

BAB II TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

(Paryanta, 2017) dalam penelitiannya yang berjudul “**Sistem Informasi Administrasi Kependudukan Berbasis Web Desa Sawahan**” menjelaskan bahwa Desa Sawahan adalah sebuah desa yang membutuhkan sistem informasi administrasi kependudukan yang dapat melakukan pengolahan data terkait proses pemberian informasi tentang Desa Sawahan, informasi demografi, pendaftaran data kependudukan, data keluarga, tanggal lahir, data kematian, data mentransfer, mencetak surat lamaran warga negara, laporan Anggaran dan laporan demografis lainnya. Pengolahan data yang saat ini berjalan saat ini masih tercatat ke dalam buku data demografi. Dalam pelayanan dan informasi ke publik masih dilakukan secara manual, semua data dicatat ke dalam buku. Sedangkan untuk mengetahui masyarakat informasi sebaiknya meluangkan waktu untuk datang ke kantor. Hambatan yang dihadapi oleh proses adalah aktivitas yang memakan waktu dan risiko kesalahan dalam pencatatan data. Hal ini bisa diatasi dengan membangun sistem baru yang lebih efektif. Administrasi Penduduk Sistem Informasi Pedesaan Situs Berbasis Desa Sawahan dibangun dengan menggunakan studi kasus deskriptif Desa Sawahan. Analisis teknik kelemahan dalam sistem dengan menggunakan analisis *PIECES* (*Performance, Information, Economy, control, Efficiency and Service*). Untuk teknik pengembangan sistem menggunakan metode *waterfall*, sedangkan untuk menggambarkan arus data menggunakan *DFD* (*Data Flow Diagram*). *Software* yang digunakan untuk membangun sistem ini adalah *PHP* (*Hyertext Preprocessor*) dan *XAMPP* sebagai koneksi ke *database MySQL*

(Imaniawan, 2017) dalam penelitiannya yang berjudul “**Sistem Informasi Administrasi Kependudukan Berbasis Web Pada Desa Bogangin Sumpiuh**” menjelaskan bahwa sistem informasi Administasi Kependudukan pada Desa Bogangin Sumpiuh masih dilakukan secara manual. Dalam pengelolaan data

kependudukan masih secara konvensional dengan sistem pendataan yang sekarang dirasakan masih banyak kekurangan yang terjadi, dikarenakan sistem yang ada masih menggunakan lembaran kertas sehingga dapat menyebabkan data-data yang ada mudah hilang ataupun rusak. Pengembangan sistem berupa sistem Administrasi Kependudukan berbasis *web* pada Desa Bogangn merupakan sistem yang memberikan informasi mengenai data penduduk dan pelayanan cetak surat dengan berbasiskan *web*, sehingga membantu kecepatan dan kualitas dalam penyampaian informasi serta mempermudah dalam pelayanan cetak surat-menyurat agar lebih efektif dan efisien. Pada sistem ini, *web* hanya dapat diakses oleh user yaitu warga dan administrator. Metode yang digunakan dalam pengembangan sistem ini yaitu *SDLC (System Development Life Cycle)* dengan model proses waterfall. Dalam penelitian ini selain memaparkan kajian teori yang digunakan sebagai dasar penyusunan, juga akan dibahas mengenai perancangan sistem dan pembuatan *web* administrasi kependudukan. Sehingga diharapkan dapat diimplementasikan dalam sebuah produk yang akan memperbaiki segala kekurangan yang ada pada sistem lama.

(Syukron, 2019) dalam penelitiannya yang berjudul **“Perancangan Sistem Informasi Administrasi Kependudukan Desa Berbasis *Website* pada Desa Winong”** menjelaskan bahwa administrasi kependudukan merupakan rangkaian kegiatan penataan dan penertiban dokumen dan data kependudukan serta pendayagunaan hasilnya untuk pelayanan publik dan pembangunan sektor lain. Desa Winong sebagai salah satu instansi pemerintahan, memiliki peran yang penting yaitu sebagai pengelola data kependudukan di tingkat desa. Pengelolaan data kependudukan di Desa Winong saat ini masih belum memaksimalkan penggunaan teknologi informasi untuk pengelolaannya, sehingga masih terdapat beberapa kekurangan dan kendala yang dihadapi. Seperti masih terdapat kerangkapan data kependudukan, kesulitan dalam pencarian data, serta pembuatan laporan kependudukan. Sehingga pelayanan kepada masyarakat serta kerja dari perangkat desa menjadi kurang efektif dan efisien. Oleh karena itu dibutuhkan sebuah sistem informasi terkomputerisasi yang dapat digunakan untuk mengelola data tersebut. Metode yang digunakan untuk perancangan sistem administrasi

kependudukan yaitu dengan metode prototyping. Dengan adanya sistem informasi administrasi kependudukan yang berbasis *website* ini, dapat memudahkan pengelolaan data kependudukan. Sehingga dapat terwujud tertib administrasi kependudukan yang baik di desa Winong.

(Mustika, 2021) dalam penelitiannya yang berjudul **“Sistem Informasi Administrasi Kependudukan (SIASIK) Pada Kelurahan Berbasis Web”** menjelaskan bahwa kelurahan pada umumnya sebagai pusat pelayanan masyarakat pada wilayah tertentu. Salah satu pelayanan kelurahan adalah mengurus surat administrasi kependudukan. Permasalahan utama pada Kelurahan pada umumnya belum ada sistem yang membantu dalam penyimpanan data Administrasi Kependudukan. Sistem Informasi Administrasi Kelurahan (SIASIK) bertujuan untuk melakukan pengadministrasian seperti (Surat Pindah, Surat Kematian dll). SIASIK ini juga untuk penanganan dokumen yang banyak sehingga dalam pembuatan surat pengajuan bisa lebih cepat dan efisien. Dalam pengumpulan informasi penulis melakukan pengamatan langsung dan wawancara langsung ke pihak yang berkaitan dengan administrasi kelurahan. pembuatan SIASIK menggunakan metode *SDLC Waterfall* . Di dalam pengembangannya dibuat perencanaan dimulai dari pengumpulan semua data-data serta membuat desain untuk membangun kebutuhan sistem. Bahasa pemrograman yang digunakan berbasis *web*. Keuntungan yang didapat di aplikasi ini dibuat untuk mempermudah petugas dalam hal penanganan dokumen-dokumen yang banyak, seperti kesulitan mencari dan mencetak data. Dan demi keamanan data untuk menghindari kehilangan data, kerangkapan data dan manipulasi data. SIASIK menggunakan Aplikasi *Sublime* dengan bahasa pemrograman *PHP* , *jquery*, dan *Enterprise Architect 7.0*. Hasil penelitian ini sebuah program administrasi kelurahan yang dapat membantu kegiatan administrasi kelurahan tersebut.

(Abza, 2020) dalam penelitiannya yang berjudul **“Perancangan Sistem Informasi Administrasi Kependudukan Desa Alai Kabupaten Kepulauan Meranti Berbasis Website”** menjelaskan bahwa sistem pengelolaan data administrasi kependudukan di kantor Desa Alai merupakan Sistem Informasi yang mengolah tentang data penduduk, data kematian, data kelahiran, data kedatangan

penduduk, data pindah penduduk, data surat masuk dan data surat keluar kependudukan. Pada sistem yang sedang berjalan, saat melakukan proses penyimpanan data administrasi masih menggunakan sistem manual yaitu pencatatan di buku agenda, sehingga dalam penyimpanan data administrasi tidak terkomputerisasi dengan baik. Hal ini menyebabkan sering terjadi keterlambatan saat mencari data administrasi dan saat melakukan proses pembuatan laporan administrasi setiap tahunnya. Dalam mendukung sistem informasi administrasi kependudukan Desa Alai, maka diperlukan sebuah aplikasi yang dapat memenuhi kebutuhan di Kantor Desa Alai dengan memanfaatkan teknologi berbasis *website*. Penulis menggunakan bahasa pemrograman *PHP* dan *MySQL* dalam membangun pengelolaan administrasi data kependudukan di Kantor Desa Alai. Tujuan dalam penelitian ini untuk membantu pegawai administrasi Kantor Desa Alai dalam merekap dan mengelola data kependudukan dengan basis data sehingga hasil dari penelitian ini berupa penyebaran informasi tentang data kependudukan yang dapat dilihat oleh masyarakat secara luas melalui *website* dan data surat masuk dan surat keluar kependudukan tersusun dengan rapi melalui sistem basis data.

Perbedaan penelitian terdahulu dengan penelitian sekarang dapat dilihat pada tabel 2.1

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian Sekarang

NO	JUDUL JURNAL	PERSAMAAN	PERBEDAAN
1.	Sistem Informasi Administrasi Kependudukan Berbasis <i>Web</i> Desa Sawahan	Menggunakan <i>database Mysql</i> dan Bahasa Pemrograman <i>PHP</i>	Menggunakan <i>framework codeigniter</i>

2.	Sistem Informasi Administrasi Kependudukan Berbasis <i>Web</i> Pada Desa Bogangin Sumpiuh	Menggunakan <i>database Mysql</i> dan Bahasa Pemrograman <i>PHP</i>	Menggunakan Metode UML
3.	Perancangan Sistem Informasi Administrasi Kependudukan Desa Berbasis <i>Website</i> pada Desa Winong	Menggunakan <i>database Mysql</i> , Bahasa Pemrograman <i>PHP</i> , dan menggunakan metode penelitian <i>Prototype</i>	Menggunakan Metode Waterfall
4	Sistem Informasi Administrasi Kependudukan (SIASIK) Pada Kelurahan Berbasis <i>Web</i>	Menggunakan <i>database Mysql</i> dan Bahasa Pemrograman <i>PHP</i>	-
5	Perancangan Sistem Informasi Administrasi Kependudukan Desa Alai Kabupaten Kepulauan Meranti Berbasis <i>Website</i>	Menggunakan <i>database Mysql</i> dan Bahasa Pemrograman <i>PHP</i>	-

2.2. Landasan Teori

2.2.1. Pengertian Sistem

Sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Sistem juga merupakan kumpulan elemen-elemen saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan (Kristanto, 2019).

2.2.2. Sistem Informasi

Sistem Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Dengan kata lain sumber dari informasi adalah data. Data menggambarkan suatu kejadian yang sering terjadi, dimana data tersebut akan diolah dan akan diterapkan dalam sistem menjadi input yang berguna dalam suatu sistem. Data merupakan bentuk yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga perlu suatu model yang nantinya akan dikelompokkan dan diproses untuk menghasilkan informasi (Kristanto, 2019).

2.2.3. Administrasi Kependudukan

Administrasi merupakan pelayanan atau pengabdian terhadap subjek tertentu. Pelayanan administrasi kependudukan diartikan sebagai pelayanan di bidang kependudukan yang diberikan oleh aparat pemerintah dan non pemerintah dari tingkat pusat sampai tingkat desa atau Kelurahan, RW dan RT. Pada pelayanan administrasi kependudukan, aparat pemerintah dan non pemerintah memberikan pelayanan misalnya pengurusan izin nikah, permohonan KTP, surat keterangan, dan kartu keluarga, dan surat-surat kependudukan yang lain (Daryanto, 2005).

2.2.4. Website

Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi *internet* sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standar yaitu HTML. *Script* HTML ini akan diterjemahkan oleh *website browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang (Abdulloh, 2018).

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebut *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext* (Batubara, 2015).

2.2.5. Database

Database adalah kumpulan banyak data yang saling terkait dan terkumpul dalam satu tempat yang sama dan dipakai oleh sistem aplikasi yang dikontrol secara terpusat serta memiliki nilai yang berharga bagi pemilik (Setiawan, 2017).

Basis data atau *database* adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam *database* komputer, dibutuhkan sistem manajemen basis data (*database management system*) (Suharyanto, dkk, 2017).

2.2.6 Analisis dan Desain Berorientasi Objek

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Desain sistem

dapat didefinisikan sebagai penggambaran dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Sedangkan objek dapat diartikan sebagai suatu entitas yang memiliki identitas, *state* dan *behavior*. Dari pengertian di atas dapat disimpulkan bahwa metode analisis dan desain berorientasi objek merupakan suatu metode untuk memeriksa kebutuhan dari sudut pandang kelas-kelas dan objek kemudian mempelajari permasalahan dengan menspesifikasikannya atau mengobservasi permasalahan tersebut sesuai dengan kombinasi antara struktur data dan perilaku dalam satu entitas. Metode analisa dan desain berorientasi objek menawarkan sebuah kerangka kerja yang baik untuk skenario (Jakimi, dkk, 2011).

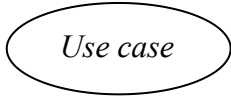
Metode analisis dan desain berorientasi objek memberikan konsistensi keamanan di seluruh siklus hidup pengembangan sistem dari analisis kebutuhan sampai implementasi (Kyung, dkk, 2014).

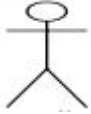

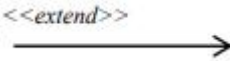

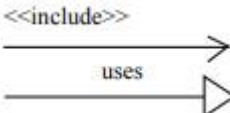
2.2.7 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa, dkk, 2018).

Pada Tabel 2.2 adalah simbol-simbol yang digunakan dalam *use case diagram* disertai dengan keterangan fungsinya:

Tabel 2.2 Simbol-Simbol *Use case Diagram*

No.	Simbol	Keterangan
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor.
2.	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.

		
3.	Asosiasi 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi.
4.		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
5.	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> yang mana fungsi yang satu lebih umum dari yang lainnya.
6.		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya.







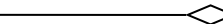
(Sumber: Rosa, dkk, 2018)

2.2.8 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan *class diagram* agar antara dokumentasi perancangan dan perangkat lunak sinkron (Rosa, dkk, 2018).

Pada Tabel 2.3 adalah simbol-simbol yang digunakan dalam *class diagram* disertai dengan keterangan fungsinya:

Tabel 2.3 Simbol-Simbol *Class Diagram*

No.	Simbol	Keterangan
1.	<p>Kelas</p> 	Merupakan kelas pada struktur sistem.
2.	<p>Antarmuka / <i>Interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p>Asosiasi</p> 	Merupakan relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	<p>Asosiasi berarah</p> 	Merupakan relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	<p>Generalisasi/<i>generalization</i></p> 	Merupakan relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6.	<p>Kebergantungan / <i>Dependency</i></p> 	Merupakan relasi antar kelas dengan makna kebergantungan antarkelas.
7.	<p>Agregasi / <i>Aggregation</i></p> 	Merupakan relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).



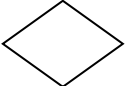



(Sumber: Rosa, dkk, 2018)

2.2.9 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas berfungsi untuk menggambarkan aktivitas sistem bukan yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa, dkk, 2018).

Pada Tabel 2.4 adalah simbol-simbol yang digunakan dalam *activity diagram* disertai dengan keterangan fungsinya:

Tabel 2.4 Simbol-Simbol *Activity Diagram*

No.	Simbol	Keterangan
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>Decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>Join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.


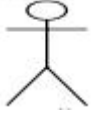
(Sumber: Rosa, dkk, 2018)


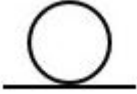
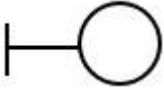




2.2.10 Sequence Diagram

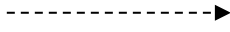
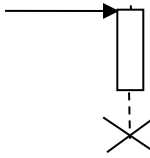
Diagram sekuen atau *sequence diagram* merupakan diagram yang menggambarkan kelakuan dari objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Rosa, dkk, 2018).

Pada Tabel 2.5 adalah simbol-simbol yang digunakan dalam *sequence diagram* disertai dengan keterangan fungsinya:

Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Keterangan
1.	<p>Garis / <i>Lifeline</i></p> 	Menyatakan kehidupan suatu objek.
2.	<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p><u>nama objek : nama</u> <u>kelas</u></p> </div>	Menyatakan objek yang berinteraksi pesan.
3.	<p>Aktor</p>  <p>nama aktor atau</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p><u>nama aktor</u></p> </div> <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

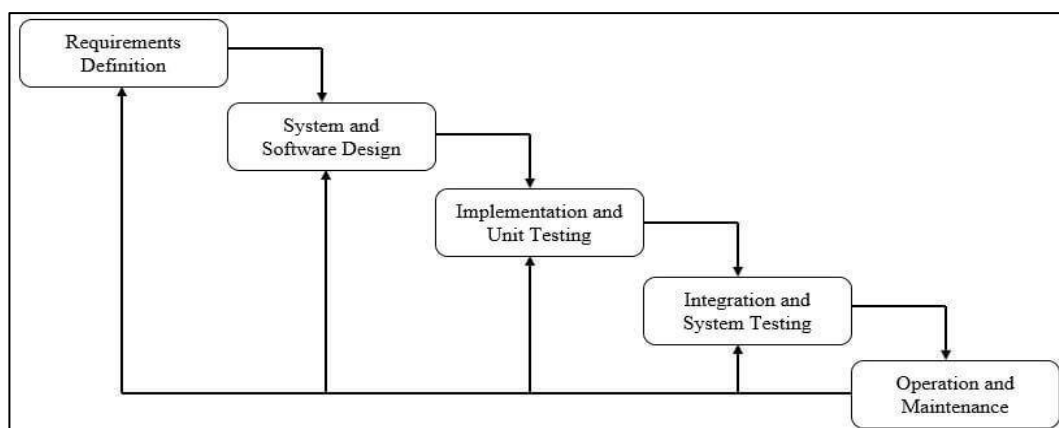
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	 <i>Entity Class</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.
6.	 <i>Boundary Class</i>	Digunakan untuk menggambarkan sebuah <i>form</i> .
7.	 <i>Control Class</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan tabel.
8.	Pesan tipe <i>create</i> <code><<create>></code> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
9.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi atau metode, karena ini memanggil operasi atau metode maka operasi atau metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
10.	Pesan tipe <i>send</i> 1 : masukan 	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang

		dikirim.
11.	<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
12.	<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber: Rosa, dkk, 2018)

2.2.11 Metode Pengembangan Sistem

Waterfall model merupakan sebuah *software development lifecycle* yang memiliki jenis linear. Memiliki beberapa *fase* seperti contohnya *requirement analysis* (analisis kebutuhan), *design* (perancangan), *coding* (pembuatan kode program), *testing* (pengujian) dan implementasi dimana umumnya satu *fase* tidak diulang kembali dan berlanjut ke *fase* berikutnya sampai *fase* sebelumnya selesai (Prihandoyo, 2018). Adapun model pengembangan digambarkan pada gambar 2.1:



Gambar 2.1 Model Waterfall

(Sumber: Prihandoyo, 2018)

2.2.12 HyperText Markup Language (HTML)

HTML merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman *website* yang dapat diakses atau ditampilkan menggunakan *web browser*. HTML sendiri secara resmi lahir pada tahun 1989 oleh tim Berners Lee dan dikembangkan oleh *World Wide Web Consortium (W3C)*, yang kemudian pada tahun 2004 dibentuklah *Web Hypertext Application Technology Working Group (WHATG)* yang hingga kini bertanggung jawab akan perkembangan bahasa HTML (Setiawan, 2017).

HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar *website* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa *tag-tag* yang menyusun setiap elemen dari *website*. HTML berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai *layout* yang diinginkan. HTML biasanya disimpan dalam sebuah *file* berekstensi *.html*. Untuk mengetikkan *script* HTML dapat menggunakan *text editor* seperti *Notepad* sebagai paling bentuk sederhana atau *text editor* khusus yang dapat mengenali setiap unsur *script* HTML dan menampilkannya dengan warna yang berbeda sehingga mudah dibaca, seperti *Notepad++*, *Sublime Text*, dan masih banyak lagi aplikasi lain yang sejenis (Abdulloh, 2018).

HTML singkatan dari *Hyper Text Markup Language*, yaitu *script* yang berupa *tag-tag* untuk membuat dan mengatur struktur *website*. Beberapa tugas utama HTML dalam membangun *website* diantaranya sebagai berikut:

- a) Menentukan *layout website*.
- b) Memformat *text* dasar seperti pengaturan paragraf, dan format *font*.
- c) Membuat *list*.
- d) Membuat tabel.
- e) Menyisipkan gambar, video, dan audio.
- f) Membuat *link*.
- g) Membuat formulir (Josi, 2017).

2.2.13 Cascading Style Sheets (CSS)

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen *website* yang berfungsi mengatur elemen HTML dengan berbagai *property* yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Sebagian orang menganggap CSS bukan termasuk salah satu bahasa pemrograman karena memang strukturnya sederhana, hanya berupa kumpulan-kumpulan aturan yang mengatur *style* elemen HTML.

Cara kerja CSS dalam memodifikasi HTML dengan memilih elemen HTML yang akan diatur kemudian memberikan *property* yang sesuai dengan tampilan yang diinginkan. Dalam memberikan aturan pada elemen HTML, *script* CSS terdiri dari atas 3 bagian yaitu *selector* untuk memilih elemen yang akan diberi aturan, *property* yang merupakan aturan yang diberikan dan *value* sebagai nilai dari aturan yang diberikan (Abdulloh, 2018).

Selain itu, CSS juga menjelaskan bagaimana elemen-elemen HTML ditampilkan pada *layer* dengan lebih mulus. CSS dapat menghemat banyak pekerjaan dan mengontrol tata letak beberapa halaman *web* sekaligus. CSS dapat ditambahkan ke elemen HTML dalam 3 cara, yaitu:

1. *Inline*, yaitu dengan menggunakan atribut *style* dari elemen HTML.
2. *Internal*, yaitu dengan menggunakan elemen `<style>` yang ditempatkan pada area `<head>`.
3. *External*, yaitu dengan menggunakan *file* CSS eksternal (Saputra, 2019).

2.2.14 JavaScript (JS)

JavaScript merupakan bahasa pemrograman *website* yang pemrosesnya dilakukan di sisi *client*. Karena berjalan disisi *client*, *JavaScript* dapat dijalankan hanya dengan menggunakan *browser*. Berbeda dengan *PHP* yang bekerja disisi *server*, untuk menjalankan *script JavaScript* tidak memerlukan *refresh* pada *browser*. *JavaScript* biasanya dijalankan ketika ada *event* tertentu yang terjadi pada halaman *website*. Baik *event* yang dilakukan oleh *user*, maupun *event* yang terjadi karena adanya perubahan pada halaman *website* (Abdulloh, 2018).

2.2.15 JQuery

JQuery merupakan salah satu dari sekian banyak *JavaScript library*, yaitu kumpulan fungsi *JavaScript* yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode *JavaScript* (Abdulloh, 2018).

JQuery adalah kumpulan fungsi-fungsi *JavaScript* yang sudah dibentuk sebagai suatu objek. Sehingga penggunaan *jQuery* ini bisa dikategorikan sebagai suatu *library* yang nantinya kita hanya perlu menggunakan fungsi-fungsi di dalam *library* tersebut.

JQuery pertama kali dirilis oleh John Resig pada tahun 2006. Pada perkembangannya *jQuery* tidak sekedar sebagai *library javascript*, namun memiliki keandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak *developer web* menggunakannya. *jQuery* dikenal dengan slogan “*Write less, do more*” artinya penulisan kode yang sedikit tetapi memiliki beberapa aksi (*action*) (Hidayatullah, 2017).

2.2.16 Perl Hypertext Preprocessor (PHP)

PHP berasal dari kata *Perl Hypertext Preprocessor*, yaitu bahasa pemrograman yang bersifat *universal* dan dapat digunakan bersamaan dengan bahasa pemrograman HTML. PHP digunakan untuk melakukan penanganan, pembuatan dan pengembangan sebuah situs *web* (Josi, 2017).

Perl Hypertext Preprocessor atau disingkat PHP adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP juga dapat diintegrasikan dengan HTML, *JavaScript*, *JQuery*, *Ajax*. Namun, pada umumnya PHP lebih banyak digunakan bersamaan dengan *file* bertipe HTML. Dengan menggunakan PHP anda bisa membuat *website* yang dinamis dengan disertai manajemen *database*-nya (Hidayatullah, 2017).

PHP merupakan kependekan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *website* yang dapat disisipkan dalam *script* HTML dan bekerja disisi *server*. Tujuan dari bahasa ini adalah membantu para pengembang *website* untuk membuat *website* dinamis dengan cepat.

Agar dapat menjalankan PHP harus menyediakan perangkat lunak berikut sebagai :

1. *Web server (Apache, IIS, Personal Web Server/PWS).*
2. *PHP server.*
3. *Database server (MySQL, Interbase, MS SQL, dan lain-lain).*

Aplikasi diatas tidak perlu diinstal satu per-satu, karena biasanya telah tersedia paket aplikasi yang telah menyediakan ketiganya dalam satu paket aplikasi seperti XAMPP atau *Appserv* (Abdulloh, 2018).

2.2.17 MySQL

MySQL merupakan sebuah *Relational Database Management System* (RDBMS) yang bersifat *open-source*. Perangkat lunak *database* pada umumnya disandingkan dengan bahasa pemrograman *server web* seperti PHP atau JSP. MySQL (*My Structured Query Language*) adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut dengan DBMS (*Database Management System*), sifat DBMS ini ialah *open-source* (Josi, 2017).

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya (Ibrahim, dkk, 2018).

Salah satu keunggulan MySQL adalah kemudahan penggunaan dan pengelolannya, untuk mengakses *database* MySQL dapat dilakukan dengan menggunakan berbagai MySQL *client*. Selain menggunakan MySQL *client* bawaan berbasis *command line*, saat ini banyak sekali *tools* yang dikembangkan oleh pihak ketiga untuk mempermudah pengelolaan *database* MySQL (Solichin, 2016).

2.2.18 XAMPP

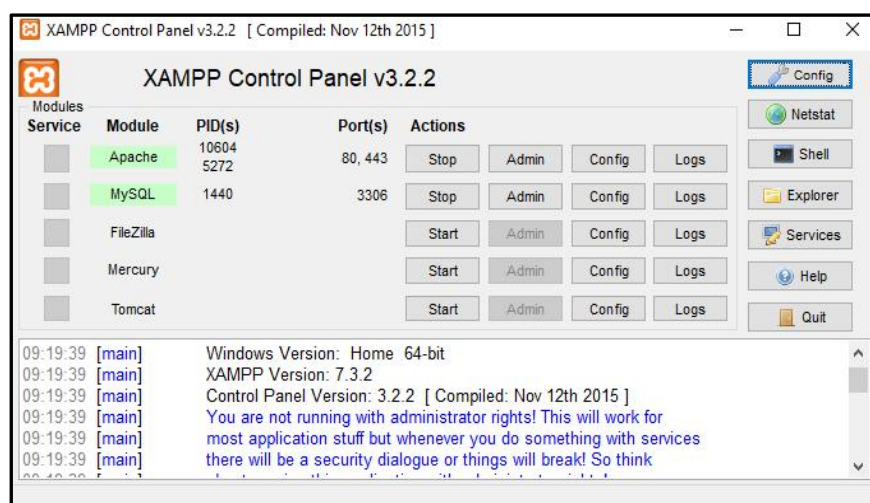
XAMPP adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah *server*. Kegunaan xampp yaitu untuk membuat jaringan lokal sendiri dalam artian kita dapat membuat *website* secara *offline* untuk masa coba-coba di komputer sendiri. Disebut *server* karena dalam hal ini komputer

yang akan kita pakai harus memberikan pelayanan untuk mengakseskan *web*, untuk itu komputer kita harus menjadi *server* (Josi, 2017).

XAMPP banyak digunakan sistem operasi seperti Windows, Linux, Mac dan Solaris sehingga tidak masalah ketika berpindah-pindah sistem operasi. Kata XAMPP sendiri berasal dari:

- a. X yang berarti *cross platform* karena XAMPP bisa dijalankan di windows, Linux, Mac, dsb.
- b. A yang artinya *Apache* sebagai *web server*-nya.
- c. M yang berarti MySQL sebagai *Database Management System* (DBMS)-nya.
- d. PP yang berarti PHP dan *Perl* sebagai bahasa yang didukung (Hidayatullah, 2017).

Adapun Gambar 2.2 merupakan tampilan XAMPP Control Panel:

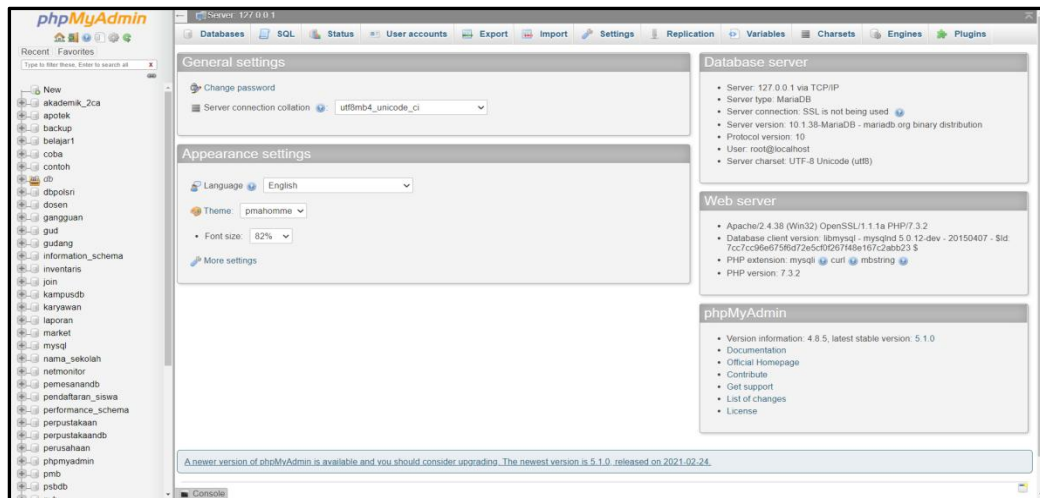


Gambar 2.2 Tampilan XAMPP Control Panel

2.2.19 PHPMYAdmin

PHPMyAdmin adalah sebuah perangkat lunak bebas (*open-source*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun *internet*. *PHPMyAdmin* mendukung berbagai operasi MySQL. Diantaranya (mengelola basis data, tabel-tabel, *fields*, relasi (*relations*), indeks, pengguna (*user*), perijinan (*permissions*), dan lain-lain) (Rozaq, dkk, 2015). Adapun Gambar 2.3 merupakan

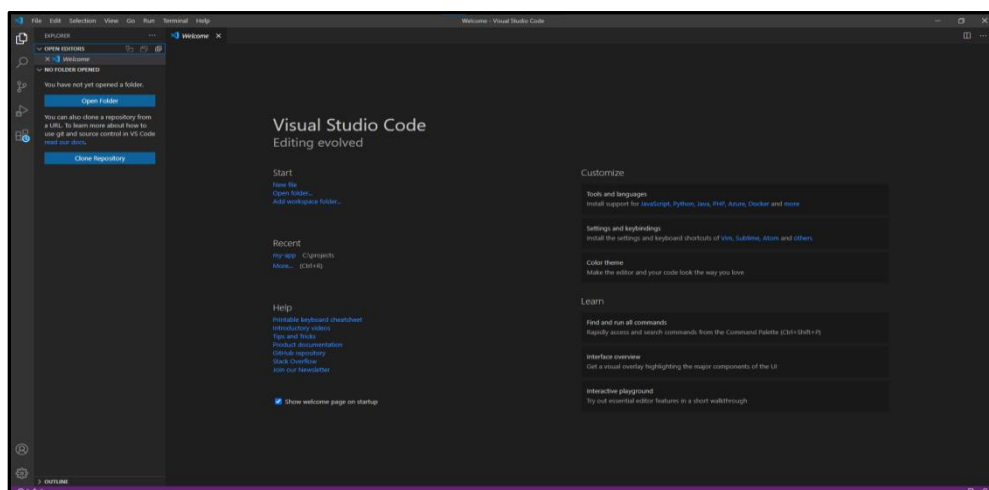
tampilan *dashboard* *PHPMyAdmin*:



Gambar 2.3 Tampilan *Dashboard* *PHPMyAdmin*

2.2.20 Visual Studio Code

Visual Studio Code (VS Code) adalah kode editor yang bersifat *open-source* untuk pengembangan dan *debugging* aplikasi *cloud* dan situs *web modern* yang tersedia secara gratis di Linux, OS X dan Windows. VS Code mendukung lebih dari 30 bahasa pemrograman, *markup* dan *database* yang berbeda, beberapa di antaranya adalah PHP, HTML, CSS dan SQL (Gamma, 2016). Adapun Gambar 2.4 merupakan tampilan *visual studio code*:



Gambar 2.4 Tampilan *Visual Studio Code*

2.2.21 Framework

Framework merupakan kumpulan fungsi–fungsi/prosedur–prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal (Supardi, dkk, 2018).

2.2.22 Bootstrap

Bootstrap merupakan salah satu *framework* CSS yang paling populer dari sekian banyak *framework* CSS yang ada. *Bootstrap* memungkinkan desain sebuah *website* menjadi *responsive* sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik. *Bootstrap* juga membuat proses pengaturan desain menjadi lebih cepat karena tidak perlu lagi banyak menulis CSS, bahkan hampir tidak perlu, kecuali memerlukan pengaturan desain yang berbeda dengan *style bootstrap*. *Bootstrap* telah didukung oleh hampir semua *browser* baik pada *desktop* maupun *mobile* (Abdulloh, 2018).

2.2.23 CodeIgniter

CodeIgniter adalah aplikasi *open-source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan pengembang (*developer*) untuk membuat aplikasi *web* dengan cepat dan mudah dibandingkan dengan membuatnya dari awal (Supardi, dkk, 2018).

Framework Codeigniter dikembangkan oleh Rick Ellis, CEO Ellislab, Inc. kelebihan dari *framework CodeIgniter* jika dibandingkan dengan *framework* lain adalah sebagai berikut:

- 1) Gratis (*Open-Source*)

Kerangka kerja *Codeigniter* memiliki lisensi dibawah *Apache/BSD open-source* sehingga bersifat bebas atau gratis.

- 2) Berukuran kecil

Ukuran yang kecil merupakan keunggulan tersendiri jika dibandingkan *framework* lain yang berukuran besar dan membutuhkan *resource* yang

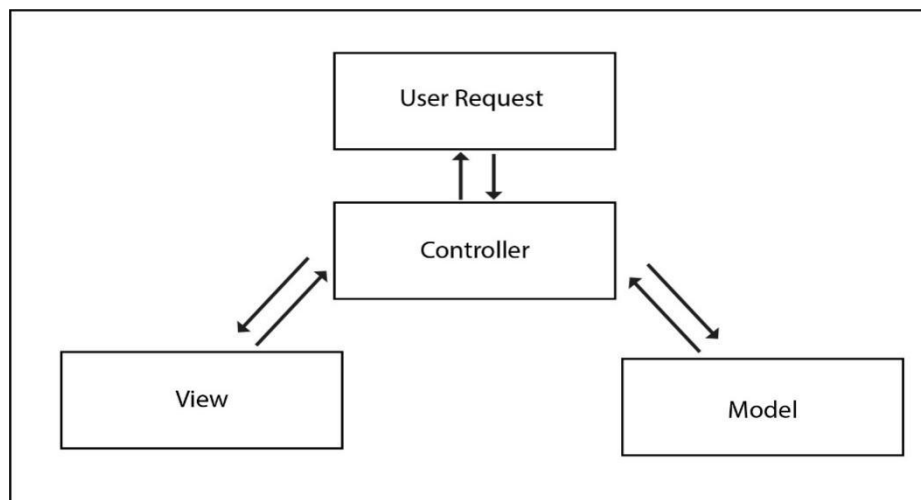
besar dan juga dalam eksekusi maupun penyimpanannya.

3) Menggunakan konsep M-V-C

Codeigniter merupakan konsep M-V-C (*Model-View-Controller*) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Dengan konsep ini kode PHP, *query* MySQL, *Javascript* dan CSS dapat saling dipisah-pisahkan sehingga ukuran *file* menjadi lebih kecil dan lebih mudah dalam perbaikan kedepannya atau *maintenance*.

- a) *Model* merupakan kode program (berupa OOP *class*) yang digunakan untuk berhubungan dengan *database* MySQL sekaligus untuk memanipulasinya (*input-edit-delete*).
- b) *View* merupakan kode program berupa *template* atau PHP untuk menampilkan data pada *browser*.
- c) *Controller* merupakan kode program (berupa OOP *class*) yang digunakan untuk mengontrol aliran atau dengan kata lain sebagai pengontrol *model* dan *view* (Destiningrum, dkk, 2017).

Adapun Gambar 2.5 merupakan alur dari program aplikasi berbasis *CodeIgniter* yang menggunakan konsep M-V-C:



Gambar 2.5 Konsep Aliran M-V-C

(Sumber: Destiningrum, dkk, 2017)

Pada versi 4 proses instalasi bisa dilakukan dengan dua cara :

a. Cara manual :

1. Mendownload *source file codeigniter*
2. Meng-*extract file* hasil *download*
3. Selanjutnya memindahkannya ke *root directory web server*

b. Menggunakan *composer* dengan perintah :

1. `composer create-project`
2. `codeigniter4/appstarter project-root` (Putri, dkk, 2021).

2.2.24 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program. Dalam pengujian ini, tester menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.

Pada *Black Box Testing* ini dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh customer. *Black-box testing* ini lebih menguji ke tampilan luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh pengguna. Pengujian ini tidak melihat dan menguji *source code program*. *Black-box testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya hanya terfokus pada informasi *domain*.

Jadi metode ini bertujuan untuk memeriksa, setelah tahap akhir proyek, apakah perangkat lunak atau aplikasi berfungsi dengan baik, dan melayani penggunaannya secara efisien. Biasanya, penguji mencari fungsi yang hilang atau salah; antarmuka, kinerja, inisialisasi program dan kesalahan keluar; struktur data atau kesalahan akses basis data eksternal (Syafnidawaty, 2020).