengacu pada sebuah komputerisasi atau *smartphone* yang diinginkan atau diharapkan.

Berdasarkan pengertian aplikasi di atas, maka dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang dibuat dengan tujuan untuk mempermudah penggunaanya dalam mengerjakan suatu pekerjaan berdasarkan kemudahan dengan memanfaatkan teknik pemrosesan data yang telah di program sebelumnya.

2.2.2 Manajemen Aset

Menurut Siregar (2004:65) “manajemen aset adalah suatu profesi atau keahlian yang belum sepenuhnya berkembang atau populer di lingkungan pemerintahan maupun di satuan kerja atau instansi”. Menurut Siregar (2004:517) "Manajemen aset merupakan rangkaian aktivitas yang dilakukan dalam mengelola aset agar dapat memberikan manfaat secara optimal dengan cara yang efektif dan efisien".

Menurut Gima (2013:98), manajemen aset adalah suatu ilmu dan seni untuk memandu pengelolaan kekayaan yang mencakup proses perencanaan

kebutuhan aset, mendapatkan, inventarisasi, legal audit, menilai, mengoperasikan, memelihara, membaharukan atau menghapuskan, hingga mengalihkan aset secara efektif dan efisien.

Menurut Gima (2013:43) berpendapat bahwa manajemen aset adalah suatu ilmu dan seni untuk melakukan kegiatan pengelolaan aset yang mencakup proses merencanakan kebutuhan aset, mendapatkan, menginvestasi, legal audit, menilai, mengoperasikan, memelihara, membaharui atau menghapuskan hingga mengalihkan aset secara efektif dan efisien.

Pada dasarnya tujuan asset management adalah untuk membantu perusahaan dalam mengambil keputusan yang tepat sehingga aset dapat dikelola secara efektif dan efisien. Adapun beberapa tujuan manajemen aset adalah sebagai berikut (Gima, 2013:23):

- a. Untuk memastikan status kepemilikan suatu aset.
- b. Untuk menginventarisasi kekayaan dan masa pakai aset yang dimiliki.
- c. Untuk menjaga agar nilai aset tetap tinggi dan memiliki usia hidup yang panjang.
- d. Untuk meminimalisasi biaya selama umur suatu aset.
- e. Untuk memastikan suatu aset dapat menghasilkan keuntungan yang maksimum.
- f. Untuk mencapai penggunaan dan pemanfaatan aset secara optimal.
- g. Untuk keperluan pengamanan aset.
- h. Sebagai acuan dalam menyusun neraca dalam akuntansi

Dalam pelaksanaan asset management terdapat 8 tahapan yang harus dilakukan sehingga siklus dapat terbentuk. Adapun beberapa tahapan manajemen aset adalah sebagai berikut (Gima, 2013:25):

- a. Perencanaan Kebutuhan Aset

Ini adalah tahap awal proses asset management dimana dilakukan perencanaan mengenai apa saja hal yang dibutuhkan dalam mengelola aset. Misalnya kebutuhan untuk pengadaan, inventarisasi, perawatan, dan lain sebagainya.

- b. Pengadaan Aset

Pada tahap ini dilakukan kegiatan pengadaan aset, misalnya barang atau jasa yang diperoleh dengan biaya sendiri atau pihak lain.

c. Inventarisasi Aset

Pada tahap ini terdapat rangkaian kegiatan berupa identifikasi kualitas dan kuantitas aset, baik secara fisik/ non fisik maupun secara yuridis/ legal. Masing-masing aset didokumentasikan dan diberi kode tertentu untuk keperluan pengelolaan aset tersebut.

d. Legal Audit Aset

Pada tahap ini dilakukan audit mengenai status aset, sistem dan prosedur pengadaan, sistem dan alur pengalihan. Selain itu, identifikasi kemungkinan terjadinya masalah legalitas juga dilakukan pada tahap ini dan sekaligus mempersiapkan solusinya.

e. Pengoperasian dan Pemeliharaan Aset

Pada tahap ini setiap aset yang dimiliki digunakan untuk melakukan tugas dan pekerjaan sesuai dengan fungsinya untuk mencapai tujuan perusahaan.

f. Penilaian Aset

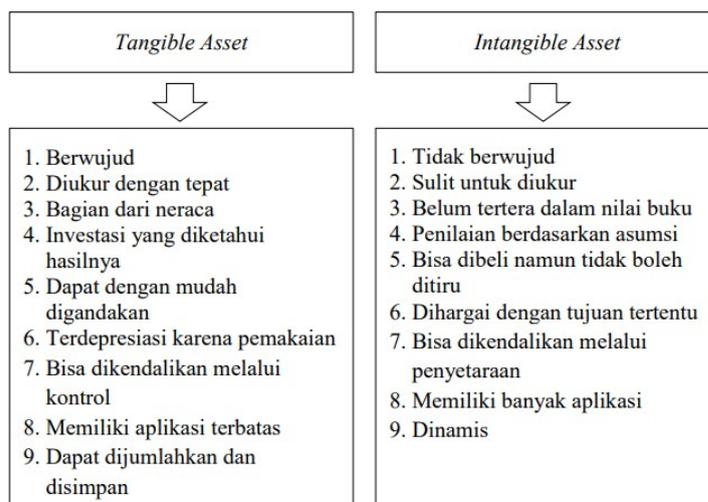
Pada tahap ini pihak asset management menentukan nilai aset yang dimiliki sehingga perusahaan mengetahui dengan jelas nilai kekayaan yang dimiliki, yang dialihkan maupun yang dihapuskan.

g. Penghapusan Aset

Pada tahap ini perusahaan akan menilai aset apa saja yang dianggap tidak menguntungkan dan akan dihapuskan.

h. Pembaharuan Aset

Pada banyak kasus aset yang dianggap tidak produktif bisa diperbaharui sehingga dapat dimanfaatkan lagi sampai umur ekonomisnya berakhir. Pembaharuan atau peremajaan tersebut dilakukan dalam bentuk perbaikan atau penggantian suku cadang sehingga aset dapat bekerja seperti kondisi semula.



Sumber: Siregar (2016:39)

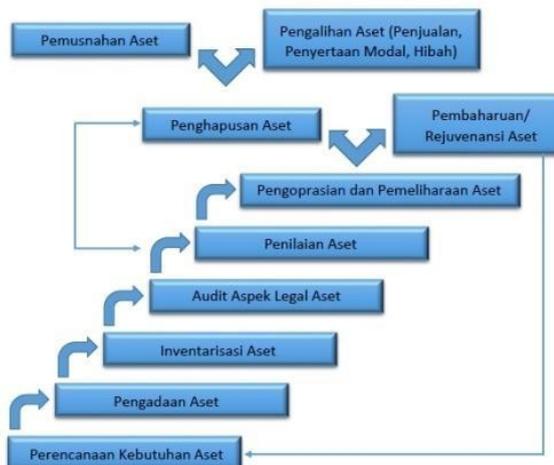
Gambar 2.1 Perbedaan *Tangible Asset* dan *Intangible Asset*

Terdapat tiga tujuan utama dalam kegiatan pengelolaan aset. Berikut merupakan penjelasan dari tujuan kegiatan manajemen/ pengelolaan aset (Siregar, 2004:77):

- a. Efisiensi pemilikan dan pemanfaatan Pengelolaan aset yang baik dan sesuai dengan tugas pokok dan fungsinya akan berimplikasi pada pemanfaatan aset yang optimal.
- b. Nilai ekonomis serta potensi yang terjaga Aset yang dikelola dengan baik akan menjaga nilai ekonomis aset tetap stabil dan akan meningkatkan keuntungan dari segi pendapatan maupun pencapaian tujuan perusahaan/ organisasi.
- c. Objektivitas dalam pengawasan, pengendalian peruntukan, penggunaan, serta pengalihan kekuasaan. Pengawasan terhadap aset akan lebih terarah apabila pengelolaan aset dilakukan dengan baik. Pengawasan juga dilakukan dengan tujuan membantu dalam pencapaian tujuan dari aset tersebut.

2.2.3 Siklus Aset

Siklus hidup aset merupakan suatu komponen penting yang harus diperhatikan dalam pelaksanaan pengelolaan aset. Menurut Gima (2013:45) “siklus hidup aset memiliki beberapa fase yang berurutan, diawali dari tahap perencanaan kebutuhan, pengadaan, inventarisasi, lalu proses legal audit, penilaian, dan pengoprasian aset termasuk didalamnya proses pemeliharaan, pembaharuan/ rejuvenasi dan penghapusan”. Siklus aset ini diakhiri dengan pemusnahan aset atau pengalihan aset yang dapat dilakukan dengan cara penjualan, penyertaan modal ataupun penghibahan aset. Penjelasan lebih lanjut dapat dilihat pada gambar 2.2 di bawah ini (Sugiama, 2013:27):



Sumber: Siklus alur aset (Sugiama, 2013:27)

Gambar 2.2 Siklus Manajemen Aset

Berikut ini adalah uraian dari tahapan siklus alur manajemen aset (Sugiama, 2013:260-264):

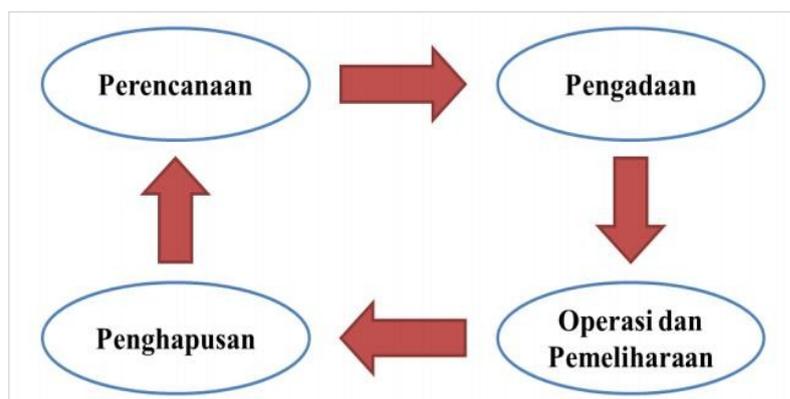
- a. Perencanaan kebutuhan aset kegiatan perencanaan kebutuhan aset merupakan langkah awal sekaligus langkah paling strategis pada proses manajemen aset. Perencanaan kebutuhan aset harus dapat memberikan informasi serta menjawab tingkat kebutuhan perusahaan atas aset tetap yang akan dikelola baik pengelolaan jangka pendek maupun jangka panjang. Perencanaan kebutuhan yang dilakukan secara tepat diharapkan dapat meminimalkan resiko kerugian perusahaan dimasa yang akan datang serta mampu meningkatkan keuntungan bagi perusahaan. Proses perencanaan kebutuhan aset tetap untuk tiap perusahaan memiliki tingkat kompleksitas yang berbeda-beda, tergantung pada skala aset tetap yang dibutuhkan.
- b. Pengadaan aset Pengadaan aset adalah serangkaian kegiatan untuk memperoleh atau mendapatkan aset yang dilaksanakan baik dengan melaksanakan sendiri secara langsung oleh pihak internal perusahaan, maupun oleh pihak luar yang ditunjuk sebagai mitra penyedia/pemasok aset bersangkutan.
- c. Inventarisasi “Berdasarkan Peraturan Pemerintah Nomor 27 Tahun 2014 Tentang Pengelolaan Barang Milik Daerah/ Negara, inventarisasi adalah kegiatan untuk melakukan pendataan, pencatatan, dan pelaporan hasil pendataan Barang Milik Daerah/ Negara.

- d. Legal Audit aset “Legal audit aset atau uji tuntas hukum adalah pemeriksaan (audit) untuk mendapatkan gambaran jelas dan menyeluruh terutama mengenai status kepemilikan, sistem dan prosedur penguasaan (penggunaan dan pemanfaatan), pengalihan aset, mengidentifikasi kemungkinan terjadinya berbagai permasalahan hukum, serta mencari solusi atas masalah hukum tersebut. Kegiatan legal audit aset merupakan salah satu kegiatan penting dalam pengelolaan aset, mengingat seringnya terjadi permasalahan sengketa penguasaan atas aset yang terjadi seperti pengakuan hak milik aset oleh pihak lain, penyerobotan batas lahan dan masalah pengguna lahan ilegal.”
- e. Penilaian aset “Berdasarkan Peraturan Pemerintah Nomor 27 Tahun 2014 Tentang Pengelolaan Barang Milik Daerah/ Negara, penilaian adalah proses kegiatan untuk memberikan suatu opini nilai atas suatu objek penilaian berupa Barang Milik Negara/Daerah pada saat tertentu.” Aset tetap termasuk ke dalam kategori jenis aset berwujud, penilaian yang dilakukan akan memberikan estimasi penilaian terhadap aset berdasarkan kondisi aset pada saat kegiatan penilaian dilakukan.
- f. Pengoperasian dan pemeliharaan aset “Pengoperasian aset adalah sebuah proses atau serangkaian kegiatan yang secara khusus terdiri dari langkah-langkah mendasar dalam sebuah pekerjaan atau kumpulan pekerjaan untuk memfungsikan/memakai aset bersangkutan. Sedangkan pemeliharaan aset sebuah sistem yang mencakup kombinasi dari sekumpulan aktivitas yang dilengkapi oleh beragam sumberdaya untuk menjamin agar aset yang bersangkutan dapat berfungsi sebagaimana diharapkan. Pengoperasian aset dapat berupa penggunaan dan pemanfaatan aset tetap, penggunaan aset yakni pengoperasian aset tetap sesuai dengan perencanaan kebutuhan aset yang telah ditetapkan sebelumnya, sedangkan pemanfaatan aset yakni pengoperasian aset yang keluar dari tupoksi aset itu sendiri. Pemanfaatan aset tetap dapat berupa sewa, Kerjasama Pemanfaatan aset (KSP), Kerjasama Operasi aset, Bangun Serah Guna aset (BSG), Bangun Guna Serah aset (BGS) dan pinjam pakai aset lahan.”
- g. Pembaharuan/ rejuvenasi Setelah aset dioperasikan maka tahap selanjutnya yakni tahap pembaharuan/rejuvenasi aset. Tahap pembaharuan/rejuvenasi aset merupakan proses kegiatan yang dilaksanakan ketika kinerja aset sudah tidak

sesuai dengan yang diharapkan. Kegiatan rejuvinasi berarti melakukan tindakan agar aset kembali seperti keadaan semula atau *to restore to youthful vigor or appearance*. Namun fase ini tidak berlaku untuk aset lahan, mengingat lahan merupakan jenis aset yang tidak dapat dihancurkan/di rejuvenasi.

- h. Penghapusan aset dan pengalihan Aset yang sudah tidak memungkinkan lagi diperbaharui maka harus dihapuskan atas pertimbangan ekonomi atau fungsinya. Tindak lanjut dari proses penghapusan yakni pengalihan aset tetap yang dapat dilakukan dengan cara menjual aset tersebut, menghibahkan, atau melakukan penyertaan aset bersangkutan.

Menurut Hindrawan dkk, (2006:119) “siklus hidup fisik dari suatu aset atau kelompok aset memiliki empat fase, yaitu perencanaan, pengadaan (*acquisition*), operasi dan pemeliharaan, serta penghapusan (*disposal*)”. Gambar 2.3 menunjukkan siklus hidup aset (Hindrawan, 2006:119).



(Sumber: Hindrawan, dkk, 2006)

Gambar 2.3 Siklus Hidup Aset

2.2.4 Sistem Informasi Geografis

Sistem informasi geografis merupakan sebuah data yang mengacu pada posisi, objek dan hubungan diantaranya dalam ruang bumi. SIG secara khusus dibuat untuk mengelola data yang bereferensi keruangan dan merupakan sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis seperti data yang diidentifikasi menurut lokasinya dalam sebuah *database*. SIG terdiri dari suatu komponen yang terdiri dari perangkat keras, perangkat lunak, sumber daya manusia dan data yang bekerja bersama secara efektif untuk memasukkan, menyimpan, memperbaiki, memperbaharui, mengelola, memanipulasi, mengintegrasikan, menganalisa dan

menampilkan data dalam suatu informasi berbasis geografis. Adanya aplikasi SIG dapat menjawab beberapa pertanyaan mengenai lokasi, kondisi, trend, pola dan pemodelan dan penayangan data lokasi secara digital bahkan dapat diintegrasikan secara beragam mulai dari citra satelit, foto, dan peta (Hutagulung, 2020:79) .

2.2.5 WebGIS

Ovan & Saputra (2020) menyatakan bahwa aplikasi berbasis *web* adalah sebuah aplikasi yang dapat diakses menggunakan *web browser* atau penjelajah *web* yaitu tidak memerlukan lisensi pada saat menggunakan aplikasi tersebut. Abdulloh (2016), *website* adalah sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk digital, baik berupa teks, gambar, video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet.

SIG berbasis *web* adalah sebuah aplikasi sistem informasi geografis yang dapat diaplikasikan dan dijalankan dengan menggunakan jaringan internet pada suatu *web browser*. Aplikasi ini berjalan dengan internet dalam suatu jaringan global, dalam jaringan LAN yang telah memiliki *web server*. Dengan menggunakan teknologi *WebGIS* dapat dengan mudah ditampilkan peta wisata di wilayah tertentu yang dilengkapi dengan ulasan terhadap objek dan disajikan dalam sebuah *website*. Keunggulan dari *webGIS* adalah lebih mudah dalam menyajikan data yang biasa diakses secara online tanpa menggunakan bantuan software GIS (Hutagulung dan Azlan, 2020:93).

2.2.6 Mapbox

Mapbox adalah sebuah *web service* yang menyediakan informasi tentang *geocoding* dan *direction* dari dua buah *node*. *Mapbox* dipilih sebagai *web service* dikarenakan penggunaan *API* cenderung lebih mudah, serta informasi yang dihasilkan cukup lengkap, mulai dari *distance*, *coordinates*, *street name*, hingga *way points*. dengan memanfaatkan *mapbox*, dapat menentukan jarak kedua buah titik yang sudah dikombinasikan (Atmojo, 2018).

Mapbox API adalah salah satu layanan penyedia peta digital untuk berbagai *platform*. Untuk data pemetaan sendiri *Mapbox* menggunakan *base tiles* dari *OpenStreetMaps* dan juga dari beberapa sumber lainnya seperti *NASA*, *DigitalGlobe*, dan *USGS* (Rais & Setiawan, 2018).

2.2.7 Database

Database digunakan untuk mengatasi permasalahan pengolahan data dengan metode pengarsipan berkas dan *database* sangat diperlukan dalam membangun sistem informasi pada sebuah perusahaan, sehingga dengan *database* akan dapat meningkatkan kinerja perusahaan (Yanto, 2016:5). Berikut pengertian *database* menurut beberapa ahli, yaitu:

- a. Sebuah sistem yang berfungsi untuk menyimpan dan mengolah sekumpulan data (Enterprise, 2018:6).
- b. Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah (Yanto, 2016:5).
- c. Kumpulan berbagai data dan informasi yang ada dan disimpan di suatu media tertentu, umumnya pada komputer (Habibi & Aprilian, 2019:19).
- d. Suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta memelihara data secara lengkap pada sebuah organisasi atau perusahaan, sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk kepentingan proses pengambilan keputusan (Lubis, 2016:3)

1. Struktur *Database*

Adapun struktur *database* beserta penjelasannya sebagai berikut (Habibi dan Aprilian, 2019:20):

- a. *Hierarchical, software* DBMS didesain untuk menggunakan struktur data tertentu agar dapat memberi para pemakai akhir akses yang cepat dan mudah ke informasi yang disimpan dalam *database*.
- b. *Network*, struktur ini memungkinkan hubungan banyak ke banyak antar catatan, model jaringan dapat mengakses elemen data dengan mengikuti salah satu dari beberapa jalur, karena elemen data atau catatan apapun dapat dihubungkan ke banyak elemen.
- c. *Relational*, model relasional adalah model yang paling banyak digunakan dari ketiga struktur *database*. Model ini digunakan oleh kebanyakan *software* DBMS mikrokomputer dan juga sistem skala menengah dan

mainframe. Dalam model ini, semua elemen data dalam *database* disimpan dan dipandang dalam bentuk tabel-tabel sederhana.

d. *Object Oriented*.

e. *Mult Dimensional*.

2. Tujuan Membangun *Database*

Adapun tujuan dibangun *database* sebagai berikut (Yanto, 2016:72):

- a. Kecepatan dan Kemudahan, *database* memungkinkan untuk dapat menyimpan data atau melakukan perubahan terhadap data atau menampilkan kembali data tersebut secara lebih efektif.
- b. Efisiensi Ruang Penyimpanan, penggunaan *database* memungkinkan efisiensi ruang penyimpanan dapat dilakukan dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi antar kelompok data yang saling berhubungan.
- c. Keakuratan, pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan, keunikan data, secara ketat dapat diterapkan dalam sebuah basis data sangat berguna untuk menekan ketidakakuratan penyimpanan data.
- d. Ketersediaan, pemanfaatan jaringan komputer membuat data yang berada pada suatu lokasi juga dapat diakses bagi lokasi lain.
- e. Kelengkapan, kelengkapan data yang disimpan memiliki sifat relatif, bias jadi saat ini dianggap sudah lengkap, namun belum tentu pada suatu saat akan dianggap lengkap.
- f. Keamanan, aspek keamanan dapat diterapkan dengan ketat, dengan begitu akan dapat menentukan pemakai basis data serta objek-objek di dalamnya.
- g. Kebersamaan Pemakaian, *database* yang dikelola dengan aplikasi *multiuser* dapat memenuhi kebutuhan ini.

2.2.8 Bahasa Pemrograman

Bahasa pemrograman merupakan kumpulan aturan yang disusun sedemikian rupa sehingga memungkinkan pengguna komputer membuat program yang dapat dijalankan dengan aturan tersebut dan dapat dikelompokkan seperti pendekatan dari notasi bahasa pemrograman tersebut, apakah lebih dekat ke bahasa mesin atau bahasa manusia (Utami & Sukrisno, 2005:7).

Bahasa pemrograman merupakan prosedur penulisan, ada tiga *record* dalam penulisan bahasa *pemrograman* (Fridayanthie & Mahdiati, 2016):

1. *Syntax* adalah aturan penulisan bahasa tersebut (tata bahasa).
2. *Semantic* adalah arti atau maksud yang terkandung di dalam *statement* tersebut.
3. Kebenaran logika adalah berhubungan dengan benar tidaknya urutan *statement*.

Berikut adalah beberapa bahasa pemrograman yang digunakan dalam perancangan sistem informasi pada skripsi ini yaitu HTML, PHP, CSS, dan SQL.

2.2.8.1 Hypertext Markup Language (HTML)

Menurut Jumardi (2019:15) HTML adalah “sebuah bahasa yang digunakan untuk membuat halaman *web*, menampilkan berbagai informasi didalam sebuah penjelajah *web internet* dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.”

Menurut Jubilee Enterprise (2016:45) HTML adalah “sebuah teks berbentuk link dan mungkin juga foto atau gambar yang saat di klik, akan membawa si pengakses internet dari satu dokumen ke dokumen lainnya”. Sedangkan menurut Setiawan & Mulyanti (2020) HTML adalah “sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman *website* yang dikembangkan untuk membuat halaman *website* yang dapat diakses atau ditampilkan menggunakan Web Browser”.

Menurut Jubilee Enterprise (2016:47) HTML adalah “sebuah teks berbentuk link dan mungkin juga foto atau gambar yang saat di klik, akan membawa si pengakses internet dari satu dokumen ke dokumen lainnya.” Adapun struktur dasar HTML menurut Jubilee Enterprise (2018:12) yaitu:

1. Tag DTD atau DOCTYPE
DTD (*Document Type Declaration*) merupakan tag paling awal dari struktur HTML, yang berfungsi untuk memberitahu *browser* bahwa dokumen berjenis HTML.
2. Tag HTML
Tag ini merupakan tag pembuka dari seluruh halaman *web* yang akan dibuat. Selanjutnya semua kode HTML yang membentuk desain sebuah halaman *website* harus berada dalam tag HTML ini.
3. Tag HEAD

Tag ini berfungsi secara teknis, bagian atau elemen yang ditulis pada tag HEAD umumnya berbentuk keterangan teknis seperti definisi judul halaman, kode-kode CSS, JavaScript, deskripsi halaman, dan kode-kode lainnya yang tidak terlihat sebagai salah satu desain di dalam sebuah halaman.

4. Tag BODY

Tag BODY digunakan untuk meletakkan semua elemen yang akan terlihat di dalam halaman *website* pada saat halaman itu diakses oleh *browser*.

2.2.8.2 Hypertext Preprocessor (PHP)

Menurut Sidik (2017) PHP merupakan “bahasa pemrograman *script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan *editor* teks atau editor HTML.”

Menurut Zahni dan Warman (2013:) PHP merupakan “bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*, artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja.”

Menurut Anhar (2017:3) PHP merupakan “*script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*”. Menurut Jubilee Entreprise (2017:1) PHP merupakan “bahasa pemrograman berjenis *server-side*, hal ini berarti PHP akan diproses oleh *server* yang olahannya akan dikirim kembali ke *browser*.”

PHP biasanya digunakan untuk membuat situs *web* dinamis dan sebagai bahasa pemrograman pelengkap untuk menghasilkan *web* yang interaktif, agar dapat menyesuaikan tampilan konten tergantung dengan situasi, menyimpan data dalam *database*, membuat halaman yang berubah-ubah sesuai *input* dari *user*, dan memproses *form*. Untuk pembuatan *web* kode PHP biasanya disisipkan kedalam dokumen HTML atau disebut sebagai *scripting language* atau pemrograman *script* (Habibi dan Aprilian, 2019:6)

Adapun fungsi dari PHP menurut Habibi dan Aprilian (2019:6) sebagai berikut:

1. Mempersingkat Tatanan HTML dan CSS

Untuk membangun sebuah halaman *web* dinamis, PHP dapat berfungsi untuk mempersingkat penggunaan tatanan HTML dan CSS serta dapat mengatur beberapa baris yang dibutuhkan atau ditampilkan.

2. *Input Data*

Bahasa pemrograman PHP menjadikan *user* dapat memasukkan data dan menyimpan dalam sistem *database*.

3. Pengelolaan *Cookie* dan *Session*

Cookie dan *session* digunakan untuk menyimpan informasi pengguna. Sebagai contoh proses *cookie* pada saat sistem menyimpan *username* dan *password* di *browser* sehingga *user* tidak perlu mengisinya berulang-ulang.

4. *Compress Text*

PHP memungkinkan programmer untuk mengompresi teks yang panjang menjadi lebih pendek dengan fungsi `gzcompress ()` dan mengembalikannya dengan fungsi `gzuncompress ()`.

2.2.8.3 Cascading Sheet Style

Menurut Rerung (2018:133) CSS adalah “bahasa yang dapat digunakan untuk mendefinisikan bagaimana suatu bahasa *markup* ditampilkan pada suatu media dimana bahasa *markup* ini salah satunya adalah HTML”. Menurut Rerung (2018:133) CSS juga dapat dikatakan “kumpulan kode yang untuk mendesain halaman *website* agar lebih menarik dilihat”.

CSS digunakan oleh para pengembang perangkat lunak berbasis *website* untuk menentukan warna, tata letak *font*, dan semua aspek yang mendukung di halaman *web* di situs. Cara kerja CSS sama halnya dengan HTML tetapi di dalam kode program HTML hanya menambahkan baris kode untuk pemanggilan dokumen dari CSS yang dimaksud (Azminuddin et al., 2019:36). CSS dapat menghasilkan tampilan *web* yang indah dan menarik. Bahasa pemrograman CSS lebih mudah dan ringkas dari pada menggunakan atribut di tiap tag HTML. CSS juga bisa digunakan untuk menciptakan *web* dengan tampilan yang dinamis dan fleksibel (Salamah, 2021).

2.2.7.4 JavaScript

Menurut Constantianus dan Suteja (2005) *JavaScript* adalah kode-kode program kecil yang dapat digunakan untuk membuat halaman *web* terlihat dinamis. Dengan menggunakan *Javascript* dapat menambahkan beberapa fitur yang dapat membuat tampilan lebih menarik serta dapat juga membatasi akses dari pengguna.

Dengan *JavaScript*, navigasi menu bias diatur efek grafisnya, seperti *scroll down* menu *header web*.

2.2.8.5 Structured Query Language (SQL)

Menurut Fridayanthie & Mahdiati (2016) SQL merupakan “bahasa terstruktur yang digunakan untuk mengelola *database* beserta isinya”. SQL merupakan suatu bahasa permintaan terstruktur karena pada penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi bernama ANSI. SQL adalah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional, bahasa ini merupakan bahasa standar yang digunakan dalam pengelolaan basis data relasional. SQL dapat juga diartikan sebagai antarmuka standar untuk sistem pengelolaan basis data relasional, sehingga SQL memungkinkan seorang pengguna untuk mengakses informasi tanpa mengetahui dimana lokasinya atau bagaimana informasi tersebut disusun. SQL juga merupakan bahasa pemrograman yang dirancang khusus untuk mengirimkan suatu perintah *query* terhadap sebuah *database* (Mardiani et al., 2016).

Adapun perintah dasar SQL didefinisikan kedalam tiga jenis yaitu (Rerung, 2018:27):

1. DDL (*Data Definition Language*), paket bahasa yang merumuskan tentang apa dan bagaimana *database* dibentuk. Hasil kompilasi dari perintah DDL adalah satu set dari tabel yang disimpan dalam *file* khusus disebut *data dictionary* atau *directory*. DDL dapat mendefinisikan *database*, tabel dan *view* dengan pernyataan *CREATE*, *ALTER* dan *DROP*.
2. DCL (*Data Control Language*), sebuah skema basis data yang digunakan untuk mengatur hak-hak pada objek basis data. DCL dapat membuat perintah-perintah yang akan digunakan untuk pengaturan hak seperti *GRANT* dan *REVOKE*.
3. DML (*Data Manipulation Language*), berupa jenis instruksi SQL yang berkaitan dengan data yang ada dalam tabel, tentang bagaimana memasukkan data, menghapus, memperbarui, serta membaca data yang tersimpan dalam *database*. Perintah DML SQL meliputi *INSERT*, *SELECT*, *UPDATE* dan *DELETE*.

2.2.9 Pengujian Perangkat Lunak

Menurut Hariyanto (2004) pengujian perangkat lunak merupakan “proses pemeriksaan atau evaluasi sistem komponen sistem secara manual atau otomatis untuk

memverifikasi apakah sistem memenuhi kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan antara hasil yang diharapkan dengan hasil yang terjadi.”

Menurut Pressman & Maxim (2014:459), *blackbox testing* merupakan pengujian tingkah laku yang memusatkan pada kebutuhan fungsional perangkat lunak. Teknik pengujian *black box* memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi yang tidak benar, kesalahan antarmuka, kesalahan struktur data, kesalahan inisialisasi dan akhir program.

Adapun proses pengujian *black box testing* menurut Jaya (2018) sebagai berikut:

- a. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
- b. Memilih jenis *input* yang memungkinkan menghasilkan *output* benar serta jenis *input* yang memungkinkan *output* salah pada perangkat lunak yang sedang di uji.
- c. Menentukan *output* dari jenis *input*.
- d. Melakukan pengujian *input-input* yang telah diseleksi.
- e. Melakukan pengujian.
- f. Membandingkan *output* yang dihasilkan dengan *output* yang diharapkan.
- g. Menentukan fungsionalitas yang seharusnya ada pada perangkat lunak yang diuji.

2.3 Tools Pendukung

2.3.1 Unified Modelling Language (UML)

Menurut Wanto et al.,(2020:46) pemodelan data merupakan “tahap untuk menerjemahkan keadaan data yang ada di dunia nyata untuk diklasifikasikan ke dalam struktur data”. Pemodelan data pada penelitian ini menggunakan ERD (*Entity Relationship Diagram*) dan UML (*Unified Modelling Language*).”

Adapun pemodelan data yang digunakan pada penelitian ini adalah UML, berikut beberapa definisi tentang UML yaitu:

- a. UML merupakan teknologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Hendini, 2016).

- b. UML merupakan sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem (Mulyani, 2016:244)
- c. UML merupakan salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Fridayanthie dan Mahdiati, 2016).

Beberapa literatur menyatakan bahwa UML memiliki sebelas diagram yang terbagi dalam dua kelompok yaitu 7 diagram struktur (*structure diagram*) yang merupakan diagram yang menggambarkan struktur statis suatu sistem. 4 diagram perilaku (*behavior diagram*) yang menunjukkan dinamika suatu sistem tentang bagaimana kolaborasi dan interaksi antar objek serta kondisi internal suatu objek (Hidayat, 2019:50).

Namun pada penelitian ini, penulis hanya menggunakan empat jenis diagram yaitu *usecase diagram*, *activity diagram*, *class diagram* dan *sequence diagram*. Adapun penjelasan dari diagram-diagram tersebut yaitu:

a. *Use Case Diagram*

Use Case mendeskripsikan sebuah interaksi antara satu aktor atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem yang akan dibuat (Fridayanthie dan Mahdiati, 2016). *Use Case diagram* menggambarkan fungsionalitas dari sebuah sistem. *Use Case* bermakna sebagai cara-cara aktor menggunakan sistem. Tujuan utama diagram ini adalah membantu pengembangan model untuk memvisualisasikan kebutuhan fungsional sistem termasuk relasi para aktor terhadap proses-proses esensial yang ada. Aktor adalah mereka yang berinteraksi dengan sistem yang akan dikembangkan (Purnomo, 2019:66).

b. *Activity Diagram*

Activity diagram digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada di perangkat lunak. *Activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Fridayanthie dan Mahdiati, 2016). Menurut Kesuma & Kholifah (2019) *activity diagram* merupakan “state diagram yang khusus, dimana sebagian

besar transisi di *trigger* oleh selesainya *state* sebelumnya (*internal processing*).”

c. *Class Diagram*

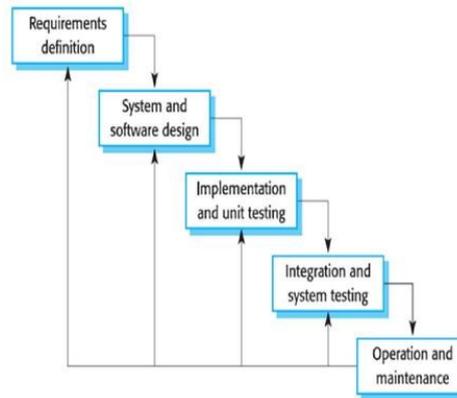
Class diagram dipakai untuk menunjukkan bagaimana entitas yang berbeda terkait satu sama lain, *class diagram* menunjukkan struktur statis dari sebuah model. *Class* dipakai untuk menggambarkan sebuah abstraksi dari sehimpunan objek yang serupa, objek mempunyai kesamaan struktural atau mempunyai atribut yang sama dan mempunyai kesamaan perilaku atau operasi-operasi yang sama. *Class* dapat berdiri sendiri, namun pada umumnya memiliki hubungan dengan kelas-kelas lainnya yang berupa asosiasi, agregasi, dan generalisasi atau spesialisasi (Purnomo, 2019:67).

d. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan atau perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antara objek (Kesuma dan Kholifah, 2019). *Sequence diagram* menyajikan bagaimana objek berkomunikasi satu sama lain sepanjang waktu simulasi. Ide utamanya adalah menyajikan interaksi antara objek pada suatu sekuens atau urutan tertentu, dan sekuens dimulai dari awal sampai akhir proses. *Sequence diagram* tidak perlu menyajikan keseluruhan interaksi yang mungkin, cukup kalau mengambil salah satu kemungkinan interaksi diantara objek-objek tersebut (Adityo, 2019:71).

2.3.2 Waterfall

Model *waterfall* merupakan model klasik yang sederhana dengan aliran sistem yang linier-keluaran dari tahap sebelumnya merupakan masukan untuk tahap berikutnya. Model *waterfall* memiliki proses berurutan dan terus mengalir ke bawah (seperti air terjun) yang melewati fase-fase *requirement* (analisa kebutuhan), *desain* sistem (*system design*), *coding* dan *testing*, penerapan program, pemeliharaan (Habibi dan Karnovi, 2020:31).



Sumber: Habibi dan Karnovi (2020:31)

Gambar 2.4 Metode *Waterfall*

Berikut adalah penjelasan dari tahapan-tahapan *waterfall* sebagai berikut (Habibi dan Karnovi, 2020:33):

1. *Requirement*

Langkah ini merupakan analisis terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau *study literature*. *System analyst* akan menggali informasi dari *user* sehingga akan tercipta sebuah sistem yang bisa melakukan tugas-tugas yang diinginkan oleh *user*.

2. *Design System*

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan *detail* (algoritma) prosedural.

3. *Coding and Testing*

Coding merupakan penerjemahan *design* ke dalam bahasa yang bisa dikenali komputer. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem dan kemudian bisa diperbaiki.

4. *Transition Phase*

Tahapan ini adalah tahapan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi dapat digunakan oleh *user*.

2.3.3 Sublime Text

Sublime text merupakan teks editor berbasis *Python*, teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel. *Sublime text* biasa digunakan untuk menyunting *source code* yang sedang dikerjakan. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur sintaks *highlight* hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas. *Sublime text* memiliki *minimap* pada sisi kanan atas untuk mempermudah *developer* dalam menemukan kode ataupun penulisan kode (Habibi et al., 2020:10).

2.3.4 MySQL

Menurut Anhar (2017:21) *MySQL* adalah “salah satu *Database Management System* (DBMS) dari sekian banyak DBMS seperti *oracle*, *MS SQL*, *Postgresql*, dan lainnya”. Menurut Arief dalam Fridayanthie dan Mahdiati (2016) *MySQL* adalah “salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya.”

MySQL masuk ke dalam jenis RDBMS (*Relational Database Management System*), maka dari itu istilah baris, kolom, tabel, dipakai pada *MySQL*. *MySQL* merupakan *database engine* atau *server database* yang mendukung bahasa *database*. *MySQL* adalah sebuah perangkat lunak sistem pengelolaan basis data SQL atau DBMS yang *multithread* dan *multi user* (Fitri, 2020). Adapun kelebihan *MySQL* menurut Fitri (2020), yaitu:

1. *Speed*

MySQL menyediakan sistem basis data berkecepatan tinggi yang sempurna untuk proyek-proyek kecil hingga menengah.

2. *Open Source*

MySQL dapat digunakan secara gratis. Meskipun ada juga untuk versi komersial yang tentu sudah diberikan tambahan fitur berupa kemampuan spesifik dan layanan *technical support* dari *MySQL*.

3. *Scalability*

MySQL dapat menangani *database* dengan skala besar yaitu dengan jumlah *record* lebih dari 50 juta.

4. *Connectivity* dan *Security*

Database MySQL dapat diakses dari semua tempat di internet dengan hak akses tertentu. *MySQL* adalah *database* yang menggunakan enkripsi *password*, jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.

5. *Flexibility* atau *Portability*

MySQL dapat digunakan untuk mengembangkan aplikasi berbasis *desktop* maupun aplikasi berbasis *web* dengan menggunakan teknologi yang beragam.

6. *Cross Platform Operating System*

MySQL dapat berjalan stabil diberbagai sistem operasi. Apabila diperlukan proses migrasi data antar sistem operasi dapat dilakukan dengan mudah.

2.3.5 XAMPP

Menurut Wicaksono dalam Fridayanthie dan Mahdiati (2016) *XAMPP* adalah “sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* di komputer lokal.”

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Dengan menginstall *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, *PHP* dan *MySQL* secara manual, *XAMPP* akan menginstallasi dan mengkonfigurasi nya secara otomatis (Fathoroni et al., 2020).

2.3.6 Framework CodeIgniter

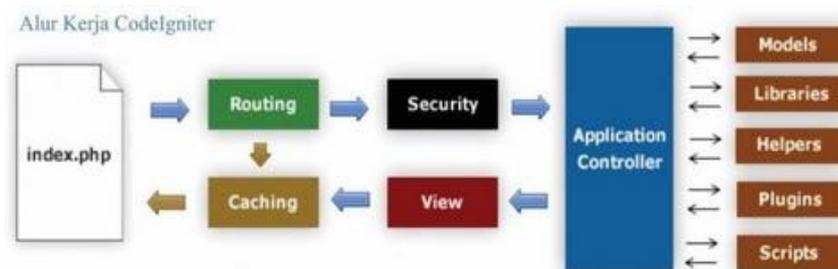
Menurut Fauzan dan Roza (2020) *Framework CodeIgniter* merupakan *framework* PHP yang kuat dan sedikit *bug*, *codeigniter* dibangun untuk para *developer* dengan bahasa pemrograman PHP sebagai alat untuk membuat sebuah situs *website* dengan fitur yang lengkap. *CodeIgniter* merupakan *framework* PHP yang dibuat berdasarkan MVC (*Model View Controller*), dimana MVC merupakan sebuah pola desain arsitektur pengembangan aplikasi yang memisahkan dan mengelompokkan beberapa kode sesuai dengan fungsinya MVC membagi aplikasi ke dalam tiga bagian fungsional yaitu *model*, *view* dan *controller*.

Menurut Prawiro et al., (2020) *CodeIgniter* memberikan kumpulan *library* untuk program yang sering digunakan dan untuk mengakses *library* tersebut terbilang cukup mudah. *Framework* ini memberikan tampilan yang baik untuk

halaman utama pada sistem juga dapat diaplikasikan metode *user centered design* yang merupakan metode untuk pengembangan pada tampilan *web* sehingga dapat mempermudah dalam mendesain tampilan *web* yang digunakan untuk pembobotan kriteria.

Menurut Habibi dan Aprilian (2019:90) *framework* merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempercepat dan mempermudah pekerjaan seorang programmer, tanpa harus membuat fungsi atau *class* dari awal. *CodeIgniter* adalah sebuah *web application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi php dinamis. *CodeIgniter* menjadi sebuah *framework* PHP dengan model MVC (*Model-View-Controller*) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembangan untuk membuat sebuah aplikasi *web*. *CodeIgniter* memiliki dokumentasi yang lengkap disertai contoh implementasi kodenya.

Adapun alur kerja *framework CodeIgniter* dikutip dari Habibi dan Aprilian (2019:91).



Sumber: Habibi dan Aprilian (2019:91)

Gambar 2.4 Alur Kerja CodeIgniter

a. *Index.php*

Index.php berfungsi sebagai *file* pertama dalam program yang akan dibaca oleh program.

b. *The Router*

Router akan memeriksa *HTTP request* untuk menentukan hal apa yang harus dilakukan oleh program.

c. *Cache File*

Apabila dalam program terdapat *cache file*, maka *file* tersebut akan langsung dikirim ke *browser*. *File cache* inilah yang dapat membuat sebuah *website* dapat dibuka dengan cepat. *Cache file* dapat melewati proses sebenarnya yang harus dilakukan oleh program *codeigniter*.

d. *Security*

Sebelum *file controller* di *load* keseluruhan, *HTTP request* dan data yang di *submit* oleh *user* akan disaring terlebih dahulu melalui fasilitas *security* yang dimiliki oleh *codeigniter*.

e. *Controller*

Controller akan membuka *file model*, *core libraries*, *helper* dan semua *resources* yang dibutuhkan dalam program tersebut.

f. *View*

Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam *view file* dan mengirimkannya ke *browser* supaya dapat dilihat. Apabila *file view* sudah ada yang di *cache*, maka *file view* baru yang belum di *cache* akan melakukan *update file view* yang sudah ada.

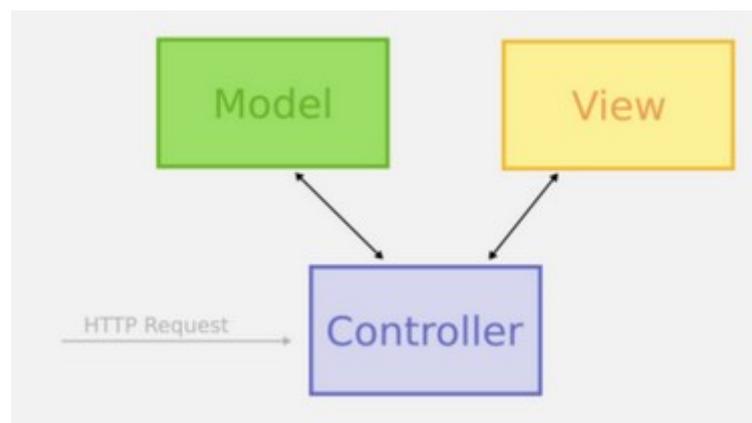
Menurut Habibi dan Aprilian (2019:93) MVC adalah “sebuah pola desain (*design pattern*) arsitektur pengembangan aplikasi yang memisahkan dan mengelompokkan beberapa kode sesuai dengan fungsinya.”

MVC membagi aplikasi ke dalam tiga bagian fungsional (Daqiqil Id,2011), yaitu:

- a. *Model* berhubungan dengan data dan interaksi ke *database* atau *web service*. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa *database* maupun data lain, misalnya dalam bentuk *file* teks, *file XML* maupun *web service*. Biasanya di dalam model akan berisi *class* dan fungsi untuk mengambil, melakukan *update* dan menghapus data *website*. Sebuah aplikasi *web* biasanya menggunakan *database* dalam menyimpan data, maka pada bagian *model* biasanya akan berhubungan dengan perintah-perintah *query SQL*.
- b. *View* berhubungan dengan segala sesuatu yang akan ditampilkan ke *end-user*. Bisa berupa halaman *web*, *rss*, *JavaScript* dan lain-lain. Pada *view*, harus menghindari adanya logika atau pemrosesan data. Di dalam *view* hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. *View* dapat dikatakan

sebagai halaman *website* yang di buat dengan menggunakan HTML dan bantuan CSS atau *JavaScript*. Di dalam *view* jangan pernah ada kode untuk melakukan koneksi ke *database*. *View* hanya dikhususkan untuk menampilkan data-data hasil dari *model* dan *controller*.

- c. *Controller* bertindak sebagai penghubung data dan *view*. Di dalam *controller* inilah terdapat *class-class* dan fungsi-fungsi yang memproses permintaan dari *view* ke dalam struktur data di dalam *model*. *Controller* juga tidak boleh berisi kode untuk mengakses *database* karena tugas mengakses *database* telah diserahkan kepada bagian *model*. Tugas *controller* adalah menyediakan berbagai variabel yang akan ditampilkan di *view*, memanggil model untuk melakukan akses ke *database*, menyediakan penanganan kesalahan/*error*, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap *input*.



Sumber: Habibi dan Aprilian (2019:93)

Gambar 2.5 Konsep Model View Controller Framework CodeIgniter

Adapun alur kerjanya meliputi (Habibi dan Aprilian, 2019:93):

- a. Mulai;
- b. *User* mengirim *request* ke *web*;
- c. *File* yang pertama kali di eksekusi adalah *index.php*;
- d. Kemudian *request* akan diteruskan oleh *routers.php*;
- e. *Routers.php* akan mencari *cache* di *server*, apabila terdapat *cache* maka *cache* itu yang akan dikirim sebagai balasan (*reponse*). Apabila tidak ada *cache* barulah *request* diteruskan ke *controller*;

- f. *Controller* akan bertanggung jawab untuk mengambil data dari *model* dan merendernya ke dalam *view* dengan menggunakan *library*, *plugin*, dan *helper* yang ada;
- g. Hasil *render (view)* di kirim ke pengguna dan di simpan dalam *cache*, apabila fitur *cache* aktif;
- h. Selesai.

Id (2021) menyatakan: pada eksekusi PHP, biasanya akan melakukan “load” semua *library* dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk di eksekusi oleh PHP. Untuk kasus sederhana, cara tersebut tidak akan menimbulkan masalah. Tetapi jika aplikasi tersebut menjadi kompleks, maka pemeliharannya jika tidak di dukung oleh arsitektur *software* yang bagus. Hal tersebut bisa terjadi disebabkan oleh *code* yang sama namun berulang-ulang dan kode tidak konsisten. *Browser* berinteraksi melalui *controller*. Kemudian *controller* akan menerima dan membalas semua *request* dari *browser*. Untuk data maka *controller* akan meminta ke *model* dan untuk *UI/template* akan meminta ke *view*. Jadi otak dari aplikasi ada di *controller*, muka aplikasi ada pada *view* dan data pada *model*. Ketika *browser* meminta sebuah halaman *web* maka *router* akan mencarikan *controller* mana yang harus menangani *request* tersebut. Setelah itu barulah *controller* menggunakan model untuk mengakses data dan *view* untuk menampilkan data tersebut.