

# IMPLEMENTASI MODEL-VIEW-CONTROLLER (MVC) PADA UJIAN ONLINE MELALUI PENERAPAN FRAMEWORK CODEIGNITER

Yohanes Rasuliano Laberto Kelen <sup>a</sup>,  
Benyamin Jago Belalawe <sup>b</sup>

<sup>a</sup> STIKOM Uyelindo Kupang, Kayu Putih - NTT,  
<sup>b</sup> STIKOM Uyelindo Kupang, Kayu Putih - NTT

<sup>a</sup> [exchellenz@live.com](mailto:exchellenz@live.com), <sup>b</sup> [pbcrewx@gmail.com](mailto:pbcrewx@gmail.com)

## ABSTRAK

Pendidikan termasuk salah satu bidang yang turut mengalami peningkatan dalam konteks penyelenggaraan ujian. Ujian konvensional yang masih menggunakan kertas perlahan-lahan mulai ditinggalkan. Penyelenggaraan ujian yang terkomputerisasi pun menjadi terobosan yang bukan lagi baru di zaman ini.

*Model, view, controller* (MVC) merupakan sebuah arsitektur pengembangan aplikasi yang menekankan kepada 3 komponen penting. Ketiga komponen tersebut masing-masing memiliki fokus perhatian, tanggung jawab dan logika sehingga mempercepat kinerjanya. *Framework codeigniter* merupakan salah satu *framework* terkenal yang sudah mendukung MVC. Dengan *framework* codeigniter, pengembangan aplikasi *web* dapat dilakukan dengan cepat. Sekolah Menengah Kejuruan Informatika Uyelindo Kupang pun merupakan salah satu sekolah dengan basis kejuruan informatika komputer. Sekolah ini dapat menerapkan ujian yang terkomputerisasi dengan sebuah sistem yang dinamakan sistem ujian *online*. Sistem ini dibangun dengan *framework* codeigniter.

Hasil implementasinya adalah sebuah sistem ujian yang berbasis *web* dengan *framework* codeigniter yang dapat mempermudah guru dan siswa dalam melaksanakan ujian yang fleksibel. Ujian ini dapat dilaksanakan dimana saja dengan waktu yang sudah ditetapkan oleh guru dan tidak lagi menggunakan kertas sebagai media pelaksanaan ujian. Harapan dengan adanya sistem ujian seperti ini, proses tatap muka di kelas menjadi lebih efektif dan guru dapat mengetahui kemampuan siswa dengan kuis yang bisa dikerjakan dari rumah.

**Keywords :** *codeigniter, framework, model view controller, MVC, ujian online.*

## 1. PENDAHULUAN

Peranan teknologi informasi sudah berkembang sedemikian pesat di segala bidang kehidupan manusia. Pendidikan termasuk salah satu bidang yang turut mengalami peningkatan dalam konteks penyelenggaraan ujian. Kecepatan, kepraktisan dan juga kemudahan yang tidak ditemukan pada ujian yang masih menggunakan kertas dapat bergeser ke arah komputerisasi dengan menerapkan ujian *online*.

Dengan penyelenggaraan ujian yang terkomputerisasi, diharapkan terjadi penekanan biaya terhadap penggunaan sumber daya kertas. Pelaksanaan ujian *online* yang terhubung dengan jaringan internet pun menjadi lebih fleksibel karena

bisa dilakukan di mana saja dan kapan saja. Siswa pun dapat menguji kemampuan belajar akan mata pelajaran dan secara tidak langsung efektifitas guru dalam pengajaran di sekolah menjadi lebih baik.

*Model, view, controller* (MVC) merupakan sebuah arsitektur pengembangan aplikasi yang menekankan kepada 3 komponen penting. Ketiga komponen tersebut masing-masing memiliki fokus perhatian, tanggung jawab dan logika sehingga mempercepat kinerjanya (Harianja, 2010). *Framework Codeigniter* merupakan salah satu *framework* terkenal yang sudah mendukung MVC. Dengan *framework* codeigniter, pengembangan aplikasi *web* dapat dilakukan dengan cepat.

## 2. METODE PENELITIAN

Metodologi adalah proses mendapat data sampai dengan pembuatan perancangan program yang akan dirancang atau dibangun. Dalam melakukan penelitian penulis menggunakan beberapa metode untuk mendapatkan data yaitu: metode kepustakaan, yakni mencari literatur atau sumber-sumber pustaka pendukung yang mampu memberikan informasi yang memadai; metode wawancara, yakni melakukan pembicaraan dengan pihak terkait.

### A. Tinjauan umum software

#### 1) Hypertext Preprocessor

Menurut Peranginangin (2006), PHP singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* yang disisipkan pada dokumen *html*. Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP merupakan software *open-source* yang disebarluaskan dan dilisensikan secara gratis.

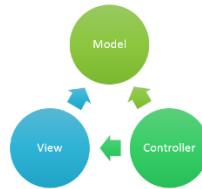
#### 2) MySQL

Kadir (2010) menyatakan MySQL tergolong sebagai DBMS (*Database Management System*). Perangkat lunak ini bermanfaat untuk mengelola data dengan cara yang sangat fleksibel dan cepat.

### B. Tinjauan pustaka

#### 1) MVC

MVC (*Model, View* dan *Controller*) yang merupakan sebuah arsitektur untuk membuat sebuah program. Tujuan dari pembagian program ke dalam tiga bagian besar ini adalah untuk memisahkan fokus perhatian, tanggung jawab, dan logika ke dalam bagian masing-masing. (Harianja, 2010).



Gambar 1. *Model, view, controller*

- a) *Model* adalah bagian dari aplikasi yang mengimplementasikan logika untuk domain data aplikasi. Umumnya, objek *model* digunakan untuk mengambil data dari *database* atau menyimpan data ke dalam *database*.
- b) *View* adalah komponen yang menampilkan antarmuka (*user interface*) untuk pengguna aplikasi. Antarmuka ini dibuat berdasarkan data dari *model*.
- c) *Controller* merupakan komponen yang digunakan untuk menangani interaksi pengguna, bekerja dengan *model*, dan memilih *view* mana yang digunakan untuk me-render data.

#### 2) Framework

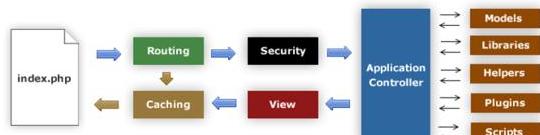
*Framework* adalah kumpulan kode program yang disimpan pada *file* yang berbeda yang dapat menyederhanakan operasi yang berulang-ulang. Kelebihan dengan adanya *framework* adalah dapat melakukan pengembangan aplikasi menjadi seragam, dimana pengembang dituntut untuk mengikuti alur kerja yang ditetapkan oleh *framework*. (Harianja, 2010).

#### 3) Codeigniter

Sidik (2012) menyatakan bahwa codeigniter (CI) adalah *framework* pengembangan aplikasi (*application development framework*) yang memiliki suatu kerangka yang sistematis untuk bekerja atau membuat program dengan menggunakan PHP. Codeigniter menyediakan sekumpulan *library* yang banyak untuk keperluan menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengakses *library*-nya.

- a) *index.php* berlaku sebagai *controller* di depan, yang menginisiasi sumber

- daya yang dibutuhkan untuk menjalankan codeigniter,
- Router* akan memeriksa *request HTTP* untuk menentukan apa yang harus dilakukan dengan *request* tersebut,
  - Jika ada *file cache*, maka *file cache* langsung dikirimkan kepada *browser*, dengan memotong eksekusi sistem normal,
  - Untuk keamanan, sebelum suatu aplikasi *controller* dipanggil, *request HTTP* dan data yang dikirimkan oleh pengguna disaring (filter) terlebih dahulu,
  - Controller* akan memuat *model*, *library*, *helper*, dan sumber daya lainnya yang diperlukan untuk memproses *request* tersebut.
  - View* terakhir di-*render* kemudian dikirimkan kepada *browser* untuk dilihat. Jika *caching* didefinisikan, maka *view* akan di-*cache* terlebih dahulu, setelah itu baru dikirimkan kepada *browser*.



Gambar 2. Flowchart codeigniter mengimplementasikan MVC

### C. Perancangan sistem

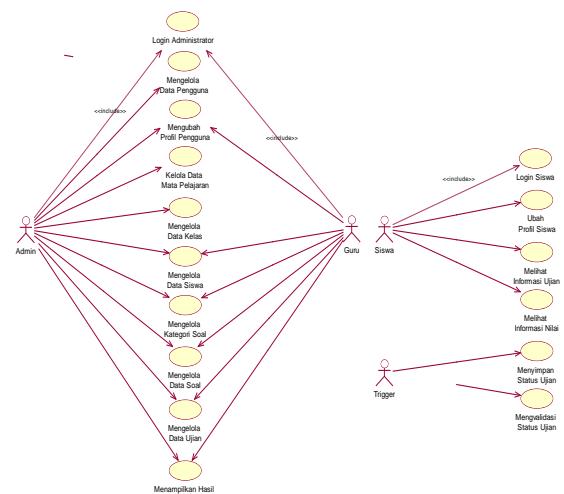
*Unified Modeling Language* (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem. UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2005).

UML memiliki beberapa diagram yang mampu membantu pengembang mengkomunikasikan sistem yang akan mereka buat, diagram-diagram tersebut

antara lain adalah *use case*, *class diagram*, dan *sequence diagram*.

#### 1) Use case diagram

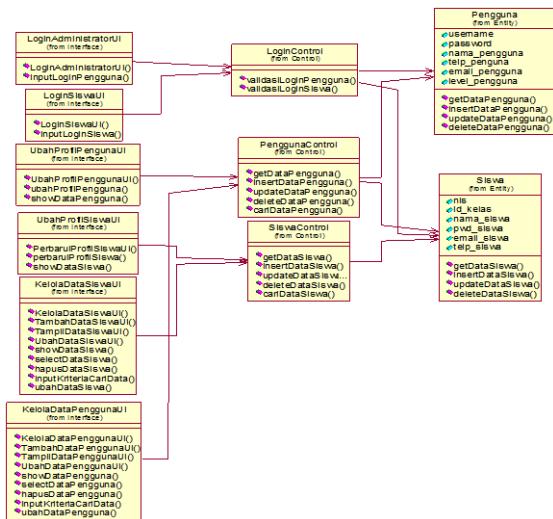
*Use case* merupakan penjelasan fungsi dari sebuah sistem melalui perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan jenis interaksi antara *user (actor)* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan hubungan antar *actor* dengan sistem disebut dengan *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna (Munawar, 2005).



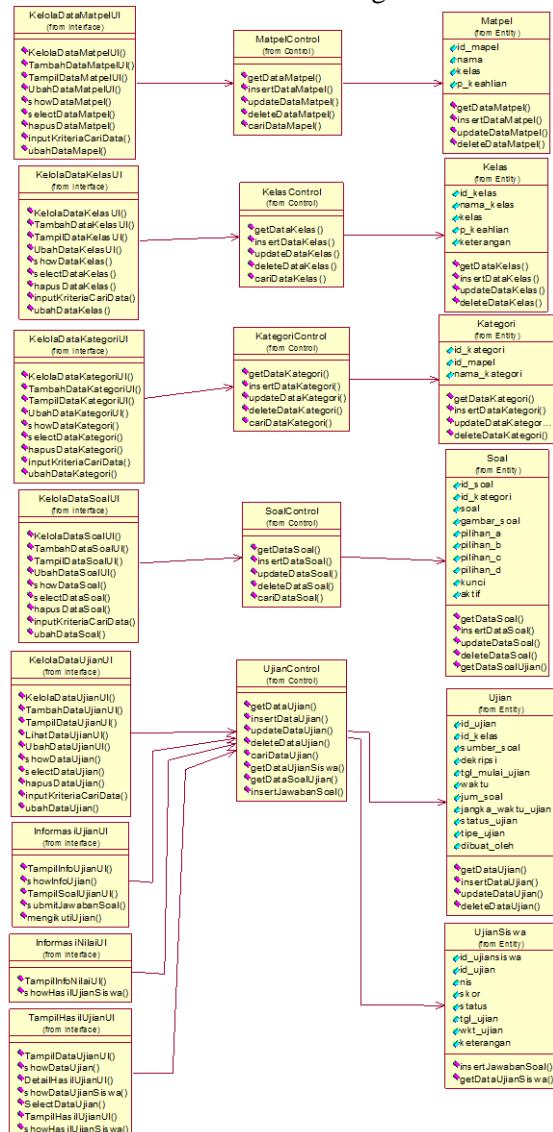
Gambar 3. Use case diagram

#### 2) Class diagram

*Class diagram* merupakan representasi sebuah gambar yang memperlihatkan *attribut* atau *property* serta operasi yang dimiliki oleh suatu objek dan menggambarkan hubungan objek lainnya. *Class* biasanya digunakan untuk mendefinisikan objek-objek bisnis. *Class* seperti ini biasanya mendefinisikan model *database* dari suatu aplikasi (Munawar, 2005).



Gambar 5. class diagram



Gambar 6. class diagram (lanjutan)

### 3) Sequence diagram

Irwanto (2006) menyatakan bahwa digunakannya *sequence diagram* karena kita ingin melihat rangkaian urut-urutan *event* yang terjadi. Pada fase tersebut kita masih dimungkinkan untuk melakukan pemeriksaan secara literasi operasi-operasi dan atribut-atribut yang masih kurang. Jika ditemukan masih ada operasi yang kurang, maka tambahkan operasi tersebut ke dalam ke dalam *class diagram*.

### 4) Desain basis data

Berdasarkan *class diagram* maka dapat dibentuk 8 tabel. Berikut ini adalah rincian tabel-tabel yang dihasilkan.

Tabel 1. Tabel pengguna

Nama	Tipe	Panjang
username	varchar	30
password	varchar	50
nama_pengguna	varchar	40
telp_pengguna	varchar	12
email_pengguna	varchar	30
level_pengguna	varchar	10

Tabel 2. Tabel kelas

Nama	Tipe	Panjang
id_kelas	int	6
nama_kelas	varchar	40
kelas	int	2
p_keahlilan	varchar	3
keterangan	text	-

Tabel 3. Tabel mata pelajaran

Nama	Tipe	Panjang
id_mapel	int	5
nama_mapel	varchar	40
kelas	int	2
p_keahlilan	varchar	3
keterangan	text	-

Tabel 4. Tabel siswa

Nama	Tipe	Panjang
nis	int	5
id_kelas	int	6
nama_siswa	varchar	40
pwd_siswa	varchar	50
email_siswa	varchar	30
telp_siswa	varchar	12

Tabel 5. Tabel kategori soal

Nama	Tipe	Panjang
------	------	---------

id_kategori	int	5
id_mapel	int	5
nama_kategori	varchar	35

**Tabel 6.** Tabel soal

Nama	Tipe	Panjang
id_soal	int	10
id_kategori	int	5
soal	text	-
gambar_soal	varchar	70
pilihan_a	text	-
pilihan_b	text	-
pilihan_c	text	-
pilihan_d	text	-
pilihan_e	text	-
kunci	varchar	2
aktif	varchar	2

**Tabel 7.** Tabel ujian

Nama	Tipe	Panjang
id_ujian	int	10
id_kelas	int	6
sumber_soal	int	5
deskripsi	varchar	40
tgl_mulai_ujian	date	-
waktu	varchar	8
jum_soal	int	5
jangka_waktu_ujian	date	-
status_ujian	varchar	30
tipe_ujian	varchar	10
dibuat_oleh	varchar	30

**Tabel 8.** Tabel ujian siswa

Nama	Tipe	Panjang
id_ujiansiswa	int	10
id_ujian	int	10
nis	int	5
skor	int	10
status	varchar	15
tgl_ujian	date	-
wkt_ujian	int	-
keterangan	varchar	45

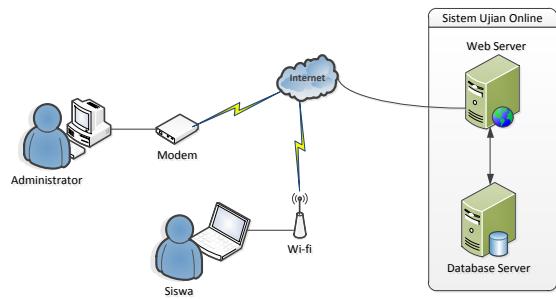
### 3. HASIL DAN PEMBAHASAN

#### A. Arsitektur perangkat lunak

Sistem ujian *online* merupakan sebuah perangkat lunak yang digunakan untuk membantu siswa melaksanakan ujian pada SMK Informatika Uyelindo Kupang. Sistem ujian ini dibuat menggunakan *framework codeigniter* yang menerapkan *Model, View, Controller* (MVC), PHP

(*Hypertext Preprocessor*) sebagai bahasa pemrograman dan MySQL sebagai *Database Management System* (DBMS).

Pada sistem ini arsitektur perangkat lunak yang digunakan berupa *web database server*, dimana segala permintaan (*request*) pengguna diproses oleh *web server* dan semua data tersimpan pada *database server*. Pengguna sistem yakni *administrator* dan siswa dapat mengakses sistem ujian *online* melalui *browser* pada *personal computer* (PC) atau *notebook* yang terhubung dengan jaringan internet melalui *modem* atau *wi-fi*. *Web server* akan memproses data pada *database server* ketika pengguna melakukan permintaan ke sistem ujian *online*.



**Gambar 7.** Arsitektur sistem ujian *online*

#### B. Rekomendasi server

Perangkat lunak sistem ujian *online* ini pada realisasinya akan diakses oleh guru dan para siswa. Hal tersebut akan meningkatkan *traffic* ketika mengakses sistem ini dan berpotensi mengakibatkan sistem menjadi *down*. Maka dari itu, rekomendasi *server* yang bisa disarankan untuk sistem ujian *online* sebagai berikut :

**Tabel 9.** Rekomendasi *server*

No	Komponen	Rekomendasi
1.	<i>Server Host</i>	Server IIX
2.	<i>Disk space</i>	1 GB
3.	<i>Bandwidth</i>	50 GB
4.	<i>Web server</i>	LiteSpeed Web Server
5.	<i>PHP support</i>	PHP 5.3.x Support
6.	<i>Database</i>	MySQL 5 Databases

#### C. Antarmuka sistem

##### 1) Halaman *login*

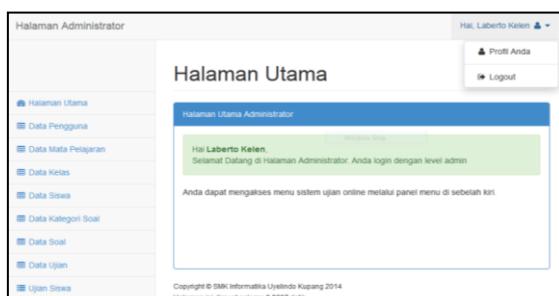
Halaman *login administrator* adalah halaman yang ditampilkan pada saat

halaman *administrator* diakses pertama kali oleh pengguna (pengguna belum melakukan *login* pada sistem). Pada halaman ini, pengguna harus mengisi *username* dan *password* yang sudah terdaftar dalam *database system*.

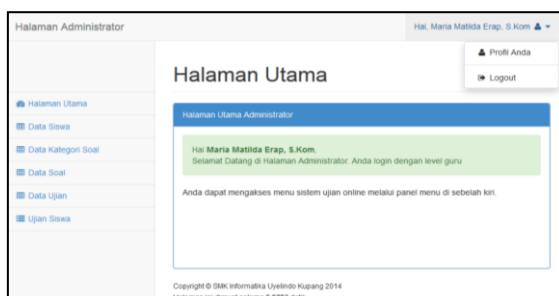


**Gambar 8.** Halaman login administrator

Halaman utama *administrator* adalah halaman utama ketika pengguna berhasil *login* ke dalam sistem. Pada halaman ini, terdapat menu yang terdapat pada sisi kiri halaman. Ada 2 jenis *administrator*, yakni halaman *administrator* yang diakses oleh pengguna dengan level admin dan level guru.



**Gambar 9.** Halaman utama admin



**Gambar 10.** Halaman utama guru

Halaman *login* siswa adalah halaman yang ditampilkan pada saat halaman siswa diakses pertama kali oleh siswa (siswa belum melakukan *login* pada sistem). Pada

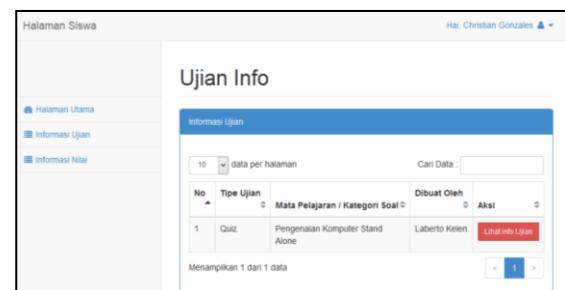
halaman ini, siswa harus mengisi Nomor Induk Siswa (NIS) dan *password* yang sudah terdaftar dalam *database* sistem.



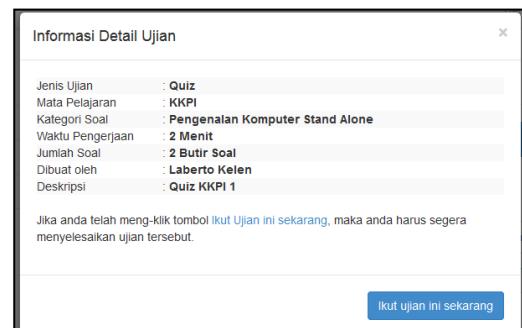
**Gambar 11.** Halaman login siswa

## 2) Halaman ujian

Halaman informasi ujian adalah halaman yang menampilkan informasi ujian yang telah dibuat kepada siswa. Siswa dapat mengklik tombol lihat info ujian untuk melihat informasi detail ujian.



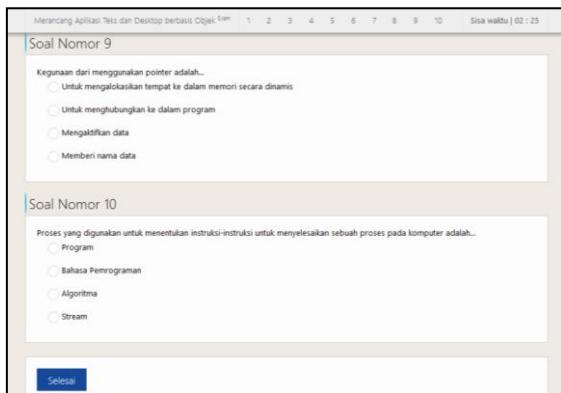
**Gambar 12.** Halaman ujian yang tersedia



**Gambar 13.** Konfirmasi mengikuti ujian

Halaman soal ujian adalah halaman yang menampilkan soal-soal ujian. Halaman ini diciptakan otomatis sesuai jenis ujian (*exam/quiz*) dan soal-soal yang disajikan

kepada siswa ditampilkan secara acak (*random*).



**Gambar 14.** Tampilan lembar soal ujian ketika diakses

#### 4. KESIMPULAN

Pada tahapan implementasi MVC pada sistem ujian *online* dengan *framework* codeigniter ini membuat pelaksanaan ujian pada SMK Informatika Uyelindo Kupang menjadi lebih mudah dan fleksibel karena siswa bisa melakukan ujian dari mana saja dan sesuai waktu yang ditetapkan guru. Penerapan sistem ini diharapkan dapat

meminimalisir penggunaan kertas dalam pelaksanaan ujian.

#### DAFTAR PUSTAKA

- [1] Peranginangin, Kasiman, Aplikasi Web dengan PHP dan MySQL, Penerbit Andi, Yogyakarta, 2006.
- [2] Kadir, Abdul, Mudah Mempelajari Database MySQL, Penerbit Andi, Yogyakarta, 2010.
- [3] Harianja, Jhonson P., Sistem Ujian Adaptif untuk Seleksi Penerimaan Mahasiswa Baru pada Perguruan Tinggi Berbasis Web melalui Penerapan Teknik MVC dengan menggunakan PHP Framework Codeigniter, Universitas Maritim Raja Ali Haji, Jurnal, 2010.
- [4] Sidik, Betha, Framework CodeIgniter, Penerbit Informatika, Bandung, 2012.
- [5] Munawar, Pemodelan Visual dengan UML, Penerbit Graha Ilmu, Yogyakarta, 2005.
- [6] Irwanto, Djon, Perancangan Object Oriented Software dengan UML, Penerbit Andi, Yogyakarta, 2006.