

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, «tanggal» «bulan» 2020

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Pascal Alfadian Nugroho

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 «tanggal» «bulan» 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, «bulan» 2020

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 Code Igniter	5
2.1.1 Controller	5
2.1.2 Model	6
2.1.3 View	7
2.1.4 <i>Kelas</i> pada Code Igniter	7
2.2 Phpspreadsheet	10
2.2.1 <i>Kelas</i> pada Phpspreadsheet	10
2.2.2 Instalasi Phpspreadsheet	12
2.2.3 Contoh Penulisan Nilai pada <i>cell</i>	13
2.2.4 Contoh Penulisan Spreadsheet ke Xls	13
2.3 Bootstrap	13
2.3.1 Pemasangan Bootstrap	13
3 ANALISIS	15
3.1 Analisis Kebutuhan Perangkat Lunak <i>Bluetape</i>	15
3.2 Bluetape	16
3.2.1 Instalasi	16
3.2.2 Struktur Bluetape	16
3.2.3 Pengaturan Dasar Bluetape	16
3.2.4 Database	16
DAFTAR REFERENSI	17
A KODE PROGRAM	19
B HASIL EKSPERIMEN	21

DAFTAR GAMBAR

2.1	Flowchart MVC	5
B.1	Hasil 1	21
B.2	Hasil 2	21
B.3	Hasil 3	21
B.4	Hasil 4	21

DAFTAR TABEL

2.1	Tabel format yang didukung oleh phpspreadsheet	10
2.2	<i>Browser Mobile</i> yang mendukung <i>bootstrap</i>	13
2.3	<i>Browser Desktop</i> yang mendukung <i>bootstrap</i>	13
3.1	Tabel analisis kebutuhan pengguna perangkat <i>bluetape</i>	15

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*¹ adalah aplikasi web yang dibuat oleh dosen dan mahasiswa informatika. Aplikasi ini dibuat menggunakan *Hypertext Preprocessor* atau lebih dikenal dengan *PHP*². *Database management system* atau DBMS yang digunakan adalah *MYSQL*³. *Bluetape* menggunakan *framework Bootstrap*⁴ dan *Codeigniter*⁵. *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 otomatisasi 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*

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

²<https://www.php.net/>

³<https://www.mysql.com/>

⁴<https://getbootstrap.com/>

⁵<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.

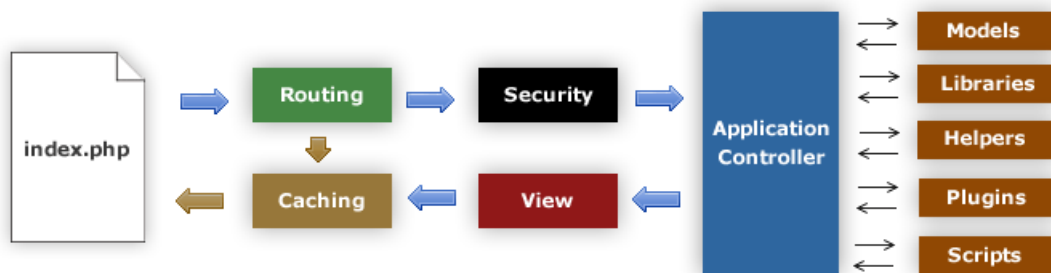
BAB 2

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

```
<?php echo $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:

- `$model(mixed)`: Nama *model*.
- `$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()`

2.1.4.6 Session

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 tersedia 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])**
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.

```
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.

BAB 3

ANALISIS

3.1 Analisis Kebutuhan Perangkat Lunak *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
1	Memperbaiki ekspor ke xls pada EntriJadwalDosen	#1	Akan diimplementasi
2	Memperbaharui <i>google api</i> dan <i>phpspreadsheet</i>	#2	Akan diimplementasi
3	Tampilan <i>chart</i> pada ManajemenCetakTranskrip	#3	Akan diimplementasi
4	Tampilan <i>chart</i> pada ManajemenPerubahanKuliah	#4	Akan diimplementasi
5	Mahasiswa dengan format NPM baru tidak dapat login dan LihatJadwalDosen	#5	Akan diimplementasi
6	Kolom pada EntriJadwalDosen dan LihatJadwalDosen tidak seragam	#6	Akan diimplementasi
7	Fungsi Tab pada LihatJadwalDosen tidak berfungsi	#7	Akan diimplementasi
8	Pengelompokkan rekap perubahan jadwal pada mata kuliah yang sama	#8	
9	Forum Q&A pada <i>bluetape</i>	#9	
10	<i>Scheduling</i> matakuliah pada googlemet/zoom	#10	Tidak diimplementasi
11	Mengubah atau membatalkan permohonan	#11	Akan diimplementasi
12	Menambahkan jam kuliah selesai di perubahan kelas	#12	Akan diimplementasi
13	Memperbaiki form dan link yang tidak aktif	#13	Tidak diimplementasi
14	Mengintegrasikan <i>bluetape</i> dengan SSO UNPAR	#14	Tidak diimplementasi
15	Menambah list permintaan pada <i>bluetape</i>	#15	Tidak diimplementasi
16	Pengajuan surat keterangan aktif kuliah secara online	#16	
17	Notifikasi email untuk mahasiswa jika permintaan sudah diselesaikan	#17	Akan diimplementasi
18	Dapat melihat profil mahasiswa	#18	Tidak diimplementasi
19	Halaman histori dan request transkrip terpisah	#19	
20	Fitur bahasa indonesia dan inggris	#20	
21	Pagination tidak terstyle dengan baik	#22	Akan diimplementasi
22	Format Datetimepicker tidak konsisten	#23	Akan diimplementasi

Tabel 3.1: Tabel analisis kebutuhan pengguna perangkat *bluetape*

3.2 Bluetape

Bluetape adalah aplikasi + framework untuk membuat urusan-urusan paper-based di FTIS UNPAR menjadi paperless.¹

3.2.1 Instalasi

3.2.2 Struktur Bluetape

3.2.3 Pengaturan Dasar Bluetape

3.2.4 Database

¹<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