

«SKRIPSI/TUGAS AKHIR»

«JUDUL BAHASA INDONESIA»



«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

PROGRAM STUDI «MATEMATIKA/FISIKA/TEKNIK INFORMATIKA»
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
«tahun»

«FINAL PROJECT/UNDERGRADUATE THESIS»

«JUDUL BAHASA INGGRIS»



«Nama Lengkap»

NPM: «10 digit NPM UNPAR.»

DEPARTMENT OF «MATHEMATICS/PHYSICS/INFORMATICS»
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
«tahun»

LEMBAR PENGESAHAN

«JUDUL BAHASA INDONESIA»

«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

«pembimbing utama/1»

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

«Ketua Program Studi»

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa «skripsi/tugas akhir» dengan judul:

«JUDUL BAHASA INDONESIA»

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» «tahun»

Meterai Rp. 6000

«Nama Lengkap»
NPM: «10 digit NPM UNPAR»

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 BlueTape	5
2.2 CodeIgniter	5
2.2.1 <i>System Requirements</i> Server Untuk Menjalankan CodeIgniter	6
2.2.2 Cara Instalasi CodeIgniter	6
2.2.3 Cross-site Request Forgery (CSRF)	6
2.2.4 Model-View-Controller	7
2.2.5 Controller	7
2.2.6 View	8
2.2.7 Model	9
2.2.8 Kelas Migrasi	10
2.2.9 Flow Chart Codeigniter	11
2.3 Cross Site Request Forgery	12
2.4 Zurb Foundation	13
2.4.1 Top Bar	13
2.4.2 Reveal Modal	13
2.4.3 Variabel Sass	14
2.4.4 Scrolling Table	14
2.5 Google OAuth 2.0	16
2.6 PHPExcel	17
3 ANALISIS	21
3.1 Analisis Kebutuhan Data	21
3.1.1 Studi Pustaka	21
3.2 BlueTape	21
3.2.1 Modul Saat Ini	21
3.2.2 Pengguna Aplikasi	24

3.3	Diagram Use Case	27
4	PERANCANGAN DAN IMPLEMENTASI	31
4.1	Perancangan Basis Data	31
4.1.1	Model Relasional	31
4.1.2	Perancangan Tabel	32
4.1.3	Perancangan Rinci	32
4.2	Perancangan Antarmuka	35
4.2.1	Perancangan Antarmuka Entri Jadwal Dosen	35
4.2.2	Perancangan Antarmuka Edit Jadwal Dosen	36
4.2.3	Perancangan Antarmuka Lihat Jadwal Dosen	37
	DAFTAR REFERENSI	39
	A KODE PROGRAM	41
	B HASIL EKSPERIMEN	43

DAFTAR GAMBAR

1.1	Template jadwal dosen	1
2.1	Flow Chart Codeigniter	11
2.2	Skenario OAuth menggunakan token akses	16
2.3	Arsitektur PHPExcels	17
3.1	Tampilan Cetak Transkrip	22
3.2	Tampilan Manajemen Transkrip	22
3.3	Tampilan Menu Perubahan Kuliah	23
3.4	Tampilan Menu Manajemen Perubahan Kuliah	24
3.5	Diagram Use Case	27
4.1	Model Relasional	31
4.2	Perancangan Antarmuka Entri Jadwal Dosen	36
4.3	Perancangan Antarmuka Edit Jadwal Dosen	37
4.4	Perancangan Antarmuka Lihat Jadwal Dosen	38
B.1	Hasil 1	43
B.2	Hasil 2	43
B.3	Hasil 3	43
B.4	Hasil 4	43

DAFTAR TABEL

2.1	Daftar Preferensi di Kelas Migrasi	10
2.2	Daftar Variabel Sass Untuk <i>Reveal Modal</i>	14
2.3	Daftar Variabel Sass Untuk Kelas Tabel	15
4.1	Perancangan Tabel Jadwal	32

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Aplikasi *Blue Tape* adalah aplikasi sederhana yang memiliki tujuan utama untuk mengubah berbagai pekerjaan *paper-based* di FTIS UNPAR menjadi *paperless*. Selain itu aplikasi ini memiliki beberapa kegunaan lainnya seperti mengautentikasi mahasiswa dan staf UNPAR via OAuth 2.0 ke Google (layanan OAuth ke Google ini juga dapat digunakan untuk menentukan hak akses yang bisa dilihat dari email pengguna) dan *Pilot Project* untuk permohonan transkrip ke Tata Usaha . Aplikasi ini merupakan aplikasi berbasis web dengan memanfaatkan *Codeigniter* dan *Zurb Foundation*. Selain itu aplikasi *Blue Tape* ini didesain sebagai *framework* agar dapat ditambahkan layanan-layanan baru. Untuk menambahkan layanan baru sudah tersedia menu khusus, developer cukup menambahkan layanan baru dalam bentuk modul. Untuk saat ini *Blue Tape* baru memiliki layanan untuk *Transcript Request / Manage* yang memiliki fungsi untuk melakukan permohonan serta pencetakan transkrip mahasiswa.

Pada saat ini untuk menginformasikan jadwalnya masing-masing, dosen harus mencetak *hardcopy*-nya dengan template seperti pada gambar di bawah.

JADWAL AKTIVITAS DOSEN					
Dosen:					
	Senin	Selasa	Rabu	Kamis	Jumat
7 - 8					
8 - 9					
9 - 10					
10 - 11					
11 - 12					
12 - 13					
13 - 14					
14 - 15					
15 - 16					
16 - 17					
Keterangan:		Waktu Konsultasi			
		Jika Dijadwalkan			

Gambar 1.1: Template jadwal dosen

Jadwal tersebut akan ditempelkan pada dinding ruangan masing-masing dosen. Sedangkan bila menggunakan *Blue Tape* maka dosen tidak perlu lagi mencetak jadwalnya tersebut karena

mahasiswa dapat melihat jadwal setiap dosen di dalam aplikasi ini. Maka dari itu aplikasi ini membuat pencatatan jadwal dosen menjadi *papeless*.

Pada Skripsi ini akan ditambahkan dua modul yaitu modul entri jadwal untuk dosen informatika dan modul lihat jadwal dosen untuk mahasiswa ke dalam aplikasi Blue Tape. Modul-modul tersebut berfungsi untuk melakukan hal-hal yang berhubungan dengan pembangkitan jadwal dosen. Modul dosen memiliki beberapa fungsi diantaranya: input jadwal mingguan dosen(jadwal dapat berupa jadwal konsultasi, jadwal konsultasi tentatif ataupun jadwal rutin), mencatat *update* terakhir jadwal dosen dan mengeksport jadwal dosen ke XLS. Modul Umum sendiri memiliki fungsi untuk melihat jadwal seluruh dosen dan mengeksport jadwal dosen ke XLS.

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas dalam penelitian ini :

- Bagaimana mengintegrasikan autentikasi mahasiswa maupun staf UNPAR yang mengakses *Blue Tape*?
- Bagaimana cara mencatat, *update* dan melihat jadwal dosen di *Blue Tape*?
- Bagaimana mengeksport jadwal dosen ke XLS sesuai template yang saat ini berlaku?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini :

- Mengimplementasikan autentikasi pengguna yang mengakses *Blue Tape*
- Membuat modul entri jadwal dosen dan modul lihat jadwal dosen yang berfungsi untuk menginput jadwal mingguan, *update* dan melihat jadwal dosen
- Mengimplementasikan kode-kode yang diperlukan untuk memasukkan data-data yang ada di dalam PHP ke dalam *file* XLS.

1.4 Batasan Masalah

Dalam penelitian ini ditetapkan batasan-batasan masalah sebagai berikut:

1.5 Metodologi

Metode penelitian yang akan digunakan dalam skripsi ini adalah:

1. Studi literatur mengenai:

- bahasa pemrograman PHP
- *framework* Codeigniter
- modul *Zurb Foundation*, PHPExcels dan *regular expression*
- Prosedur pembangkitan jadwal dosen

2. Analisis kebutuhan aplikasi dengan mengenali metode pencatatan jadwal dosen saat ini dan mengimplementasikannya ke dalam modul tersebut

3. Membangun modul aplikasi yang sesuai dengan kebutuhan dosen dan mahasiswa dalam pembangkitan jadwal dosen agar aplikasi yang dibuat dapat membantu kedua pihak dalam mengakses informasi-informasi yang berkaitan dengan jadwal dosen . Pembuatan modul aplikasi ini dibagi menjadi empat tahap :

- Analisis kebutuhan modul
- Perancangan modul
- Implementasi
- Pengujian modul

1.6 Sistematika Pembahasan

Untuk penulisan skripsi ini akan dibagi dalam enam bagian sebagai berikut :

Bab 1 Pendahuluan berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.

Bab 2 Landasan Teori berisi dasar-dasar teori yang akan digunakan dalam pembuatan aplikasi pembangkit jadwal dosen. Dasar-dasar teori yang akan digunakan diantaranya adalah bahasa pemrograman PHP, framework Codeigniter, Zurb Foundation dan PHPExcel.

Bab 3 Analisis berisi analisis kebutuhan data, analisis sistem yang sudah ada sekarang dan analisis sistem usulan

Bab 4 Perancangan

Bab 5 Implementasi dan Pengujian

Bab 6 Kesimpulan dan Saran

BAB 2

LANDASAN TEORI

2.1 BlueTape

Aplikasi *Blue Tape* adalah perangkat lunak *open source* sederhana yang memiliki tujuan utama untuk mengubah berbagai pekerjaan *paper-based* di FTIS UNPAR menjadi *paperless*. Selain itu perangkat lunak ini memiliki beberapa kegunaan lainnya seperti mengautentikasi mahasiswa dan staf UNPAR via OAuth 2.0 ke Google (layanan OAuth ke Google ini juga dapat digunakan untuk menentukan hak akses yang bisa dilihat dari email pengguna) dan *Pilot Project* untuk permohonan transkrip ke Tata Usaha . Aplikasi ini merupakan aplikasi berbasis web dengan memanfaatkan *Codeigniter* dan *Zurb Foundation*.

Perangkat lunak *Blue Tape* ini didesain sebagai *framework* yang terdiri dari beberapa layanan yang dipisahkan ke dalam modul-modul. Pemisahan layanan ke dalam modul-modul dibuat dengan tujuan agar pemeliharaan perangkat lunak lebih mudah dan juga mempermudah cara untuk menambahkan layanan baru ke dalam BlueTape. Sudah ada layanan yang aktif di BlueTape saat ini yaitu *Transcript Request / Manage* yang memiliki fungsi untuk melakukan permohonan serta pencetakan transkrip mahasiswa.

2.2 CodeIgniter

CodeIgniter adalah *framework* pengembangan aplikasi untuk *developer* yang membangun situs web menggunakan PHP. Tujuannya adalah untuk memungkinkan Anda mengembangkan proyek lebih cepat, daripada bila *developer* menulis kode dari awal, dengan menyediakan banyak kumpulan *library* untuk tugas-tugas yang sering dibutuhkan dan juga menyediakan tampilan sederhana serta struktur logika untuk mengakses *library-library* tersebut. CodeIgniter memungkinkan *developer* untuk fokus secara kreatif pada proyek *developer* dengan cara meminimalkan jumlah kode yang dibutuhkan untuk setiap tugas yang diberikan. [1]

CodeIgniter dirancang untuk memenuhi kebutuhan :

- *Framework* dengan tapak keberadaan yang kecil
- performa yang baik
- kompatibilitas akun *hosting* yang luas yang dapat berjalan di berbagai versi dan konfigurasi PHP
- *Framework* yang hampir tidak membutuhkan konfigurasi
- *Framework* yang tidak membutuhkan *command line*
- *Framework* yang tidak mengikuti aturan pengkodean yang ketat
- membutuhkan solusi yang sederhana
- dokumentasi yang menyeluruh

2.2.1 *System Requirements* Server Untuk Menjalankan CodeIgniter

Server disarankan sudah menggunakan PHP versi 5.6 atau versi-versi setelahnya. CodeIgniter dapat berjalan pada PHP versi lama, namun ada kemungkinan muncul masalah-masalah yang berkaitan dengan performa dan keamanan.

Database yang didukung oleh CodeIgniter adalah sebagai berikut :

- MySQL (5.1+) melalui `mysql`, `mysqli` dan `pdo` drivers
- Oracle melalui `oci8` dan `pdo` drivers
- PostgreSQL melalui `postgre` dan `pdo` drivers
- MS SQL melalui `mssql`, `sqlsrv` (versi 2005 dan setelahnya) dan `pdo` drivers
- SQLite melalui `sqlite` (versi 2), `sqlite3` (versi 3) dan `pdo` drivers
- CUBRID melalui `cubrid` dan `pdo` drivers
- Interbase/Firebird melalui `ibase` dan `pdo` drivers
- ODBC melalui `odbc` dan `pdo` drivers

2.2.2 Cara Instalasi CodeIgniter

Instalasi CodeIgniter dilakukan dalam 4 langkah :

- *Unzip package* CodeIgniter tersebut
- Unggah folder CodeIgniter dan data-datanya ke dalam server. Pada umumnya *index.php* berada pada *root* server.
- Buka *application/config/config.php* menggunakan program pengolah teks dan tentukan URL-nya. Jika akan digunakan enkripsi, tentukan kunci enkripsinya.
- Jika *database* akan digunakan, buka *application/config/database.php* menggunakan program pengolah teks dan atur *database* anda.

2.2.3 Cross-site Request Forgery (CSRF)

Perlindungan terhadap CSRF dapat dilakukan dengan cara mengubah nilai variabel berikut menjadi TRUE pada file **application/config/config.php**

```
$config['csrf_protection'] = TRUE;
```

Jika *form helper* digunakan di view aplikasi, maka field csrf akan secara otomatis dimasukkan ke dalam form tersebut. Kode dibawah ini akan secara otomatis dimasukkan oleh CodeIgniter ketika `form_open()` dipanggil.

```
<input type="hidden" name="<?=$csrf['name'];?>" value="<?=$csrf['hash  
↪ '];?>" />
```

Token dapat diatur agar selalu meregenerasi setiap terjadi *submission* atau juga dapat diatur agar selalu tetap sama sepanjang masa hidup *cookie* CSRF-nya. Regenerasi token akan membuat keamanan sistem lebih kuat, namun dapat menyebabkan masalah navigasi web (seperti navigasi *back/forward* halaman ke halaman, membuka tab baru, dan lain-lain). Regenerasi token dapat diatur dengan mengubah nilai variabel :

```
$config['csrf_regenerate'] = TRUE;
```

2.2.4 Model-View-Controller

CodeIgniter didasari pola pengembangan *Model-View-Controller* atau MVC. MVC memisahkan logika aplikasi dengan tampilannya.

- **Model** merepresentasikan struktur data. Pada umumnya kelas-kelas model menampung fungsi-fungsi untuk mengambil, memperbarui atau memasukan data ke dalam basis data.
- **View** menampilkan informasi ke pengguna.
- **Controller** berfungsi sebagai perantara antara model dan view.

2.2.5 Controller

Sebuah *controller* adalah kelas yang dinamakan demikian agar dapat diasosiasikan dengan URI. sebagai contoh URI "example.com/index.php/blog/" , CodeIgniter akan mencari *controller* bernama Blog.php dan menjalankannya. Nama *controller* harus diawali dengan huruf kapital. Selain itu *controller* juga harus *extend* kelas "CI_Controller".

Contoh yang benar :

```
<?php
class Blog extends CI_Controller {

}
```

Contoh yang salah :

```
<?php
class blog extends CI_Controller {

}
```

2.2.5.1 Method

Untuk menjalankan suatu method, maka developer perlu menuliskannya pada segmen kedua URI. Contoh "example.com/index.php/blog/comments" maka akan dijalankan method comments() pada controller blog.php. Method yang akan dijalankan bila bagian kedua URI kosong adalah method index(). Jika URI mengandung lebih dari dua segment, segment-segment tersebut akan dimasukan ke dalam method sebagai parameter.

2.2.5.2 Default Controller

CodeIgniter dapat diperintahkan untuk menjalankan *default controller* jika tidak terdapat URI, pada umumnya terjadi ketika hanya terdapat permintaan menggunakan URL dasar *website*. Penentuan *default controller* terdapat pada file "application/config/routes.php" dan set variabel . Nama *controller* tersebut adalah 'blog', maka ketika index.php dijalankan tanpa menspesifikasikan URI akan dijalankan *controller* 'blog'.

2.2.5.3 Memproses Output

CodeIgniter memiliki kelas *output* yang mengurus pengiriman data ke *web browser* secara otomatis. Untuk kasus-kasus saat pengguna ingin mengubah cara pengiriman data tersebut, CodeIgniter menyediakan caranya dengan menambahkan method bernama "__output()" ke *controller* terkait. Jika controler memiliki method bernama "__output()" maka controller tersebut akan selalu dipanggil oleh kelas "output".

Contoh penggunaan method "__output()" :

```
public function _output(\$output)
{
    echo $output;
}
```

2.2.5.4 Private Method

Method-method dengan tipe *private* tidak dapat diakses oleh publik. Method ini hanya dapat diakses oleh method lain dalam *controller* yang sama. Selain itu method ini juga tidak akan dapat diakses melalui URL.

Contoh penulisan *private method*:

```
private function _utility()
{
    // kode program
}
```

Method di atas tidak dapat diakses dengan cara pemanggilan method pada umumnya seperti :

```
example.com/index.php/blog/_utility/
```

2.2.5.5 Mengorganisir Controller-controller ke Dalam Sub Direktori

Di CodeIgniter pengguna dapat mengorganisir *controller-controller* ke dalam sub direktori. Untuk melakukannya cukup dengan membuat sub direktori di dalam direktori *application/controllers/* dan simpan kelas-kelas *controller* ke dalamnya. Ketika menggunakan fitur ini, pengguna harus menspesifikasikan foldernya ke dalam URL.

Contoh ada controller yang berlokasi di:

```
application/controllers/products/Shoes.php
```

Untuk memanggil controller tersebut URI harus menuliskan nama sub direktorinya :

```
example.com/index.php/products/shoes/show/123
```

2.2.6 View

View adalah sebuah halaman web, bagian-bagian halaman (seperti *header*, *footer*, *sidebar*, dan-lain-lain) atau bagian dari *view* lainnya. *View* tidak pernah dipanggil secara langsung, melainkan harus melalui controller karena dalam framework MVC controller berfungsi sebagai pengatur.

Controller dapat memanggil view menggunakan potongan kode seperti di bawah ini :

```
$this->load->view('name');
```

Jika 1 halaman terdiri dari beberapa view, CodeIgniter juga menyediakan cara untuk memanggil banyak view sekaligus:

```
<?php

class Page extends CI_Controller {

    public function index()
    {
        $data['page_title'] = 'Your title';
```

```
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('content', $data);
        $this->load->view('footer');
    }

}
```

Bila ada data lain yang akan dikirimkan ke view dari controller, gunakan potongan kode seperti di bawah ini :

```
$this->load->view('name', $data);
```

2.2.7 Model

Model adalah kelas PHP yang dirancang untuk bekerja dengan informasi-informasi di *database*. Sebagai contoh, misalkan CodeIgniter digunakan untuk mengatur sebuah blog maka model mengandung fungsi-fungsi seperti *select*, *insert*, *update* data-data blog.

2.2.7.1 Anatomi dari Model

Model disimpan di direktori **application/models**. Model dapat disimpan dalam sub-direktori jika dibutuhkan. Bentuk dasar kode pada kelas model adalah seperti di bawah ini:

```
class Model_name extends CI_Model {

    public function __construct()
    {
        parent::__construct();
        //constructor code
    }

}
```

Model_name adalah nama dari kelasnya. Huruf pertama dari nama kelas harus huruf kapital dengan huruf-huruf setelahnya ditulis menggunakan huruf kecil. Kelas model harus dipastikan meng-*extend* kelas Model dasar. Nama file harus sama dengan nama kelasnya.

2.2.7.2 Load Model

Pada umumnya model akan di-*load* di dalam method-method *controller*. Untuk *load* model diperlukan method berikut:

```
$this->model->model('model_name');
```

Jika model terdapat pada sub-direktori, perlu dimasukan *path*-nya ke dalam direktori 'models'. Misalnya model terdapat di *application/models/blog/Queries.php* maka untuk *load* model gunakan :

```
$this->load->model('blog/queries');
```

Bila ada sebuah model yang akan digunakan secara global pada aplikasi pengguna, CodeIgniter bisa diperintahkan untuk *auto-load* model tersebut ketika instalasi sistem. Untuk melakukan hal ini, tambahkan model ke dalam *autoload array* yang berlokasi di "*application/config/autoload.php*".

2.2.7.3 Menghubungkan Model Dengan *Database*

Ketika Model dipanggil, model tidak akan terhubung ke database secara otomatis. Ada dua cara untuk menghubungkan model dengan *database* :

1. Menghubungkan secara otomatis ketika instalasi sistem dilakukan menggunakan fitur auto-load seperti disebutkan di bagian *Load Model* di atas.
2. Menghubungkan *database* secara manual. Untuk menghubungkan *database* dengan model, pengguna perlu untuk memanggil *database*-nya di *controller* atau di *model* itu sendiri dengan menuliskan potongan kode:

```
$this->load->database();
```

Parameter-parameter yang tersedia untuk potongan kode di atas adalah:

- Nama dari database yang dituju berupa array atau *Data Source Name*(DSN) string.
- *Boolean* untuk menentukan id koneksi perlu dikembalikan ke model atau tidak.
- *Boolean* untuk menentukan perlunya dibuat *Query Builder* atau tidak. Jika tidak ditentukan, nilai dari parameter ini adalah TRUE.

2.2.8 Kelas Migrasi

Kelas ini digunakan untuk memudahkan pengubahan *database* secara terstruktur dan terorganisir dengan baik. Dengan cara ini maka pengguna tidak perlu memberi tahu pengguna lainnya yang sedang mengembangkan aplikasi yang sama untuk melakukan perubahan *database* secara manual, pengguna lain cukup menjalankan kelas migrasi ini. Kelas ini akan secara otomatis membuat tabel bernama "*migration*" di *database* yang mencatat migrasi-migrasi mana saja yang sudah dijalankan, yang perlu dilakukan secara manual oleh pengguna adalah untuk memperbarui *file* aplikasinya dan memanggil *\$this->migration->current()* untuk menentukan migrasi mana yang perlu dijalankan. Untuk mengatur preferensi dari kelas migrasi terdapat di **application/config/migration.php**.

Preferensi	Default	Opsi	Deskripsi
migration_enabled	FALSE	TRUE / FALSE	Menentukan migrasi dijalankan atau tidak.
migration_path	APPPATH.migrations/'	None	Alamat ke folder migrasi.
migration_version	0	None	Versi migrasi yang digunakan.
migration_table	migrations	None	Nama tabel migrasi di <i>database</i> .
migration_auto_latest	FALSE	TRUE / FALSE	Menentukan dijalanannya migrasi otomatis atau tidak.
migration_type	'timestamp'	'timestamp' / 'sequential'	Tipe penomoran kelas migrasi yang digunakan.

Tabel 2.1: Daftar Preferensi di Kelas Migrasi

2.2.8.1 Nama Kelas Migrasi

Setiap migrasi dijalankan sesuai urutan numerik baik dengan urutan menaik atau menurun tergantung dari metode yang digunakan. Ada dua cara penomoran yang bisa dilakukan yaitu :

- **Sekuensial**: setiap migrasi dinomori secara sekuensial, dimulai dengan angka **001**. Setiap nomor harus terdiri dari 3 digit dan tidak ada jeda di angka tersebut. Cara penomoran ini adalah satu-satunya cara yang digunakan sebelum versi CodeIgniter 3.0 .
- **Timestamp**: setiap migrasi menggunakan *timestamp* ketika kelas migrasinya dibuat. Format penomrannya adalah YYYYMMDDHHIISS.
 - YYYY : tahun terdiri dari 4 digit
 - MM : bulan terdiri dari 2 digit
 - DD : hari terdiri dari 2 digit
 - HH : jam terdiri dari 2 digit
 - II : menit terdiri dari 2 digit
 - SS : detik terdiri dari 2 digit

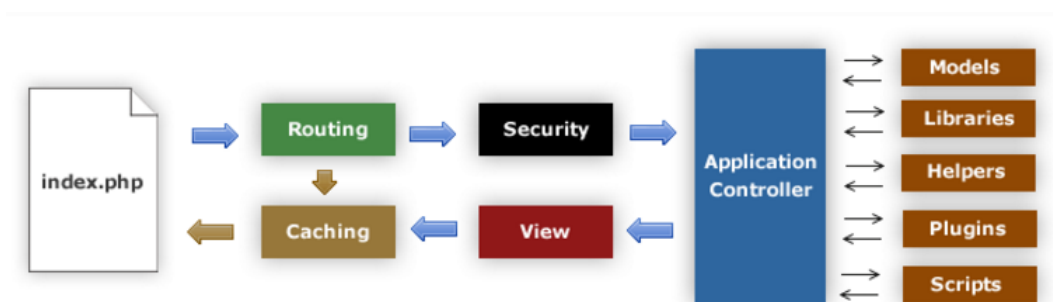
Cara ini digunakan untuk menghindari konflik penomoran ketika bekerja dalam tim. Cara penomoran ini merupakan cara yang lebih banyak digunakan sejak versi CodeIgniter 3.0 dan setelahnya.

Urutan penomran dapat diatur menggunakan `$config['migration_type']` di file `application/config/migration.php` . Apapun cara yang dipilih, pisahkan nama file dan penomoran kelasnya menggunakan tanda garis bawah. Sebagai contoh:

- 001_add_blog.php (penomoran sekuensial)
- 20121031100537_add_blog.php (penomoran dengan *timestamp*)

2.2.9 Flow Chart Codeignter

Gambar dibawah mengilustrasikan aliran data dalam sistem : ¹



Gambar 2.1: Flow Chart Codeignter

1. `index.php` berfungsi sebagai controller depan. Menginisialisasi sumber daya yang dibutuhkan untuk menjalankan CodeIgniter
2. Router memeriksa permintaan HTTP untuk menentukan apa yang akan dilakukan pada permintaan tersebut.

¹<https://www.codeigniter.com/userguide3/overview/appflow.html>

3. Jika ada cache file, maka akan dikirim langsung ke browser. Melewati cara eksekusi sistem yang normal.
4. Security. Sebelum controller aplikasi dimuat, permintaan HTTP dan data-data pengguna yang telah diserahkan disaring untuk keamanan.
5. Controller memuat model, pustaka inti (*core libraries*), pembantu dan sumber daya lain yang dibutuhkan untuk memproses permintaan khusus.
6. Kemudian tampilan akhir dibuat dan dikirim ke web browser untuk dilihat. Jika caching diaktifkan, maka tampilan dimasukkan ke dalam cache terlebih dahulu sehingga pada permintaan selanjutnya tampilan tersebut dapat diakses lebih cepat.

2.3 Cross Site Request Forgery

"Cross Site Request Forgery" atau "Session Riding" adalah teknik penyerangan yang mengeksploitasi otentikasi implisit. Penyerangan ini dilakukan dengan cara menyebabkan browser korban membuat *http requests* tersembunyi ke sumber daya - sumber daya terlarang. Pada kasus ketika *request* untuk sumber daya tersebut berhasil, akan menyebabkan aplikasi web terkait untuk melakukan tindakan-tindakan penyerangan tersebut secara terus menerus. Penyerangan ini bertujuan untuk berbagai hal berikut seperti mengubah *field-field* di dalam *database*, mengirim email atau mengubah bagian-bagian aplikasi. Semua aksi tersebut dilakukan menggunakan token otentikasi milik korban.

Pada umumnya *web browser* memiliki kebijakan untuk memperbolehkan *website-website* untuk mengirim *HTTP request* ke alamat jaringan manapun. Akibat dari kebijakan ini, maka penyerang dapat mengendalikan konten-konten yang dianggap oleh *browser* bukan di bawah kendali pengguna. Konten yang dapat dikendalikan oleh penyerang diantaranya :

- Konektivitas Jaringan. Misalkan korban menggunakan *firewall*, maka penyerang dapat mempengaruhi browser pada mesin korban untuk mengirim *network request* ke mesin-mesin lain yang menggunakan firewall juga yang tidak dapat diakses secara langsung melalui mesin milik penyerang.
- Membaca Status Browser. *Request-request* yang dikirimkan ke jaringan melalui *browser* pada umumnya berisi status browser seperti cookies, sertifikat klien atau data otentikasi sederhana.
- Menulis Status Browser. Penyerang membuat *browser* untuk menulis *network request*. *Browser* juga akan bereaksi pada respon dari website yang dituju. Hal ini mengakibatkan *browser* memodifikasi beberapa bagian status browser.

Serangan CSRF tidak terbatas pada satu *request* palsu saja. Alur kerja yang membutuhkan serangkaian *request* http juga rentan terhadap serangan ini selama kondisi-kondisi yang diperlukan terpenuhi. Misalnya bila setiap konten dan *identifier* dalam setiap langkah pada alur kerja *web form* diketahui dan alur kerja dari *website* tersebut tidak memiliki mekanisme lain untuk melacak cara kerja langkah per-langkah *web form*-nya, melainkan hanya menggunakan *identifier* dari *session* saja. Jika kondisi tersebut dipenuhi, maka penyerang dapat membuat serangkaian *iframe-iframe* tersembunyi yang mengandung *web form* berbahaya. *Form-form* tersebut lalu dikirimkan secara sekuensial melalui JavaScript menggunakan *event onload* dari *iframe* untuk membuat server yang menerimanya mengira bahwa *form-form* tersebut diisi oleh user dengan benar.

2.4 Zurb Foundation

Foundation adalah kumpulan pola desain HTML, CSS dan Javascript yang dapat digunakan untuk membuat website. Hal tersebut untuk membantu *developer* agar tidak perlu menulis kode yang sama berulang kali. Selain membantu menghemat waktu, Foundation juga membantu *developer* untuk menulis kode dengan lebih baik. Foundation dapat bekerja pada berbagai media seperti komputer desktop, laptop, tablet, dan telepon genggam.^[2]

Komponen-komponen dalam Foundation sendiri ada beberapa macam diantaranya sebagai berikut:²

- Grid untuk mempermudah pembagian halaman
- Desain tombol yang bermacam-macam. Desain tombol ini dapat diubah-ubah dengan cara menambahkan kelas.
- Navigasi untuk mempermudah pengunjung aplikasi dalam menggunakan aplikasinya.
- Plugins JavaScript untuk mempermudah *developer* dalam membuat tampilan aplikasinya.

2.4.1 Top Bar

Top bar adalah *wrapper* sederhana untuk komponen-komponen menu website. Top bar dapat memiliki 2 bagian yaitu bagian kiri (**.top-bar-left**) dan bagian kanan (**.top-bar-right**). Pada layar yang kecil bagian top bar ini bisa menjadi di atas atau di bawah sisi lainnya.

2.4.2 Reveal Modal

Pada dasarnya modal hanyalah sebuah kontainer kosong, jadi pengguna dapat memasukan konten apapun ke dalamnya. Untuk membuat modal, buat kelas **.reveal** dan atribut **data-reveal** lalu beri id unik ke kontainernya tersebut seperti contoh di bawah ini :

```
<div class="reveal" id="exampleModal1" data-reveal>
    ...
</div>
```

Untuk membuka modal, tambahkan atribut **data-open** ke elemen apapun. Nilai dari **data-open** adalah id dari modalnya.

```
<p><button class="button" data-open="exampleModal1">Click me for a modal</button>
↪ ></p>
```

Jika tidak diberi aturan tambahan, pada dasarnya modal akan ditutup jika pengguna menekan area di luar dari modal atau ketika tombol ESC ditekan. Untuk memberi tombol "*close*" di menu modal tambahkan atribut **data-close** ke elemen yang berisi pemicu modal. Contoh dapat dilihat pada potongan kode di bawah.

```
<button class="close-button" data-close aria-label="Close modal" type="button">
    <span aria-hidden="true">&times;</span>
</button>
```

²<http://foundation.zurb.com/sites/docs/v/5.5.3/>

2.4.3 Variabel Sass

Berikut adalah variabel-variabel yang dapat digunakan untuk mengkostumisasi modal :

Nama	Tipe	Nilai <i>Default</i>	Deskripsi
\$reveal-background	Color	\$white	Warna <i>default</i> latar belakang modal
\$reveal-width	Number	600px	Lebar <i>default</i> modal, tanpa memakai kelas apapun
\$reveal-max-width	Number	\$global-width	Lebar maximum modal
\$reveal-padding	Number	\$global-padding	<i>Default</i> padding di dalam modal
\$reveal-border	Number	\$global-radius	Nilai <i>default</i> radius untuk sebuah modal
\$reveal-zindex	Number	1005	nilai z-index untuk modal.
\$reveal-overlay-background	Color	rgba(\$black,0,45)	Warna latar belakang penutup modal

Tabel 2.2: Daftar Variabel Sass Untuk *Reveal Modal*

2.4.4 Scrolling Table

Scrolling table digunakan bila banyak data yang ada pada tabel. Dengan menggunakan ini maka isi tabel dapat digeser secara horizontal. Untuk menggunakan tabel jenis ini, deklarasikan kelas **table-scroll** seperti contoh di bawah:

```
<div class="table-scroll">
  <table></table>
</div>
```

2.4.4.1 Variabel Sass

Berikut adalah daftar variabel Sass yang dapat dikustomisasi untuk mengubah tampilan komponen-komponen dari tabel

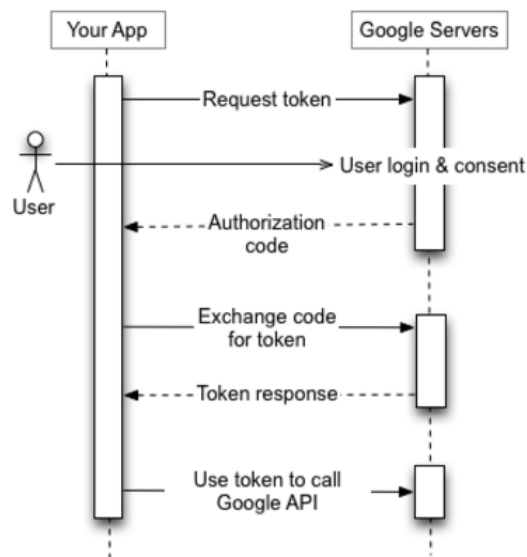
Nama	Tipe	Nilai <i>Default</i>	Deskripsi
<code>\$table-background</code>	Color	<code>\$white</code>	Warna <i>default</i> tabel
<code>\$table-color-scale</code>	Number	5%	Skala warna gelap pada tabel bergaris
<code>\$table-border</code>	List	1px solid smart-scale(<code>\$table-background</code> , <code>\$table-color-scale</code>)	Tipe batas tabel yang digunakan
<code>\$table-padding</code>	Number	rem-calc(8 10 10)	<i>Default</i> padding di tabel
<code>\$table-hover-scale</code>	Number	2%	Skala warna gelap baris tabel ketika ditunjuk kursor
<code>\$table-row-hover</code>	List	darken(<code>\$table-background</code> , <code>\$table-hover-scale</code>)	Warna baris pada tabel ketika ditunjuk oleh kursor.
<code>\$table-row-stripe-hover</code>	List	darken(<code>\$table-background</code> , <code>\$table-color-scale</code> + <code>\$table-hover-scale</code>)	Warna baris gelap pada tabel bergaris ketika ditunjuk kursor
<code>\$table-is-striped</code>	Boolean	TRUE	Jika bernilai TRUE maka tabel menggunakan tipe tabel bergaris
<code>\$table-striped-background</code>	Color	smart-scale(<code>\$table-background</code> , <code>\$table-color-scale</code>)	Warna baris gelap pada tabel bergaris
<code>\$table-stripe</code>	Keyword	even	Nilai untuk memunculkan baris gelap pada tabel, kecuali pada bagian <i>header</i> . Jika bernilai <i>even</i> , maka baris genap akan memiliki warna latar. Jika bernilai <i>odd</i> maka baris ganjil yang akan memiliki warna latar
<code>\$table-head-background</code>	Color	smart-scale(<code>\$table-background</code> , <code>\$table-color-scale</code> / 2)	Warna latar belakang <i>header</i> tabel
<code>\$table-head-row-hover</code>	List	darken(<code>\$table-head-background</code> , <code>\$table-hover-scale</code>)	Warna <i>header</i> ketika ditunjuk kursor
<code>\$table-foot-background</code>	Color	smart-scale(<code>\$table-background</code> , <code>\$table-color-scale</code>)	Warna latar belakang <i>footer</i>
<code>\$table-foot-row-hover</code>	List	darken(<code>\$table-foot-background</code> , <code>\$table-hover-scale</code>)	Warna latar belakang <i>footer</i> ketika ditunjuk kursor
<code>\$table-head-font-color</code>	Color	<code>\$body-font-color</code>	Warna teks yang berada di <i>header</i>
<code>\$table-foot-font-color</code>	Color	<code>\$body-font-color</code>	Warna teks yang berada di <i>footer</i>
<code>\$show-header-for-stacked</code>	Boolean	false	Nilai untuk menentukan penggunaan header ketika menggunakan <i>stacked table</i>
<code>\$table-stack-breakpoint</code>	Breakpoint	medium	Batas pemicu perubahan jenis tabel dari tipe <i>mobile</i> menjadi <i>desktop</i> atau sebaliknya

Tabel 2.3: Daftar Variabel Sass Untuk Kelas Tabel

2.5 Google OAuth 2.0

Google OAuth 2.0 merupakan salah satu protokol dari Google Sign-in. Google OAuth 2.0 digunakan oleh Google API untuk otorisasi dan autentikasi. Secara garis besar, cara pemakaian Google OAuth 2.0 adalah sebagai berikut : ³

1. Dapatkan OAuth 2.0 credential dari konsol Google API. OAuth 2.0 credential seperti client ID dan client secret yang diketahui oleh Google dan aplikasi pengguna, dapat didapatkan di halaman <https://console.developers.google.com/> .
2. Dapatkan token akses dari Google Authorization Server. Sebelum aplikasi dapat mengakses data pribadi menggunakan Google API, aplikasi tersebut harus mendapat token akses yang memberikan akses ke API. Satu token akses dapat memberikan berbagai macam akses ke banyak API. Variable parameter "*scope*" mengendalikan kumpulan-kumpulan sumber daya dan operasi yang telah diperbolehkan untuk diakses oleh token akses. Selama masa permintaan token akses, aplikasi mengirimkan satu atau lebih nilai ke dalam parameter "*scope*". Ada beberapa cara untuk melakukan permintaan, tergantung dari tipe aplikasi yang sedang dibuat. Sebagai contoh aplikasi JavaScript dapat meminta token akses menggunakan *redirect* dari *browser* yang mengarah ke Google, sementara aplikasi lain yang terinstall di dalam perangkat yang tidak memiliki browser menggunakan *web service* untuk melakukan permintaan. Beberapa permintaan membutuhkan tahap autentikasi yang meminta pengguna untuk masuk ke akun Google mereka. Setelah masuk ke dalam akun, pengguna akan diminta jika mereka bersedia untuk memberikan izin ke aplikasi yang sedang melakukan permintaan tersebut.



Gambar 2.2: Skenario OAuth menggunakan token akses

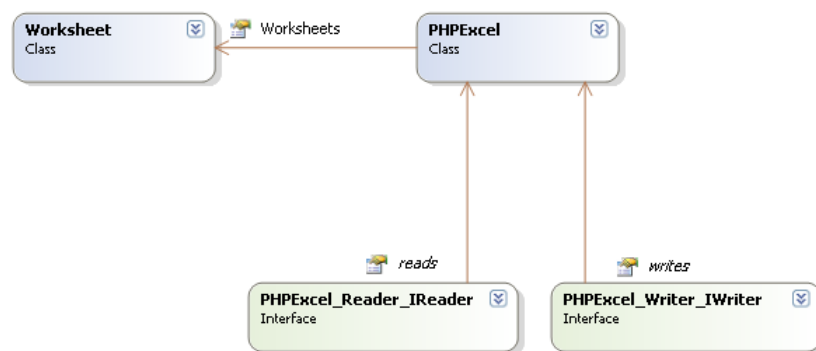
3. Mengirim akses token ke API. Setelah aplikasi mendapatkan akses token, aplikasi tersebut akan mengirim token ke Google API dalam bentuk HTTP *authorization header*. Jika memungkinkan, aplikasi dapat mengirim token-token sebagai parameter *URI query-string*. Pengiriman dalam bentuk parameter URI tidak disarankan karena parameter URI dapat tersimpan dalam *log* yang tidak aman. Token akses hanya berlaku untuk kumpulan operasi dan sumber daya yang dideskripsikan dalam parameter *scope* di permintaan token.

³<https://developers.google.com/identity/protocols/OAuth2>

4. Jika dibutuhkan, token akses dapat di-*refresh* karena token akses memiliki masa berlaku terbatas. Jika aplikasi membutuhkan akses ke Google API lebih dari masa berlaku satu buah token, aplikasi dapat mendapatkan token *refresh*. Token *refresh* memungkinkan aplikasi untuk mendapatkan token akses baru.

2.6 PHPEXcel

PHPExcel adalah suatu proyek yang menyediakan berbagai kelas-kelas untuk pemrograman bahasa PHP yang memungkinkan *developer* untuk menulis dan membaca dari berbagai macam bentuk *spreadsheet* seperti Excel (BIFF) .xls, Excel 2007 (OfficeOpenXML) .xlsx, CSV, Libre/OpenOffice Calc .ods, Gnumeric, PDF, HTML, dan lain-lain. Proyek ini dibangun sesuai standar Microsoft OpenXML dan PHP.[3]



Gambar 2.3: Arsitektur PHPEXcel

Untuk menjalankan PHPEXcel, diperlukan :

- PHP versi 5.2.0 keatas
- PHP extension php_zip diaktifkan
- PHP extension php_xml diaktifkan
- PHP extension php_gd2 diaktifkan

2.6.0.1 Membuat *Spreadsheet*

Untuk membuat *spreadsheet* pengguna perlu menggunakan kelas PHPEXcel. Kelas PHPEXcel ini merupakan bagian inti dari PHPEXcel, kelas ini merepresentasikan *workbook* yang akan dibuat. Pada umumnya ada dua cara untuk membuat *workbook* sebagai berikut :

1. Memuat *workbook* dari file *spreadsheet* yang sudah ada.

Cara paling mudah untuk memuat sebuah *workbook* adalah dengan memerintahkan *PHPExcel IO Factory* untuk mengidentifikasi file *workbook*-nya lalu memuatnya dengan cara memanggil *static method* `load()` dari kelas "*PHPExcel_IOFactory*". Contoh cara pemuatan *workbook* di dalam kode aplikasi adalah sebagai berikut:

```

$inputFileName = './sampleData/example1.xls';

/** Load $inputFileName ke dalam obyek PHPEXcel */
$objPHPExcel = PHPEXcel_IOFactory::load($inputFileName);
  
```

Method `load()` bekerja dengan cara mengidentifikasi terlebih dahulu tipe filenya, lalu menginstansiasi sebuah *loader* untuk tipe file terkait. *Loader* ini lalu digunakan untuk memuat *file workbook*-nya dan menyimpan data-data lalu membentuknya ke dalam obyek PHPEXcel.

Pada awalnya method `load()` akan memuat *loader* yang dibutuhkan sesuai dari ekstensi *file workbook*, namun method ini akan memeriksa secara mendalam file tersebut sebelum memulai pemuatan *workbook*-nya ke dalam obyek PHPExcel. Sebagai contoh bila sebuah file merupakan file CSV namun memiliki ekstensi `.xls`, maka file tersebut akan menolak *loader* Excel5 yang biasanya digunakan untuk file `.xls`. Bila hal tersebut terjadi, maka method `load()` akan terus mencoba memuat *workbook* tersebut menggunakan *loader-loader* lainnya sampai ditemukan *loader* yang sesuai untuk file tersebut. *Loader* yang sudah cocok tersebut kemudian akan digunakan untuk membaca *file*-nya.

Format file *spreadsheet* yang didukung oleh PHPExcel adalah sebagai berikut:

- BIFF (Excel5)
- SpreadsheetML (Excel2003XML)
- OfficeOpenXML (Excel2007)
- Open Document Format (OOCalc)
- Multiplan SYLK
- Gnumeric
- CSV
- HTML

2. Membuat *workbook* baru secara manual.

Untuk membuat *workbook* baru, instansiasi obyek PHPExcel di dalam kode aplikasi.

```
$objPHPExcel = new PHPExcel();
```

Workbook baru yang dibuat akan memiliki satu buah *worksheet*.

2.6.0.2 Mengakses Sel

Untuk memasukan atau mengubah nilai dalam suatu sel berdasarkan koordinatnya dapat dilakukan dengan menggunakan method `setCellValue()`.

Contoh :

```
// Memasukan nilai string ke sel A1
$objPHPExcel->getActiveSheet()->setCellValue('A1','PHPExcel');

// Memasukan nilai numerik ke sel A2
$objPHPExcel->getActiveSheet()->setCellValue('A2',12345.6789);

// Memasukan nilai boolean ke sel A3
$objPHPExcel->getActiveSheet()->setCellValue('A3',TRUE);

// Memasukan suatu formula ke sel A4
$objPHPExcel->getActiveSheet()->setCellValue(
    'A4',
    '=IF(A3, CONCATENATE(A1," ", A2), CONCATENATE(A2," ", A1))'
);
```

Tujuh tipe data yang didukung MS Excel :

- string
- bilangan

- boolean
- null
- formula
- eror
- *inline string*

Ketika method *setCellValue()* atau *setValue()* dipanggil, PHPEXcel akan memakai tipe data yang sesuai untuk tipe data null, boolean, float atau integer yang didapat dari PHP. PHPEXcel juga bisa mengubah string yang dikirimkan dari PHP ke Excel menjadi tipe data yang lebih sesuai. Sebagai contoh string yang terdiri dari angka-angka saja akan diubah menjadi tipe data numerik, atau string yang memiliki awalan tanda sama dengan "=" akan dianggap sebagai formula.

Konversi-konversi tipe data tersebut ditangani oleh sebuah "*value binder*" yang dapat diubah-ubah sesuai keinginan pengguna bila pengguna ingin mengatur cara-cara konversinya. PHPEXcel standar juga menyediakan "*advanced value binder*" yang menangani konversi-konversi yang lebih kompleks seperti mengonversi string menjadi bilangan pecahan seperti "3/4" menjadi nilai numerik (untuk kasus ini menjadi 0,75). Fitur ini berguna ketika memuat data dari csv atau memasukan nilai ke dalam sel dari *database*.

Beberapa format yang ditangani oleh *advanced value binder* adalah sebagai berikut:

- TRUE atau FALSE dikonversi menjadi boolean
- String yang berisi nilai numerik akan diubah menjadi bilangan
- Pecahan akan diubah menjadi bilangan
- Persentase akan diubah menjadi bilangan yang dibagi 100
- Tanggal dan waktu akan diubah menjadi nilai *timestamp* di Excel
- Ketika string mengandung karakter yang memerintahkan pembautan baris baru (n), maka sel akan diatur menjadi menggunakan style "*wrap*"

2.6.0.3 Memasukan Tanggal dan atau Waktu ke Dalam Sel

Nilai tanggal dan waktu disimpan dalam rupa *timestamp* (bilangan desimal biasa) di dalam Excel. Bilangan tersebut kemudian dibungkus oleh bilangan lain yang menentukan format penulisan tanggalnya. Jadi untuk memasukan tanggal ke dalam sel, perlu dihitung *timestamp* yang benar, dan menentukan bilangan pembungkusnya.

```
// Mendapatkan tanggal dan waktu saat ini
$dateTimeNow = time();
$excelDateValue = PHPEXcel_Shared_Date::PHPToExcel( $dateTimeNow );
// Memasukan tanggal dan waktu ke dalam sel A6
$objPHPEXcel->getActiveSheet()->setCellValue(
    'A6',
    $excelDateValue
);
// Mengatur format bilangan pembungkus sehingga timestamp dalam Excel bisa
    ↪ ditampilkan dalam format yang dapat dibaca oleh manusia
$objPHPEXcel->getActiveSheet()->getStyle('A6')
    ->getNumberFormat()
```

```
->setFormatCode(
    PHPExcel_Style_NumberFormat::FORMAT_DATE_DATETIME
);
```

2.6.0.4 Memasukan Data Numerik yang Diawali Angka Nol

Pada umumnya PHPExcel secara otomatis akan mendeteksi tipe dari nilai yang dimasukkan dan mengubahnya menjadi tipe data numerik di Excel. Tipe konversi ini ditangani oleh *value binder*. Karena bilangan tidak memiliki awal angka nol, maka jika ada nilai numerik yang memiliki awalan angka nol (misalnya nomor telpon), maka nilai tersebut akan kehilangan angka-angka nol yang berada di depan.

Untuk mencegah konversi demikian, ada beberapa cara untuk melakukannya diantaranya sebagai berikut:

1. Menentukan secara manual di dalam kode agar tipe datanya tidak dikonversi ke dalam bilangan.

```
// Memasukan nilai numerik ke dalam sel A8, tetapi memerintahkan
    ↳ PHPExcel untuk menyimpan nilai tersebut dalam tipe data String
$objPHPExcel->getActiveSheet()->setCellValueExplicit(
    'A8',
    "01513789642",
    PHPExcel_Cell_DataType::TYPE_STRING
);
```

2. Cara lainnya adalah dengan menentukan format bilangan di dalam worksheet agar menampilkan bilangan dengan awalan nol.

```
// Memasukan nilai numerik ke dalam sel A9
$objPHPExcel->getActiveSheet()->setCellValue('A9', 1513789642);
// Menentukan format sel untuk menampilkan nilai dalam selnya berupa
    ↳ bilangan 11 digit dengan awalan nol
$objPHPExcel->getActiveSheet()->getStyle('A9')
    ->getNumberFormat()
    ->setFormatCode(
        '00000000000'
    );
```

3. Cara kedua juga dapat digunakan untuk memisahkan digit-digit ke dalam grup agar lebih mudah dibaca.

```
// Memasukan nilai numerik ke dalam sel A10
$objPHPExcel->getActiveSheet()->setCellValue('A10', 1513789642);
//Mengatur format sel untuk menampilkan nilai dalam selnya berupa
    ↳ bilangan 11 digit dengan awalan nol
$objPHPExcel->getActiveSheet()->getStyle('A10')
    ->getNumberFormat()
    ->setFormatCode(
        '0000-000-0000'
    );
```

BAB 3

ANALISIS

3.1 Analisis Kebutuhan Data

Pengumpulan data dalam skripsi ini dilakukan dengan cara studi pustaka.

3.1.1 Studi Pustaka

Mempelajari cara pengembangan perangkat lunak menggunakan *Framework* BlueTape yang berbasis CodeIgniter. Selain itu juga mempelajari *library-library* pembantunya diantara lain : PHPExcel , Google OAuth dan ZurbFoundation. Tujuan studi pustaka ini untuk memahami secara rinci cara-cara untuk menambahkan layanan berbentuk modul ke dalam BlueTape dan membangun layanan tersebut menggunakan *library-library* yang disebutkan sebelum ini.

3.2 BlueTape

Aplikasi *Blue Tape* adalah perangkat lunak *open source* sederhana yang memiliki tujuan utama untuk mengubah berbagai pekerjaan *paper-based* di FTIS UNPAR menjadi *paperless*. Selain itu perangkat lunak ini memiliki beberapa kegunaan lainnya seperti mengautentikasi mahasiswa dan staf UNPAR via OAuth 2.0 ke Google (layanan OAuth ke Google ini juga dapat digunakan untuk menentukan hak akses yang bisa dilihat dari email pengguna) dan *Pilot Project* untuk permohonan transkrip ke Tata Usaha . Aplikasi ini merupakan aplikasi berbasis web dengan memanfaatkan *Codeigniter* dan *Zurb Foundation*.

Perangkat lunak *Blue Tape* ini didesain sebagai *framework* yang terdiri dari beberapa layanan yang dipisahkan ke dalam modul-modul. Pemisahan layanan ke dalam modul-modul dibuat dengan tujuan agar pemeliharaan perangkat lunak lebih mudah dan juga mempermudah cara untuk menambahkan layanan baru ke dalam BlueTape. Sudah ada layanan yang aktif di BlueTape saat ini yaitu *Transcript Request / Manage* yang memiliki fungsi untuk melakukan permohonan serta pencetakan transkrip mahasiswa.

3.2.1 Modul Saat Ini

Pada saat skripsi ini ditulis, sudah ada dua modul yang ada di BlueTape yaitu Perubahan Kuliah dan *Transcript Request / Manage*.

3.2.1.1 *Transcript Request / Manage*

Modul *Transcript Request/Manage* merupakan salah satu dari dua modul yang sudah aktif di BlueTape ketika skripsi ini ditulis. Modul ini memiliki fungsi utama sebagai alat bagi mahasiswa untuk memohon pencetakan transkrip ke Tata Usaha dengan tampilan seperti di bawah ini :

Permohonan Baru

Yang memohon: NPM: Nama:

Tipe Transkrip: Keperluan:

[Kirim Permohonan](#)

Histori Permohonan

ID	Status	Tanggal Permohonan	Tipe Transkrip	Tanggal Jawab/Cetak	Keterangan	Aksi
----	--------	--------------------	----------------	---------------------	------------	------

Gambar 3.1: Tampilan Cetak Transkrip

Permintaan Transkrip

Cari NPM: [Cari](#)

ID	Status	Tanggal Permohonan	Tipe Transkrip	NPM	Aksi
#1	MENUNGGU	Senin, 16 Oktober 2017	DPS_ID	-	View Edit Print Delete

Gambar 3.2: Tampilan Manajemen Transkrip

Untuk memohon pencetakan transkrip pada halaman tersebut mahasiswa diperlukan untuk melakukan :

1. Memilih salah satu dari 3 tipe transkrip yang ada. Ada 3 tipe transkrip: Daftar Perkembangan Studi (DPS) Bahasa Inggris, DPS Bahasa Indonesia dan Lembar Hasil Studi Semester Akhir.
2. Lalu mahasiswa mengisi keterangan keperluan pencetakan transkrip di *field* keperluan.
3. Setelah kedua hal tersebut dipilih dan diisi , tekan tombol "Kirim Permohonan" untuk memohon pencetakan transkrip ke Tata Usaha.

Tata usaha memiliki beberapa opsi untuk merespon permintaan yang sudah dikirim mahasiswa tadi :

1. Melihat detail permintaan transkrip
2. Menolak permintaan transkrip
3. Mencetak detail permintaan transkrip
4. Menghapus permintaan transkrip

3.2.1.2 Perubahan Kuliah

Perubahan kuliah adalah modul kedua yang sudah aktif di BlueTape. Modul ini berfungsi sebagai alat bagi mahasiswa untuk mengirimkan permintaan perubahan kuliah dari mahasiswa ke karyawan tata usaha. Selain itu karyawan tata usaha juga dapat menggunakan modul ini untuk mengatur permintaan-permintaan yang sudah dikirim dari mahasiswa tadi.

The screenshot displays the 'Perubahan Kuliah' (Class Change) menu. The top navigation bar includes links for 'Cetak Transkrip', 'Manajemen Cetak Transkrip', 'Perubahan Kuliah' (active), 'Manajemen Perubahan Kuliah', 'Entri Jadwal Dosen', 'Lihat Jadwal Dosen', and 'Logout'.

The main content area is divided into two sections:

Permohonan Baru

This section contains a form for submitting a new request. The fields are as follows:

- Pemohon:**
- Nama:**
- Kode MK:**
- Nama Mata Kuliah:**
- Kelas:**
- Jenis Perubahan:**
- Dari Hari & Jam:**
- Dari Ruang:**
- Keterangan Tambahan:**
- Menjadi Hari & Jam:**
- Menjadi Ruang:**


At the bottom of the form are two buttons: **Kirim Permohonan** (blue) and **Tambah Pertemuan Ekstra** (grey).






Histori Permohonan

This section displays a table of request history:

ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Tanggal Jawab	Keterangan	Aksi
#1	TUNGGU	Senin, 16 Oktober 2017	AlF311	Diganti			

Gambar 3.3: Tampilan Menu Perubahan Kuliah

	Cetak Transkrip	Manajemen Cetak Transkrip	Perubahan Kuliah	Manajemen Perubahan Kuliah	Entri Jadwal Dosen	Lihat Jadwal Dosen	Logout
---	-----------------	---------------------------	------------------	-----------------------------------	--------------------	--------------------	--------

Permohonan Perubahan Kuliah					
ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Aksi
#1	MENUNGGU	Senin, 16 Oktober 2017	AIF311	Diganti	    

Gambar 3.4: Tampilan Menu Manajemen Perubahan Kuliahp

Untuk membuat permintaan perubahan kuliah, mahasiswa perlu mengisi setiap *field* yang ada di menu. Setelah semua *field* diisi, mahasiswa bisa memilih untuk mengirim permohonan atau tambah pertemuan.

- Tombol "Kirim Permohonan" akan memeriksa apakah data yang sudah dimasukan sudah benar, bila sudah maka data permohonan akan dikirim ke server dan apabila ada yang masih salah maka sistem akan menandai bagian yang salah.
- Tombol "Tambah Pertemuan Ekstra" akan memunculkan 2 *field* baru yaitu *field* "Menjadi Hari dan jam" dan *field* "Menjadi Ruang". Bila mahasiswa ingin mengirimkan permintaan penambahan pertemuan, tetap harus menekan tombol "Kirim Permohonan" setelah kedua *field* yang baru tersebut juga diisi.

Di sisi tata usaha , modul ini memiliki fungsi untuk :

- Menolak permintaan perubahan kuliah
- Menyetujui permintaan perubahan kuliah
- Mencetak detail perubahan kuliah
- Melihat detail perubahan kuliah
- Menghapus permintaan perubahan kuliah

3.2.2 Pengguna Aplikasi

Pada bagian ini akan dijelaskan tipe-tipe pengguna aplikasi.

3.2.2.1 Mahasiswa FTIS

Mahasiswa FTIS adalah semua mahasiswa Fakultas Teknik Informasi dan Sains. Saat ini golongan Mahasiwa FTIS baru memiliki satu akses yaitu untuk memnita transkrip.

3.2.2.2 Tata Usaha UNPAR

Tata Usaha UNPAR adalah golongan pengguna yang bekerja sebagai staff tata usaha di Universitas Katolik Parahyangan. Di BlueTape, Tata Usaha UNPAR memiliki akses fitur-fitur sebagai berikut :

- Mengatur permintaan transkrip dari mahasiswa
- Mengatur permintaan perubahan kuliah

3.2.2.3 Staf UNPAR

Staf UNPAR adalah para pekerja dan karyawan di Universitas Katolik Parahyangan. Untuk saat ini golongan staf UNPAR hanya memiliki akses fitur untuk meminta perubahan kuliah.

3.2.2.4 Mahasiswa Informatika

Mahasiswa Informatika adalah pengguna yang memiliki kepentingan untuk melihat jadwal-jadwal semua dosen Informatika. Dengan mengetahui jadwal dosen, maka mahasiswa dapat mengatur jadwal bimbingan atau konsultasi lainnya dengan dosen terkait secara lebih teratur. Di Bluetape, mahasiswa memiliki akses-akses pada fitur :

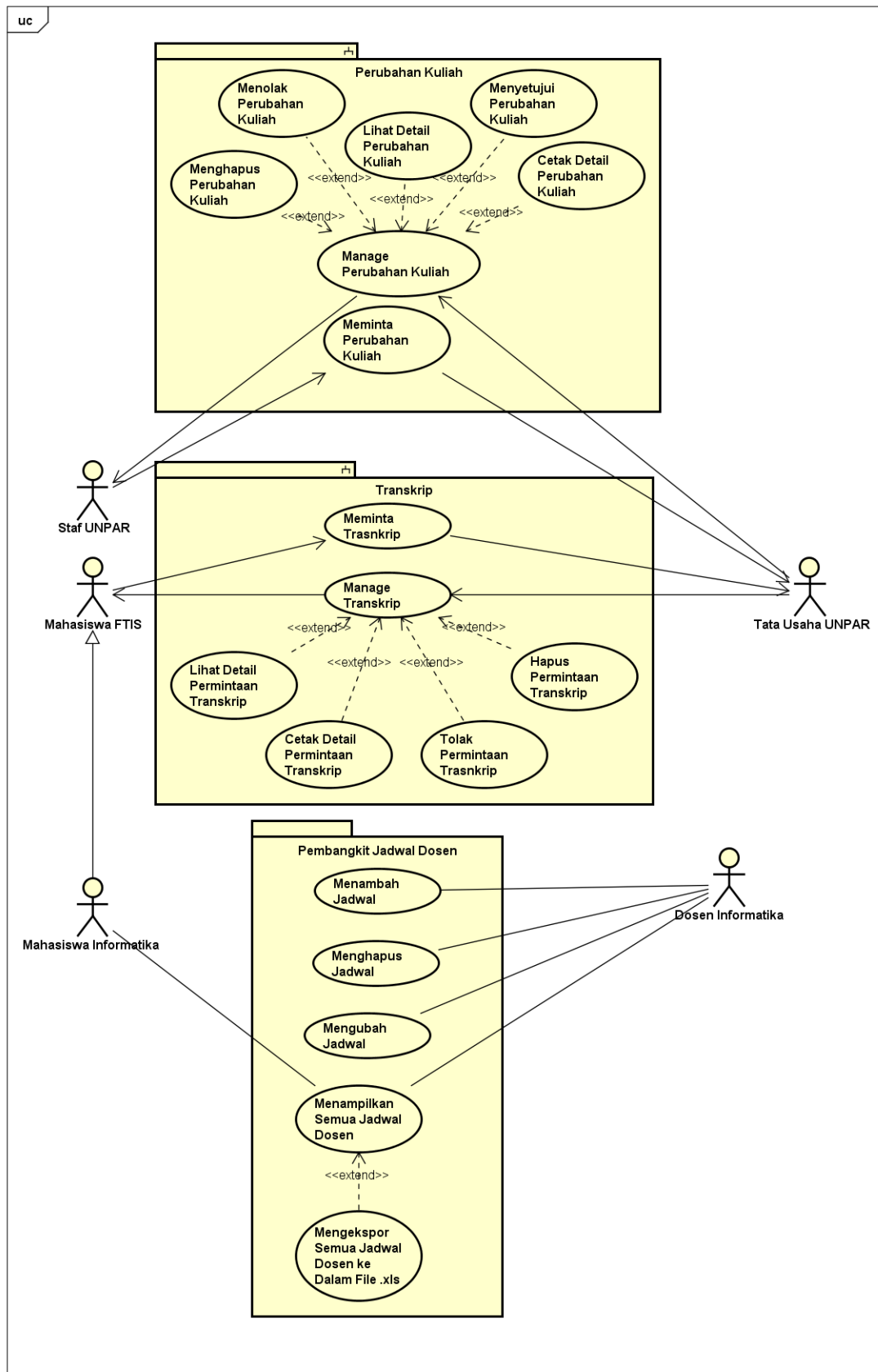
- Melihat semua jadwal dosen Informatika
- Mengekspor semua jadwal dosen Informatika ke dalam tipe file .xls
- Meminta transkrip nilai

3.2.2.5 Dosen Informatika

Dosen merupakan pengguna yang akan memasukan jadwal-jadwalnya ke dalam BlueTape agar dapat dilihat mahasiswa. Dosen memiliki akses pada fitur :

- Memasukan jadwal ke dalam BlueTape
- Mengubah jadwal yang sudah dimasukan
- Menghapus jadwal yang sudah dimasukan
- Melihat semua jadwal dosen , termasuk jadwal miliknya sendiri
- Mengeskpor semua jadwal dosen ke dalam tipe file .xls

3.3 Diagram Use Case



Gambar 3.5: Diagram Use Case

Berikut ini adalah penjelasan skenario dari diagram *use case* di atas

1. Skenario Meminta Perubahan Kuliah

- Aktor : Staf UNPAR, Tata Usaha UNPAR
- Skenario Normal
 1. Staf UNPAR mengisi data-data yang diperlukan untuk melakukan perubahan kuliah
 2. Sistem mengirim data-data tersebut ke server
 3. Tata Usaha menerima data yang dikirim oleh Staf UNPAR untuk diproses
- Skenario *Exception*
 1. Staf UNPAR memasukan data secara tidak lengkap
 2. Sistem menampilkan pesan eror karena data tidak lengkap

2. Skenario Melihat Detail Perubahan Kuliah

- Aktor : Tata Usaha UNPAR
- Skenario Normal
 1. Aktor memilih data perubahan kuliah
 2. Aktor memilih menu lihat detail
 3. Sistem menampilkan semua data-data permintaan mahasiswa dengan npm terkait

3. Skenario Menolak Perubahan Kuliah

- Aktor : Staf UNPAR, Tata Usaha UNPAR
- Skenario Normal
 1. Tata Usaha UNPAR memilih data perubahan kuliah
 2. Tata Usaha UNPAR memilih menu tolak perubahan kuliah
 3. Tata Usaha UNPAR mengisi data alasan penolakan
 4. Sistem mengirim pesan berisi alasan penolakan ke staf UNPAR
 5. Sistem menghapus permintaan perubahan kuliah dari *database*

4. Skenario Menyetujui Perubahan Kuliah

- Aktor : Staf UNPAR, Tata Usaha UNPAR
- Skenario Normal
 1. Tata Usaha UNPAR memilih data perubahan kuliah
 2. Tata Usaha UNPAR memilih menu tolak perubahan kuliah
 3. Tata Usaha UNPAR mengisi keterangan penyetujuan
 4. Sistem mengirim pesan berisi keterangan penyetujuan ke staf UNPAR
 5. Sistem menghapus permintaan perubahan kuliah dari *database*

5. Skenario Menghapus Perubahan Kuliah

- Aktor : Tata Usaha UNPAR
- Skenario Normal
 1. Tata Usaha UNPAR memilih data perubahan kuliah
 2. Tata Usaha UNPAR memilih menu hapus
 3. Sistem menghapus permintaan perubahan kuliah dari *database*

6. Skenario Cetak Detail Perubahan Kuliah

- Aktor : Tata Usaha UNPAR
 - Skenario Normal
 1. Tata Usaha UNPAR memilih data perubahan kuliah
 2. Tata Usaha UNPAR memilih menu cetak
 3. Sistem mencetak detail permintaan perubahan kuliah
7. Skenario Meminta Transkrip
- Aktor : Mahasiswa FTIS
 - Skenario Normal
 1. Mahasiswa FTIS mengisi data keterangan dan tipe transkrip
 2. Sistem menyimpan data-data tersebut ke server
8. Skenario Lihat Detail Permintaan Transkrip
- Aktor : Tata Usaha UNPAR
 - Skenario Normal
 1. Tata Usaha UNPAR memilih data permintaan
 2. Tata Usaha UNPAR memilih menu lihat detail
 3. Sistem menampilkan detail data permintaan
9. Tolak Permintaan Transkrip
- Aktor : Tata Usaha UNPAR
 - Skenario Normal
 1. Tata Usaha UNPAR memilih data permintaan
 2. Tata Usaha UNPAR memilih menu tolak permintaan
 3. Sistem menampilkan menu keterangan penolakan
 4. Tata Usaha UNPAR mengisi data keterangan penolakan
 5. Sistem mengirim pesan berisi keterangan penolakan ke Mahasiswa FTIS.
10. Hapus Permintaan Transkrip
- Aktor : Tata Usaha UNPAR
 - Skenario Normal
 1. Tata Usaha UNPAR memilih data permintaan
 2. Tata Usaha UNPAR memilih menu hapus
 3. Sistem menampilkan pesan konfirmasi
 4. Tata Usaha UNPAR mengkonfirmasi penghapusan
 5. Sistem menghapus data permintaan dari *database*
11. Skenario Menambah Jadwal
- Aktor : Dosen Informatika
 - Skenario Normal
 1. Dosen Informatika mengisi data jadwal
 2. Sistem menyimpan data jadwal tersebut ke dalam *database*
12. Skenario Mengubah Jadwal
- Aktor : Dosen Informatika

- Skenario Normal

1. Dosen Informatika memilih jadwal yang akan diubah
2. Sistem menampilkan menu pengubahan berisi data jadwal yang dipilih tadi
3. Dosen Informatika mengubah data-data yang ingin diubah
4. Sistem menyimpan hasil perubahan ke dalam database

13. Skenario Menghapus Jadwal

- Aktor : Dosen Informatika

- Skenario Normal

1. Dosen Informatika memilih jadwal yang akan dihapus
2. Sistem menampilkan menu pengubahan berisi data jadwal yang dipilih
3. Dosen Informatika menekan tombol hapus
4. Sistem menghapus data jadwal tersebut yang ada di database

14. Skenario Menampilkan Semua Jadwal Dosen

- Aktor : Mahasiswa Informatika , Dosen Informatika

- Skenario Normal

1. Aktor memilih menu lihat jadwal dosen
2. Sistem memuat semua data jadwal-jadwal dari *database*
3. Sistem mengelompokkan setiap jadwal berdasarkan pemiliknya
4. Sistem membuat tab-tab yang merepresentasikan setiap dosen yang sudah menyimpan jadwal ke dalam database
5. Sistem memasukan jadwal-jadwal ke dalam tab-tab sesuai nama pemiliknya.

15. Skenario Mengekspor Semua Jadwal Dosen ke Dalam File .xls

- Aktor : Mahasiswa Informatika , Dosen Informatika

- Skenario Normal

1. Aktor memilih menu lihat jadwal dosen
2. Sistem memuat semua data jadwal-jadwal dari *database*
3. Sistem mengelompokkan setiap jadwal berdasarkan pemiliknya
4. Sistem membuat tab-tab yang merepresentasikan setiap dosen yang sudah menyimpan jadwal ke dalam database
5. Sistem memasukan jadwal-jadwal ke dalam tab-tab sesuai nama pemiliknya.
6. Aktor menekan tombol ekspor
7. Sistem mengkonversi semua data jadwal dari bentuk php ke dalam bentuk *spreadsheet* .xls
8. Sistem menampilkan menu pemilihan lokasi penyimpanan *file* .xls
9. Aktor menekan tombol simpan
10. Sistem menyimpan file .xls tersebut di lokasi yang sudah dipilih oleh aktor.

- Skenario Exception

1. Aktor memilih menu lihat jadwal dosen
2. Sistem memuat semua data jadwal-jadwal dari *database*
3. Sistem tidak menerima data apapun dari *database*
4. Sistem men-*disable* tombol ekspor

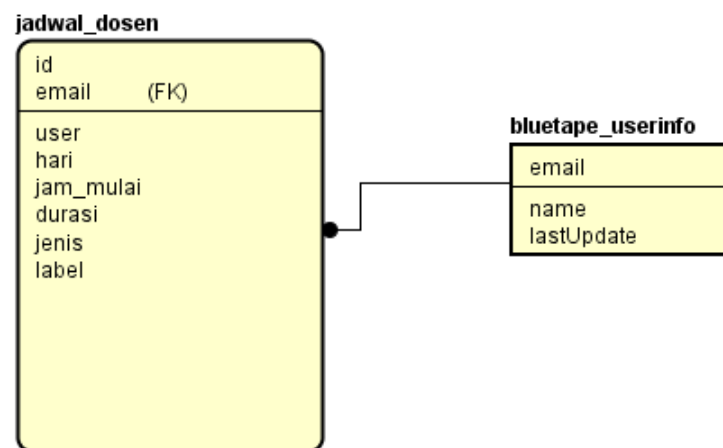
BAB 4

PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan Basis Data

4.1.1 Model Relasional

Dari analisis yang telah dilakukan, maka dibuatlah model relasional seperti pada gambar 4.1



Gambar 4.1: Model Relasional

4.1.2 Perancangan Tabel

Atribut	Tipe Data	Ukuran	Primary Key	Foreign Key	Null	Keterangan
id	int	11	ya	tidak	tidak	id jadwal
user	varchar	256	tidak	ya	tidak	pemilik jadwal
hari	int	11	tidak	tidak	tidak	hari berlangsungnya jadwal
jam_mulai	int	11	tidak	tidak	tidak	jam berlangsungnya jadwal
durasi	int	11	tidak	tidak	tidak	lama jadwal berlangsung
jenis	varchar	255	tidak	tidak	tidak	jenis kegiatan jadwal
label	varchar	100	ya	tidak	tidak	nama kegiatan

Tabel 4.1: Perancangan Tabel Jadwal

4.1.3 Perancangan Rinci

4.1.3.1 Controller EntriJadwalDosen

Berikut adalah perancangan kelas Controller EntriJadwalDosen

Nama Method	insert
Parameter Output	insert_success, insert_failed
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk memasukan jadwal
Algoritma	<ul style="list-style-type: none"> • proses menerima input-input data dari user • proses mengecek apakah data-data yang dimasukan sudah valid • proses mengecek apakah jadwal yang bertabrakan dengan jadwal lain atau tidak • bila jadwal bertabrakan dengan jadwal lain maka proses menampilkan pesan "Jadwal gagal dimasukan. Sudah ada jadwal lain pada waktu ini" • bila tidak ada masalah , proses akan memasukan data ke dalam database

Nama Method	edit
Parameter Output	edit_success, edit_failed
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk mengubah jadwal yang sudah dimasukan sebelumnya
Algoritma	<ul style="list-style-type: none"> • proses menerima input-input data dari user • proses mengecek apakah data-data yang dimasukan sudah valid • proses mengecek apakah waktu baru dari jadwal yang diubah bertabrakan dengan jadwal lain atau tidak • bila jadwal bertabrakan dengan jadwal lain maka proses menampilkan pesan "Pengubahan gagal. Sudah ada jadwal lain pada waktu ini" • bila tidak ada masalah, maka jadwal akan memperbarui data yang ada di database.
Nama Method	delete
Parameter Output	deleted
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk menghapus jadwal
Algoritma	<ul style="list-style-type: none"> • proses menerima pilihan jadwal yang akan dihapus oleh user. • proses mengeluarkan form konfirmasi untuk memastikan user untuk menghapus jadwal yang bersangkutan • bila user memilih "ya", maka data jadwal yang bersangkutan akan dihapus dari database. • bila pilihan user "tidak" maka proses akan menampilkan halaman sebelum user menekan tombol <i>delete</i>.

4.1.3.2 Controller LihatJadwalDosen

Berikut adalah perincian rancangan kelas Controller LihatJadwalDosen

Nama Method	index
Parameter Output	-
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk memperlihatkan jadwal ke user
Algoritma	<ul style="list-style-type: none"> • proses memuat semua data jadwal dari database. • proses mengelompokkan jadwal-jadwal dari database tersebut berdasarkan pemiliknya. • proses membuat tab-tab setiap tab diberi nama dosen yang sudah memasukan jadwal ke dalam sistem • proses memasukan setiap jadwal yang sudah dipisahkan berdasarkan pemilik ke dalam tab-tab sesuai dengan nama pemilik yang bersangkutan. • secara default akan ditampilkan jadwal dosen yang pertama kali dimuat oleh proses. • bila user menekan tab, proses akan menampilkan semua jadwal dosen terkait

Nama Method	ekspor
Parameter Output	-
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk mengkonversi jadwal dari php ke dalam tipe file .xls
Algoritma	<ul style="list-style-type: none"> • proses memuat semua data jadwal dari database. • proses mengelompokkan jadwal-jadwal dari database tersebut berdasarkan pemiliknya. • proses membuat tab-tab di dalam spreadsheet setiap tab diberi nama dosen yang sudah memasukan jadwal ke dalam sistem • proses memasukan setiap jadwal yang sudah dipisahkan berdasarkan pemilik ke dalam tab-tab di speradsheet sesuai dengan nama pemilik yang bersangkutan. • secara default akan ditampilkan jadwal dosen yang pertama kali dimuat oleh proses. • bila user menekan tab, proses akan menampilkan semua jadwal dosen terkait

4.1.3.3 Model JadwalDosen

Berikut adalah perincian rancangan kelas Model JadwalDosen

Nama Method	add_jadwal
Parameter Output	-
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk memasukan jadwal ke dalam <i>database</i>
Algoritma	<ul style="list-style-type: none"> • proses menerima data dari user • data-data dimasukan ke dalam <i>database</i>
Nama Method	update_jadwal
Parameter Output	-
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk mengupdate data jadwal di <i>database</i> berdasarkan pilihan user
Algoritma	<ul style="list-style-type: none"> • proses menerima data dan id_jadwal • proses mengupdate data di <i>database</i> berdasarkan id_jadwal
Nama Method	delete_jadwal
Parameter Output	-
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk menghapus data jadwal dari <i>database</i> berdasarkan pilihan user
Algoritma	<ul style="list-style-type: none"> • proses menerima id_jadwal • proses menghapus data jadwal di <i>database</i> berdasarkan id_jadwal
Nama Method	get_jadwal
Parameter Output	-
Tabel yang berhubungan	jadwal_dosen
Deskripsi	Proses untuk mengambil semua data jadwal
Algoritma	<ul style="list-style-type: none"> • proses memuat semua data jadwal dari <i>database</i>. • proses mengirim semua data yang suda dimuat ke controller pemanggil

4.2 Perancangan Antarmuka

Pada bagian ini akan dibahas rancangan antarmuka Aplikasi Pembangkit Jadwal Dosen.

4.2.1 Perancangan Antarmuka Entri Jadwal Dosen

Perancangan tampilan untuk modul Entri Jadwal Dosen dapat dilihat pada gambar di bawah ini

skripsi.com

← → ↻ 🔍 ☰

Entri Jadwal Dosen Lihat Jadwal Dosen

Tambah Jadwal

Hari: Durasi: Label:

Jam Mulai: Jenis:

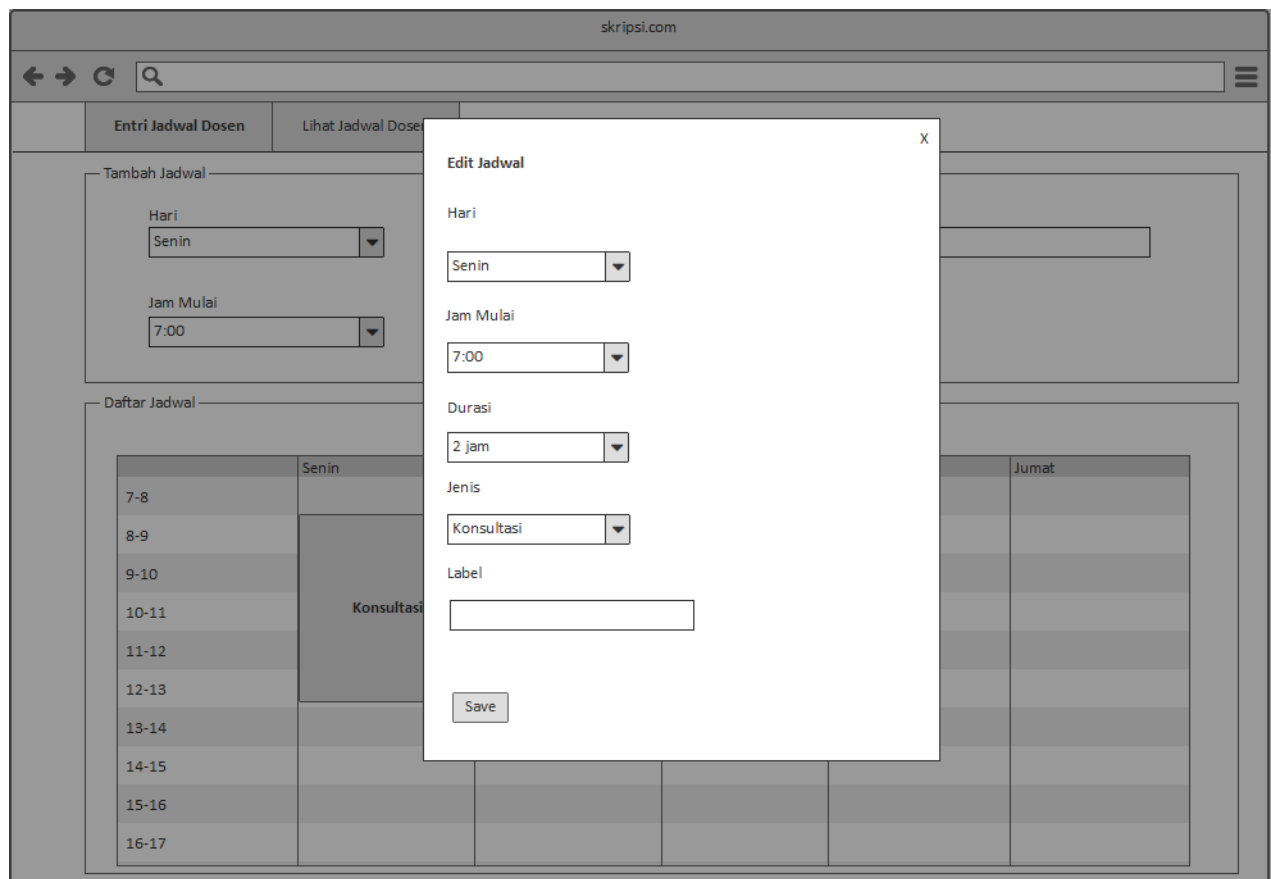
Daftar Jadwal

	Senin	Selasa	Rabu	Kamis	Jumat
7-8					
8-9	Konsultasi				
9-10					
10-11					
11-12					
12-13					
13-14					
14-15					
15-16					
16-17					

Gambar 4.2: Perancangan Antarmuka Entri Jadwal Dosen

4.2.2 Perancangan Antarmuka Edit Jadwal Dosen

Perancangan antarmuka untuk *pop-up* pengubahan data jadwal yang sudah dimasukan dapat dilihat pada gambar berikut



Gambar 4.3: Perancangan Antarmuka Edit Jadwal Dosen

4.2.3 Perancangan Antarmuka Lihat Jadwal Dosen

Perancangan antarmuka untuk modul Lihat Jadwal Dosen dapat dilihat pada gambar di bawah ini

skripsi.com

← → ↻ 🔍 ☰

Entri Jadwal Dosen **Lihat Jadwal Dosen**

Nama Dosen 0	Nama Dosen 1	Nama Dosen 2
	Senin	Selasa
7-8		Rabu
8-9	Konsultasi	Kamis
9-10		Jumat
10-11		
11-12		
12-13		
13-14		
14-15		
15-16		
16-17		

Ekspor ke XLS

Gambar 4.4: Perancangan Antarmuka Lihat Jadwal Dosen

DAFTAR REFERENSI

- [1] Codeigniter 3 (2017) *Codeigniter 3.1.4 Documentation*. British Columbia Institute of Technology. 5112 Bobby Hicks Hwy, Gray, TN 37615, USA.
- [2] Foundation 6 (2017) *Foundation v6.3.1 Documentation*. Zurb Inc. 100 W Rincon Ave, Campbell, CA 95008, USA.
- [3] Version 1.8.0 (2014) *PHPExcel Formula Function Reference Developer Documentation*. PHPOffice. 100 W Rincon Ave, Campbell, CA 95008, USA.

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf

```

Listing A.2: MyCode.java

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }

```


LAMPIRAN B

HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4