

«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
2.1.1 Login	3
2.1.2 Dosen	3
2.1.3 Dosen Informatika	4
2.1.4 Mahasiswa	4
2.1.5 Staf Tata Usaha	5
2.2 CodeIgniter	5
2.2.1 Application Flow Chart	5
2.2.2 CodeIgniter URLs	6
2.2.3 Model	6
2.2.4 View	7
2.2.5 Controller	9
2.3 Zurb Foundation 6	11
2.3.1 Struktur File	11
2.3.2 Sistem Grid pada Foundation	11
2.3.3 <i>Navigation</i> dan <i>Media Attributes</i>	12
2.3.4 Komponen CSS	14
2.3.5 Komponen JavaScript	17
2.4 Bootstrap 4	18
2.4.1 Sistem Grid Bootstrap	18
2.4.2 Konten	20
2.4.3 Komponen	22
A KODE PROGRAM	29
B HASIL EKSPERIMEN	31

DAFTAR GAMBAR

2.1	Flow Chart Aplikasi CodeIgniter	6
2.2	Struktur File Zurb Foundation	11
2.3	Grid pada Zurb Foundation	12
2.4	<i>Basic Navigation Menu</i> pada Foundation	13
2.5	Menu <i>align to right in Foundation</i>	13
2.6	Menu <i>align to center in Foundation</i>	13
2.7	Menu <i>active state menu in Foundation</i>	13
2.8	Menu <i>active state menu in Foundation</i>	14
2.9	Basic Button pada Foundation	14
2.10	Coloring Button pada Foundation	15
2.11	Basic Table pada Foundation	16
2.12	Text Input pada Foundation	17
2.13	Grid pada Zurb Foundation	18
2.14	Grid pada Bootstrap	18
2.15	Pilihan kelas grid pada Bootstrap	19
2.16	Tabel default pada Bootstrap	20
2.17	Tabel dengan thead yang dimodifikasi pada Bootstrap	21
2.18	<i>Menyelaraskan gambar ke kanan dan kiri pada bootstrap</i>	22
2.19	Forms Basic pada Bootstrap	23
2.20	Forms Basic pada Bootstrap	23
2.21	Disabled Basic pada Bootstrap	24
2.22	Button pada Bootstrap	24
2.23	Tombol <i>dropdown</i> pada Bootstrap	25
2.24	Navigation Bar pada Bootstrap	25
2.25	Modal pada Bootstrap	26
2.26	Ikon <i>Coffee</i> pada Font Awesome	27
2.27	Ikon <i>Coffee</i> pada Font Awesome	27
B.1	Hasil 1	31
B.2	Hasil 2	31
B.3	Hasil 3	31
B.4	Hasil 4	31

DAFTAR TABEL

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 merupakan aplikasi berbasis web, berguna sebagai aplikasi yang menunjang proses administrasi dalam lingkungan FTIS UNPAR. Web ini dapat diakses pada <http://www.bluetape.azurewebsites.net>.

2.1.1 Login

Halaman utama aplikasi BlueTape akan mengarahkan *user* untuk *login* dengan menggunakan Google, user akan login dengan melihat beberapa kondisi ini:

- Apabila *user* belum pernah login menggunakan akun UNPAR(xxx@student.unpar.ac.id atau yyy@unpar.ac.id) maka *user* akan diminta untuk memasukan email UNPAR dan password
- Apabila user sudah pernah login menggunakan akun UNPAR, maka *user* akan diminta untuk memilih akun beserta password.
- User akan terhubung otomatis dengan akun @gmail.com. Apabila BlueTape menolak autentikasi user maka: User akan diminta untuk buka halaman Gmail lalu klik avatar di kanan atas dan memilih akun UNPAR yang tepat pada *button* "Add Account"

User akan melihat beberapa menu sesuai dengan *role* user, sebagai mahasiswa, staf TU, dll.

2.1.2 Dosen

Perubahan Kuliah

Modul perubahan kuliah berguna untuk mengirimkan permintaan perubahan mata kuliah yang dikirim oleh dosen kepada staf Tata Usaha. Kolom - kolom yang terdapat dalam modul ini:

- Kode MK (Mata Kuliah)
- Nama Mata Kuliah
- Kelas
- Jenis perubahan (diganti / tambahan / ditiadakan)
- Dari (hari/jam dan ruang) dan ke(hari/jam dan tempat)
- Keterangan

Apabila ada kolom yang belum dapat diisi(contoh : dosen belum tahu tempat kelas pengganti) maka kolom kelas dapat dikosongkan. Dosen juga dapat membuat lebih dari 1 kelas pengganti, dengan mengklik tombol "Tambah Pertemuan Ekstra". Setelah dosen klik "Kirim Permohonan", maka sistem akan mengirim permohonan ke halaman BlueTape bagian Tata Usaha untuk diperiksa, disetujui, dan dicetak sebagai pengumuman. Jika staf Tata Usaha telah selesai mengkonfirmasi(atau menolak), maka dosen akan mendapatkan e-mail notifikasi.

2.1.3 Dosen Informatika

Entri Jadwal Dosen

Dosen informatika dapat menggunakan menu ini untuk mengisi jadwal mingguan. Hasil dari pengisian jadwal dapat diekspor ke XLS, atau dapat dilihat oleh mahasiswa informatika melalui portal BlueTape.

Tambah Jadwal

Pada bagian entri jadwal, dosen informatika dapat mengisi hari, jam mulai, durasi, label, dan sejenisnya. Berikut ini jenis yang dapat dipilih:

- Konsultasi : Waktu yang dosen siapkan untuk konsultasi mahasiswa. Pada tabel akan diberi *background* berwarna kuning.
- Terjadwal: Kegiatan mingguan dosen informatika yang telah terjadwal. Contoh : rapat jurusan
- Kelas : Kelas kuliah maupun praktikum.

Lalu dosen dapat klik tombol "Tambah" untuk menambahkan

Ubah/Hapus Jadwal

Dosen dapat mengubah atau menghapus jadwal yang tertera pada tabel. Lalu Pop up window akan terbuka dengan pilihan-pilihan yang sesuai dengan permintaan dosen.

Hapus Semua

Tombol "Delete All" dapat digunakan untuk menghapus secara cepat seluruh jadwal yang telah dosen buat sebelumnya. Penggunaan tombol ini biasa nya digunakan pada awal semester, dimana jadwal yang dosen miliki berubah seluruhnya.

Ekspor ke XLS

Tombol "Ekspor ke XLS" berfungsi untuk membuat file XLS untuk jadwal dosen.

2.1.4 Mahasiswa

Cetak Transkrip

Mahasiswa dapat menggunakan menu ini untuk mengirimkan permohonan cetak transkrip Mahasiswa mengirimkan permohonan pencetakan transkrip dengan mengisi kolom-kolom pada formulir "Permohonan Baru". Mahasiswa hanya dapat mengirimkan permohonan:

- Maksimal 1x dalam satu semester (kecuali permohonan ditolak)
- Jika ada permohonan yang belum dijawab.

Mahasiswa Informatika

Lihat Jadwal Dosen






Mahasiswa dapat melihat jadwal mingguan seluruh dosen dengan memilih nama dosen pada seleksi tab, dan tabel jadwal dosen akan ditampilkan pada bagian bawah halaman. Selain itu tabel juga berisi informasi tanggal terakhir dosen meng-*update* jadwal sehingga mahasiswa dapat melihat apakah jadwal tersebut merupakan jadwal semester ini atau semester lalu. Lalu terdapat tombol

"Ekspor ke XLS" pada halaman lihat jadwal dosen, sehingga mahasiswa dapat menyimpan atau mencetak jadwal tersebut.




2.1.5 Staf Tata Usaha

Manajemen Perubahan Kuliah

Staf Tata Usaha dapat melakukan manajemen permintaan perubahan kuliah. Sebuah tabel akan menampilkan daftar permohonan dengan menampilkan tanggal kapan permohonan dibuat. Setiap daftar permohonan akan memiliki beberapa tombol :





-  berfungsi untuk melihat detail permohonan sehingga dapat menentukan apakah permohonan disetujui atau tidak.
-  berfungsi untuk membuka pop-up print-out pengumuman.
-  berfungsi sebagai konfirmasi bahwa pengumuman telah dicetak dan disebar.
-  berfungsi untuk menyatakan bahwa permohonan ditolak. Staf Tata Usaha akan mengisi alasan mengapa permohonan ditolak sehingga tidak membingungkan pemohon.
-  berfungsi untuk menghapus permohonan **secara permanen**. Staf Tata Usaha diimbau agar tidak menggunakan tombol ini kecuali dalam keadaan terpaksa.

Manajemen Cetak Transkrip

Staf Tata Usaha dapat melihat daftar permintaan transkrip dalam bentuk tabel. Keterangan mengenai transkrip dapat dilihat menggunakan tombol  (detail). Selain itu terdapat dua pilihan jawaban dalam setiap daftar permintaan yaitu  (tolak) dan  (cetak). Masing-masing tombol memerlukan keterangan tambahan mengenai alasan mengapa transkrip dapat dicetak maupun ditolak.

Modul ini berguna untuk manajemen permohonan cetak transkrip. Terdapat sebuah tabel yang menampilkan daftar permohonan dengan tanggal yang terurut. Staf Tata Usaha dapat mencari daftar permintaan berdasarkan NPM pemohon.

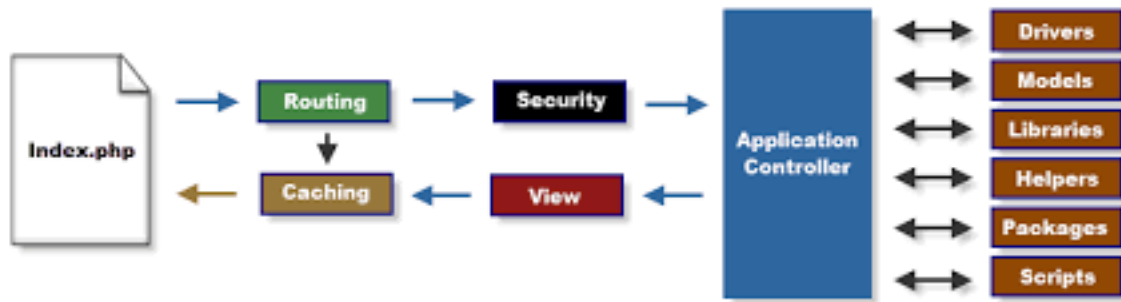
Beberapa tombol yang tersedia untuk setiap permohonan :

-  berfungsi untuk melihat detail permohonan sehingga dapat menentukan apakah permohonan disetujui atau tidak.
-  berfungsi untuk membuka pop-up print-out pengumuman. Dalam pop-up akan disediakan sebuah link menuju halaman percetakan transkrip pada SIAkad.
-  berfungsi untuk menyatakan bahwa permohonan ditolak. Staf Tata Usaha akan mengisi alasan mengapa permohonan ditolak sehingga tidak membingungkan pemohon.
-  berfungsi untuk menghapus permohonan **secara permanen**. Staf Tata Usaha diimbau agar tidak menggunakan tombol ini kecuali dalam keadaan terpaksa.

2.2 CodeIgniter

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 / class / function / ID

1. Segmen pertama menyatakan kelas *controller* yang harus dipanggil.
2. Segmen kedua menyatakan fungsi kelas, atau metode, yang harus dipanggil.
3. Segmen ketiga dan setiap segmen setelahnya menyatakan ID dan variabel apa pun yang akan diteruskan ke controller.

2.2.3 Model

Model merepresentasikan struktur data. 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 );
```

listing only

2.2.4 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>
```

2.2.5 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:

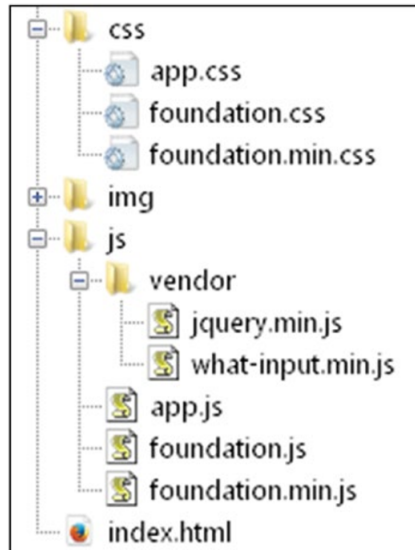
```
example.com/index.php/blog/index/
```

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>

```



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

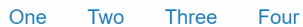
Semua versi menu terdiri dari sebuah `` yang diisi oleh beberapa elemen ``. Secara default, menu akan berorientasi horizontal.[?]

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>

```


 A horizontal navigation menu with four items: "One", "Two", "Three", and "Four". All items are in a light blue font and are left-aligned.
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>
```


 A horizontal navigation menu with four items: "One", "Two", "Three", and "Four". All items are in a light blue font and are right-aligned.
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>
```


 A horizontal navigation menu with four items: "One", "Two", "Three", and "Four". All items are in a light blue font and are center-aligned.
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>
```


 A horizontal navigation menu with three items: "Home", "About", and "Nachos". The "Home" item is highlighted with a dark blue background and white text, while "About" and "Nachos" are in light blue text on a white background.
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>
```

Learn More View All Features Save Delete

Gambar 2.9: Basic Button pada Foundation

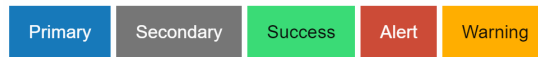
Warna dapat ditambahkan untuk memberikan *buttons* arti yang bermakna.

```
<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 eget metum</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</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</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>
</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 paradisematic 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

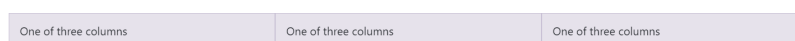
Modal hanyalah wadah kosong, sehingga *developer* dapat menaruh segala jenis konten di dalamnya, seperti teks ke formulir hingga video ke seluruh *grid*. Untuk membuat modal, tambahkan kelas `.reveal`, atribut `data-reveal`, dan ID yang unik ke dalam *container*.

```
<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

2.4.1 Sistem Grid Bootstrap

Sistem grid Bootstrap menggunakan *container*, *rows*, dan *columns* untuk tata letak dan penyelarasan konten. Selain itu sistem ini dibangun dengan *flexbox* dan seluruhnya *responsive*.



Gambar 2.14: Grid pada Bootstrap


```

<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>

```

Dalam contoh diatas akan dibuat tiga kolom yang memiliki lebar yang sama baik dalam *device small, medium, large* dan *extra large* menggunakan kelas grid yang sudah ditentukan sebelumnya oleh Bootstrap. Penggunaan `.container` akan membuat kolom berada ditengah halaman.

Secara detail, bootstrap bekerja dengan cara:

- *Container* disediakan agar konten berada ditengah halaman dan mengisi konten tersebut secara horizontal. Penggunaan `.container` untuk menentukan lebar pixel secara responsif atau `.container-fluid` untuk membuat lebar: 100% di semua ukuran *viewport* dan perangkat.
- Sebuah baris akan membungkus kolom - kolom. Setiap kolom akan memiliki *padding* secara horizontal yang disebut **gutter** untuk mengatur jarak antar kolom.
- Penggunaan flexbox akan membuat lebar pada kolom tidak perlu dispesifikasikan. Misalnya empat variabel dari `.col-sm` akan secara otomatis membuat lebar kolom sebesar 25%.
- Kelas kolom menunjukkan jumlah kolom yang ingin digunakan, dengan maksimal 12 kolom per baris. Apabila *developer* menginginkan tiga kolom yang memiliki lebar yang sama maka dapat menggunakan `.col-4`.
- Lebar kolom diatur dalam persentase, sehingga kolom akan memiliki lebar yang berubah-ubah dan ukuran bergantung dengan elemen *parent* nya.

Pilihan Grid

Bootstrap menggunakan px untuk grid breakpoint dan lebar container. Ini dikarenakan lebar *viewport* ditentukan dengan satuan pixels. Berikut ini tabel yang menjelaskan penggunaan kelas grid dalam berbagai perangkat :

	Extra small ≤576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Gambar 2.15: Pilihan kelas grid pada Bootstrap

2.4.2 Konten

Tabel

Dengan penggunaan kelas `.table` pada seluruh tag `<table>` maka *style* pada bootstrap akan diterapkan, sehingga setiap tabel yang *nested* akan diatur sesuai dengan *parent* nya.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Gambar 2.16: Tabel default pada Bootstrap

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

Opsi Tabel dengan *head* yang dimodifikasi

Penggunaan kelas `.thead-light` atau `.thead-dark` dapat digunakan untuk membuat `<thead>` menjadi abu muda atau abu tua.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter
#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Gambar 2.17: Tabel dengan thead yang dimodifikasi pada Bootstrap

```

<table class="table">
  <thead class="thead-dark">
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>

<table class="table">
  <thead class="thead-light">
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>

```

```

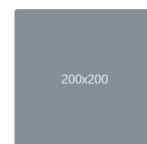
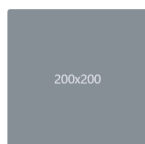
</thead>
<tbody>
  <tr>
    <th scope="row">1</th>
    <td>Mark</td>
    <td>Otto</td>
    <td>@mdo</td>
  </tr>
  <tr>
    <th scope="row">2</th>
    <td>Jacob</td>
    <td>Thornton</td>
    <td>@fat</td>
  </tr>
  <tr>
    <th scope="row">3</th>
    <td>Larry</td>
    <td>the Bird</td>
    <td>@twitter</td>
  </tr>
</tbody>
</table>

```

Gambar

Gambar dalam Bootstrap akan memiliki sifat *responsive* dengan menerapkan kelas `.img-fluid` serta mengatur lebar gambar dengan properties `max-width: 100%` dan `height: auto`. Sehingga gambar tidak pernah lebih besar dari *parent* nya.

Developer dapat menyelaraskan (align) sebuah gambar ke kiri atau kanan dengan **helper float classes** atau **text alignment classes**.



Gambar 2.18: Menyelaraskan gambar ke kanan dan kiri pada bootstrap

```

<img src = "... " class="rounded float-left" alt = "... ">
<img src = "... " class="rounded float-right" alt = "... ">

```

2.4.3 Komponen

Formulir

Form pada Bootstrap menyediakan beragam tipe input sesuai dengan kebutuhan *user*. Contohnya penggunaan kelas `email` untuk *input* email atau `number` untuk input berupa angka.

Form Controls

Developer dapat membuat form menggunakan kelas `.form-control`. Kelas ini terdiri dari beberapa tag seperti tag `<input>`, `<select>` dan `<textarea>`.

```

<form>
  <div class="form-group">
    <label for="exampleFormControlInput1">Email address</label>
    <input type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
  </div>
  <div class="form-group">
    <label for="exampleFormControlSelect1">Example select</label>
    <select class="form-control" id="exampleFormControlSelect1">
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
    </select>
  </div>
  <div class="form-group">
    <label for="exampleFormControlSelect2">Example multiple select</label>
    <select multiple class="form-control" id="exampleFormControlSelect2">
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
    </select>
  </div>
  <div class="form-group">
    <label for="exampleFormControlTextarea1">Example textarea</label>
    <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
  </div>
</form>

```

The screenshot shows a rendered Bootstrap 4 form. It contains four fields stacked vertically. The first field is labeled 'Email address' and is a text input with the placeholder 'name@example.com'. The second field is labeled 'Example select' and is a single-select dropdown menu with '1' selected. The third field is labeled 'Example multiple select' and is a multiple-select dropdown menu showing options 2, 3, 4, and 5. The fourth field is labeled 'Example textarea' and is a text area with three rows.

Gambar 2.19: Forms Basic pada Bootstrap

Column Sizing

Bootstrap memungkinkan *developer* untuk menempatkan sejumlah `.col` di dalam baris `.row` atau `.form` dengan lebar tertentu. Misalnya ada tiga buah kolom, kolom pertama memiliki lebar 7 dengan menggunakan kelas `.col-7` maka dua kolom sisanya akan memiliki lebar yang memenuhi baris tersebut.

The screenshot shows a rendered Bootstrap 4 form with three text input fields arranged horizontally in a single row. The first field is labeled 'City' and has a width of 7 columns. The second field is labeled 'State' and the third is labeled 'Zip', both with a width of 1 column each, totaling 9 columns.

Gambar 2.20: Forms Basic pada Bootstrap

```

<form>
  <div class="form-row">
    <div class="col-7">
      <input type="text" class="form-control" placeholder="City">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="State">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="Zip">
    </div>
  </div>
</form>

```

```

    </div>
  </div>
</form>

```

Disabled Forms

Penambahan atribut boolean `disabled` pada sebuah input membuat *user* tidak bisa mengisi data pada *field* tersebut. Untuk non-aktifkan seluruh *field* pada sebuah kolom dapat menambahkan atribut `disabled` pada tag `<fieldset>`.



Gambar 2.21: Disabled Basic pada Bootstrap

```

<form>
  <fieldset disabled>
    <div class="form-group">
      <label for="disabledTextInput">Disabled input</label>
      <input type="text" id="disabledTextInput" class="form-control" placeholder="Disabled input">
    </div>
    <div class="form-group">
      <label for="disabledSelect">Disabled select menu</label>
      <select id="disabledSelect" class="form-control">
        <option>Disabled select</option>
      </select>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" id="disabledFieldsetCheck" disabled>
      <label class="form-check-label" for="disabledFieldsetCheck">
        Can't check this
      </label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </fieldset>
</form>

```

Tombol

Bootstrap memasukan beberapa button dengan *style* yang sudah didefinisikan sebelumnya, membuat setiap button akan memiliki makna nya sendiri.



Gambar 2.22: Button pada Bootstrap

```

<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

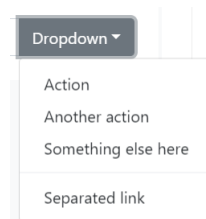
<button type="button" class="btn btn-link">Link</button>

```

Button with Dropdowns

```
<div class="input-group mb-3">
  <div class="input-group-prepend">
    <button class="btn btn-outline-secondary dropdown-toggle" type="button" data-toggle="dropdown"
      aria-haspopup="true" aria-expanded="false">Dropdown</button>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Action</a>
      <a class="dropdown-item" href="#">Another action</a>
      <a class="dropdown-item" href="#">Something else here</a>
      <div role="separator" class="dropdown-divider"></div>
      <a class="dropdown-item" href="#">Separated link</a>
    </div>
  </div>
  <input type="text" class="form-control" aria-label="Text input with dropdown button">
</div>

<div class="input-group">
  <input type="text" class="form-control" aria-label="Text input with dropdown button">
  <div class="input-group-append">
    <button class="btn btn-outline-secondary dropdown-toggle" type="button" data-toggle="dropdown"
      aria-haspopup="true" aria-expanded="false">Dropdown</button>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Action</a>
      <a class="dropdown-item" href="#">Another action</a>
      <a class="dropdown-item" href="#">Something else here</a>
      <div role="separator" class="dropdown-divider"></div>
      <a class="dropdown-item" href="#">Separated link</a>
    </div>
  </div>
</div>
```



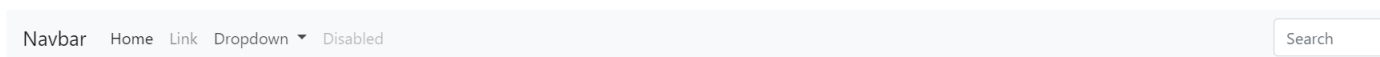
Gambar 2.23: Tombol *dropdown* pada Bootstrap

Navigation Bar

Navbar pada Bootstrap terdiri dari beberapa sub-komponen yang bisa digunakan sesuai dengan kebutuhan:

- **.navbar-brand** : Komponen untuk menampilkan nama perusahaan, nama produk atau nama proyek.
- **.navbar-nav** : Komponen untuk membuat navigasi memiliki lebar yang memenuhi layar.
- **.navbar-toggler** : Komponen yang digunakan bersamaan dengan plugin untuk membuat efek jatuh dan perilaku navigasi lainnya.
- **.form-inline** : Komponen untuk pengaturan formulir dan aksi.
- **.collapse.navbar-collapse** : Komponen untuk mengelompokkan dan menyembunyikan *navigation bar* dengan sebuah breakpoint induknya.

Berikut ini merupakan semua sub-komponen yang termasuk dalam navigation bar, navbar mengimplementasikan tema **light-themed** yang secara otomatis menyembunyikan menu pada breakpoint **lg**



Gambar 2.24: Navigation Bar pada Bootstrap

```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
          data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          Dropdown
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdown">
          <a class="dropdown-item" href="#">Action</a>
          <a class="dropdown-item" href="#">Another action</a>
          <div class="dropdown-divider"></div>
          <a class="dropdown-item" href="#">Something else here</a>
        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
    </form>
  </div>
</nav>

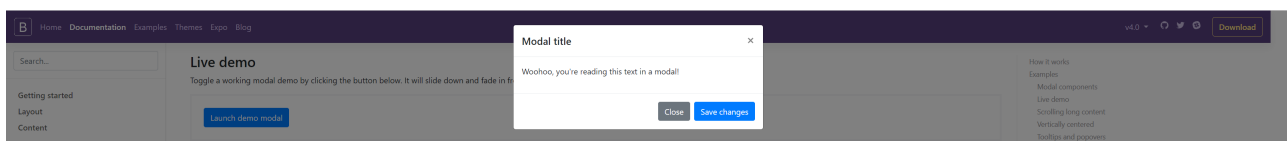
```

Modal

Bagaimana Modal bekerja :

- Modal dibangun dengan HTML, CSS dan Javascript.
- Menekan modal "backdrop" otomatis menutup komponen modal.
- Bootstrap hanya mendukung satu modal dalam sebuah window pada satu waktu. Penggunaan modal yang bercabang dalam Bootstrap dipercaya memberikan *user experience* yang buruk.
- Modal menggunakan **position: fixed** yang diletakkan pada posisi teratas dalam kode agar terhindar dari *bug* yang disebabkan elemen lain yang memiliki posisi *fixed*.

Komponen modal terdiri dari modal header modal body dan modal footer (opsional).



Gambar 2.25: Modal pada Bootstrap

```

<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal">
  Launch demo modal
</button>

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog"
  aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        ...
      </div>
    </div>
  </div>
</div>

```



```

    </div>
    <div class="modal-footer">
      <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
      <button type="button" class="btn btn-primary">Save changes</button>
    </div>
  </div>
</div>
</div>

```

Ikon

Bootstrap tidak memiliki *library* ikon secara *default*, sehingga ikon yang digunakan diambil dari **Font Awesome**. Penggunaan ikon dengan menggunakan tag `<i>` yang disertai dengan kelas `fa` (font-awesome).

```
<i class="fa fa-coffee"></i>
```



Gambar 2.26: Ikon *Coffee* pada Font Awesome

Alert

Alert menyediakan pesan umpan balik untuk user untuk berbagai tipe pesan peringatan yang tersedia. Untuk gaya yang sesuai *developer* dapat menggunakan delapan kelas yang tersedia.



Gambar 2.27: Ikon *Coffee* pada Font Awesome

```

<div class="alert alert-primary" role="alert">
  This is a primary a l e r t check it out!
</div>
<div class="alert alert-secondary" role="alert">
  This is a secondary a l e r t check it out!
</div>
<div class="alert alert-success" role="alert">
  This is a success a l e r t check it out!
</div>

```

```
<div class="alert alert-danger" role="alert">
  This is a danger alert check it out!
</div>
<div class="alert alert-warning" role="alert">
  This is a warning alert check it out!
</div>
<div class="alert alert-info" role="alert">
  This is a info alert check it out!
</div>
<div class="alert alert-light" role="alert">
  This is a light alert check it out!
</div>
<div class="alert alert-dark" role="alert">
  This is a dark alert check it out!
</div>
```

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