

«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 Codeigniter	5
2.1.1 Model-View-Controller	5
2.1.2 Controller	5
2.1.3 View	6
2.1.4 Model	6
2.1.5 Flow Chart Codeigniter	7
2.2 Zurb Foundation	7
2.3 Google OAuth 2.0	8
2.4 PHPExcel	9
DAFTAR REFERENSI	11
A KODE PROGRAM	13
B HASIL EKSPERIMEN	15

DAFTAR GAMBAR

1.1	Template jadwal dosen	1
2.1	Flow Chart Codeigniter	7
2.2	Skenario OAuth menggunakan token akses	8
2.3	Arsitektur PHPExcel	9
B.1	Hasil 1	15
B.2	Hasil 2	15
B.3	Hasil 3	15
B.4	Hasil 4	15

DAFTAR TABEL

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 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

2.1.1 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.

2.1.2 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.

2.1.2.1 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 dimasukkan ke dalam method sebagai parameter.

2.1.2.2 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'.

2.1.3 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.

2.1.4 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.

2.1.4.1 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.

2.1.4.2 Load Model

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

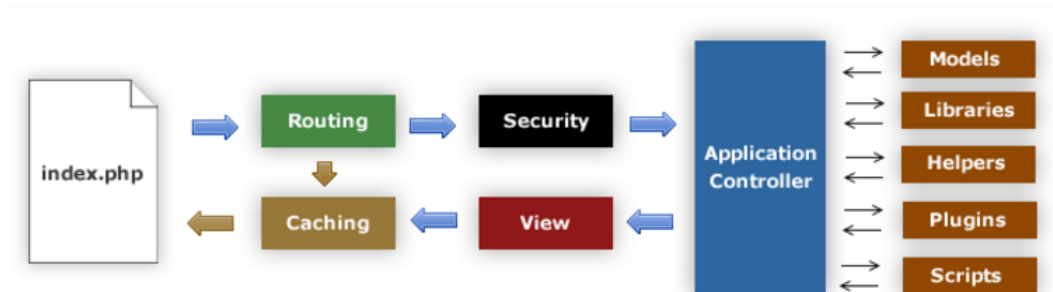
```
$this->model->model( 'model_name' );
```

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 digunakan :

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

2.1.5 Flow Chart Codeigniter

Gambar dibawah mengilustrasikan aliran data dalam sistem : ¹



Gambar 2.1: Flow Chart Codeigniter

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.
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 dimasukan ke dalam cache terlebih dahulu sehingga pada permintaan selanjutnya tampilan tersebut dapat diakses lebih cepat.

2.2 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.

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

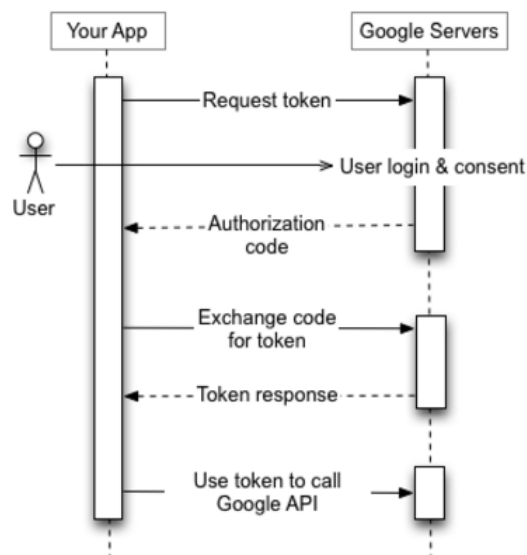
²<http://foundation.zurb.com/sites/docs/v/5.5.3/>

- Navigasi untuk mempermudah pengunjung aplikasi dalam menggunakan aplikasinya.
- Plugins JavaScript untuk mempermudah *developer* dalam membuat tampilan aplikasinya.

2.3 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

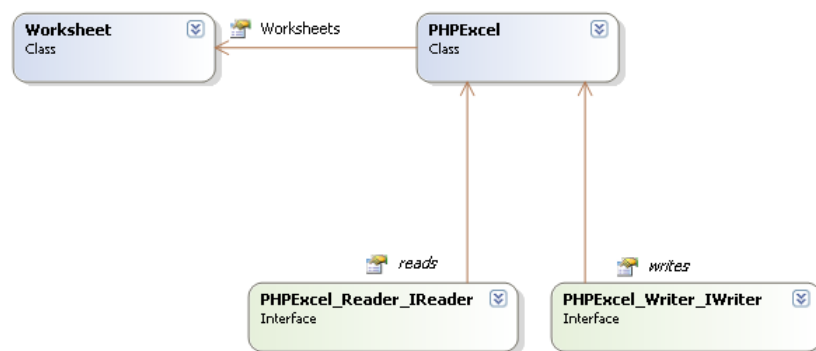
³<https://developers.google.com/identity/protocols/OAuth2>

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.

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.

2.4 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 perangkat lunak sebagai berikut :

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

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