

BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

landasan dasar pada teori umum sebagai pengetahuan dalam membangun sistem informasi pendaftaran *online* pada sekolah clarissa berbasis *web*.

2.1.1 Pendaftaran siswa baru secara *online*

Pendaftaran siswa baru merupakan salah satu kegiatan yang dilakukan oleh semua instansi baik negeri maupun swasta. setiap tahun, semua institusi menghadapi kegiatan pendaftaran siswa baru. kegiatan manajemen pendidikan sangat dipengaruhi oleh jumlah siswa yang diterima oleh lembaga tersebut. modal kerja lembaga tersebut berasal dari mahasiswa atau siswa yang diterima oleh lembaga pendidikan (Bariah & Putera, 2020).

Pendaftaran *online* merupakan sistem yang dirancang untuk mengotomatisasi seleksi penerimaan mahasiswa baru, dan otomatisasi soal dimulai dari proses registrasi, seleksi, dan penyajian, hasil seleksi yang dibuat dengan online secara real time (Angraina Fitri & Sulistio, 2020). Salah satu fokus penelitian ini adalah bagian siswa baru Clarissa *School* yang belum dijalankan secara *online* Proses sistem pendaftaran siswa baru pada sekolah clarissa meliputi:

1. Halaman awal: halaman menu pendaftaran, login dan menu pendaftaran menampilkan Formulir pengisian biodata untuk kebutuhan data calon siswa seperti data orang tua, identitas calon siswa, dan asal pendidikan.

Pada halaman login, staff akan memasukkan kata sandi dan masuk ke kedalaman dashboard staff

2. Dashboard staff

a. Menu calon siswa

1. Pada halaman ini admin dapat melihat formulir online siswa yang masuk sesuai kategori yaitu TK, SD, SMP.
2. Staff dapat melihat detail formulir dan mengimport data yang ada di formulir ke menu kelola test karena calon siswa yang mendaftar harus mengikuti test tertulis terlebih dahulu.
3. Selain itu di menu ini juga terdapat fitur searching sehingga memudahkan staff untuk mencari data calon siswa yang spesifik. Terdapat informasi total formulir yang masuk.

b. Halaman kelola test

1. Pada menu ini bertujuan untuk *staff* menginput nilai test yang akan di kerjakan oleh staff sekolah.
2. selain itu pada menu ini juga *staff* bisa membuat jadwal test dan *equipment* yang dibawa oleh calon siswa ke sekolah dan jadwal ini akan muncul pada *website* informasinya yang dapat di cek oleh calon peserta.
3. Setelah *staff* menginput semua hasil nilai test tertulis maka akan di peroleh hasil secara otomatis nama-nama calon siswa yang lulus dan nama-nama yang tidak lulus.

4. Calon siswa yang lulus datanya dapat disimpan oleh staff secara otomatis dengan sekali klik satu persatu dan di beri akses untuk mengeditnya
 5. Setelah klik simpan data akan tersimpan kedalam *database* siswa dengan syarat calon peserta sudah menyelesaikan urusan administrasi dan melapor ke *staff* pengelola kesiswaan dengan menunjukkan bukti transaksi.
- c. Menu data siswa
1. Pada menu ini halaman akan menampilkan data-data siswa yang ada di dalam database siswa.
 2. staff di beri akses untuk mengedit data siswa serta menghapusnya.
- d. Form pendaftaran
1. Form ini diperuntukkan kepada calon peserta yang akan mendaftar diwajibkan mengisi semua data yang dibutuhkan sesuai panduan yang ada di form.
 2. Data yang tersimpan akan di kirim ke *staff* akan di proses lagi.

Adapun sistem pendaftaran *online* pada penelitian sebelumnya adalah pada SMK bina putra jakarta merupakan bentuk satuan pendidikan formal yang menyelenggarakan pendidikan kejuruan pada jenjang pendidikan menengah sebagai lanjutan dari Sekolah menengah pertama atau bentuk lain yang sederajat, atau sebagai kelanjutan dari hasil belajar yang dipersepsikan sederajat/setara dengan pendaftaran SMP secara *online* yang dilakukan sebagai berikut:

1. dapat mengisi biodata diri, calon siswa mengakses menu upload seperti dapat mengunggah ijazah, akte kelahiran, ktp Ayah kandung, atau wali. dan foto terbaru
2. kebutuhan staff sekolah akan sistem staff mengakses menu daftar pendaftar, mengelola biodata pendaftar , dapat menambah data pendaftar, dapat mengubah data pendaftar, dapat mencetak data pendaftar, dapat menghapus data pendaftaran, mengelola dokumen yang diupload (Sidik & Rahmawati, 2018).

2.1.2 *Unified Modelling Language (UML)*

Unified Modelling Language (UML) merupakan bahasa metode pemodelan visual untuk sistem atau perangkat lunak dengan berparadigma berorientasi objek. Konsep dasar dari UML terdiri atas struktur klasifikasi, *dynamic behavior*, dan model manajemen. UML didefinisikan diagram-diagram sebagai *Use case diagram*, *Class diagram*, *Activity diagram*, *Sequence diagram*, (Rahmawati, Nuryadi, & Aserih, 2021).

UML disimpulkan menjadi keseluruhan notasi grafis yang didukung oleh meta model tunggal yang membantu pendeskripsian dan disain yang dibangun menggunakan pemrograman berorientasi objek (Juliyanto, Rusman, & Ramanda, 2020). Menurut widodo prabowop “*Unified Modeling Language (UML)* adalah cara untuk menentukan, memvisualisasikan membangun dan mendokumentasikan artifact (informasi yang dibuat atau didapatkan melalui tahap pembuatan perangkat lunak, artefak digambarkan seperti berupa model, penjelasan, atau peralatan perangkat lunak)”. UML menyediakan notasi pemodelan Sistem multi-

sudut, UML juga digunakan tidak hanya di buat untuk pemodelan perangkat lunak, tetapi juga dipergunakan dalam semua bidang yang dilakukan dan dibutuhkan dipemodelan (Tedyyana, Ratnawati, & Kurniati, 2019). dalam Pengembangan sistem ada beberapa diagram UML sering digunakan yaitu:

1. Diagram *use case*

Use case merupakan gambaran dari fungsionalitas yang diharapkan dari sistem dan mewakili hubungan aktivitas terhadap aktor dengan sistem. dalam sebuah *use case*, terdapat aktor yang merupakan deskripsi entitas manusia atau sistem yang bekerja pada sistem menjelaskan hubungan antara sejumlah besar aktor eksternal dan kasus penggunaan yang disediakan oleh sistem. Bagaimana kasus penggunaan dijelaskan hanya dari apa yang dilihat aktor dari luar atau keadaan lingkungan sistem seperti yang dilihat oleh pengguna *System Functional Use Case Diagram*, manfaatnya untuk mendapatkan gambaran konkrit dari aspek dinamis dari sistem. *Use case* diagram digunakan untuk mengumpulkan kebutuhan sistem karena efek dari faktor internal dan eksternal.(Munawar, 2018).

Secara umum tujuan dari diagram *use case* dapat dideskripsikan sebagai berikut:

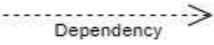
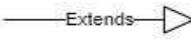
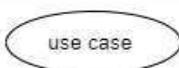
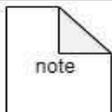
- a. Digunakan untuk mengumpulkan persyaratan dari sistem.
- b. Untuk mendapatkan tampilan dari luar sistem.
- c. Mengidentifikasi faktor-faktor yang mempengaruhi baik di dalam maupun di luar sistem untuk menunjukkan interaksi aktor dalam sistem.

Adapun Skenario *use case*, yang fungsinya untuk menjelaskan apa yang Sistem atau sub sistem lakukan contoh tabel dibawah sebagai berikut:

Tabel 2. 1 Skenario use case

<i>Use case name</i>	Nama <i>use case</i>
<i>Goal</i>	adalah tujuan dari <i>use case</i> tersebut
<i>Pre-condition</i>	adalah kondisi-kondisi yang harus terpenuhi sebelum <i>use case</i> tersebut di eksekusi.
<i>Post-condition</i>	adalah kondisi-kondisi yang harus terpenuhi setelah <i>use case</i> tersebut di eksekusi <i>use case</i>
<i>Failed end condition</i>	adalah <i>actor</i> utama yang berpasitipasi apabila terjadi kegagalan dalam eksekusi <i>use case</i>
<i>Primary actor</i>	adalah <i>actor</i> utama yang berpasitipasi dalam <i>use case</i> tersebut
<i>flow/basic path</i>	adalah arus kerja utama dengan asumsi semua berjalan lancar.
<i>Alternate flow</i>	adalah arus kerja alternatif dari arus kerja utama.

Penjelasan gambaran dan simbol beserta keterangan diagram *use case* ada pada gambaran berikut:

NO	SIMBOL	KETERANGAN SIMBOL
1.	 Actor	menspesifikasikan peran pengguna mainkan waktu berinteraksi di use case
2.	 Dependency	perubahan elemen mandiri yang akan memengaruhi elemen lainnya yang bergantung pada elemen ini.
3.	 Extends	menspesifikasikan use case bahwa untuk memperluas target perilaku dari use case sumber pada satu titik yang telah diberikan.
4.	 association	apa yang menghubungkan antara objek satu dengan objek lainnya
5.	 sistem	Menspesifikasikan paket yang menampilkan sistem secara terbatas
6.	 use case	deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
7.	 note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
8.	 Generalisasi	hubungan antara objek anak (Descendent) dan objek di atasnya objek induk (Ancestor) berbagi perilaku dan struktur data.

Gambar 2. 1 Simbol Diagram *Usecase*

2. Activity Diagram

Merupakan alur kegiatan serta gambaran pada sistem yang sedang berjalan. Diagram aktivitas merupakan aspek dinamis dari sistem yang menjelaskan penting dari UML. Bagi, diagram aktivitas dimaksudkan untuk menangkap perilaku dinamis sistem dengan menunjukkan aliran pesan dari

satu aktivitas ke aktivitas lainnya (Munawar, 2018). Tujuan diagram aktivitas secara umum, dapat dijelaskan sebagai berikut:

1. menjelaskan aliran aktivitas dari sistem.
2. Jelaskan tahap kegiatan dari satu kegiatan ke kegiatan lainnya.
3. Menjelaskan paralelisme sistem, percabangan, dan aliran paralel.

No	simbol	Nama simbol	keterangan Simbol
1		<i>Activity</i>	Bagaimana masing-masing kelas memperlihatkan antarmuka dan berinteraksi satu sama lain
2		<i>Initial node</i>	bagaimana objek dibentuk atau diawali
3		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Decision</i>	pilihan untuk pengambilan keputusan
5		<i>Fork node</i>	menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu

Gambar 2. 2 *Activity* diagram

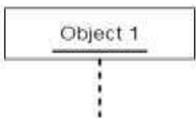
3. *Sequence* Diagram

Menjelaskan hubungan dalam bentuk pesan yang dijelaskan dari waktu ke waktu antara objek di dalam dan di sekitar sistem.

Secara umum, diagram urutan menunjukkan garis waktu alur pesan dari satu objek ke objek lainnya (Munawar, 2018).

Tujuan dari sequence diagram-diagram secara spesifik dapat dijelaskan yaitu:

- a. Model interaksinya dapat mendukung antara objek aktif didalam suatu sistem.
- b. Model interaksinya dapat mendukung antara *instance* objek di dalam kolaborasi yang dapat mewujudkan use case tersebut.
- c. Model interaksinya dapat didukung antara objek didalam kolaborasi yang mewujudkan operasi tersebut.
- d. Menunjukkkan sebagai bentuk interaksi *generic* (menunjukkan seluruh jalur yang hanya melalui interaksi). atau contoh spesifik dari suatu interaksi.

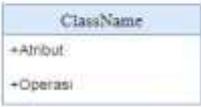
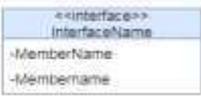
NAMA	SIMBOL	KETERANGAN
Objek/aktor		sebuah objek yang berasal dari kelas, atau dapat dinamai dengan kelasnya saja. aktor termasuk objek. garis putus-putus menunjukkan garis hidup suatu objek
Aktivitas		menunjukkan masa hidup dari objek
pesan		Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. interaksi antar objek ditunjukkan pada bagian operasi pada diagram kelas
Return		pesan kembalian dari komunikasi antar objek.

Gambar 2. 3 *Sequence* diagram

4. Class diagram

deskripsi struktur dan deskripsi setiap kelas, paket, dan objek terkait seperti pewarisan dan asosiasi. Menurut (Suendri, 2018) *class* diagram memiliki tiga (3) area utama adalah:

- Nama kelas: Area ini berisi nama yang ditetapkan untuk kelas.
- Attribut*: integritas ditambahkan ke kelas. Nilai kelas yang dimiliki jumlah atribut yang terbatas untuk diproses.
- Operasi: proses yang bisa diselesaikan oleh kelas di mana Individu atau kelas lainnya.

Simbol	Nama Simbol	Keterangan Simbol
	kelas	Kelas pada struktur sistem
	Antar muka/ <i>Interface</i>	Sama dengan konsep didalam pemrograman berorientasi objek
	<i>Directed Association</i>	Relasi antar kelas dengan makna kelas satu digunakan dengan kelas lainnya, asosiasi juga disertai dengan <i>multiplicity</i>
	<i>Depedency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas
	<i>Generalization</i>	Relasi antar kelas dengan makna <i>generalisasi spesialisasi</i> (Umum-Khusus)
	<i>Assoasiation</i>	Relasi antar kelas dengan makna umum, <i>asosiasi</i> biasanya sekaligus disertai <i>multiplicity</i>
	<i>Agregasi</i>	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Gambar 2. 4 *Class* Diagram

2.1.3 Aliran sistem informasi

Aliran Sistem Informasi (ASI) dimaksud disini adalah diagram yang menunjukkan dan mengarahkan aliran kerja dan keseluruhan sistem (Ismael, 2017). Gambar ini mendeskripsikan setiap urutan program serta langkah-langkah yang terdapat pada sistem, namun model yang bisa digunakan dalam alur sistem informasi adalah:

Gambar Simbol	Nama Simbol	Keterangan Simbol
	<i>Document</i>	merupakan formulir yang digunakan untuk merekam data yang menunjukkan input dan output, baik untuk proses manual maupun komputer
	<i>processing symbol</i>	Proses yang menggunakan komputer dimana pengolahan data dilakukan secara online
	Arsip	untuk menggambarkan penyimpanan data, baik dalam bentuk arsip atau file komputer dapat ditulis f au a
	<i>Manual operation</i>	digunakan untuk menggambarkan kegiatan manual atau pekerjaan yang dilakukan tanpa menggunakan komputer
	Disket	Menggambarkan input dan output dengan menggunakan disket
	<i>Decision</i>	Simbol pemilihan proses berdasarkan kondisi yang baik
	<i>manual input</i>	simbol untuk pemasukan data secara manual melalui online keyoard
	simbol penghubung	digunakan untuk menunjukkan sambungan dari bagan alir yang terputus d halaman yang masih sama atau halaman lainnya
	<i>Flow direction</i>	simbol yang berguna untuk menghubungkan antara simbol yang satu dengan yang lainnya

Gambar 2. 5 Simbol Aliran Sistem Informasi

2.2 Teori Khusus

Ada beberapa landasan teori khusus untuk membangun rancang bangun sistem informasi pendaftaran online pada sekolah Clarissa yang akan dibuat yaitu:

2.2.1 Website

Website merupakan alat *hypertext* yang menampilkan seperti teks, gambar, suara serta animasi dalam bentuk *multimedia*. data tersebut saling berhubungan. situs *web* yang dimaksud disini berupa semua halaman *web* yang terdapat dalam domain, termasuk area penyimpanan data dan informasi yang berisi topik tertentu (Utami & Hidayati, 2021) menurut dari jurnal penelitian Maslan, Arnomo, Teknik, & Batam, 2015 *website* merupakan situs kumpulan halaman web yang diringkas menjadi domain atau subdomain.

World Wide Web merupakan sebuah Fasilitas yang disediakan di Internet. *Web* adalah dunia virtual di Internet, dengan jutaan informasi tentang semua konten yang dikemas dalam bentuk dokumen *hypertext*. *hypertext* merupakan konsep dasar dari World Wide Web, sehingga dokumen *hypertext* adalah dokumen yang dapat dilihat dari halaman *web* kehalaman *web* satu halaman *web* lain-lainnya menggunakan tautan (Nurjaman & Yasin, 2020)

2.2.2 PHP

PHP merupakan *language* pemrograman yang banyak digunakan untuk menangani pembuatan dan pengembangan *Web*, serta dapat dibuat dalam bahasa HTML. Ini berarti sintaks dan perintah yang diserahkan akan dijalankan semuanya di *server*, tetapi terdapat di halaman HTML biasa, sehingga *script*

tidak terlihat di *client*. PHP dirancang untuk digunakan dengan *server database*, dan metode produksinya sangat memudahkan pengerjaan arsip HTML yang bisa memproses ke *database*. Dalam bahasa *scripting* ini mempunyai tujuan untuk membangun suatu *software* yang dihasilkan oleh PHP biasanya akan membagikan hasil pada *web browser*, namun keseluruhan metodenya berjalan di *server* (Susilo, 2018).

PHP juga merupakan bahasa standar dalam pengembangan *web*, selain *website*, lebih banyak aplikasi *web* yang dikembangkan menggunakan bahasa PHP (Moutaouakkil & Mbarki, 2020).

2.2.3 XAMPP

Extensible messaging and presence protocol (XMPP) merupakan gabungan teknologi yang dimanfaatkan untuk pesan instan, kehadiran, obrolan grup, panggilan suara dan video, hingga *middlewar* (Asmara & Saputra, 2019) awalnya xampp dikembangkan sebagai alternatif *open source* dan pesan instan terdesentralisasi, tetapi kemudian xampp dapat digunakan untuk berbagai keperluan. Sifat xampp memungkinkan protokol dikembangkan melebihi dari perancangan utamanya. diperluas melampaui desain aslinya.

Salah satu perkembangan protokol xampp adalah menyediakan banyak informasi terkini dalam komunikasi mesin-ke-mesin. Pengembangan ini mengizinkan entitas mesin mengirimkan status aktivitasnya sampai-sampai entitas mesin lain tidak penting mengomunikasikan data saat entitas mitra komunikasi *offline*, mekanisme ini boleh menyedikitkan kuantitas data yang dikirimkan pada

jaringan, yang secara tak refleks dapat meninggikan kapasitas jaringan. sifat *asynchronous* dari karakteristik xampp berpengaruh atas kemunculan data, data masuk dalam bentuk aliran, batas diperlukan metode yang berlainan untuk menampilkannya dengan benar di halaman *web*. Penelitian ini mengusulkan model aplikasi *Web* menggunakan xampp untuk menunjukkan data *asynchronous* dengan baik pada aplikasi *Web* (Asmara & Saputra, 2019).

2.2.4 Database

Basis data atau *database* merupakan gabungan data yang samasama berhubungan dalam organisasi maupun suatu perusahaan dan dikelola dengan memasang perangkat komputer yang berguna dalam memberikan informasi kepada pengguna dan mempermudah pengolahan data (H. Bahroni, 2020)

2.2.5 CodeIgniter

Menurut Betha Sidik mengatakan *Codeigniter* adalah *framework* PHP *open source* yang berperan dalam metode MVC (*Model, View, Controller*) demi melancarkan aplikasi berbasis *web* bagi *developer* dan *programmer* tidak mesti melakukan dari awal. Situs sah *codeigniter* menyatakan bahwa *codeigniter* adalah kerangka kerja PHP yang energik dengan lumayan *bug*. *Codeigniter* dibuat kepada pengembang yang menggunakan pemrograman PHP *language*, alat pembuatan *web* berfitur lengkap. (Rahmawati et al., 2021).

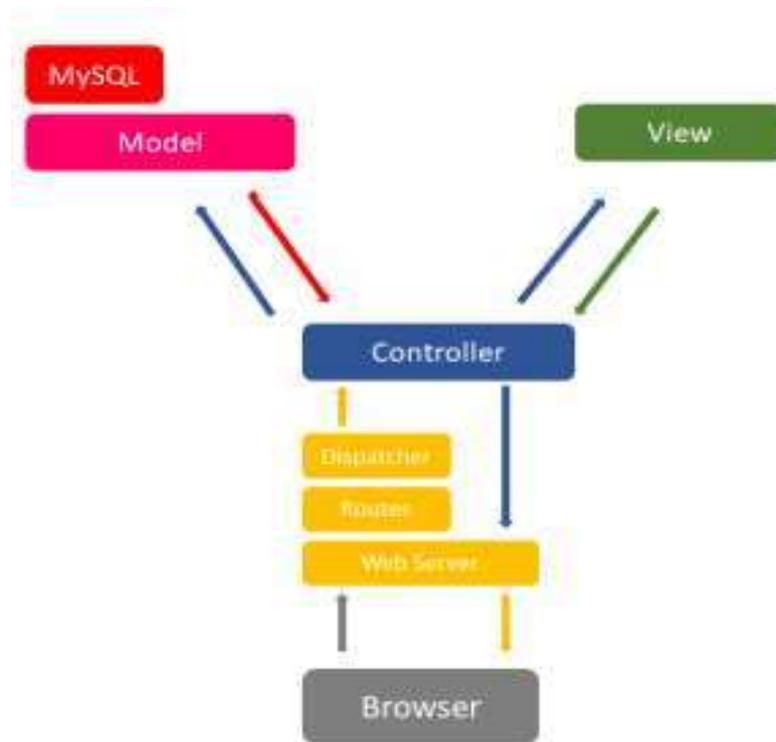
CodeIgniter atau CI merupakan kerangka kerja bahasa pemrograman PHP, yang dirancang untuk menyederhanakan dan memperkenalkan pemrograman PHP terstruktur dengan konsep MVC. *Model, view, dan controller* merupakan

kepenjangan dari MVC. menggunakan konsep MVC sebagai acuan pembuat *script*, programmer tidak perlu repot membaca dokumen yang ditulis oleh penulis *script* sebelumnya (Marlina, Masnur, & Dirga.F, 2021).

1. Komponen MVC pattern

Didalam buku (Yosef Murya Kusuma Ardhana, 2018, hal. 13–14) dideskripsikan suatu konsepsi yang cukup populer dalam pengembangan aplikasi *web*. Membangun MVC *pattern* terdapat tiga jenis komponen yaitu:

- a. Model, berkaitan dengan database atau *web servis* untuk mempresentasikan bentuk data dari *software* basis data maupun data lain yaitu dengan memanipulasi data dilakukan seperti; *insert, update, delete* dan *search* membenahi konfirmasi dari faktor *controller*, akan tetapi enggak boleh berhubungan langsung pada bagian *view* yang terlihat perintah-perintah *query SQL* yang biasanya berhubungan.
- b. *View, end-user* yang segala sesuatunya akan diperoleh. Biasanya berupa halaman web, rss, *javascript*. tampilan didedikasikan hanya untuk menampilkan data dari model dan pengontrol. Bagian ini tidak memiliki akses langsung ke bagian model.
- c. *controller* adalah pengatur interaksi model-*view* Pengontrol mengatur variabel yang ditampilkan dalam tampilan, menerima permintaan dan data dari pengguna, dan membantu menentukan apa yang ditangani aplikasi *Controller*.



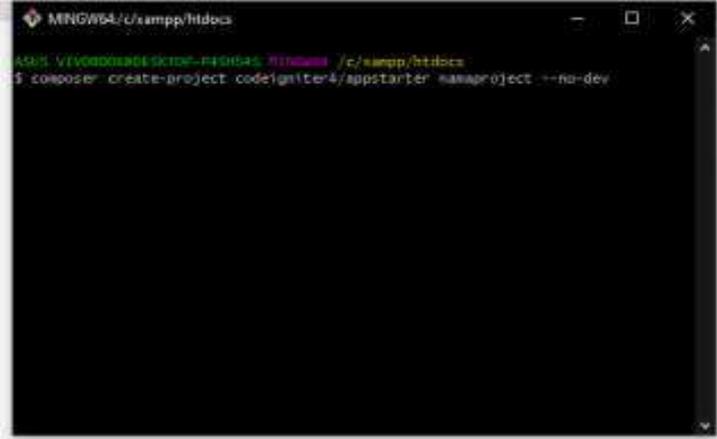
Gambar 2. 6 Cara kerja MVC

2. Instalasi CodeIgniter

CodeIgniter yang penulis gunakan untuk merancang aplikasi versi terbaru adalah CodeIgniter4. tahap *instalasi* CodeIgniter4 berbeda dengan CodeIgniter3, yang memungkinkan penulis untuk menginstal CodeIgniter4 online, Jika ada pembaruan dari kerangka kerja melalui terminal, Anda perlu mengubah kerangka lagi dan mengkonfigurasi ulang dengan Situs *web* yang kami rancang, tetapi perlu diperbarui Integrasikan langsung secara online melalui terminal atau composer Secara otomatis di situs *web* kami.

Tahap pengistalan *framework* codeigniter 4 dapat di uraikan sebagai berikut:

- a. Sebelum menginstal CI4 pastikan kita sudah menginstal terlebih dahulu composer di computer kita. Yaitu bisa didapatkan di [Composer \(getcomposer.org\)](https://getcomposer.org)
- b. Juga, pastikan kita sudah memiliki *server* lokal di komputer. yang paling sering digunakan adalah xampp untuk pengguna *Windows* dan *Linux* dan MAMP untuk pengguna *Mac*.
- c. Arahkan folder melalui *command prompt* ke `C:\xampp\htdocs` atau ditempat kita menginstal *local server* nya dan ketikkan `$ composer create-project codeigniter4/appstarter namaproject --no-dev`. Lalutekan *enter* dan tunggu prosesnya.

A screenshot of a terminal window with a black background and green text. The window title is 'MINGW64/C:/xampp/htdocs'. The prompt is '\$' and the command entered is 'composer create-project codeigniter4/appstarter namaproject --no-dev'. The cursor is at the end of the command.

```
MINGW64/C:/xampp/htdocs
$ composer create-project codeigniter4/appstarter namaproject --no-dev
```

Gambar 2. 7 instalasi codeigniter 4

- d. *Framework* berhasil diinstal.

2.2.6 *Bootstrap*

Bootstrap adalah salah satu kerangka kerja HTML, CSS, dan JS yang paling banyak digunakan oleh pengembang *web*. *Bootstrap* adalah template disain dengan fitur lebih. mempersiapkan alat untuk membuat tata letak halaman dengan mudah yang dapat Anda modifikasi dalam tampilan HTML dasar untuk membuat halaman *web* yang dikembangkan untuk komponen lain (Yuniva; Syafi'i, 2018).

Bootstrap merupakan kerangka kerja untuk konstruksi responsif desain *web*. Artinya tampilan *web* yang dibentuk oleh *bootstrap* akan mencocokkan ukuran layar didesktop, tablet atau *browser* seluler. kita dapat mengaktifkan atau menonaktifkan fitur ini sesuai keinginan kita. oleh karena itu, kita bisa melakukan *Web* hanya pada tampilan *desktop*, ketika dirender melalui *browser* seluler, Tampilan *web* yang saya buat tidak muat di layar. anda juga dapat menggunakan *bootstrap* untuk membangun *web* dinamis atau statis. *Bootstrap* menyediakan satu set komponen antarmuka dasar yang dirancang untuk membuat antarmuka yang menarik, bersih, dan ringan. selain komponen antarmuka kelas, *Bootstrap* juga memiliki fungsi *grid*. Ini digunakan untuk mengatur tata letak halaman *web* dan sangat mudah dan cepat digunakan. *Bootstrap* juga memberikan fleksibilitas untuk menggunakan *Bootstrap* untuk mengembangkan tampilan *website* yang kita buat. menambahkan *class* dan CSS itu sendiri untuk mengubah tampilan *bootstrap* (Nurjaman & Yasin, 2020)

1. Instalasi Bootstrap

Download file framework di <https://getbootstrap.com/>



Gambar 2. 8 instalasi bootstrap

Sesudah itu ekstrak filenya dan copy folder css dan js ke dalam website kita.

2.2.7 MySQL

Mysql merupakan yang dapat menampung data pada sistem *database* banyak yang bisa di akses setiap pengguna, karena dimanfaatkan dalam sistem SQL. Pada dasarnya SQL berperan sebagai penghubung terhadap program *database* dan digunakan dalam bahasa pemrograman. dengan hadirnya SQL, mempermudah dan tidak pernah mengalami kerumitan dalam menghubungkan aplikasi yang dibuat serta pemrograman jaringan (Bendo, Kuncoro, Sanjaya, & Aji, 2020). *Mysql* merupakan sistem manajemen basis data relasional yang paling populer, Ini adalah salah satu aplikasi RDBMS terbaik dalam aplikasi *web* (Zhu, Xu, & Gao, 2020).

MySQL merupakan database untuk *server*, *server* ramai dijadikan untuk merancang dan membangun *database* dalam sistem, Ini karena *MySQL* menggunakan SQL untuk mengakses *database* sebagai bahasa dasarnya. dengan

kode sumber (kode yang digunakan untuk membuat Mysql), file yang dapat dieksekusi atau kode yang dapat langsung dieksekusi di sistem (Sudrajat, 2021). *Mysql* merupakan standar terkini dalam pengembangan aplikasi berbasis *web*, memungkinkan akses data yang mudah dan jarak jauh, antarmuka pengguna (UI) untuk mengakses *database* dirancang dan diprogram dengan PHP (Welsh, Mezhuyev, & Irsa, 2020). menurut jurnal dari (Imam Rosyadi & Arum sari, 2018) adapun kelebihan mysql bisa di jelaskan beberapa bagian yaitu:

1. *Linux* dan *windows* terhadap MySQL bisa Ia bekerja secara stabil di berbagai sistem operasi.
2. MySQL didistribusikan secara gratis di bawah GNU *General Public License* (GPL). dengan secara *open source*.
3. Karena sifatnya multi-pengguna, Mysql bisa dimanfaatkan oleh banyak pengguna secara bersamaan tanpa menimbulkan masalah.
4. Mysql mempunyai kelancaran sangat bagus untuk memproses *query* (perintah SQL). Dalam hal lain bisa memproses lebih banyak SQL per satuan waktu.