

# SKRIPSI

**ANALISA DAN IMPLEMENTASI PERBAIKAN PERANGKAT LUNAK BLUETAPE**



**Stephen Hadi**

**NPM: 2017730016**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2020**



# UNDERGRADUATE THESIS

## IMPROVEMENT ANALYSIS AND IMPLEMENTATION OF BLUETAPE SOFTWARE



Stephen Hadi

NPM: 2017730016

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2020



# LEMBAR PENGESAHAN

ANALISA DAN IMPLEMENTASI PERBAIKAN PERANGKAT LUNAK BLUETAPE

Stephen Hadi

NPM: 2017730016

Bandung, 25 Februari 2020

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Pascal Alfadian, M.Comp.

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### **ANALISA DAN IMPLEMENTASI PERBAIKAN PERANGKAT LUNAK BLUETAPE**

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 25 Februari 2020

Meterai Rp. 6000
---------------------

**Stephen Hadi**  
NPM: **2017730016**





## ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

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.

**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, Februari 2020

Penulis





# DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
<b>1 PENDAHULUAN</b>	<b>1</b>
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
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Code Igniter . . . . .	5
2.1.1 Controller . . . . .	5
2.1.2 Model . . . . .	6
2.1.3 View . . . . .	7
2.1.4 Kelas pada Code Igniter . . . . .	7
2.2 Phpspreadsheet . . . . .	11
2.2.1 Kelas pada Phpspreadsheet . . . . .	11
2.2.2 Instalasi Phpspreadsheet . . . . .	13
2.2.3 Contoh Penulisan Nilai pada <i>cell</i> . . . . .	14
2.2.4 Contoh Penulisan Spreadsheet ke Xls . . . . .	14
2.3 Bootstrap . . . . .	14
2.3.1 Pemasangan Bootstrap . . . . .	14
2.3.2 Container . . . . .	15
2.3.3 Grid . . . . .	15
2.3.4 Table . . . . .	15
2.3.5 Pagination . . . . .	16
2.3.6 Modal . . . . .	16
2.3.7 Form . . . . .	17
2.3.8 Navbar . . . . .	18
2.3.9 Nav . . . . .	18
2.4 JQuery . . . . .	19
<b>3 ANALISIS</b>	<b>21</b>
3.1 Analisis Program Usulan <i>Bluetape</i> . . . . .	21
3.1.1 Fitur "Expor ke XLS" pada Menu EntriJadwalDosen Menghasilkan File <i>Corrupt</i> . . . . .	22

3.1.2	<i>Update google api/phpspreadsheet</i> . . . . .	22
3.1.3	Fitur <i>chart</i> pada ManajemenCetakTranskrip . . . . .	23
3.1.4	Fitur <i>chart</i> pada ManajemenPerubahanKuliah . . . . .	23
3.1.5	Mahasiswa dengan NPM Baru Tidak Dapat <i>Login</i> dan LihatJadwalDosen . . . . .	23
3.1.6	Kolom pada EntriJadwalDosen dan LihatJadwalDosen Tidak Seragam . . . . .	23
3.1.7	Fungsi tab pada LihatJadwalDosen tidak Berfungsi . . . . .	23
3.1.8	Pengelompokkan Rekap Perubahan Jadwal pada Mata Kuliah yang Sama . . . . .	23
3.1.9	Menyediakan Forum Q&A pada <i>Bluetape</i> . . . . .	23
3.1.10	<i>Scheduling</i> Matakuliah pada <i>googlemeet/zoom</i> . . . . .	23
3.1.11	Mengubah atau Membatalkan Permohonan . . . . .	23
3.1.12	Menambahkan Jam Kuliah Selesai di Perubahan Kelas . . . . .	23
3.1.13	Memperbaiki <i>Form</i> dan <i>Link</i> yang Tidak Aktif . . . . .	23
3.1.14	Mengintegrasikan <i>Bluetape</i> dengan SSO UNPAR . . . . .	23
3.1.15	Menambah <i>List</i> Permintaan pada <i>Bluetape</i> . . . . .	23
3.1.16	Pengajuan Surat Keterangan Aktif Kuliah Secara Online . . . . .	24
3.1.17	Notifikasi E-mail Terhadap Mahasiswa saat Permintaan Selesai . . . . .	24
3.1.18	Dapat Melihat Profil Mahasiswa . . . . .	24
3.1.19	Halaman Histori dan Request Transkrip Terpisah . . . . .	24
3.1.20	Fitur Bahasa Indonesia dan Inggris . . . . .	24
3.1.21	Pagination tidak Ter- <i>style</i> dengan baik . . . . .	24
3.1.22	Format Datetimepicker tidak Konsisten . . . . .	24
3.2	<i>Bluetape</i> . . . . .	24
3.2.1	Instalasi . . . . .	24
3.2.2	Struktur <i>Bluetape</i> . . . . .	24
3.2.3	Pengaturan Dasar <i>Bluetape</i> . . . . .	24
3.2.4	Database . . . . .	24

<b>DAFTAR REFERENSI</b>	<b>25</b>
-------------------------	-----------

<b>A KODE PROGRAM</b>	<b>27</b>
-----------------------	-----------

<b>B HASIL EKSPERIMEN</b>	<b>29</b>
---------------------------	-----------

## DAFTAR GAMBAR

2.1	Flowchart MVC . . . . .	5
2.2	Gambar bootstrap-grid . . . . .	15
2.3	Gambar bootstrap-grid . . . . .	18
2.4	Navigasi dari <code>nav-tabs</code> . . . . .	18
3.1	Antarmuka halaman <code>EntriJadwalDosen</code> . . . . .	22
B.1	Hasil 1 . . . . .	29
B.2	Hasil 2 . . . . .	29
B.3	Hasil 3 . . . . .	29
B.4	Hasil 4 . . . . .	29



## DAFTAR TABEL

2.1	Tabel format yang didukung oleh phpspreadsheet . . . . .	11
2.2	<i>Browser Mobile</i> yang mendukung <i>bootstrap</i> . . . . .	14
2.3	<i>Browser Desktop</i> yang mendukung <i>bootstrap</i> . . . . .	14
3.1	Tabel analisis kebutuhan pengguna perangkat <i>bluetape</i> . . . . .	21



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Berkembangnya FTIS UNPAR disertai dengan tersedianya semakin banyak matakuliah muncul permasalahan baru di bagian bidang administrasi. Jika dosen ingin meniadakan perkuliahan atau mengganti perkuliahan akan tidak efisien jika melakukan panggilan atau *email* ke pihak tata usaha. Hal tersebut akan memberatkan pihak tata usaha. Mahasiswa yang ingin melakukan pengajuan transkrip akan membuang waktu dan tenaga, karena saat mahasiswa ingin melakukan pengajuan transkrip maka mahasiswa harus datang ke tata usaha melakukan pengajuan dan menunggu beberapa hari untuk mendapatkan hasil transkrip tersebut. Mahasiswa yang tidak memiliki perkuliahan pada hari tersebut harus datang hanya untuk melakukan pengajuan hal ini selain membuang waktu juga membuang biaya transportasi.

*Bluetape*<sup>1</sup> adalah aplikasi web yang dibuat oleh dosen dan mahasiswa informatika. Aplikasi ini dibuat menggunakan *Hypertext Preprocessor* atau lebih dikenal dengan *PHP*<sup>2</sup>. *Database management system* atau DBMS yang digunakan adalah *MYSQL*<sup>3</sup>. *Bluetape* menggunakan *framework Bootstrap*<sup>4</sup> dan *Codeigniter*<sup>5</sup>. *Bluetape* berguna untuk membantu kegiatan administrasi FTIS UNPAR. Aplikasi ini dapat melakukan *transkrip request/manage* dan *request perubahan kuliah/manage*. Sehingga jika dosen ingin meniadakan/mengganti perkuliahan dapat dilakukan dengan mudah tanpa harus membuat email ataupun melakukan panggilan. Mahasiswa dapat melakukan permintaan transkrip nilai tanpa tatap muka sehingga mahasiswa hanya perlu datang ke UNPAR saat ingin mengambil hasil dari transkrip tersebut, selain itu mahasiswa juga dapat melihat jadwal dosen. Dengan adanya sistem otomasi pada kegiatan administrasi tentunya pekerjaan tata usaha menjadi lebih ringan.

Aplikasi *Bluetape* digunakan oleh mahasiswa FTIS, dosen, dan tata usaha. Dari pengguna-pengguna tersebut tentu akan ada hal yang disukai dan hal yang tidak disukai. Seperti ada fitur yang bermasalah, ada fitur yang kurang atau hanya saran untuk fitur kedepannya yang akan memudahkan kegiatan administrasi dalam FTIS UNPAR. Dengan adanya masukan-masukan dari pengguna maka aplikasi *Bluetape* dapat ditingkatkan penggunaannya, memperbaiki kelemahan dari *Bluetape*.

Pada topik skripsi ini akan dilakukan analisis untuk perbaikan dan penambahan fitur untuk aplikasi *Bluetape*, penambahan fitur yang akan disurvei kepada semua pengguna *Bluetape*. Pengguna yang akan disurvei tidak semua. Hanya perwakilan-perwakilan dari pihak dosen, tata usaha, dan mahasiswa. Selanjutnya akan dianalisa bersama pembimbing untuk menentukan fitur yang akan dirancang dan diimplementasi. Survei tidak terbatas hanya pada fitur tambahan, fitur-fitur yang tidak menghasilkan sesuai kegunaannya akan diperbaiki juga.

Sebelum melakukan perancangan dan implementasi diperlukan untuk *setup* aplikasi *Bluetape* pada komputer sendiri dan ada persyaratan yang harus dipenuhi yaitu mendaftarkan diri ke *google*

---

<sup>1</sup><https://github.com/ftisunpar/BlueTape>

<sup>2</sup><https://www.php.net/>

<sup>3</sup><https://www.mysql.com/>

<sup>4</sup><https://getbootstrap.com/>

<sup>5</sup><https://codeigniter.com/>

*OAuth* diakrenakan *login* pada *Bluetape* menggunakan akun *gmail* UNPAR. Selanjutnya juga akan dipelajari *framework CodeIgniter* dan *Bootstrap* yang menjadi dasar dari aplikasi ini.

## 1.2 Rumusan Masalah

Rumusan masalah yang telah diidentifikasi sebagai berikut:

1. Apa sajakah kebutuhan yang diinginkan oleh pengguna *bluetape*?
2. Bagaimana menganalisa, merancang, dan mengimplementasi *feedback-feedback* dari pengguna?
3. Bagaimana melakukan pengujian setelah mengimplementasi *feedback-feedback* dari pengguna?

## 1.3 Tujuan

Tujuan pembuatan dan penulisan skripsi ini adalah sebagai berikut:

1. Mendapatkan *feedback-feedback* dari pengguna terkait kebutuhan yang diinginkan.
2. Menganalisa, merancang, dan mengimplementasi *feedback-feedback* dari pengguna
3. Melakukan pengujian setelah mengimplementasi semua *feedback* yang memungkinkan untuk diimplementasi

## 1.4 Batasan Masalah

Penulisan dan pembuatan skripsi ini memiliki batasan sebagai berikut:

1. Penelitian ini tidak menjelaskan tentang cara bagaimana melakukan *deploy* aplikasi pada server.
2. Fitur-fitur yang membutuhkan lebih dari 1 semester untuk diimplementasi yang sebelumnya telah di diskusikan oleh pembimbing.

## 1.5 Metodologi

Metedologi yang akan dilakukan pada penulisan skripsi ini sebagai berikut:

1. Mempelajari *framework CodeIgniter* dan *Bootstrap*.
2. Melakukan survei ke pengguna.
3. Menganalisa kuisisioner dari pengguna bersama pembimbing untuk menetapkan fitur tambahan yang akan dikerjakan
4. Merancang dan mengimplementasi fitur tambahan tersebut.
5. Melakukan pengujian dan perbaikan selama 1 semester.
6. Menulis dan menyelesaikan dokumen skripsi.



## 1.6 Sistematika Pembahasan

Rencana pembahasan penelitian ini sebagai berikut:

- Bab 1** Pendahuluan membahas hal-hal dasar pada skripsi yang berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, sistematika pembahasan.
- Bab 2** Landasan teori berisi dasar-dasar teori meliputi : *Codeigniter*, *PHPspreadsheet*, *Bootstrap*.
- Bab 3** Analisis akan menganalisa survei dari pengguna.
- Bab 4** Perancangan berisi perancangan antarmuka untuk fitur yang akan diimplementasikan.
- Bab 5** Implementasi dan pengujian berisi hasil-hasil implementasi dan pengujian secara fungsional dan eksperimental.
- Bab 6** Kesimpulan dan saran berisi kesimpulan dari penelitian ini dan saran untuk pengembangan berikutnya.



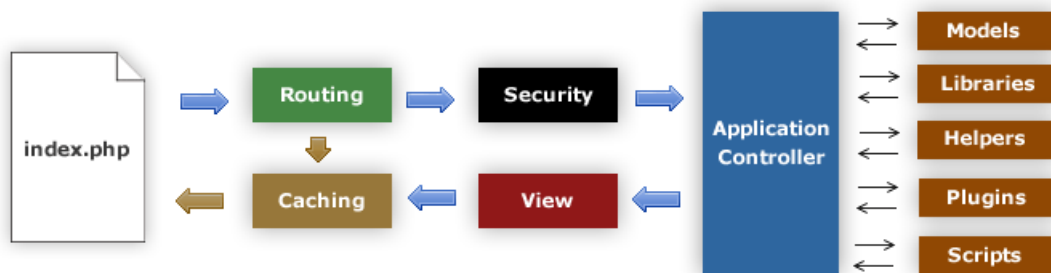
## BAB 2

### LANDASAN TEORI

Pada bab ini akan berisi landasan-landasan teori yang dipakai pada penelitian ini.

#### 2.1 Code Igniter

CodeIgniter[1] adalah *framework* untuk pembuat website yang menggunakan *PHP*. *CodeIgniter* mempermudah *developer* untuk meminimalisir penggunaan kode untuk mengakses suatu fungsi. Seperti untuk mengambil data pada *database*, mengakses file *php* lainnya. Penggunaan *framework CodeIgniter* juga mudah. *Developer* tidak perlu melakukan banyak konfigurasi-konfigurasi saat melakukan *setup*. *CodeIgniter* juga memberikan dokumentasi yang lengkap. Permasalahan *routing* sudah diselesaikan oleh *framework* ini. *Framework* ini secara otomatis akan mengarah ke file dalam *directory controllers* sesuai dengan *path-abempty* pada *URI* dan menjalankan *method index()*.



Gambar 2.1: Flowchart MVC

*CodeIgniter* menerapkan arsitektur *MVC* yang dapat dilihat pada gambar 2.1, file *index.php* berfungsi mengatur routing dan mengarahkan ke *application controller* yang berada di *directory controller* dan melalui *controller* akan dipanggil *models*, *libraries*, *helpers*, *etc* yang dibutuhkan dengan perintah `$this->load->«apa_yang_mau_diload»('«nama_file»')`. Masukan akan diolah melalui *models* dan hasil yang sudah siap akan dikirim ke *view* melalui *controller*. Fitur tambahan dari arsitektur *CodeIgniter* adalah saat *router* memeriksa *HTTP request* jika *cache* tersedia maka akan dikirimkan *cache* tersebut dan jika tidak ada *cache* maka *security* akan memeriksa dan melakukan filter terhadap *HTTP request* seperti pada gambar 2.1.

##### 2.1.1 Controller

*Controller* adalah pusat dari aplikasi, *controller* menangani apa yang harus dilakukan dari *HTTP request*. Dalam *CodeIgniter* untuk menginisiasi *controller* cukup menulis nama kelas diikuti dengan `extends CI_Controller`.

*CodeIgniter* secara otomatis akan menjalankan *method index()* jika tidak diperintahkan untuk menjalankan *method* tertentu. Untuk menjalankan *method* lain hanya perlu ditambahkan

*path-abempty* seperti `example.com/index.php/Welcome/«nama_method»`. fungsi diatas akan mengembalikan file `welcome_message.php` pada direktori *view*. Developer dapat menaruh parameter pada *view* tersebut.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Welcome extends CI_Controller {

    public function __construct(){
        parent::__construct();
        $this->load->database();
        $this->load->model(contoh_model);
    }

    public function index()
    {
        $t = "hello";
        $this->load->view('welcome_message', array(
            't' => $t));
    }
}
```

Fungsi *constructor* yang dijalankan pada *codeigniter* harus memanggil `parent::__construct()`. Dalam contoh diatas juga dapat dilakukan *load model* dan *database*. File *model* akan berada di direktori *model* sedangkan untuk `$this->load->database()` akan melakukan *load* pada *database* menggunakan parameter yang ada di direktori `/config/database.php`. Begitu juga dengan kebutuhan-kebutuhan lainnya dapat dilakukan dengan *method* `$this->load`.

### 2.1.2 Model

*Model* berfungsi sebagai *logic* dari aplikasi. *Model* pada *CodeIgniter* bersifat opsional, tetapi disediakan untuk *developer* yang ingin menggunakan MVC[1].

```
class Blog_model extends CI_Model {

    public function get_last_ten_entries()
    {
        $query = $this->db->get('entries', 10);
        return $query->result();
    }
}
```

*Model* pada *CodeIgniter* harus diikuti dengan `extends CI_Model`. Hal yang berurusan terhadap *database* dapat dilakukan dengan perintah `$this->db->query('«isi query»')` atau untuk mempermudah, beberapa fungsi *MYSQL* dasar disediakan oleh *CodeIgniter*. *Method* bisa langsung digunakan seperti `$this->db->get('«nama tabel»')` untuk mengambil semua nilai dari tabel tersebut. Untuk dapat mengakses *database* maka harus dipasang *database* yang akan digunakan pada `config/database.php`.

```
$config['hostname'] = 'localhost';
$config['username'] = 'myusername';
$config['password'] = 'mypassword';
```

```

$config['database'] = 'mydatabase';
$config['dbdriver'] = 'mysqli';
$config['dbprefix'] = '';
$config['pconnect'] = FALSE;
$config['db_debug'] = TRUE;

$this->load->model('model_name', '', $config);

```

File `database.php` diatas menyimpan kredensial dari *database* yang digunakan. Mulai dari *hostname, username, password dll*.

### 2.1.3 View

*View* tidak pernah dipanggil secara langsung, *view* harus dipanggil melalui *controller*[1]. *View* pada *CodeIgniter* ditaruh pada direktori *view*. Pemanggilan *view* menggunakan *method* `$this->load->view('nama_view')`. Jika *controller* ingin mengirimkan data kepada *view* maka perlu dilakukan:

```

$t = "hello";
$this->load->view('welcome_message', array('t' => $t));

```

Selanjutnya untuk menampilkan data tersebut ke halaman pada *view* dapat memanggil variabel `$t`.

### 2.1.4 Kelas pada Code Igniter

#### 2.1.4.1 CI\_Loader

*CI\_Loader* adalah kelas yang berfungsi untuk *load elements*. *Elements* dapat berupa *libraries, view files, drivers, helpers, models*. Kelas ini sudah diinisialisasi secara otomatis oleh *code igniter*. *Method* yang digunakan pada *CI\_Loader*:

- `library($library [, $params = NULL [, $object_name=NULL]])`  
*Method* ini berfungsi untuk *load* kelas yang disediakan oleh *codeigniter*.  
*Parameter*:
  - `$library(mixed)`: Nama *library* dalam bentuk *string*.
  - `$params(array)`: Parameter tambahan dalam bentuk *array*.
  - `$object_name(string)`: Parameter tambahan untuk merubah nama objek.

*Return*: *CI\_Loader*.

- `view($view [, $vars = array() [, return=FALSE]])`  
*Method* ini berfungsi untuk *load* file tampilan.  
*Parameter*:
  - `$view(string)`: Nama *view*.
  - `$vars(array)`: Parameter tambahan berupa variabel *associative array*.
  - `$return(bool)`: Apakah data dari tampilan dikembalikan.

*Return*: *mixed*.

- `model($model [, $name = "[, $db_conn=FALSE]])`  
*Method* ini berfungsi untuk *load* file *model*.  
*Parameter*:
  - `$model`: Nama *model*.
  - `$name`: Nama *model* dalam bentuk *array*.
  - `$db_conn`: Apakah koneksi database diperlukan.

- `$model(mixed)`: Nama *model*.
- `$name(string)`: Parameter tambahan untuk menamai objek *model*.
- `$db_conn(string)`: Parameter tambahan untuk konfigurasi database.

*Return*: `CI_Loader`.

- `database([$params="[, $return=FALSE[, $query_builder=NULL]])`

*Method* ini berfungsi untuk melakukan *load* kelas *database*.

*Parameter*:

- `$params(mixed)`: Parameter tambah untuk konfigurasi dari *database*.
- `$return(bool)`: Apakah mengembalikan objek *database*.
- `$query_builder(bool)`: Apakah akan *load query builder*.

*Return*: `CI_Loader` jika `$return=FALSE`, `CI_DB` jika `$return=TRUE`.

- `config($file[, $use_sections=FALSE[, $fail_gracefully=FALSE]])`

*Method* ini akan meneruskan ke *method* `$this->config->load()`.

*Parameter*:

- `$file(string)`: Nama file dari konfigurasi.
- `$use_sections(bool)`: Apakah nilai dari konfigurasi akan *load* di *section* sendiri.
- `$fail_gracefully(bool)` Apakah akan mengembalikan *False* jika gagal.

*Return*: `bool`.

#### 2.1.4.2 CI\_Config

*CI\_Config* adalah kelas yang digunakan untuk mengambil konfigurasi yang dibuat. Kelas ini dapat digunakan dengan awalan `$this->config`. *Method* yang digunakan pada *CI\_Config*:

`item($item[, $index=""])`

*Parameter*:

- `$item(string)`: Nama konfigurasi.
- `$index(string)`: Nama indeks.

#### 2.1.4.3 CI\_Email

*CI\_Email* adalah kelas yang berfungsi untuk mempermudah pengiriman *email* yang disediakan oleh *codeigniter*. *Load* kelas *CI\_Email* dapat dilakukan dengan:

```
$this->load->library('email');
```

Setelah *load email library*, untuk menggunakan kelas *CI\_Email* diawali dengan `$this->email`. *Method* yang digunakan pada *CI\_Email*:

- `from($from[, $name="[, $return_path=NULL]])`

*Method* ini digunakan untuk memasang *email address* dan nama dari pengirim.

*Parameter*:

- `$from(string)`: Memasang alamat pengirim *email*.
- `$name(string)`: Memasang nama pengirim *email*.
- `$return_path(string)`: Alamat *email* tambahan jika *email* gagal dikirim.

*Return:* CI\_Email.

- **to(\$to)**  
*Method* ini digunakan untuk menentukan alamat *email* tujuan.  
*Parameter:* \$to(*mixed*), string yang dipisahkan oleh koma untuk setiap penerima atau *array* dari alamat email.  
*Return:* CI\_Email.
- **subject(\$subject)**  
*Method* ini menulis *subject* dari *email*.  
*Parameter:* \$subject(*string*), subjek *email*.  
*Return:* CI\_Email.
- **message(\$body)**  
*Method* ini digunakan memasang pesan.  
*Parameter:* \$body(*string*), *email message body*.  
*Return:* CI\_Email.
- **send([\$auto\_clear=TRUE])**  
*Method* ini digunakan untuk mengirim *email*.  
*Parameter:* \$auto\_clear(*bool*), apakah pesan akan dibuang setelah pengiriman.  
*Return:* TRUE jika sukses, FALSE jika gagal.

#### 2.1.4.4 CI\_Input

*CI\_Input* adalah kelas yang digunakan untuk menyaring dan membersihkan data masukan untuk meningkatkan keamanan. Kelas ini juga menyediakan *method* bantuan untuk mengambil data masukan. Kelas ini sudah diinisialisasi secara otomatis. *Method* yang digunakan dari kelas *CI\_Input*:

- **post([\$index=NULL[, \$xss\_clean=NULL]])**  
Membaca masukan dengan tipe *post*.  
*Parameter:*
  - \$index(*mixed*): Parameter *name*.
  - \$xss\_clean(*bool*): Apakah akan menggunakan *XSS filtering*.*Return:* \$\_POST jika tidak ada parameter masukan, nilai dari masukan jika ditemukan, atau NULL jika tidak ditemukan.
- **server(\$index[, \$xss\_clean=NULL ])**  
*Method* ini berguna untuk mengambil *server data* seperti \$\_SERVER.  
*Parameter:*
  - \$index(*mixed*): Value *name*.
  - \$xss\_clean(*bool*): Apakah akan menggunakan *XSS filtering*.*Return:* \$\_SERVER item *value* jika ditemukan, NULL jika tidak.

#### 2.1.4.5 CI\_Migration

Kelas ini dapat digunakan untuk merubah struktur *database* dengan terstruktur. Untuk menggunakan kelas ini diperlukan pembuatan kelas baru dengan **extends** CI\_Migration. *Method* yang digunakan dari kelas ini:

- `latest()`  
*Method* ini akan *migrate* ke versi migrasi yang terbaru.  
*Parameter*: Tidak ada.  
*Return*: *Current version string* jika sukses, *FALSE* jika gagal.
- `error_string()`  
*Method* ini mengembalikan semua *error* dalam *string* saat *migrate*.  
*Parameter*: Tidak ada.  
*Return*: *string*.

#### 2.1.4.6 CI\_Session

*Session* digunakan untuk mengurus *user's state* dan melihat aktivitas *user* saat menelusuri situs. *Session* sebaiknya diinisialisasi pada konstruktor di *controller*. Menginisiasi *session* dengan cara `$this->load->library('session')`. Sedangkan penggunaannya dapat menggunakan perintah `$this->session`. *Method* yang digunakan pada kelas *CI\_Session*:

- `set_flashdata($data[, $value=NULL])`  
*Method* ini berguna untuk menyimpan data ke `$_SESSION` *superglobal* dan menandai sebagai *flashdata*. *Flashdata* adalah *session data* yang hanya tersedia untuk permintaan berikutnya dan setelah itu akan dibuang.  
*Parameter*:
  - `$data(mixed)`: *Array* dengan pasangan *key/value* untuk menyimpan ke *session data* atau *key* jika jumlah data hanya satu.
  - `$value(mixed)`: Nilai yang ingin disimpan jika `$data` adalah *key*.*Return*: *void*.
- `flashdata([$key=NULL])`  
*Method* ini mengambil nilai dari spesifik `$_SESSION` yang telah ditandai sebagai *flashdata*.  
*Parameter*: `$key(mixed)`, *key* dari *flashdata*.  
*Return*: Nilai dari *item key* tertentu, atau *array* semua *flashdata*.
- `set_userdata($data[, $value=NULL])`  
*Method* ini berguna untuk menyimpan data ke `$_SESSION` *superglobal*.  
*Parameter*:
  - `$data(mixed)`: *Array* dengan pasangan *key/value* untuk menyimpan ke *session data* atau *key* jika jumlah data hanya satu.
  - `$value(mixed)`: Nilai yang ingin disimpan jika `$data` adalah *key*.*Return*: *void*.
- `userdata([$key=NULL])`  
*Method* ini berguna untuk mengambil *value* untuk `$_SESSION` *item* atau *array* semua data user jika parameter dikosongkan.  
*Parameter*: `$key(mixed)`, *session item key* atau *null*.  
*Return*: Nilai dari *item key* tertentu, atau *array* semua data pengguna.
- `unset_userdata($key)`  
*Method* ini berfungsi untuk membuang data dari `$_SESSION` *superglobal* dengan *key* masukan.  
*Parameter*: `$key(mixed)` *key* dari *session data* yang ingin dibuang.  
*Return*: *void*



## 2.2 Phpspreadsheet

PhpSpreadsheet adalah *library* yang ditulis dengan bahasa PHP berguna untuk membaca dan menulis file dengan jenis spreadsheet seperti Excel dan LibreOffice Calc[2]. Format-format yang didukung oleh phpspreadsheet dapat dilihat pada tabel 2.1. Pada penelitian kali ini phpspreadsheet hanya digunakan untuk menulis ke dokumen dengan *extension* `.xls`

Format	Reading	Writing
Open Document Format/OASIS (.ods)	✓	✓
Office Open XML (.xlsx) Excel 2007 and above	✓	✓
BIFF 8 (.xls) Excel 97 and above	✓	✓
BIFF 5 (.xls) Excel 95	✓	
SpreadsheetML (.xml) Excel 2003	✓	
Gnumeric	✓	
HTML	✓	✓
SYLK	✓	
CSV	✓	✓
PDF (using either the TCPDF, Dompdf or mPDF libraries, which need to be installed separately)		✓

Tabel 2.1: Tabel format yang didukung oleh phpspreadsheet

### 2.2.1 Kelas pada Phpspreadsheet

Pada penelitian kali ini kelas utama yang digunakan adalah kelas *Spreadsheet*. Kelas *Spreadsheet* dapat diakses menggunakan `use PhpOffice\PhpSpreadsheet\Spreadsheet`. *Method* yang dimiliki oleh *library* *PhpSpreadsheet* yang digunakan pada penelitian kali ini:

#### 2.2.1.1 Spreadsheet

Kelas *Spreadsheet* dapat diakses dengan menggunakan:

```
use PhpOffice\PhpSpreadsheet\Spreadsheet;
$spreadsheet = new Spreadsheet();
```

*Constructor* kelas *Spreadsheet* tidak menerima parameter apapun, dan *return value* adalah *mixed*. *Method* yang tersedia pada kelas *Spreadsheet*:

- `createSheet([$sheetIndex:null|int=null])`  
*Method* ini berfungsi untuk membuat *sheet*.  
*Parameter*: `$sheetIndex`, *index* dari sheet dikosongkan jika menaruh sheet pada *index* terakhir.  
*Return*: Kelas *Worksheet*.
- `setActiveSheetIndex($pIndex:int)`  
Memilih *sheet* yang ingin dijadikan *active* berdasarkan *index*.  
*Parameter*: `$pIndex`, tipe data *int*, *index* dari *Worksheet*.  
*Return*: Kelas *Worksheet*.
- `getActiveSheet()`  
Mengembalikan kelas *Worksheet* yang aktif.  
*Parameter*: Tidak ada.  
*Return*: Kelas *Worksheet*.

### 2.2.1.2 Worksheet

Kelas *Worksheet* adalah kelas yang mengatur nilai dari *cell*, *cell style*, judul *sheet* dll. Pembuatan kelas ini dapat dilakukan dengan:

```
$spreadsheet->createSheet();
```

Variabel *\$spreadsheet* adalah kelas *Spreadsheet*, *method* yang digunakan pada kelas *Worksheet*:

- `getStyle($pCellCoordinate:string)`  
Mengembalikan kelas *Style*.  
*Parameter*: *\$pCellCoordinate* koordinat dari *cell* atau *range*, contoh : 'A1','A1:E1'.  
*Return*: Kelas *Style*.
- `setCellValue($pCoordinate : string , $pValue : mixed )`  
Merubah suatu nilai pada *cell* tertentu.  
*Parameter*:
  - *\$pCoordinate*: Koordinat dari *cell* contoh: 'A1'.
  - *\$pValue*: Nilai baru dari *cell* tersebut.*Return*: *\$this*.
- `mergeCells($pRange:string)`  
Melakukan *merge* pada *cell*.  
*Parameter*: *\$pRange* *cell range* yang ingin dilakukan *merge*, contoh: 'A1:E1'.  
*Return*: *\$this*.
- `setTitle($pValue:string [, $updateFormulaCellReferences:bool = true ] [, $validate:bool:true ] )`  
Berfungsi untuk memberi judul pada *sheet*.  
*Parameter*:
  - *\$pValue*: *String* yang akan dijadikan nama judul.
  - *\$updateFormulaCellReferences*: *Flag* untuk menentukan *cell reference* pada *formula* akan dirubah mengikuti judul baru. Direkomendasikan untuk tidak merubah nilai variabel ini.
  - *\$validate*: Nilai asal adalah *true*, pasang nilai *false* untuk melewati validasi dari judul baru.*Return*: *\$this*.
- `getRowDimension($pRow:int [, $create:bool = true ])`  
Mengambil dimensi baris pada baris tertentu.  
*Parameter*:
  - *\$pRow*: *Index* dari baris.
  - *\$create*: Nilai *default* adalah *true*.*Return*: Kelas *RowDimension*.
- `getColumnDimension($pColumn:string [, $create : bool = true ])`  
Mengambil dimensi kolom pada kolom tertentu.  
*Parameter*:
  - *\$pColumn*: *String* dari kolom, contoh: 'A'.
  - *\$create*: Nilai *default* adalah *true*.*Return*: Kelas *ColumnDimension*.

### 2.2.1.3 Style

Kelas *Style* adalah kelas yang mengatur *style* dari suatu *cell* seperti *alignment*, *fill*, *font* dll. Kelas *Style* dapat diakses menggunakan:

```
$worksheet->getStyle('A1');
```

Variabel *\$worksheet* adalah kelas *Worksheet*, *method* yang tersedia pada kelas *Style*:

- **getFill()**  
Mengembalikan kelas *Fill*. untuk merubah *fill* pada suatu *cell* dapat memanggil **setFillType()** pada kelas *Fill*.  
*Parameter*: Tidak ada. *Return*: Kelas *Fill*.
- **getAlignment()**  
Mengembalikan kelas *Alignment*, untuk merubah *alignment* pada suatu *cell* dapat memanggil **setHorizontal()** atau **setVertical()** pada kelas *Alignment*.  
*Parameter*: Tidak ada.  
*Return*: Kelas *Alignment*.
- **getFont**  
Mengembalikan kelas *Font*, untuk merubah penebalan kata dapat menggunakan **setBold()** dengan masukan *boolean*.  
*Parameter*: Tidak ada.  
*Return*: Kelas *Font*.

### 2.2.1.4 Xls

Kelas *Xls* memiliki 2 tipe yaitu *writer* dan *reader*. Pada penelitian kali ini tipe *Xls* yang digunakan hanya tipe *writer*. Kelas *Xls* dapat diinisiasi dengan:

```
$writer = new \PhpOffice\PhpSpreadsheet\Writer\Xls($spreadsheet);
```

*Constructor* dari kelas *Xls* menerima masukan berupa kelas *Spreadsheet*, *method* dari kelas *Xls*: **Save(\$pFilename:resource|string)**, berfungsi untuk menyimpan *Spreadsheet* menjadi *file*.  
*Parameter*: *\$pFilename*, menerima masukan *resource* atau *string*.  
*Return*: *void*.

## 2.2.2 Instalasi Phpspreadsheet

Sebelum dapat menginstalasi phpspreadsheet dibutuhkan composer. Composer dapat diunduh pada [getcomposer.org](https://getcomposer.org).

```
composer require phpoffice/phpspreadsheet
```

perintah tersebut digunakan untuk membuat file *composer.json* dan menginstalasi *dependencies* tersebut.

```
<?php
require 'vendor/autoload.php';
use PhpOffice\PhpSpreadsheet\Spreadsheet;
$spreadsheet = new Spreadsheet();
```

Penggunaan phpspreadsheet secara dasar membutuhkan perintah `use PhpOffice\PhpSpreadsheet\Spreadsheet` dan `new Spreadsheet()`.

### 2.2.3 Contoh Penulisan Nilai pada *cell*

*Phpspreadsheet* menyediakan suatu *method* untuk dapat menaruh atau merubah *value* pada *cell* tertentu

```
$sheet = $spreadsheet->getActiveSheet();
$sheet->setCellValue('A1', 'PhpSpreadsheet');
```

Fungsi diatas akan menulis 'PhpSpreadsheet' pada kolom 'A1'. *PhpSpreadsheet* memiliki *method* untuk merubah nilai dari *cell* tertentu dengan menggunakan `setCellValue('kolom', 'nilai')`.

### 2.2.4 Contoh Penulisan Spreadsheet ke Xls

*Phpspreadsheet* dapat melakukan *read and write* ke banyak format. Mulai dari *xls,xlsx,csv* dll. Pada kali ini format yang akan digunakan adalah format *xls*.

```
$writer = new \PhpOffice\PhpSpreadsheet\Writer\Xls($spreadsheet);
$writer->save("05featuredemo.xls");
```

Penggunaan fungsi *write* dari *phpspreadsheet* membutuhkan kelas `\PhpOffice\PhpSpreadsheet\Writer\Xls()`. Kelas lain yang dapat digunakan untuk *read&write* adalah `\PhpOffice\PhpSpreadsheet\IOFactory::createWriter($spreadsheet, "Xls")`

## 2.3 Bootstrap

*Bootstrap* adalah *framework* paling terkenal untuk membuat *site* yang *mobile-first* dan *responsive*[3]. Dapat dilihat pada tabel 2.2 dan tabel 2.3. Hampir semua *browser* pada *desktop* dan *mobile* dapat menjalankan *bootstrap*.

	Chrome	Firefox	Safari	Android & WebView	Microsoft Edge
Android	Supported	Supported	-	Android v5.0+ Supported	Supported
IOS	Supported	Supported	Supported	-	Supported
Windows 10 Mobile	-	-	-	-	Supported

Tabel 2.2: *Browser Mobile* yang mendukung *bootstrap*

	Chrome	Firefox	Internet Explorer	Microsoft Edge	Opera	Safari
Mac	Supported	Supported	-	Supported	Supported	Supported
Windows	Supported	Supported	Supported IE10+	Supported	Supported	Not supported

Tabel 2.3: *Browser Desktop* yang mendukung *bootstrap*

### 2.3.1 Pemasangan Bootstrap

Pemasangan dilakukan dengan cara mengunduh *compiled css and jss* yang disediakan oleh *bootstrap*.

```
<head>
  <link rel = "stylesheet" href = "css/bootstrap.css">
  <script src = js/bootstrap.js></script>
</head>
```

Pemasangan *bootstrap* memerlukan *import* file *css* dan *js* dari *compiled bootstrap* yang telah diunduh.

### 2.3.2 Container

Kelas **container** adalah kelas yang dibutuhkan jika ingin menggunakan *bootstrap* grid. Kelas **container** adalah kelas *responsive* dan *fixed-width*.

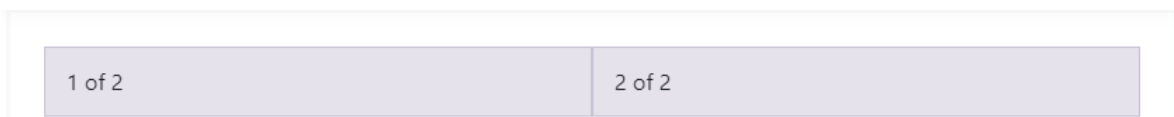
```
<div class="container"></div>
```

### 2.3.3 Grid

*Grid* pada *bootstrap* digunakan untuk mengatur *layout* dan *alignment*. *Row* adalah kelas *wrapper* dari untuk kolom. Kelas *col* akan secara otomatis membagi lebar horizontal dari layout menjadi sama dapat dilihat pada gambar 2.2. Sebagai contoh 4 *col* dalam satu *wrapper* akan membagi lebar masing masing menjadi 25%. Kelas *col* mempunyai jumlah maksimum sebanyak 12. Sehingga jika ingin membagi tiga kolom dengan lebar sama dapat menggunakan *col-4*.

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
</div>
```

Contoh gambar yang penggunaan kode diatas dapat dilihat pada gambar 2.2. Dapat dilihat bahwa kelas *col* membagi lebar horizontal menjadi sama rata. Penggunaan kelas *col-6* pada dua kolom juga akan menghasilkan gambar 2.2.



Gambar 2.2: Gambar *bootstrap-grid*

### 2.3.4 Table

Kelas **table** adalah kelas yang membantu pembuatan *table*. Tambahkan **.table** pada **<table>**. Komponen-komponen yang ada pada kelas **table**:

- **table-bordered**  
Tambahkan kelas ini jika agar semua sisi dari *cell* dan *table* memiliki *border*.
- **table-striped**  
Tambahkan kelas ini untuk tampilan *zebra-striping* di dalam **<tbody>**. Contoh penggunaan:

```
<table class="table table-bordered table-striped"></table>
```

- **table-responsive**

Gunakan kelas ini agar tampilan horizontal tabel menjadi *responsive* dan tidak *overflow*. Penggunaan **table-responsive** ditaruh pada *parent* dari *table*. Contoh penggunaan:

```
<div class="table-responsive">
  <table class="table">
    ...
  </table>
</div>
```

### 2.3.5 Pagination

*Pagination* dibuat menggunakan HTML *list* yang berada didalam **<nav>** untuk mengidentifikasi *pagination* sebagai bagian navigasi kepada pengguna. Bantuan pendukung yang disediakan oleh *bootstrap*:

- **aria-label**

**aria-label** adalah suatu elemen tambahan yang berguna untuk deskripsi dari navigasi tersebut.

- **active**

Kelas **active** digunakan untuk menandakan posisi halaman saat ini.

- **disabled**

Kelas **disabled** digunakan untuk menonaktifkan suatu *link*. Karena *CSS* untuk kelas tersebut tidak memperhitungkan navigasi dari keyboard sehingga harus ditambahkan **tabindex="-1"**.

- **page-item**

Kelas yang menandakan sebuah *page item* dan ditaruh pada *tag* **<li>**.

- **page-link**

Kelas yang menandakan sebuah *item link* dan ditaruh pada *tag* **<a>**.

Contoh penggunaan dari *pagination*:

```
<nav aria-label="example">
  <ul class="pagination">
    <li class="page-item disabled">
      <a class="page-link" href="#" tabindex="-1">1</a>
    </li>
    <li class="page-item">
      <a class="page-link" href="#">2</a>
    </li>
    <li class="page-item">
      <a class="page-link" href="#">3</a>
    </li>
  </ul>
</nav>
```

### 2.3.6 Modal

*Modal* dibuat dengan *html*, *css*, *javascript*. *Modal* akan merubah *scroll* pada **<body>** sehingga sasaran *scroll* hanya pada *modal*. *Modal* akan tertutup jika pengguna mengklik diluar *modal*. *Modal* pada *bootstrap* memiliki struktur yang sama, struktur *modal*:

```
<div class="modal" tabindex="-1" role="dialog">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      ...
    </div>
  </div>
</div>
```

Penggunaan *modal* dapat dibuat menjadi lebih rapi dengan menambahkan `modal-header`, `modal-body`, dan `modal-footer` didalam `modal-content`.

### 2.3.7 Form

*Bootstrap* menyediakan kelas yang dapat digunakan untuk menampilkan *form* yang konsisten pada *browser* ataupun *smartphone*. Kelas kelas yang disediakan dan digunakan pada *form*:

- **form-control**  
Kelas ini berfungsi untuk *style*, *focus state*, dan *sizing* untuk *form input* yang bersifat teks. Seperti: `<input>`, `<select>`, dan `textarea`.
- **readonly**  
Elemen ini dapat dimasukkan pada *input* untuk mencegah perubahan nilai dari *input*.

```
<input class="form-control" type="text" readonly>
```

- **form-group**  
*Form group* adalah kelas untuk menambah struktur pada *form* dengan mengelompokkan *label*, *controls*, dan *form validation message*.
- **Form grid**  
Penggunaan kelas *bootstrap form* juga dapat digabungkan dengan *grid* pada *bootstrap*, dengan menggunakan kelas *row*. seperti:

```
<form>
  <div class="row">
    <div class="col">
      <input type="text" class="form-control">
    </div>
    <div class="col">
      <input type="text" class="form-control">
    </div>
  </div>
</form>
```

Kode diatas akan menghasilkan *form* dengan model *grid* seperti pada gambar 2.2.

- **col-form-label**  
Kelas `col-form-label` digunakan pada tag `<label>` jika *label* dan *input* berada pada satu baris, sehingga antar *label* menjadi sejajar secara vertikal, seperti pada gambar 2.3.

Email	<input type="text" value="Email"/>
Password	<input type="password" value="Password"/>

Gambar 2.3: Gambar *label form* yang sejajar.

### 2.3.8 Navbar

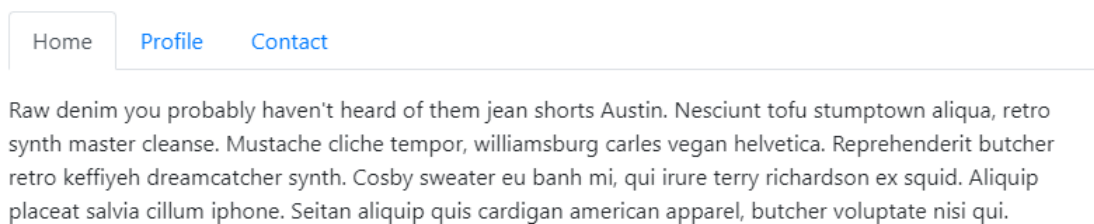
*Navbar* adalah sebuah komponen pada *bootstrap* yang mempermudah mengatur *layout* dari *navigation bar*. Komponen-komponen yang dibutuhkan *navbar*:

- **navbar-expand**  
`navbar-expand{-sm|-md|-lg|-xl}` dibutuhkan untuk *responsive collapsing* dan penggunaan kelas *color scheme*. Kelas ini ditaruh pada *nav tag*.
- **navbar-nav**  
Penggunaan kelas `navbar-nav` akan membuat *navigation* pada *navbars* akan menempati ruang horizontal sebanyak mungkin.
- *Color scheme*  
Penggunaan *color scheme* dapat dilakukan dengan `navbar-dark` atau `navbar-light`. Penggunaan kelas ini harus diikuti dengan `navbar-expand`.

### 2.3.9 Nav

*Nav* adalah komponen yang mengatur navigasi dengan memasukkan kelas `nav`. Kelas `nav` dibuat dengan *flexbox* dan menjadi dasar untuk membuat berbagai macam navigasi. Kelas `nav` dapat ditaruh pada tag `<nav>` atau `<ul>`. Komponen-komponen yang digunakan dari fungsi *nav*:

- **nav-tabs**  
Penggunaan kelas `nav` dan ditambahkan `nav-tabs` akan menghasilkan navigasi seperti pada gambar 2.4



Gambar 2.4: Navigasi dari `nav-tabs`

- **tab-content**  
Penggunaan kelas `tab-content` memerlukan *bootstrap.js*. Kelas ini adalah kelas pembungkus dari sekumpulan konten yang dibagi berdasarkan navigasi. Dalam gambar 2.4 adalah bagian yang berisi sekumpulan teks.
- **tab-pane**  
Kelas `tab-pane` berada di dalam kelas `tab-content` yang membungkus isi konten.



- `data-toggle`

Atribut `data-toggle` dipasang dengan nilai `data-toggle="tab"` agar navigasi dapat diklik dan melakukan navigasi.

- `nav-item`

Kelas ini digunakan pada *item* dari navigasi. Contoh penggunaan:

```
<li class="nav-item"></li>
```

- `nav-link`

Kelas ini digunakan pada *link* dari navigasi. Contoh penggunaan:

```
<a class="nav-link">link</a>
```

Struktur penggunaan kelas-kelas pada *nav* secara dasar:

```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#" >Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Profile</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Contact</a>
  </li>
</ul>
<div class="tab-content">
  <div class="tab-pane">...</div>
  <div class="tab-pane">...</div>
  <div class="tab-pane">...</div>
</div>
```

Penambahan atribut `data-toggle="tab"` pada *tag* `<a>`, memungkinkan navigasi dapat diklik. Kode diatas akan menghasilkan navigasi pada gambar 2.4.

## 2.4 JQuery



## BAB 3

### ANALISIS

#### 3.1 Analisis Program Usulan *Bluetape*

Pengerjaan untuk menganalisis *bluetape* dilakukan dengan melakukan survei dengan pengguna-pengguna aplikasi tersebut. Survei dilakukan melalui *google form* dan *link* dari survei tersebut dibagikan melalui *mailing list* dosen dan grup informatika UNPAR angkatan 2017. Pertanyaan pada survei tersebut cukup singkat. Pertanyaan hanya menanyakan nama dan tipe pengguna *bluetape* dan disediakan suatu text field untuk memasukkan *feedback* dari pengguna. Adapun *feedback* yang telah terkumpulkan tersedia pada tabel 3.1.

No	Deskripsi	Issue Number	Status	Analisis
1	Fitur "Expor ke XLS" pada menu EntriJadwalDosen menghasilkan file <i>corrupt</i>	#1	Akan diimplementasi	3.1.1
2	<i>Update google api / phpspreadsheet</i>	#2	Akan diimplementasi	3.1.2
3	Fitur <i>chart</i> pada ManajemenCetakTranskrip	#3	Akan diimplementasi	3.1.3
4	Fitur <i>chart</i> pada ManajemenPerubahanKuliah	#4	Akan diimplementasi	3.1.4
5	Mahasiswa dengan NPM baru tidak dapat login dan LihatJadwalDosen	#5	Akan diimplementasi	3.1.5
6	Kolom pada EntriJadwalDosen dan LihatJadwalDosen tidak seragam	#6	Akan diimplementasi	3.1.6
7	Fungsi Tab pada LihatJadwalDosen tidak berfungsi	#7	Akan diimplementasi	3.1.7
8	Pengelompokkan rekap perubahan jadwal pada mata kuliah yang sama	#8		3.1.8
9	Menyediakan forum Q&A pada <i>bluetape</i>	#9		3.1.9
10	<i>Scheduling</i> matakuliah pada googlemet/zoom	#10	Tidak diimplementasi	3.1.10
11	Mengubah atau membatalkan permohonan	#11	Akan diimplementasi	3.1.11
12	Menambahkan jam kuliah selesai di perubahan kelas	#12	Akan diimplementasi	3.1.12
13	Memperbaiki form dan link yang tidak aktif	#13	Tidak diimplementasi	3.1.13
14	Mengintegrasikan <i>bluetape</i> dengan SSO UNPAR	#14	Tidak diimplementasi	3.1.14
15	Menambah list permintaan pada <i>bluetape</i>	#15	Tidak diimplementasi	3.1.15
16	Pengajuan surat keterangan aktif kuliah secara online	#16		3.1.16
17	Notifikasi email untuk mahasiswa jika permintaan sudah diselesaikan	#17	Akan diimplementasi	3.1.17
18	Dapat melihat profil mahasiswa	#18	Tidak diimplementasi	3.1.18
19	Halaman histori dan request transkrip terpisah	#19		3.1.19
20	Fitur bahasa indonesia dan inggris	#20		3.1.20
21	Pagination tidak terstyle dengan baik	#22	Akan diimplementasi	3.1.21
22	Format Datetimepicker tidak konsisten	#23	Akan diimplementasi	3.1.22

Tabel 3.1: Tabel analisis kebutuhan pengguna perangkat *bluetape*

*feedback* dari pengguna Pada tabel 3.1 selanjutnya dibuat menjadi *issues* pada *github*<sup>1</sup>. *Feedback* dari pengguna selanjutnya akan dianalisa apakah akan diimplementasi berdasarkan jumlah fitur yang dibutuhkan dan waktu pengerjaan.

<sup>1</sup><https://github.com/stephenhadi/BlueTape/issues>

### 3.1.1 Fitur "Expore ke XLS" pada Menu EntriJadwalDosen Menghasilkan File *Corrupt*

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

Tambah Jadwal

Hari: 
Durasi: 
Label:

Jam Mulai: 
Jenis:

Daftar Jadwal

	Senin	Selasa	Rabu	Kamis	Jumat
7-8			Praktikum Pemrograman Berorientasi Objek (A)		Pemrograman Komputer
8-9					
9-10	Pemrograman Berorientasi Objek (A)	(dengan perjanjian)	Pemrograman Berorientasi Objek (B)	Layanan Berbasis Web	(dengan perjanjian)
10-11					
11-12	Pemrograman Berorientasi Objek (B)	Rapat Jurusan (jika ada)	Persiapan Materi Minggu Berikutnya	(dengan perjanjian)	
12-13					
13-14	Praktikum Pemrograman Komputer (oleh Erwin)				
14-15					
15-16	Keamanan Informasi				(dengan perjanjian)
16-17					

Terakhir diupdate pada : 26 Agustus 2019

Gambar 3.1: Antarmuka halaman EntriJadwalDosen

Fitur ini terdapat pada halaman **EntriJadwalDosen** yang dapat digunakan untuk memasukan atau merubah jadwal dosen. Dapat dilihat pada gambar 3.1 terdapat sebuah fitur untuk melakukan *expore ke xls*. Dari laporan pengguna fitur tersebut akan menghasilkan file *.xls* yang *corrupt*.

Fitur tersebut akan dikerjakan. Tetapi *error file corrupt* tidak dapat direproduksi.

### 3.1.2 Update google api/phpspreadsheet

*Bluetape* saat ini menggunakan *google/apiclient* 1.0 dan *phpexcel*. Salah satu permintaan dari pengguna/pengurus *bluetape* untuk merubah *google/apiclient* 1.0 menjadi *google/apiclient* 2.4.0 dan *phpexcel* menjadi *phpspreadsheet* v1.11.0.

Fitur ini akan dikerjakan dan solusi yang ditawarkan untuk *issue* ini adalah merubah isi *composer.json* dan melakukan *composer update*.

```
"require": {
    "google/apiclient": "^2.4.0",
    "phpoffice/phpspreadsheet": "^1.11.0"
}
```

**3.1.3 Fitur *chart* pada ManajemenCetakTranskrip****3.1.4 Fitur *chart* pada ManajemenPerubahanKuliah****3.1.5 Mahasiswa dengan NPM Baru Tidak Dapat *Login* dan LihatJadwalDosen****3.1.6 Kolom pada EntriJadwalDosen dan LihatJadwalDosen Tidak Seragam**

Pada halaman *EntriJadwalDosen* dan *LihatJadwalDosen* terdapat tabel seperti pada gambar 3.1. Ukuran tabel tersebut dapat berubah sesuai dengan ukuran tulisan di dalamnya sehingga kolom dengan tulisan yang banyak tidak seragam dengan kolom yang tulisannya sedikit.

Fitur ini akan diimplementasi. Solusi yang ditawarkan adalah dengan memasang *fixed width* pada setiap kolom.

**3.1.7 Fungsi tab pada LihatJadwalDosen tidak Berfungsi****3.1.8 Pengelompokkan Rekap Perubahan Jadwal pada Mata Kuliah yang Sama****3.1.9 Menyediakan Forum Q&A pada *Bluetape*****3.1.10 *Scheduling* Matakuliah pada *googlemeet/zoom***

*Issue* ini meminta agar *bluetape* melakukan *auto include* jadwal kuliah ke calendar, lalu *autocreate* jadwal kuliah berdasarkan jadwal tersebut di *zoom/meet* selama 1 semester, dan *Request* jadwal bertemu dari mahasiswa kepada dosen pembimbing, kemudian *auto schedule zoom/meet*.

Fitur ini tidak akan diimplementasi pada skripsi ini, karena permintaan tersebut masuk ke dalam ranah Biro Teknologi Informasi (BTI) dan saat ini *bluetape* tidak memiliki kemampuan untuk integrasi jadwal kuliah dari BTI.

**3.1.11 Mengubah atau Membatalkan Permohonan****3.1.12 Menambahkan Jam Kuliah Selesai di Perubahan Kelas****3.1.13 Memperbaiki *Form* dan *Link* yang Tidak Aktif**

Masukan ini meminta perbaikan terhadap fitur yang ada:

- *form* atau link seperti no surat, daftar ruang, dan peraturan terus diupdate.
- Dibuat lebih sistematis karena tampilannya tidak seragam.
- Dibuat lebih interaktif seperti peminjaman ruang.

Masukan ini tidak akan diimplementasi, karena sepertinya fitur tersebut di luar sistem *bluetape*. Seperti no surat, daftar ruang, dan peraturan.

**3.1.14 Mengintegrasikan *Bluetape* dengan SSO UNPAR**

Masukan ini meminta fitur tambahan agar *bluetape* terintegrasi secara penuh dengan *studentportal*<sup>2</sup>. Sehingga *bluetape* dapat diakses lewat *studentportal* dan *bluetape* dapat *login* menggunakan SSO.

Fitur masukan ini tidak akan diimplementasi karena diperlukan koordinasi yang kuat dengan BTI dan *Bluetape* untuk sementara ini tidak diperuntukan untuk pengguna diluar FTIS.

**3.1.15 Menambah *List* Permintaan pada *Bluetape***

Masukan ini meminta agar *bluetape* menambah *list* permintaan. Masukan ini tidak dikerjakan karena pengguna tersebut kurang spesifik dalam menjelaskan kebutuhannya.

---

<sup>2</sup><https://studentportal.unpar.ac.id/>

### 3.1.16 Pengajuan Surat Keterangan Aktif Kuliah Secara Online

Fitur tambahan ini meminta agar *bluetape* dapat melayani pengajuan surat keterangan aktif kuliah. Karena untuk mendapatkan surat keterangan aktif kuliah masih dilakukan secara manual.

Fitur ini tidak akan diimplementasi pada skripsi ini dikarenakan waktu yang terbatas, fitur yang cukup banyak, dan juga harus koordinasi intensif dengan tata usaha dan kepala sub bagian kemahasiswaan.

### 3.1.17 Notifikasi E-mail Terhadap Mahasiswa saat Permintaan Selesai

Fitur tambahan ini meminta saat proses pengajuan telah dijawab atau ditolak oleh tata usaha, maka mahasiswa mendapatkan notifikasi e-mail.

### 3.1.18 Dapat Melihat Profil Mahasiswa

Salah satu masukan dari *bluetape* adalah menginginkan seorang mahasiswa dapat melihat profil mahasiswa lain.

Fitur ini tidak diimplementasi pada skripsi ini karena untuk saat ini *bluetape* tidak terintegrasi dengan *student/lecturer* portal.

### 3.1.19 Halaman Histori dan Request Transkrip Terpisah

### 3.1.20 Fitur Bahasa Indonesia dan Inggris

Permintaan dari pengguna adalah fitur bahasa Indonesia dan bahasa Inggris. Setelah ditanyakan kejelasan dari permintaan ini, yang dimaksudkan pengguna adalah “Jika dibutuhkan untuk transkrip nilai dalam bahasa Inggris, dulu ada pilihannya tetapi hasil transkrip tetap bahasa Indonesia”.

Fitur ini tidak diimplementasi karena fitur transkrip nilai dalam bahasa Inggris diminta untuk dihapus oleh tata usaha.

### 3.1.21 Pagination tidak Ter-*style* dengan baik

### 3.1.22 Format Datetimepicker tidak Konsisten

## 3.2 Bluetape

*Bluetape* adalah aplikasi + framework untuk membuat urusan-urusan paper-based di FTIS UNPAR menjadi paperless.<sup>3</sup>

### 3.2.1 Instalasi

### 3.2.2 Struktur Bluetape

### 3.2.3 Pengaturan Dasar Bluetape

### 3.2.4 Database

---

<sup>3</sup><https://github.com/ftisunpar/BlueTape>

## DAFTAR REFERENSI

- [1] Version 3.1.11 (2019) *CodeIgniter 3*. British Columbia Institute of Technology. 3700 Willingdon Ave.
- [2] Version 1.14.1 (2019) *PhpSpreadsheet*. PHPOffice. The Internet.
- [3] Version 4.5 (2020) *Bootstrap*. Bootstrap. San Francisco, United States.





# 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

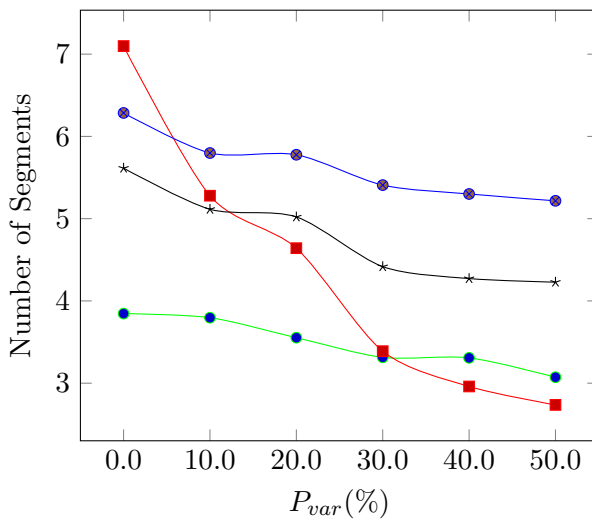
```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> 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 }
```



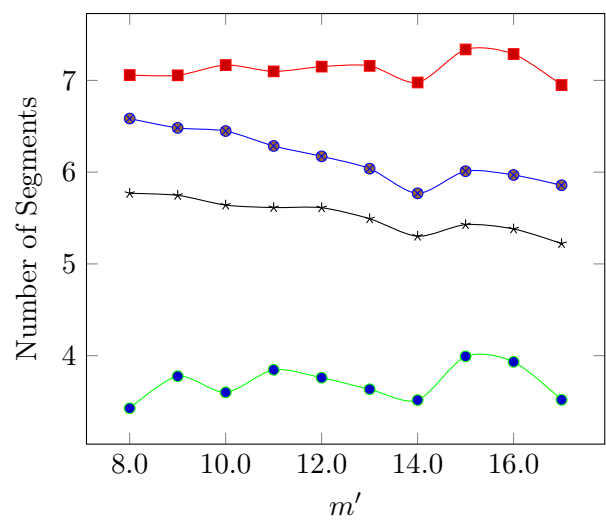
## LAMPIRAN B

### HASIL EKSPERIMEN

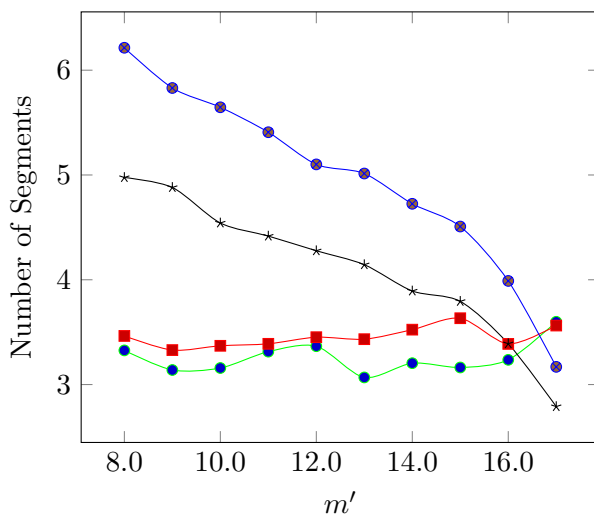
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



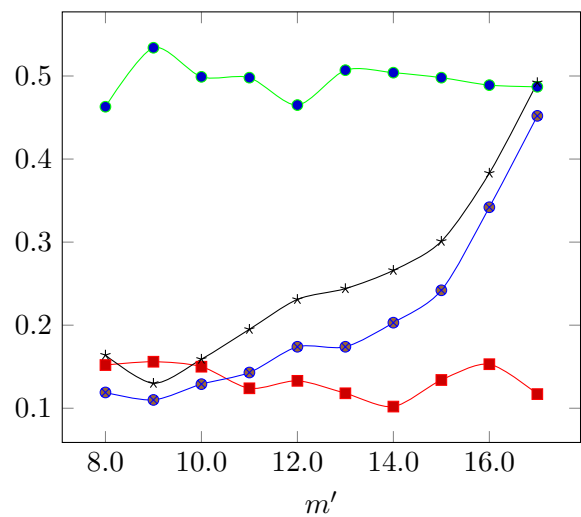
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4