

# PENGEMBANGAN APLIKASIKONVERSI MATA KULIAH MENGGUNAKAN FRAMEWORKCODEIGNITER DAN BOOTSTRAP

(Studi Kasus : Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia)

Abu Sopian

Program Studi Teknik Informatika,  
Fakultas Komputer-Universitas Mohammad Husni ThamrinJakarta, Indonesia  
ianprosia2@gmail.com

## Abstract

*Students who enroll in the Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia are not only from High School Graduates or from the same level. But students who enroll are also available from Bachelor Degree (D3) or called Level Transfer. For that, it is necessary to do equalization or conversion of courses from Bachelor Degree (D3). Currently the conversion of subjects is still done manually. For that, it is necessary to develop an application that can help in converting for subject. In this study the development of course conversion applications is used Software Development Life Cycle (SDLC) method with the waterfall model. Based on the analysis of the system design above, it can be concluded that the Model View Controller (MVC) architecture can be applied to the codeigniter framework and MVC architecture making it easier to repair changes to the source code without changing the entire logic and function properly and meet the objectives the beginning of the development of this subject conversion application.*

**Keywords:** Framework Codeigniter, Framework Bootstrap, Conversion

## I. PENDAHULUAN

Menurut data PDDIKTI tahun 2018 jumlah Perguruan Tinggi yang ada di DKI Jakarta adalah 316 Perguruan Tinggi (Kementerian Riset, Teknologi Dan Pendidikan Tinggi, 2013-2018). termasuk salah satunya Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia. Sedangkan yang berbentuk akademi jumlahnya adalah 117 Akademi. Mahasiswa yang mendaftar kuliah di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia bukan hanya mahasiswa lulusan dari sekolah menengah atas atau sederajat, termasuk didalamnya terdapat juga mahasiswa alih jenjang dari Diploma Tiga (D3) ke tingkat Strata Satu (S1) dengan program studi ilmu komunikasi yang sesuai dengan program studi di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia yaitu program studi ilmu komunikasi.

Berbeda dengan Mahasiswa dari lulusan sekolah menengah atas atau sederajat, mahasiswa lulusan Diploma Tiga (D3) atau alih jenjang yang melanjutkan studi ke jenjang strata satu (S1) di Sekolah Tinggi Ilmu Komunikasi

Profesi Indonesia perlu dilakukan penyetaraan mata kuliah terlebih dahulu atau dilakukan konversi mata kuliah dari transkrip nilai Diploma Tiga (D3) untuk diakui atau disetarakan dengan mata kuliah di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia. Tidak semua mata kuliah yang telah diambil pada Diploma Tiga (D3) dapat diakui di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia. Maka diperlukan untuk memilah mata kuliah yang diakui pada transkrip nilai mahasiswa Diploma Tiga (D3) yang akan melanjutkan studi ke jenjang strata satu (S1) di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia. Konversi adalah perubahan dari satu sistem pengetahuan ke sistem yang lainnya (Badan Pengembangan dan Pembinaan Bahasa, Kemdikbud, 2012-2018). Konversi mata kuliah adalah proses dimana mata kuliah dari kurikulum Perguruan Tinggi asal yaitu pada jenjang Diploma Tiga (D3) yang dibuktikan dengan transkrip nilai disetarakan atau dikonversi dengan kurikulum yang ada di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia.

Teknologi informasi berkembang sangat pesat sehingga penggunaan teknologi informasi yang tepat dapat menunjang keberhasilan diberbagai bidang, salah satunya percepatan dibidang konversi. Saat ini di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia untuk melakukan penyetaraan mata kuliah atau mengkonversi mata kuliah masih menggunakan pencatatan manual dengan aplikasi Microsoft Excel.

Permasalahan yang timbul dari kegiatan konversi tersebut adalah banyaknya data mata kuliah yang harus diolah dan di dipilah agar saat terjadinya proses transaksi sesuai dengan jumlah kredit mata kuliah yang tepat, berdasarkan mata kuliah yang pernah diambil di perguruan tinggi asal disetarakan atau diakui di Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia. Dan sering terjadi perbedaan hasil dari penyetaraan matakuliah mahasiswa dari perguruan tinggi yang sama dan program studi yang sama serta dari angkatan yang sama. Untuk itu diperlukan sebuah aplikasi yang dapat membantu Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia khususnya pada program studi ilmu komunikasi untuk dapat menunjang percepatan pelayanan penyetaraan mata kuliah atau konversi mata kuliah agar menghasilkan penyetaraan matakuliah yang akurat.

Perancangan aplikasi konversi mata kuliah dilakukan dengan menggunakan metode *Software Development Life Cycle (SDLC)* dengan model *waterfall*. Implementasi aplikasi konversi mata kuliah ini menggunakan *framework Codeigniter dan Bootstrap* dan menggunakan arsitektur *Model, View dan Controller (MVC)*. Serta menggunakan *Data-base Management System (DBMS) MySQL* sebagai data-base.

## II. LANDASAN TEORI

Beberapa landasan teori yang digunakan dalam melakukan perancangan aplikasi konversi matakuliah ini adalah:

### 2.1 *Software Development Life Cycle (SDLC)*

*Software Development Life Cycle (SDLC)* atau sering disebut juga *System Development*

*Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rosa & Shalahuddin, 2014). Untuk penerapan tahapan prosesnya menggunakan model *waterfall*. *Codeigniter* menerapkan konsep *Model, View, dan Controller (MVC)*.

### 2.2 *Codeigniter*

*Codeigniter* merupakan sebuah *framework* PHP yang berupa kumpulan folder dan file PHP, JavaScript, CSS, TXT, dan file berbasis web lainnya (Sofwan, 2016). Di dalam *Framework Codeigniter* terdapat kelas yang berbentuk *library* dan *helper* berfungsi untuk membantu programmer dalam mengembangkan aplikasi. Konsep dalam *Framework Codeigniter* yaitu dengan menerapkan konsep *Model, View, dan Controller (MVC)*. *MVC* adalah sebuah pola perancangan atau arsitektur yang digunakan untuk rekayasa perangkat lunak dimana para pengembang dapat memisahkan kode program kedalam tiga kelompok yaitu *Model, View* dan *Controller* (Sidik & Betha, 2012).

### 2.3 *Bootstrap*

*Bootstrap* adalah sebuah *library framework* CSS yang dibuat khusus untuk bagian pengembangan *front-end website* (Zakir, 2016). *Bootstrap* juga merupakan salah satu *framework* HTML, CSS dan javascript yang paling populer di kalangan web developer yang digunakan untuk mengembangkan sebuah website yang responsive. Sehingga halaman website nantinya dapat menyesuaikan sesuai dengan ukuran monitor device (desktop, tablet, ponsel) yang digunakan pengguna disaat mengakses website dari browser. Pada mulanya *bootstrap* bernama "*Twitter Blueprint*" yang dikembangkan oleh Mark Otto dan Jacob Thornton di Twitter sebagai kerangka kerja untuk mendorong konsistensi di alat internal.

### 2.4 *MYSQL*

*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 (Arief & Rudianto, 2011).

Mysql adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa query utamanya adalah *Transact-SQL* yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Umumnya SQL Server digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya SQL Server pada basis data besar.

## 2.5 UML

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek (Nugroho, 2015). UML membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (Fowler, 2005). UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

### 2.5.1 Use Case Diagram

*Use Case Diagram* 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. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

### 2.5.2 Diagram Aktivitas (Activity Diagram)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram ini memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem.

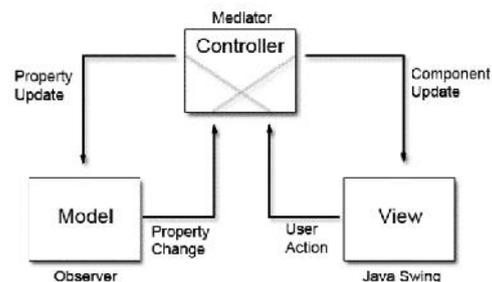
### 2.5.3 Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Class diagram

juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan obyek yang dikoneksikan. Class diagram secara khas meliputi: Kelas (Class), Relasi, Associations, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar Kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau kardinaliti.

## 2.6 Metode MVC

*Model-View-Controller* atau *MVC* adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasinya kebanyakan *framework* dalam aplikasi berbasis arsitektur *MVC*. *MVC* memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi.



Gambar 1. Model View Controller (MVC)

## III. METODE PENELITIAN

Perancangan aplikasi dibangun berdasarkan aturan dan prosedur menggunakan metode *Software Development Life Cycle (SDLC)* dengan model *waterfall* yaitu terbagi dalam 4 tahapan, yaitu : 1) Planning, 2) Analisis, 3) Desain, 4) Implementasi.



Gambar 2. Tahapan Penelitian

Tahapan penelitian dapat dijelaskan sebagai berikut:

Tahap pertama adalah planning. Planning adalah tahap perencanaan awal pengembangan aplikasi konversi dengan mengajukan proposal kepada pimpinan Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia.

Tahap kedua adalah analisis. Analisis terdiri dari analisa kebutuhan dan pengumpulan data. Analisis kebutuhan dan pengumpulan data dilakukan dengan cara observasi dan wawancara. Dari hasil observasi dan wawancara maka diperlukan sebuah aplikasi untuk dapat membantu proses penyetaraan matakuliah atau konversi matakuliah.

Tahap ketiga adalah desain. Pada tahapan desain dilakukan perancangan arsitektur dengan menggunakan MVC, perancangan aplikasi dengan menggunakan bahasa pemodelan UML dan perancangan database.

Tahap keempat adalah implementasi. Tahapan implementasi terdiri dari pengujian dan penyerahan aplikasi. Pada tahapan pengujian dilakukan dengan pendekatan *Black Box Testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program (Rosa & Shalahuddin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek, 2014). Tahapan penyerahan aplikasi adalah tahapan terakhir dalam perancangan aplikasi dimana aplikasi yang telah selesai di uji maka akan diserahkan kepada Sekolah Tinggi Ilmu Komunikasi Profesi Indonesia.

## IV. PEMBAHASAN

### 4.1 Perancangan Sistem

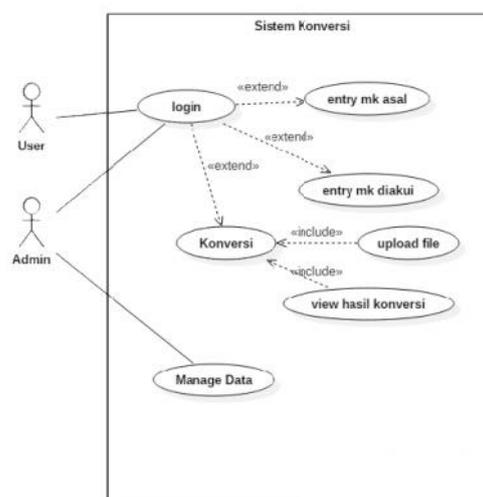
Dari analisis sistem berjalan maka dapat dilihat ada beberapa transaksi yang dilakukan yaitu :

1. *Login* untuk mengakses aplikasi konversi nilai mahasiswa.
2. *Input* atau *upload* mata kuliah asal, mata kuliah diakui serta mapping antara mata kuliah asal dan mata kuliah diakui

3. *Upload* data mahasiswa termasuk mata kuliah dan nilai untuk dilakukan proses konversi.
4. Melihat hasil konversi yang berupa data mata kuliah asal dan mata kuliah yang diakui.

#### 4.1.1 Use Case Diagram

Pembuatan *Use Case* yang dibuat setelah melakukan analisa terhadap analisis sistem berjalan dapat dilihat pada gambar 2 dibawah ini :

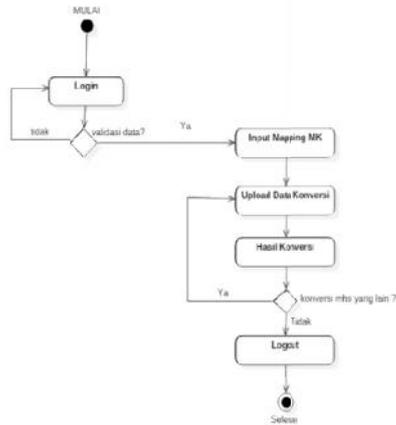


Gambar 3. Use Case Sistem Berjalan

Penjelasan gambar tersebut adalah sebagai berikut : Terdapat 2 aktor utama yaitu *User* dan *Administrator*. *User* dapat melakukan penginputan data mata kuliah asal, mata kuliah diakui, melakukan proses konversi mata kuliah dan melihat hasil konversi dari masing-masing mahasiswa yang datanya sudah diupload pada system konversi mata kuliah. Untuk *user Administrator* mempunyai kewenangan yang sama dengan *user akademik* akan tetapi pada administrator ditambah bisa manage user yang bisa mengkasas system konversi.

#### 4.1.2 Activity Diagram

Diagram ini dibuat berdasarkan aktifitas yang terjadi dalam sistem berjalan. Diagram aktifitas dapat dilihat pada gambar dibawah berikut ini:



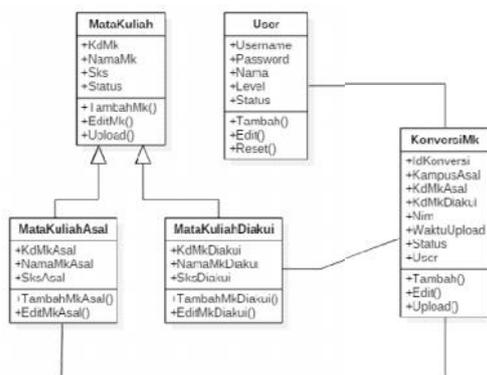
Gambar 4. Activity Diagram

Penjelasan untuk gambar diatas adalah sebagai berikut :

Setiap user yang akan melakukan proses konversi nilai mahasiswa diharuskan login ke system terlebih dahulu. Sebelum proses konversi diproses dipastikan bahwa data mata kuliah asal yang diakui harus sudah di mapping di system konversi tersebut. Data mahasiswa yang akan di konversi diupload sesuai dengan format yang sudah ditentukan kemudian diproses maka akan menghasilkan data mata kuliah asal dan mata kuliah yang diakui dari masing-masing mahasiswa yang datanya sudah diupload sebelumnya.

#### 4.1.3 Class Diagram

Class diagram yang terbentuk sesuai dengan bisnis berjalan adalah sebagai berikut :



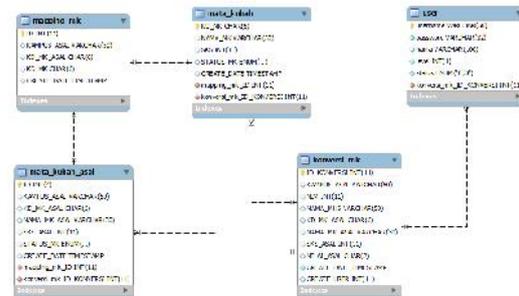
Gambar 5. Class Diagram

Penjelasan untuk gambar class diagram diatas adalah :

Yaitu menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun aplikasi. Kelas memiliki atribut dan metode. Kelas-kelas yang ada dalam perancangan aplikasi adalah kelas user, kelas matakuliah, kelas matakuliah asal, dan kelas matakuliah yang diakui serta kelas konversi matakuliah.

#### 4.2 Perancangan Database

Database yang digunakan dalam perancangan aplikasi konversi mata kuliah ini adalah MySQL. Dengan hasil desain sebagai berikut :



Gambar 6. Perancangan Database

Penjelasan untuk gambar perancangan database diatas adalah: rancangan database terdiri dari entitas, atribut dan relasi serta asosiasi. Entitas user, entitas matakuliah, entitas mping matakuliah, dan entitas matakuliah asal, serta entitas konversi matakuliah.

#### 4.3 Perancangan Aplikasi

Perancangan aplikasi adalah rancangan keluaran yang merupakan rancangan desain tampilan dari aplikasi konversi nilai mahasiswa yang informasinya berasal dari pembuat, kemudian informasi tersebut disusun dan ditampilkan kepada pengguna. Dalam merancang tampilan, rancangan harus mudah digunakan oleh pengguna agar pengguna dapat dengan mudah menggunakan fitur yang ada pada aplikasi konversi tersebut. Kemudahan akan mempengaruhi efektivitas dan efisiensi pengguna dalam menjalankan pekerjaannya.

Perancangan aplikasi dengan *framework Codeigniter* dimulai dari pembuatan *baseURL* di *config.php* yang terletak pada *folder Codeigniter*. *Base URL* adalah *link* tetap untuk mengakses *resources* dan *link* lainnya. Setelah pembuatan *base URL* dilanjutkan dengan menghubungkan *codeigniter* dengan *bootstrap*. *Bootstrap* diletakkan didalam *folder assets*. *Folder assets* dibuat untuk menampung *file* atau *folder* dari luar *folder codeigniter*.

*Model View Controller (MVC)* pada perancangan aplikasi konversi matakuliah ini adalah sebagai pola arsitektur. Dimana aplikasi dibagi tiga *folder* utama yaitu *folder Model*, *folder view* dan *folder controller*. *Request* (permintaan) *browser* ditangani oleh bagian *Controller*. *Controller* akan melakukan pemanggilan ke *Model* untuk mendapatkan data yang relevan, dan kemudian mempersiapkan data tersebut untuk ditampilkan (*View*). *Controller* memberikan data yang diperlukan kepada *view*. *View* menampilkan data dan berbagai elemen antarmuka tambahan yang diperlukan.

Aplikasi konversi matakuliah ini memiliki dua level *user* yaitu administrator dan admin dengan masing-masing *user* tersebut diatas memiliki hak akses yang berbeda. Untuk masuk kedalam aplikasi konversi matakuliah, *user* perlu melakukan *login* dengan cara memasukkan *username* dan *password*. Gambar 7 adalah tampilan halaman *login user*.



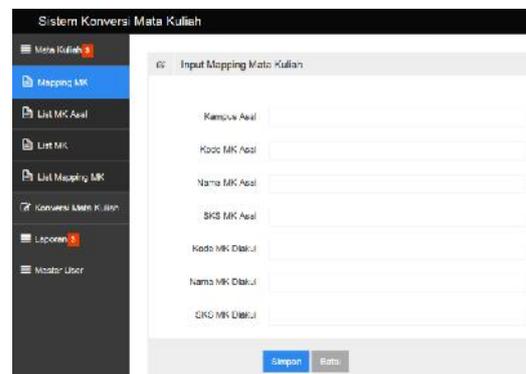
Gambar 7. Halaman Proses Login

Setelah *user* sukses dalam proses *login* dengan cara mengisikan *username* dan *password*, maka akan tampil halaman utama dari aplikasi konversi matakuliah. Gambar 8 adalah tampilan halaman utama aplikasi konversi matakuliah.

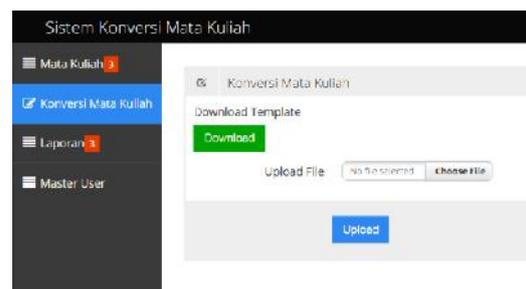


Gambar 8. Halaman Utama Aplikasi

Pada halaman utama aplikasi terdiri dari menu matakuliah, menu konversi matakuliah, menu laporan dan menu master user. Menu matakuliah terdiri dari beberapa submenu yaitu mapping matakuliah, list matakuliah asal, list matakuliah dan list mapping matakuliah.



Gambar 9. Tampilan Submenu Matakuliah



Gambar 10. Tampilan Menu Konversi Matakuliah

Setelah admin mengupload file excel yang berisi transkrip nilai mahasiswa lanjutan dari Diploma Tiga (D3), maka hasil konversi dapat dilihat pada menu laporan. Gambar 11 adalah tampilan hasil konversi. File tersebut dapat diexport kembali dalam format excel.

ID	KODE	NAMA	STATUS	WAKTU
1	001	REKAM MEDIS	2	1
2	002	REKAM MEDIS	1	1
3	003	REKAM MEDIS	1	1
4	004	REKAM MEDIS	1	1
5	005	REKAM MEDIS	1	1
6	006	REKAM MEDIS	1	1
7	007	REKAM MEDIS	1	1
8	008	REKAM MEDIS	1	1
9	009	REKAM MEDIS	1	1
10	010	REKAM MEDIS	1	1

Gambar 11. Tampilan Hasil Konversi

#### 4.4. Tahap Pengujian Aplikasi

Pengujian dilakukan dengan pendekatan *Black Box Testing*, yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian yang dilakukan oleh pengembang perangkat lunak untuk mendapatkan set kondisimasukkan yang sepenuhnya akan melaksanakan semua persyaratan fungsional suatu program.

Pengujian dengan *Black Box Testing* yaitu pengujian fungsional tanpa melihat alur eksekusi program, namun cukup dengan memperhatikan apakah setiap fungsi sudah berjalan dengan baik

Fungsi-fungsi yang diuji adalah proses login, tombol-tombol menu, penginputan matakuliah, konversi matakuliah pada menu upload transkrip nilai, menu hasil konversi dan tombol ekspor data hasil konversi ke format excel.

Dari hasil pengujian dengan menggunakan *Black Box Testing* dapat disimpulkan bahwa aplikasi konversi matakuliah yang dikembangkan dapat berfungsi sesuai dengan yang diharapkan serta aplikasi yang dikembangkan telah cukup memenuhi tujuan awal dari penelitian ini.

#### V. KESIMPULAN

Berdasarkan dari analisis rancangan sistem diatas dapat disimpulkan sebagai berikut:

1. *Model View Controll* dapat diterapkan pada *framework codeigniter* dan dengan arsitektur *MVC* mempermudah dalam perbaikan

perubahan *source code* tanpa merubah keseluruhan *logic* pemrograman.

2. Dari hasil pengujian menggunakan pengujian *Black Box Testing* bahwa aplikasi konversi matakuliah dapat berfungsi dengan baik dan memenuhi tujuan awal pengembangan aplikasi konversi matakuliah ini.

#### DAFTAR PUSTAKA

Arief, & Rudianto, M. (2011). *Pemrograman Web Dinamis Menggunakan Php dan Mysql*. Yogyakarta: Andi.

Badan Pengembangan dan Pembinaan Bahasa, Kemdikbud. (2012-2018). *About Us: kbbs*, 2.3. (E. Setiawan, Penyunting) Dipetik Agustus 1, 2018, dari kbbs Web site: <http://www.kbbs.web.id/>

Fowler, M. (2005). *Panduan Singkat Bahasa Pemodelan Objek Standar* (3th ed.). Yogyakarta: Andi.

Kementerian Riset, Teknologi Dan Pendidikan Tinggi. (2013-2018). *About Us: forlap.ristekdikti*. Dipetik Agustus 1, 2018, dari forlap.ristekdikti Web site: <https://forlap.ristekdikti.go.id/perguruantinggi/homegraphpt>

Nugroho, A. (2015). *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Bandung: Informatika.

Rosa, A. S., & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* (2nd ed.). Bandung: Informatika.

Rosa, A. S., & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek* (2nd ed.). Bandung: Informatika.

Sidik, & Betha. (2012). *Framework CodeIgniter*. Bandung: Informatika.

Sofwan, A. (2016). *About Us: Ilmu Komputer*. Dipetik Agustus 1, 2018, dari Ilmu Komputer Web site: <http://ilmukomputer.org/2010/5/02/belajar-php-dengan-framework-code-igniter>

Zakir, A. (2016). Rancang Bangun Responsive Web Layout dengan Menggunakan Bootstrap Framework. *Informatika dan Teknologi Jaringan*, 7.