

«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	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	3
2.1 BlueTape	3
2.2 CodeIgniter	3
2.2.1 Application Flow Chart	3
2.2.2 CodeIgniter URLs	4
2.2.3 Model-View-Controller	4
2.2.4 Model	4
2.2.5 View	5
2.2.6 Controller	8
2.3 Zurb Foundation 6	9
2.3.1 Struktur File	9
2.3.2 Sistem Grid pada Foundation	9
2.3.3 <i>Navigation</i> dan <i>Media Attributes</i>	10
2.3.4 Komponen CSS	12
2.3.5 Komponen JavaScript	15
2.4 Bootstrap 4	16
2.5 Template Skripsi FTIS UNPAR	16
2.5.1 Tabel	17
2.5.2 Kutipan	17
2.5.3 Gambar	18
A KODE PROGRAM	21
B HASIL EKSPERIMEN	23

DAFTAR GAMBAR

2.1	Flow Chart Aplikasi CodeIgniter	4
2.2	Struktur File Zurb Foundation	9
2.3	Grid pada Zurb Foundation	10
2.4	<i>Basic Navigation Menu</i> pada Foundation	11
2.5	Menu <i>align to right in Foundation</i>	11
2.6	Menu <i>align to center in Foundation</i>	11
2.7	Menu <i>active state menu in Foundation</i>	12
2.8	Menu <i>active state menu in Foundation</i>	12
2.9	Basic Button pada Foundation	13
2.10	Coloring Button pada Foundation	13
2.11	Basic Table pada Foundation	14
2.12	Text Input pada Foundation	15
2.13	Grid pada Zurb Foundation	16
2.14	Gambar <i>Serpentes</i> dalam format png	19
2.15	Ular kecil	19
2.16	<i>Serpentes</i> betina	20
B.1	Hasil 1	23
B.2	Hasil 2	23
B.3	Hasil 3	23
B.4	Hasil 4	23

DAFTAR TABEL

2.1	Tabel contoh	17
2.2	Tabel bewarna(1)	17
2.3	Tabel bewarna(2)	17

BAB 1

PENDAHULUAN

1.1 Latar Belakang

BlueTape merupakan aplikasi berbasis *web* yang berfungsi mengolah beberapa kebutuhan administrasi fakultas secara *paperless* yang digunakan dalam lingkungan FTIS UNPAR. Aplikasi ini mempunyai fitur untuk manajemen transkrip nilai, perubahan kuliah dan jadwal dosen. *Framework* yang digunakan dalam aplikasi BlueTape ada dua yaitu *Codeigniter* dan *Zurb Foundation*.

Foundation adalah kerangka kerja atau *Framework* untuk semua perangkat, media, dan semua aksesibilitas. *Foundation* adalah bagian dari *front-end framework* yang responsif dan membuatnya mudah untuk merancang situs web, aplikasi, dan email yang responsif dan indah yang terlihat baik di perangkat mana pun. *Foundation* bersifat semantik, mudah dibaca, fleksibel, dan sepenuhnya *customizable*. [?].

Sejak *Bootstrap* diluncurkan pada Agustus 2011, *framework* ini mulai populer. *Bootstrap* telah berkembang sepenuhnya menjadi proyek yang digerakkan oleh CSS untuk menggunakan sejumlah plugin JavaScript dan ikon yang sejalan dengan *forms* dan *buttons*. Pada dasarnya, ini memungkinkan untuk mendesain web yang responsif dan memiliki fitur grid 12-kolom, 940px-lebar yang kuat. Salah satu yang menarik adalah *build tool* di situs web *Bootstrap*, di mana *developer* dapat menyesuaikan pembangunan sesuai dengan kebutuhan *developer*, seperti memilih fitur CSS dan JavaScript yang ingin disertakan dalam situs. [?]

Pada skripsi ini akan dirubah keseluruhan antarmuka untuk setiap modul yang ada di dalam aplikasi BlueTape menggunakan *framework Bootstrap 4*. Saat ini, setiap view menggunakan template yang menampilkan nama *module*, menu navigasi, dan *flash message* (bila diperlukan).

1.2 Rumusan Masalah

Rumusan masalah yang akan dibahas dalam penelitian ini:

1. Bagaimana merubah *template* manajemen cetak transkrip, manajemen perubahan kuliah dan manajemen jadwal dosen dari framework **Zurb Foundation** ke **Bootstrap 4**
2. Bagaimana mengimplentasikan *plugin* yang tersedia di dalam *Bootstrap 4*.

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini :

1. Merubah *template* cetak transkrip nilai, *template* manajemen cetak transkrip, *template* perubahan kuliah, *module* manajemen perubahan kuliah, *modul* entri jadwal dosen dan *module* lihat jadwal dosen dengan *framework Bootstrap 4*.
2. Mengimplentasikan *plugin* yang tersedia dalam *library Bootstrap 4*.

1.4 Batasan Masalah

Dalam penelitian ini ditetapkan batasan-batasan masalah sebagai berikut.

1. Aplikasi ini tidak merubah struktur database dan file yang berisi fungsi-fungsi CRUD.
2. Aplikasi ini tidak menambah tampilan baru, hanya merubah penggunaan framework *Zurb Foundation* sesuai dengan tampilan yang sudah ada menggunakan *Bootstrap 4*

1.5 Metodologi

Metode penelitian yang digunakan dalam skripsi ini adalah :

1. Studi literatur memahami mengenai :
 - (a) *framework* CodeIgniter
 - (b) *framework* Bootstrap 4
 - (c) *framework* Zurb Foundation dan plugin - plugin nya.
2. Membangun antarmuka sesuai tampilan website BlueTape. Proses pembuatan antarmuka dibagi menjadi 3 tahap :
 - (a) Analisis tampilan antarmuka website BlueTape
 - (b) Perancangan tampilan antarmuka
 - (c) Implementasi

1.6 Sistematika Pembahasan

Untuk penulisan skripsi ini akan dibagikan dalam .. bab sebagai berikut :

Bab Pendahuluan

Bab 1 menjelaskan mengenai latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.

Bab Landasan Teori

Bab 2 berisi dasar-dasar teori pembuatan antarmuka BlueTape. Dasar-dasar teori yang digunakan diantaranya adalah pemrograman PHP, *framework Codeigniter*, *framework Zurb Foundation*, *framework Bootstrap 4*.

Bab Analisis

Bab 3 berisi analisis antarmuka yang sudah ada dan analisis antarmuka usulan.

Bab Perancangan antarmuka

Bab 4 program dan perancangan kelas-kelas program.

Bab Implementasi

Bab 5 membahas mengenai pembuatan template utama aplikasi BlueTape yaitu , pembuatan menu aplikasi dan hasil eksekusi tampilan aplikasi.

Bab Kesimpulan dan saran.

Bab 6 berisi kesimpulan setelah mengerjakan skripsi ini dan saran yang diberikan.

BAB 2

LANDASAN TEORI

2.1 BlueTape

BlueTape adalah aplikasi+framework untuk membuat urusan-urusan paper-based di FTIS UNPAR menjadi paperless. Aplikasi ini berbasis web dengan memanfaatkan CodeIgniter + ZURB Foundation.

Fitur-fitur:

- Framework disediakan untuk menambah layanan baru. Menu sudah disediakan, developer tinggal menambah dalam bentuk modul (lihat CONTRIBUTING.md)
- Layanan OAuth ke Google, memungkinkan autentikasi pengguna dan menentukan hak akses yang bisa dilihat dari alamat email pengguna, misalnya: membatasi akses ke mahasiswa Informatika ke email 73.....@student.unpar.ac.id, akses ke mahasiswa FTIS ke 7[0123].....@student.unpar.ac.id. Untuk staf TU / dosen bisa juga dengan mendaftarkan email staf / dosen.

Saat ini tersedia layanan:

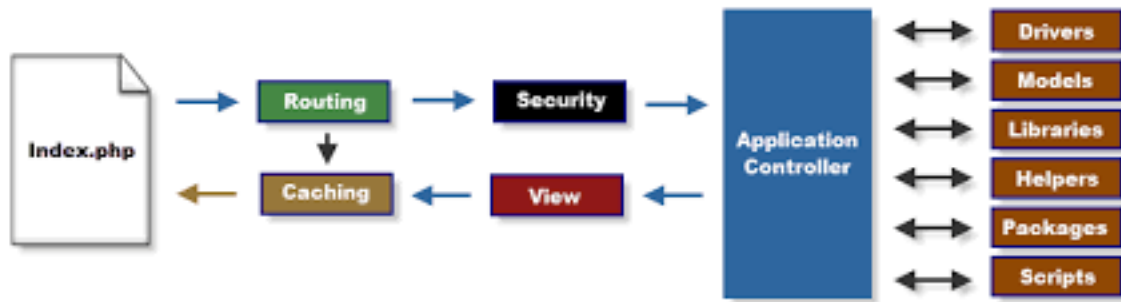
-
-

2.2 CodeIgniter

Codeigniter adalah sebuah kerangka kerja untuk pengembangan aplikasi - sebuah alat - bagi masyarakat yang ingin membangun website menggunakan PHP. Bertujuan agar proyek yang sedang dikembangkan lebih cepat daripada Anda menulis kode dari awal / *scratch*, dengan menyediakan seperangkat *libraries* untuk tugas yang umumnya digunakan, semudah sebuah *interface* dan struktur logika untuk mengakses *libraries* tersebut. Codeigniter memungkinkan untuk fokus pada proyek Anda dengan mengurangi kode yang dibutuhkan untuk tugas tertentu. [?]

2.2.1 Application Flow Chart

Gambar berikut mengilustrasikan bagaimana alur data pada sistem :



Gambar 2.1: Flow Chart Aplikasi CodeIgniter

1. index.php bertindak sebagai *front controller*, menginisiasi *base resources* yang dibutuhkan untuk menjalankan CodeIgniter.
2. Router akan memeriksa permintaan HTTP untuk menetapkan hal apa yang harus dilakukan dengan permintaan tersebut.
3. Apabila terdapat *cache*, maka *cache* tersebut akan dikirimkan langsung ke browser, dengan melewati sistem eksekusi normal.
4. Keamanan. Sebelum *controller* aplikasi dimuat, *HTTP request* dan *user* mana pun yang mengirimkan data diseleksi dahulu untuk keamanan.
5. Controller memuat *model*, *core libraries*, *helpers*, dan *resources* yang dibutuhkan untuk proses *request* yang spesifik.
6. *View* yang telah selesai dirender kemudian dikirim ke *web browser* untuk dilihat. Jika *caching* diaktifkan, tampilan dicache terlebih dahulu sehingga pada permintaan selanjutnya dapat dilayani.

2.2.2 CodeIgniter URLs

Codeigniter menggunakan pendekatan **berbasis-segmen**:^[?]

```
example.com/news/article/my_article
```

Url Segments

2.2.3 Model-View-Controller

Codeigniter berdasarkan pola pembangunan Model-View-Controller. MVC adalah pendekatan perangkat lunak yang memisahkan aplikasi logik dari presentasi. Dalam prakteknya, memungkinkan untuk *web pages* Anda berisi *scripting* yang sedikit karena presentasi terpisah dari skrip PHP.

2.2.4 Model

Model merepresentasikan struktur data Anda. Biasanya kelas *model* akan berisi fungsi yang membantu untuk *retrieve*, *insert*, dan *update* informasi di database.

Dalam Codeigniter *models* merupakan opsi yang tersedia untuk mereka yang ingin lebih menggunakan sebuah pendekatan tradisional MVC.^[?]

Anatomi Model

Kelas model akan disimpan di direktori **application/models/directory**. Kelas ini dapat bersarang didalam *sub-directories* jika Anda menginginkan tipe organisasi seperti ini. [?]

Prototipe dasar dari sebuah model kelas :

```
<?php
class Model_name extends CI_Model {

}
```

Nama file juga harus sama dengan nama kelas. Sehingga apabila kita ada kelas **User_model** maka file Anda akan seperti ini.

```
application/models/User_model.php
```

Loading a Model

Model Anda biasanya akan dimuat dan dipanggil didalam metode *controller* Anda. Untuk memuat sebuah model anda akan menggunakan metode berikut:[?]

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

Koneksi ke Database

Apabila model sudah dimuat, model tersebut tidak terhubung secara langsung ke database. Dengan cara secara manual mengatur konektifitas database melalui parameter ketiga:[?]

```
$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);
```

2.2.5 View

View adalah informasi yang sedang dilihat oleh *user*. Sebuah *View* normalnya menjadi sebuah halaman web, namun dalam CodeIgniter, sebuah *view* dapat menjadi sebuah *page fragment* seperti *header* atau *footer*. Dapat juga menjadi halaman RSS, atau tipe apapun dari "page".

Views tidak pernah dipanggil secara langsung, harus dimuat dalam sebuah *controller*. Ingat bahwa dalam *MVC framework*, *controller* bertanggung jawab untuk mengambil *view* tertentu.[?]

Membuat sebuah View

Dengan menggunakan *text editor*, buat sebuah file yang memanggil `blogview.php`, dan isi dengan kode berikut:[?]

```
<head>
    <title>My Blog</title>
</head>
<body>
    <h1>Welcome to my Blog!</h1>
</body>
</html>
```

Kemudian simpan file tersebut di `application/views/` directory.

Loading sebuah View

View dapat dimuat dengan membuat file *view* dengan syntax berikut:

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

Dimana *name* adalah nama dari file *view*.

Lalu, buka file *controller* yang dibuat sebelumnya bernama `Blog.php`, dan pindahkan *echo statement* dengan *view loading method*.

```
<?php
class Blog extends CI_Controller {

    public function index()
    {
        $this->load->view('blogview');
    }
}
```

Memuat Beberapa View

Codeigniter akan menangani beberapa panggilan dari dalam controller dengan syntax `$this->load->view()`. Apabila ada lebih dari satu panggilan yang terjadi, maka *views* akan dilampirkan secara bersamaan. Berikut ini kode yang digunakan apabila pengembang web ingin mempunyai sebuah *header view*, sebuah *menu view*, sebuah *content view*, dan sebuah *footer view*. [?]

```
<?php
class Page extends CI_Controller {

    public function index()
    {
        $data['page_title'] = 'Your title';
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('content', $data);
        $this->load->view('footer');
```

```

        }
    }
}

```

Menyimpan Views didalam *Sub-directories*

View files dapat disimpan didalam *sub-directories* dengan menyertakan nama direktori yang memuat *view*.

```
$this->load->view('directory_name/file_name');
```

Menambahkan data dinamis ke View

Data yang dikirim dari controller menuju view dalam bentuk **array** atau objek akan dilampirkan dalam parameter kedua dalam metode loading view. [?] Berikut ini penggunaan dengan array:

```

$data = array(
    'title' => 'My Title',
    'heading' => 'My Heading',
    'message' => 'My Message'
);

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

```

Kemudian, penggunaan dengan objek:

```

$data = new Someclass();
$this->load->view('blogview', $data);

```

Sehingga apabila dimasukan ke controller, kode yang ditambahkan adalah:

```

<?php
class Blog extends CI_Controller {

    public function index()
    {
        $data['title'] = "My Real Title";
        $data['heading'] = "My Real Heading";

        $this->load->view('blogview', $data);
    }
}

```

Untuk mengaksesnya dalam file HTML maka dapat digunakan syntax php

```

<html>
<head>
    <title><?php echo $title;?></title>
</head>
<body>
    <h1><?php echo $heading;?></h1>

```

```
</body>
</html>
```

2.2.6 Controller

Controller bertindak sebagai sebuah penengah antara Model, View dan *resources* lain yang dibutuhkan untuk proses *HTTP requests* dan menghasilkan sebuah halaman web.

Sebuah *controller* secara sederhana merupakan sebuah file yang dinamakan sehingga dapat dikaitkan dengan URL.[?] Misalnya untuk URL ini:

```
<?php
example.com/index.php/blog/
```

Dalam contoh diatas, *Codeigniter* berusaha menemukan *controller* bernama Blog.php dan memuatnya. Ketika sebuah nama *controller* sesuai dengan *first segment* dari sebuah URL, maka URL akan memuatnya.[?]

Kode berikut merupakan contoh dari *controller* sederhana.

```
<?php
class Blog extends CI_Controller {

    public function index()
    {
        echo 'Hello World!';
    }
}
```

Method

Dalam sebuah kelas *controller* akan terdapat beberapa method, untuk memanggil fungsi didalamnya maka dapat mengisi segmen kedua dari sebuah url. [?]

```
<?php
class Blog extends CI_Controller {

    public function index()
    {
        echo 'Hello World!';
    }

    public function comments()
    {
        echo 'Look at this!';
    }
}
```

Pemanggilan method index dapat secara otomatis dilakukan apabila segmen kedua kosong. Cara lain untuk menjalankan method comments() dapat dilakukan dengan:

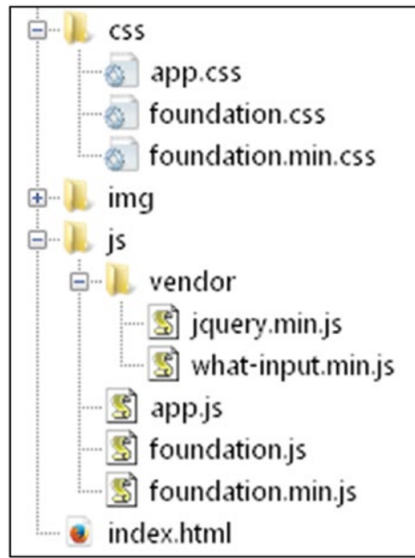
```
example.com/index.php/blog/index/
```

Kemudian untuk memuat method comment dapat dituliskan sebagai berikut:

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

2.3 Zurb Foundation 6

2.3.1 Struktur File



Gambar 2.2: Struktur File Zurb Foundation

Framework Foundation terdiri dari 3 folder utama: Folder **css** terdiri dari semua *CSS Style* yang digunakan dalam Foundation 6. Didalam folder terdapat versi yang diperkecil **foundation.min.css** atau versi yang tidak dikompresi **foundation.css**. Seluruh modifikasi *stylesheets* ditempatkan pada folder ini agar lebih terstruktur. Folder **img** tempat meletakkan semua gambar untuk proyek web. Folder **js** terdiri dari semua file Javascript yang sudah ditentukan sebelumnya.[?]

2.3.2 Sistem Grid pada Foundation

Penggunaan grid pada Foundation dapat dilakukan dengan menambahkan sebuah elemen dengan sebuah kelas **.row** sehingga akan membuat sebuah blok horizontal yang berisi kolom vertikal. Kemudian tambahkan kelas **.column** pada baris tersebut, serta tentukan masing-masing kolom dengan kelas **.small-**, **.medium-** dan **.large-**. **Foundation** adalah *mobile-first*. Kode yang dihasilkan dibuat untuk layar kecil terlebih dahulu, dan layar besar akan mewarisi *style* dari kode tersebut. [?]

```

<div class="row">
  <div class="columns small-2 large-4"><!-- ... --></div>
  <div class="columns small-4 large-4"><!-- ... --></div>
  <div class="columns small-6 large-4"><!-- ... --></div>
</div>
<div class="row">
  <div class="columns large-3"><!-- ... --></div>

```

```

<div class="columns large-6"><!-- ... --></div>
<div class="columns large-3"><!-- ... --></div>
</div>
<div class="row">
  <div class="columns small-6 large-2"><!-- ... --></div>
  <div class="columns small-6 large-8"><!-- ... --></div>
  <div class="columns small-12 large-2"><!-- ... --></div>
</div>
<div class="row">
  <div class="columns small-3"><!-- ... --></div>
  <div class="columns small-9"><!-- ... --></div>
</div>
<div class="row">
  <div class="columns large-4"><!-- ... --></div>
  <div class="columns large-8"><!-- ... --></div>
</div>
<div class="row">
  <div class="columns small-6 large-5"><!-- ... --></div>
  <div class="columns small-6 large-7"><!-- ... --></div>
</div>
<div class="row">
  <div class="columns large-6"><!-- ... --></div>
  <div class="columns large-6"><!-- ... --></div>
</div>

```

4	4	4
3	6	3
2	8	2
3	9	
4	8	
5	7	
6	6	

Gambar 2.3: Grid pada Zurb Foundation

2.3.3 *Navigation dan Media Attributes*

Komponen menu yang fleksibel pada Foundation membuat pembangunan navigasi secara umum lebih mudah karena semua pola memiliki markup yang sama.

Basic Menu

Semua versi menu terdiri dari sebuah `` yang diisi oleh beberapa elemen ``. Secara default, menu akan berorientasi horizontal.[?]

Berikut ini contoh penggunaan kode navigasi pada menu:


```
<ul class="menu">
  <li><a href="#">One</a></li>
  <li><a href="#">Two</a></li>
  <li><a href="#">Three</a></li>
  <li><a href="#">Four</a></li>
</ul>
```

One Two Three Four

Gambar 2.4: *Basic Navigation Menu* pada Foundation

Item Alignment

Secara default, setiap item dalam menu sejajar ke arah kiri. Menu dapat diubah sejajar ke arah kanan dengan menggunakan kelas `.align-right` atau ke arah tengah dengan menambahkan kelas `.align-center` ke kelas `.menu` [?]

```
<ul class="menu align-right">
  <li><a href="#">One</a></li>
  <li><a href="#">Two</a></li>
  <li><a href="#">Three</a></li>
  <li><a href="#">Four</a></li>
</ul>
```

One Two Three Four

Gambar 2.5: Menu *align to right in Foundation*

```
<ul class="menu align-center">
  <li><a href="#">One</a></li>
  <li><a href="#">Two</a></li>
  <li><a href="#">Three</a></li>
  <li><a href="#">Four</a></li>
</ul>
```

One Two Three Four

Gambar 2.6: Menu *align to center in Foundation*

Active State

Kelas `.is-active` dapat ditambahkan ke dalam tag `` untuk membuat sebuah **active state**. Active state bisa diatur dengan menandai halaman aktif secara dinamis dengan Javascript atau menerapkannya pada *server-side*. [?]

```
<ul class="menu">
  <li class="is-active"><a>Home</a></li>
  <li><a>About</a></li>
```

```
<li><a>Nachos</a></li>
</ul>
```

Home About Nachos

Gambar 2.7: Menu *active state menu in Foundation*

Text

Karena *padding* item menu digunakan pada tag `<a>`, maka saat menerapkan item yang berisi teks saja, teks tersebut akan tidak selaras. Untuk menyiasatinya, maka dapat menggunakan kelas `.menu-text` ke `` dengan menyertakan teks tanpa link.[?]

```
<ul class="menu">
  <li class="menu-text">Site Title</li>
  <li><a href="#">One</a></li>
  <li><a href="#">Two</a></li>
  <li><a href="#">Three</a></li>
</ul>
```

Site Title One Two Three

Gambar 2.8: Menu *active state menu in Foundation*

2.3.4 Komponen CSS

Alasan penggunaan *CSS framework* pada Foundation adalah komponen bawaan antarmuka pengguna. Dengan sistem grid dan komponennya, cukup mudah bagi pengembang untuk mengembangkan situs web yang rumit. [?] Beberapa komponen tersebut adalah ***button***, ***tables*** dan ***forms***. [?]

Button

Basic button dapat dibuat dengan markup minimal. Karena tombol dapat digunakan untuk banyak tujuan, penting untuk menggunakan tag yang tepat.

- Gunakan tag `<a>` jika tombolnya adalah tautan ke halaman lain, atau tautan ke jangkar di dalam halaman. Umumnya jangkar tidak memerlukan JavaScript untuk berfungsi.
- Gunakan tag `<button>` jika tombol melakukan tindakan yang mengubah sesuatu pada halaman saat ini. Elemen `<button>` hampir selalu membutuhkan JavaScript agar berfungsi. [?]

[?]

```
<!-- Anchors (links) -->
<a href="about.html" class="button">Learn More</a>
<a href="#features" class="button">View All Features</a>

<!-- Buttons (actions) -->
<button type="button" class="success button">Save</button>
<button type="button" class="alert button">Delete</button>
```



Gambar 2.9: Basic Button pada Foundation

Warna dapat ditambahkan untuk memberikan *buttons* arti yang bermakna.

```
<a class="button primary" href="#">Primary</a>
<a class="button secondary" href="#">Secondary</a>
<a class="button success" href="#">Success</a>
<a class="button alert" href="#">Alert</a>
<a class="button warning" href="#">Warning</a>
```



Gambar 2.10: Coloring Button pada Foundation

Tables

```
<table>
  <thead>
    <tr>
      <th width="200">Table Header</th>
      <th>Table Header</th>
      <th width="150">Table Header</th>
      <th width="150">Table Header</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Content Goes Here</td>
      <td>This is longer content Donec id elit non mi porta gravida at euismod</td>
      <td>Content Goes Here</td>
      <td>Content Goes Here</td>
    </tr>
    <tr>
      <td>Content Goes Here</td>
      <td>This is longer Content Goes Here Donec id elit non mi porta gravida at euismod</td>
      <td>Content Goes Here</td>
      <td>Content Goes Here</td>
    </tr>
    <tr>
      <td>Content Goes Here</td>
      <td>This is longer Content Goes Here Donec id elit non mi porta gravida at euismod</td>
      <td>Content Goes Here</td>
      <td>Content Goes Here</td>
    </tr>
  </tbody>
</table>
```

Table Header	Table Header	Table Header	Table Header
Content Goes Here	This is longer content Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here
Content Goes Here	This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here
Content Goes Here	This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.	Content Goes Here	Content Goes Here

Gambar 2.11: Basic Table pada Foundation

Hover State Penggunaan Hover State dengan menambahkan kelas `.hover` untuk sedikit menggelapkan baris tabel.

```
<table class="hover">
</table>
```

Striped Secara default, tabel akan memiliki baris yang bergaris. Untuk menghapus garis-garis tersebut dapat menggunakan kelas `.unstriped` atau dengan mengubah `$table-is-striped` ke `false` untuk menghapus semua strip pada seluruh tabel. Gunakan pula kelas `.striped` untuk menambahkan strip.[?]

Forms

Pembuatan sebuah *form* di *Foundation* didesain mudah namun fleksibel. Forms dibuat dengan kombinasi standar dari elemen `form`, serta *grid rows* dan *columns* atau *cells*. [?]

Text Inputs Tipe input berikut ini akan membuat sebuah *text field* : `text`, `date`, `datetime`, `datetime-local`, dan `week`.

```
<form>
  <div class="grid-container">
    <div class="grid-x grid-padding-x">
      <div class="medium-6 cell">
        <label>Input Label
          <input type="text" placeholder=".medium-6.cell">
        </label>
      </div>
      <div class="medium-6 cell">
        <label>Input Label
          <input type="text" placeholder=".medium-6.cell">
        </label>
      </div>
    </div>
  </div>
</form>
```



Gambar 2.12: Text Input pada Foundation

Text Inputs Penggunaan `select` menus untuk kombinasi beberapa pilihan ke dalam satu menu.

```
<label>Select Menu
  <select>
    <option value="husker">Husker</option>
    <option value="starbuck">Starbuck</option>
    <option value="hotdog">Hot Dog</option>
    <option value="apollo">Apollo</option>
  </select>
</label>
```

[?]

2.3.5 Komponen JavaScript

Foundation dilengkapi dengan komponen JavaScript untuk menambah fungsionalitas yang rumit. Komponen JavaScript dapat dimasukkan ke dalam proyek *developer* sehingga membuat pengembangan front-end lebih cepat dan lebih mudah.

Tabs

Tab semakin banyak digunakan dalam desain web karena *developer* dapat menyajikan konten secara seragam. Ini memungkinkan *developer* untuk menyimpan banyak dokumen dalam satu *window*. *developer* dapat menggunakan tab sebagai widget navigasi untuk beralih antar konten sehingga menghasilkan tata letak yang sistematis dan bersih. Komponen Tab dari Foundation membantu *developer* melakukan hal itu hanya dengan menambahkan beberapa baris kode. [?]

```
<ul class="tabs" data-tabs id="tab_component">
  <li class="tabs-title"><a href="#pub1">Section 1</a></li>
  <li class="tabs-title is-active"><a href="#pub2">Section 2</a></li>
  <li class="tabs-title"><a href="#pub3">Section 3</a></li>
  <li class="tabs-title"><a href="#pub4">Section 4</a></li>
</ul>
<div class="tabs-content" data-tabs-content="tab_component">
  <div class="tabs-panel" id="pub1">
    <p>Far far away, behind the word mountains, far from the countries
    Vokalia and Consonantia, there live the blind texts.</p>
  </div>
  <div class="tabs-panel is-active" id="pub2">
    <p> Separated they live in Bookmarksgrove right at the coast of the
    Semantics, a large language ocean. </p>
  </div>
  <div class="tabs-panel" id="pub3">
    <p>A small river named Duden flows by their place and supplies it with
```

```

the necessary regalia.</p>
</div>
<div class="tabs-panel" id="pub4">
<p>It is a paradisiacal country, in which roasted parts of sentences
fly into your mouth. </p>
</div>
</div>

```



Gambar 2.13: Grid pada Zurb Foundation

Dropdown Menu

Berfungsi untuk mengubah menu dasar menjadi menu dropdown yang dapat di-expand dengan plugin Menu Dropdown. Menu dropdown dibangun berdasarkan sintaks komponen **Menu**. Tambahkan kelas `.dropdown` dan atribut `data-dropdown-menu` ke wadah menu untuk mengatur dropdown. [?]

```

<ul class="dropdown menu" data-dropdown-menu>
  <li><a href="#">Item 1</a></li>
  <li><a href="#">Item 2</a></li>
  <li><a href="#">Item 3</a></li>
  <li><a href="#">Item 4</a></li>
</ul>

```

Reveal

Modal hanyalah wadah kosong, sehingga *developer* dapat menaruh segala jenis konten di dalamnya, seperti teks ke formulir hingga video ke seluruh *grid*. Untuk membuat modal, tambahkan kelas `.reveal`, atribut `data-reveal`, dan ID yang unik ke dalam *container*.

```

<div class="reveal" id="exampleModal1" data-reveal>
  <h1>Awesome. I Have It.</h1>
  <p class="lead">Your couch. It is mine.</p>
  <p>I'm a cool paragraph that lives inside of an even cooler modal. Wins!</p>
  <button class="close-button" data-close aria-label="Close modal" type="button">
    <span aria-hidden="true">&times;</span>
  </button>
</div>

```

2.4 Bootstrap 4

2.5 Template Skripsi FTIS UNPAR

Akan dipaparkan bagaimana menggunakan template ini, termasuk petunjuk singkat membuat referensi, gambar dan tabel. Juga hal-hal lain yang belum terpikir sampai saat ini.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

2.5.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh \LaTeX , perhatikan contoh di file bab2.tex untuk melihat bagaimana cara memaksa tabel ditempatkan sesuai keinginan kita.

Perhatikan bawa berbeda dengan penempatan judul gambar gambar, keterangan tabel harus diletakkan di atas tabel!! Lihat Tabel 2.1 berikut ini:

Tabel 2.1: Tabel contoh

	v_{start}	\mathcal{S}_1	v_{end}
τ_1	1	12	20
τ_2	1		20
τ_3	1	9	20
τ_4	1		20

Tabel 2.2 dan Tabel 2.3 berikut ini adalah tabel dengan sel yang berwarna dan ada dua tabel yang bersebelahan.

Tabel 2.2: Tabel bewarna(1)

	v_{start}	\mathcal{S}_2	\mathcal{S}_1	v_{end}
τ_1	1	5	12	20
τ_2	1	8		20
τ_3	1	2/8/17	9	20
τ_4	1			20

Tabel 2.3: Tabel bewarna(2)

	v_{start}	\mathcal{S}_1	\mathcal{S}_2	v_{end}
τ_1	1	12	5	20
τ_2	1		8	20
τ_3	1	9	2/8/17	20
τ_4	1			20

2.5.2 Kutipan

Berikut contoh kutipan dari berbagai sumber, untuk keterangan lebih lengkap, silahkan membaca file referensi.bib yang disediakan juga di template ini. Contoh kutipan:

- Buku: [?]
- Bab dalam buku: [?]

- Artikel dari Jurnal: [?]
- Artikel dari prosiding seminar/konferensi: [?]
- Skripsi/Thesis/Disertasi: [?] [?] [?]
- Technical/Scientific Report: [?]
- RFC (Request For Comments): [?]
- Technical Documentation/Technical Manual: [?] [?] [?]
- Paten: [?]
- Tidak dipublikasikan: [?] [?]
- Laman web: [?]
- Lain-lain: [?]

2.5.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen \LaTeX tidak dapat dilakukan melalui proses *drag and drop*. Perhatikan contoh pada file `bab2.tex` untuk melihat bagaimana cara menempatkan gambar. Beberapa hal yang harus diperhatikan pada saat menempatkan gambar:

- Setiap gambar **harus** diacu di dalam teks (gunakan *field* LABEL)
- *Field* CAPTION digunakan untuk teks pengantar pada gambar. Terdapat dua bagian yaitu yang ada di antara tanda [dan] dan yang ada di antara tanda { dan }. Yang pertama akan muncul di Daftar Gambar, sedangkan yang kedua akan muncul di teks pengantar gambar. Untuk skripsi ini, samakan isi keduanya.
- Jenis file yang dapat digunakan sebagai gambar cukup banyak, tetapi yang paling populer adalah tipe `eps` (lihat Gambar 2.14), tipe `JPG` (Gambar 2.15) dan tipe `PDF` (Gambar 2.16)
- Besarnya gambar dapat diatur dengan *field* SCALE.
- Penempatan gambar diatur menggunakan *placement specifier* (di antara tanda [dan] setelah deklarasi gambar. Yang umum digunakan adalah **H** untuk menempatkan gambar **sesuai** penempatannya di file `.tex` atau **h** yang berarti "kira-kira" di sini. Jika tidak menggunakan *placement specifier*, \LaTeX akan menempatkan gambar secara otomatis untuk menghindari bagian kosong pada dokumen anda. Walaupun cara ini sangat mudah, hindarkan terjadinya penempatan dua gambar secara berurutan.
 - Gambar 2.14 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
 - Gambar 2.15 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file `bab2.tex`!
 - Gambar 2.16 ditempatkan menggunakan *specifier* **h**.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

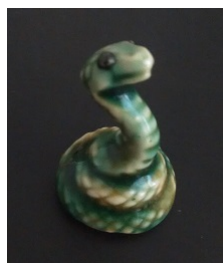
Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus

Gambar 2.14: Gambar *Serpentes* dalam format png

libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.



Gambar 2.15: Ular kecil

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt.

Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.



Gambar 2.16: *Serpentes jantan*

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