

MIGRASI ZURB FOUNDATION KE BOOTSTRAP 4

HAPSARI LAKSMI W-2015730037

1 Data Skripsi

Pembimbing utama/tunggal: **Pascal Alfadian Nugroho**

Pembimbing pendamping: -

Kode Topik : **PAN4791**

Topik ini sudah dikerjakan selama : **1 semester**

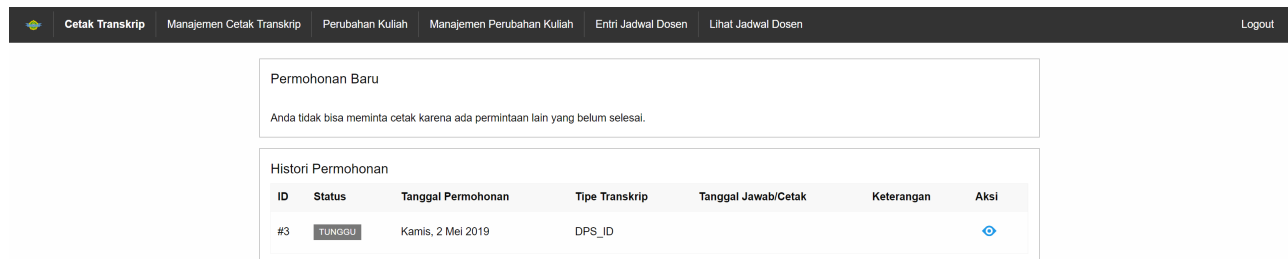
Pengambilan pertama kali topik ini pada : Semester **47 - Ganjil 19/20**

Pengambilan pertama kali topik ini di kuliah : **Skripsi 1**

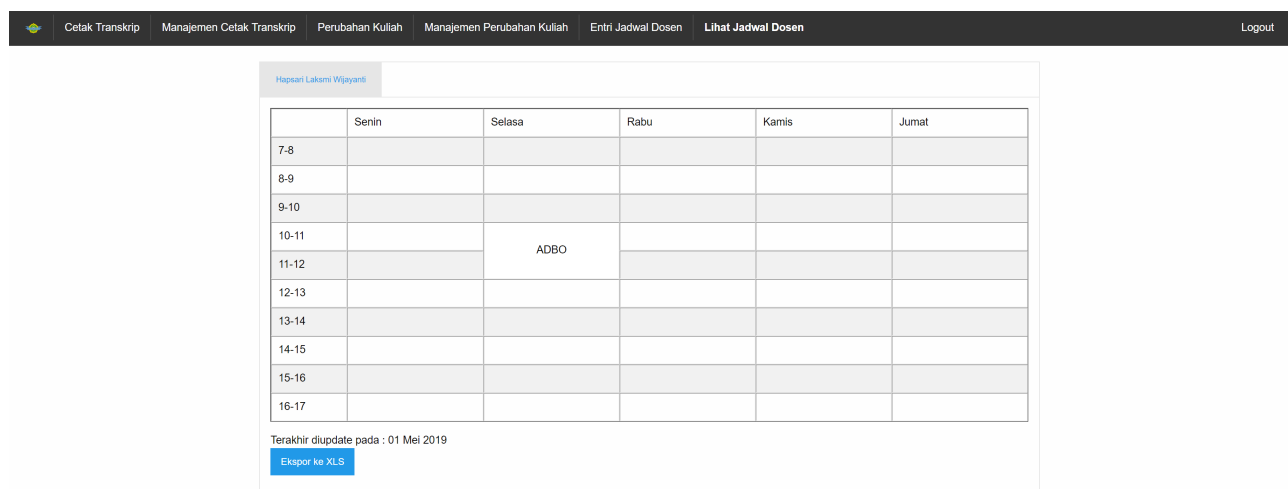
Tipe Laporan : **B** - Dokumen untuk reviewer pada presentasi dan **review Skripsi 1**

2 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*.



Gambar 1: Tampilan Cetak Transkrip



Gambar 2: Tampilan Lihat Jadwal Dosen

Permintaan Transkrip

Cari NPM: 2013730013 Cari

ID	Status	Tanggal Permohonan	Tipe Transkrip	NPM	Aksi
#3	MENUNGGU	Kamis, 2 Mei 2019	DPS_ID	-	View Edit Print Delete
#2	DITOLAK	Rabu, 1 Mei 2019	DPS_ID	2015730037	View Edit Print Delete

Gambar 3: Tampilan-Manajemen-Cetak-Transkrip

Tambah Jadwal

Hari: Durasi: Label:

Jam Mulai: Jenis: Tambah

Daftar Jadwal

	Senin	Selasa	Rabu	Kamis	Jumat
7-8	Rapat				
8-9					
9-10					
10-11		ADBO			
11-12					
12-13					
13-14					
14-15					
15-16					
16-17					

Terakhir diupdate pada : 29 Oktober 2019

Delete All Export ke XLS

Gambar 4: Tampilan Entri Jadwal Dosen

Permohonan Baru

Pemohon: Nama:

Kode MK: Nama Mata Kuliah: Kelas: Jenis Perubahan:

Dari Hari & Jam: Dari Ruang: Keterangan Tambahan:

Menjadi Hari & Jam: Menjadi Ruang:

Menjadi Hari & Jam: Menjadi Ruang: Hapus

Kirim Permohonan Tambah Pertemuan Ekstra

Histori Permohonan

ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Tanggal Jawab	Keterangan	Aksi
#1	DITOLAK	Rabu, 1 Mei 2019	AIF123	Ditiladkan	Rabu, 1 Mei 2019	tidak bisa	View

Gambar 5: Tampilan Perubahan Kuliah

Meskipun *open-source*, saat ini *Zurb Foundation* tidak sepopuler *framework Bootstrap*. *Bootstrap* adalah *Javascript framework* yang didesain untuk membantu membangun komponen *user interface* yang terdiri dari *CSS*, *JavaScript/jQuery*, dan *glyphicons*. Pembangunan *website* yang lebih cepat dan besarnya komunitas yang ada berdampak pada banyaknya pengembang *web* yang memanfaatkan *framework Bootstrap*. Sehingga jumlah proyek yang dihasilkan oleh *framework Bootstrap* lebih banyak dibanding *Zurb Foundation*.

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

Permohonan Perubahan Kuliah					
ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Aksi
#1	DITOLAK	Rabu, 1 Mei 2019	AIF123	Ditladakan	View Print Like Comment Delete

Gambar 6: Tampilan Manajemen Perubahan Kuliah

3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini :

1. Membuat 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 yang responsive untuk berbagai platform.
2. Mengimplentasikan plugin yang tersedia dalam library Bootstrap 4.

4 Rumusan Masalah

1. Bagaimana membuat template manajemen cetak transkrip, manajemen perubahan kuliah dan manajemen jadwal dosen.
2. Bagaimana mengimplentasikan plugin yang tersedia di dalam Bootstrap 4.

5 Detail Perkembangan Pengerjaan Skripsi

Detail bagian pekerjaan skripsi sesuai dengan rencan kerja/laporan perkembangan terakhir :

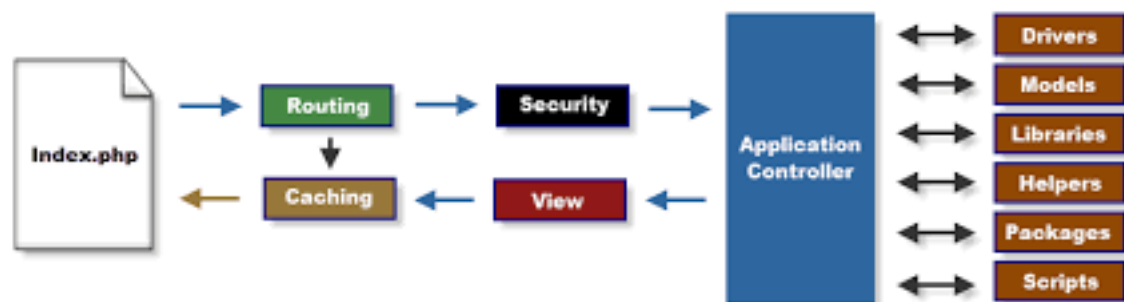
1. **Mempelajari framework PHP Codeigniter.**

Status : Ada sejak rencana kerja skripsi.

Hasil : Mempelajari konsep codeigniter dan mencoba langsung didalam proyek BlueTape

Application Flow Chart

Gambar berikut mengilustrasikan bagaimana alur data pada sistem :



Gambar 7: Flow Chart Aplikasi CodeIgniter

- (a) **Index.php** : bertindak sebagai *front controller*, menginisiasi *base resources* yang dibutuhkan untuk menjalankan CodeIgniter.

- (b) **Router** : akan memeriksa permintaan HTTP untuk menetapkan hal apa yang harus dilakukan dengan permintaan tersebut.
- (c) **Cache** : Apabila terdapat *cache*, maka *cache* tersebut akan dikirimkan langsung ke browser, dengan melewati sistem eksekusi normal.
- (d) **Security** : Sebelum *controller* dimuat, *HTTP request* dan *user* mana pun yang mengirimkan data diseleksi dahulu untuk keamanan.
- (e) **Controller** : Terdiri dari *model*, *core libraries*, *helpers*, dan *resources* yang dibutuhkan untuk proses *request* tertentu.
- (f) **View** : Tampilan yang telah selesai dirender kemudian dikirim ke *web browser* untuk dilihat. Jika *caching* diaktifkan, tampilan di *cache* terlebih dahulu sehingga pada permintaan selanjutnya dapat dilayani.¹

CodeIgniter URLs

Codeigniter menggunakan pendekatan berbasis segment :

```
1 example.com/class/function/ID
```

- (a) Segmen pertama menyatakan kelas *controller* yang harus dipanggil.
- (b) Segmen kedua menyatakan fungsi kelas, atau metode, yang harus dipanggil.
- (c) Segmen ketiga dan setiap segmen setelahnya menyatakan ID dan variabel apa pun yang akan diteruskan ke controller.

Model

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

Anatomi Model

Kelas model akan disimpan dalam direktori **application/models/directory**. Prototipe dasar dari sebuah model kelas :

```
1 <?php
2 class Model_name extends CI_Model {
3
4 }
```

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

Loading a Model

Model akan dimuat dan dipanggil didalam metode *controller*. Untuk memuat sebuah model maka dapat digunakan metode berikut:

```
1 $this->load->model('model_name');
```

¹Codeigniter 3 (2017) Codeigniter 3.1.4 Documentation. British Columbia Institute of Technology. 5112 Bobby Hicks Hwy, Gray, TN 37615, USA.

Koneksi ke Database

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

```

1      $config['hostname'] = 'localhost';
2      $config['username'] = 'myusername';
3      $config['password'] = 'mypassword';
4      $config['database'] = 'mydatabase';
5      $config['dbdriver'] = 'mysqli';
6      $config['dbprefix'] = '';
7      $config['pconnect'] = FALSE;
8      $config['db_debug'] = TRUE;
9
10     $this->load->model('model_name', '', $config);

```

listing only

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*. Dalam *MVC framework*, *controller* bertanggung jawab untuk mengambil *view* tertentu.

Membuat sebuah View

CodeIgniter memuat view dengan memanggil sebuah file php, misalkan `blogview.php`, dan *developer* dapat mengisinya dengan kode HTML sebagai berikut:

```

1      <html>
2      <head>
3      <title>My Blog</title>
4      </head>
5      <body>
6      <h1>Welcome to my Blog!</h1>
7      </body>
8      </html>

```

File tersebut akan disimpan di direktori `application/views/`.

Loading sebuah View

View dapat dimuat dengan membuat file view dengan syntax berikut:

```

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

```

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

Memuat Beberapa View

CodeIgniter dapat menangani beberapa panggilan dari dalam controller dengan menggunakan syntax :

```

1      $this->load->view()

```

Apabila ada lebih dari satu panggilan yang terjadi, maka *views* akan dilampirkan secara bersamaan. Berikut ini kode yang digunakan jika *developer* ingin mempunyai *header view*, *menu view*, *content view*, dan *footer view*.

```

1      <?php
2
3      class Page extends CI_Controller {
4
5      public function index()
6      {
7          $data['page_title'] = 'Your title';
8          $this->load->view('header');
9          $this->load->view('menu');
10         $this->load->view('content', $data);
11         $this->load->view('footer');
12     }
13
14     }

```

Menyimpan Views didalam Sub Direktori

Untuk menyimpan didalam sub direktori maka dapat menyertakan nama direktori yang memuat **view**.

```

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

```

Menambahkan data dinamis ke View

Data yang dikirim dari controller menuju view berbentuk array atau objek, sehingga akan dilampirkan dalam parameter kedua dalam metode loading view. Berikut ini penggunaan dengan array:

```

1      $data = array(
2          'title' => 'My Title',
3          'heading' => 'My Heading',
4          'message' => 'My Message'
5      );
6
7      $this->load->view('blogview', $data);

```

Kemudian, penggunaan dengan objek:

```

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

```

Sehingga apabila dimasukan ke controller, kode yang ditambahkan adalah:

```

1      <?php
2      class Blog extends CI_Controller {
3
4      public function index()
5      {
6          $data['title'] = "My Real Title";
7          $data['heading'] = "My Real Heading";
8
9          $this->load->view('blogview', $data);
10     }
11 }

```

Untuk mengaksesnya dalam file HTML maka *developer* dapat menggunakan syntax php :

```

1      <html>
2      <head>
3          <title><?php echo $title;?></title>
4      </head>
5      <body>
6          <h1><?php echo $heading;?></h1>
7      </body>
8      </html>

```

Controller

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

Sebuah *controller* secara sederhana merupakan sebuah file yang dinamakan dengan aturan tertentu sehingga dapat dihubungkan dengan sebuah URL. Misalnya untuk URL ini:

```

1 <?php
2 example.com/index.php/blog/

```

Dalam contoh diatas, *Codeigniter* berusaha menemukan *controller* bernama *Blog.php* dan lalu memuatnya. Ketika sebuah nama *controller* sesuai dengan segmen pertama dari sebuah URL, maka URL akan memuatnya.

Kode berikut merupakan contoh dari *controller* sederhana.

```

1 <?php
2 class Blog extends CI_Controller {
3
4     public function index()
5     {
6         echo 'Hello World'
7     }
8 }

```

Method

Dalam sebuah kelas *controller* akan memiliki beberapa method, lalu untuk memanggil fungsi didalamnya maka *developer* dapat mengisi segmen kedua dari sebuah url dengan sebuah method. Misalnya controller dengan dua method yaitu *index()* dan *comments()*.

```

1 <?php
2 class Blog extends CI_Controller {
3
4     public function index()
5     {
6         echo 'Hello World!';
7     }
8
9     public function comments()
10    {
11        echo 'Look at this!';
12    }
13 }

```

Pemanggilan method *index* dapat secara otomatis dilakukan apabila segmen kedua kosong. Namun ada cara lain untuk menampilkan pesan "Hello World" yang dapat dilakukan dengan:

```

1 example.com/index.php/blog/index/

```

Kemudian untuk memuat method *comment()* dapat dituliskan sebagai berikut:

```

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

```

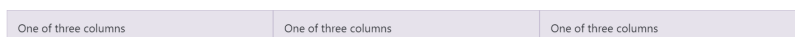
2. Mempelajari framework front-end Bootstrap 4 beserta plugin-plugin yang tersedia.

Status : Ada sejak rencana kerja skripsi.

Hasil :

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*.²



Gambar 8: Grid pada Bootstrap

²Bootstrap 4 (2019) Bootstrap v4.3.1 Documentation. Bootstrap Core Team. 77 Massachusetts Avenue Cambridge, MA 02139, USA.

```

1      <div class="container">
2          <div class="row">
3              <div class="col-sm">
4                  One of three columns
5              </div>
6              <div class="col-sm">
7                  One of three columns
8              </div>
9              <div class="col-sm">
10                 One of three columns
11            </div>
12        </div>
13    </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 detil, 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 9: Pilihan kelas grid pada Bootstrap

Tabel

Tabel Default

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 10: Tabel default pada Bootstrap

```

1  <table class="table">
2    <thead>
3      <tr>
4        <th scope="col">#</th>
5        <th scope="col">First</th>
6        <th scope="col">Last</th>
7        <th scope="col">Handle</th>
8      </tr>
9    </thead>
10   <tbody>
11     <tr>
12       <th scope="row">1</th>
13       <td>Mark</td>
14       <td>Otto</td>
15       <td>@mdo</td>
16     </tr>
17     <tr>
18       <th scope="row">2</th>
19       <td>Jacob</td>
20       <td>Thornton</td>
21       <td>@fat</td>
22     </tr>
23     <tr>
24       <th scope="row">3</th>
25       <td>Larry</td>
26       <td>the Bird</td>
27       <td>@twitter</td>
28     </tr>
29   </tbody>
30 </table>

```

Tabel dengan Garis Batas

Penggunaan kelas `.table-bordered` akan membuat tabel memiliki garis batas untuk semua sisi didalam tabel dan *cells*.

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

Gambar 11: Tabel default pada Bootstrap

```

1  <table class="table table-bordered">
2    <thead>
3      <tr>
4        <th scope="col">#</th>
5        <th scope="col">First</th>
6        <th scope="col">Last</th>
7        <th scope="col">Handle</th>
8      </tr>
9    </thead>
10   <tbody>
11     <tr>
12       <th scope="row">1</th>
13       <td>Mark</td>
14       <td>Otto</td>
15       <td>@mdo</td>
16     </tr>
17     <tr>
18       <th scope="row">2</th>
19       <td>Jacob</td>
20       <td>Thornton</td>
21       <td>@fat</td>
22     </tr>

```

```

23         <tr>
24             <th scope="row">3</th>
25             <td colspan="2">Larry the Bird</td>
26             <td>@twitter</td>
27         </tr>
28     </tbody>
29 </table>

```

Tabel dengan Warna Baris Berbeda

Penggunaan kelas `.table-striped` akan membuat tabel memiliki warna baris berbeda batas antara baris genap dan ganjil didalam tag `<tbody>`.

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

Gambar 12: Tabel default pada Bootstrap

```

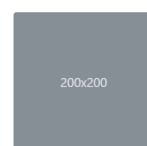
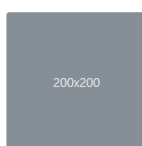
1  <table class="table table-striped">
2    <thead>
3      <tr>
4        <th scope="col">#</th>
5        <th scope="col">First</th>
6        <th scope="col">Last</th>
7        <th scope="col">Handle</th>
8      </tr>
9    </thead>
10   <tbody>
11     <tr>
12       <th scope="row">1</th>
13       <td>Mark</td>
14       <td>Otto</td>
15       <td>@mdo</td>
16     </tr>
17     <tr>
18       <th scope="row">2</th>
19       <td>Jacob</td>
20       <td>Thornton</td>
21       <td>@fat</td>
22     </tr>
23     <tr>
24       <th scope="row">3</th>
25       <td colspan="2">Larry the Bird</td>
26       <td>@twitter</td>
27     </tr>
28   </tbody>
29 </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 13: Menyelaraskan gambar ke kanan dan kiri pada bootstrap

```

1 
2 

```

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>`.

```

1 <form>
2   <div class="form-group">
3     <label for="exampleFormControlInput1">Email address</label>
4     <input type="email" class="form-control" id="exampleFormControlInput1" placeholder="
      name@example.com">
5   </div>
6   <div class="form-group">
7     <label for="exampleFormControlSelect1">Example select</label>
8     <select class="form-control" id="exampleFormControlSelect1">
9       <option>1</option>
10      <option>2</option>
11      <option>3</option>
12      <option>4</option>
13      <option>5</option>
14    </select>
15  </div>
16  <div class="form-group">
17    <label for="exampleFormControlSelect2">Example multiple select</label>
18    <select multiple class="form-control" id="exampleFormControlSelect2">
19      <option>1</option>
20      <option>2</option>
21      <option>3</option>
22      <option>4</option>
23      <option>5</option>
24    </select>
25  </div>
26  <div class="form-group">
27    <label for="exampleFormCntrlTextarea1">Example textarea</label>
28    <textarea class="form-control" id="exampleFormCntrlTextarea1" rows="3"></textarea>
29  </div>
30 </form>

```

The screenshot shows a Bootstrap form with the following elements:

- Email address:** An input field with the placeholder text "name@example.com".
- Example select:** A single-select dropdown menu with the value "1" selected.
- Example multiple select:** A multiple-select dropdown menu with options 1, 2, 3, 4, and 5. Option 1 is selected.
- Example textarea:** A text area with 3 rows and 1 column.

Gambar 14: 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 Bootstrap form with three input fields arranged horizontally:

- City:** An input field with the placeholder text "City".
- State:** An input field with the placeholder text "State".
- Zip:** An input field with the placeholder text "Zip".

Gambar 15: Forms Basic pada Bootstrap

```

1      <form>
2          <div class="form-row">
3              <div class="col-7">
4                  <input type="text" class="form-control" placeholder="City">
5              </div>
6              <div class="col">
7                  <input type="text" class="form-control" placeholder="State">
8              </div>
9              <div class="col">
10                 <input type="text" class="form-control" placeholder="Zip">
11             </div>
12         </div>
13     </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 16: Disabled Basic pada Bootstrap

```

1      <form>
2          <fieldset disabled>
3              <div class="form-group">
4                  <label for="disabledTextInput">Disabled input</label>
5                  <input type="text" id="disabledTextInput" class="form-control" placeholder="Disabled input">
6              </div>
7              <div class="form-group">
8                  <label for="disabledSelect">Disabled select menu</label>
9                  <select id="disabledSelect" class="form-control">
10                     <option>Disabled select</option>
11                 </select>
12             </div>
13             <div class="form-check">
14                 <input class="form-check-input" type="checkbox" id="disabledFieldsetCheck" disabled>
15                 <label class="form-check-label" for="disabledFieldsetCheck">
16                     Can't check this
17                 </label>
18             </div>
19             <button type="submit" class="btn btn-primary">Submit</button>
20         </fieldset>
21     </form>

```

Button

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



Gambar 17: Button pada Bootstrap

```

1      <button type="button" class="btn btn-primary">Primary</button>
2      <button type="button" class="btn btn-secondary">Secondary</button>
3      <button type="button" class="btn btn-success">Success</button>
4      <button type="button" class="btn btn-danger">Danger</button>
5      <button type="button" class="btn btn-warning">Warning</button>
6      <button type="button" class="btn btn-info">Info</button>
7      <button type="button" class="btn btn-light">Light</button>
8      <button type="button" class="btn btn-dark">Dark</button>
9
10     <button type="button" class="btn btn-link">Link</button>

```

Badge

Kelas `badge` dan `.badge-*` di dalam sebuah `<a>` akan memberikan badge yang dapat diberi atribut *hover* dan *focus*.

```

1 <a href="#" class="badge badge-primary">Primary</a>
2 <a href="#" class="badge badge-secondary">Secondary</a>
3 <a href="#" class="badge badge-success">Success</a>
4 <a href="#" class="badge badge-danger">Danger</a>
5 <a href="#" class="badge badge-warning">Warning</a>
6 <a href="#" class="badge badge-info">Info</a>
7 <a href="#" class="badge badge-light">Light</a>
8 <a href="#" class="badge badge-dark">Dark</a>

```



Gambar 18: Badge pada Bootstrap

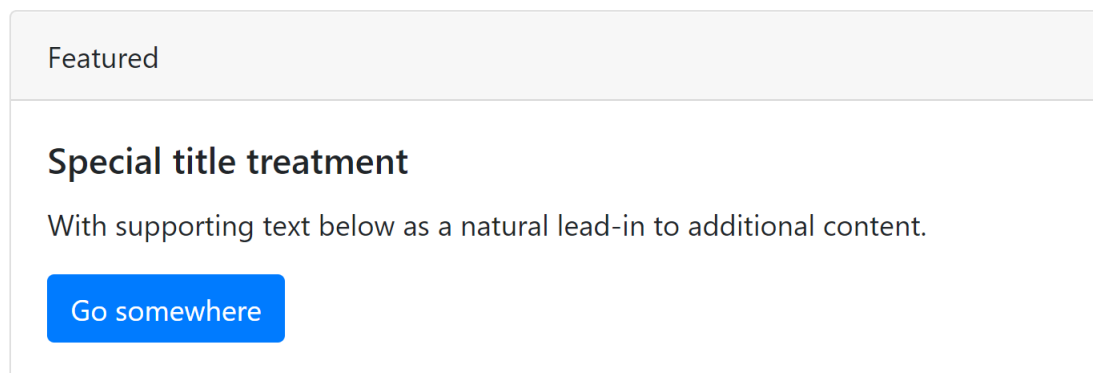
Card

Kelas `.card` adalah kontainer konten yang fleksibel dan bisa diatur lebarnya. Sebuah card memiliki sebuah *headers* dan *footers*.

```

1 <div class="card">
2   <div class="card-header">
3     Featured
4   </div>
5   <div class="card-body">
6     <h5 class="card-title">Special title treatment</h5>
7     <p class="card-text">With supporting text below as
8       a natural lead-in to additional content.</p>
9     <a href="#" class="btn btn-primary">Go somewhere</a>
10   </div>
11 </div>

```



Gambar 19: Card pada Bootstrap

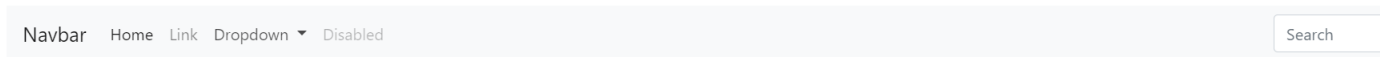
Navigation Bar

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 20: Navigation Bar pada Bootstrap

```

1      <nav class="navbar navbar-expand-lg navbar-light bg-light">
2          <a class="navbar-brand" href="#">Navbar</a>
3          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#
            navbarSupportedContent"
4              aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
5              <span class="navbar-toggler-icon"></span>
6          </button>
7
8          <div class="collapse navbar-collapse" id="navbarSupportedContent">
9              <ul class="navbar-nav mr-auto">
10                 <li class="nav-item active">
11                     <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
12                 </li>
13                 <li class="nav-item">
14                     <a class="nav-link" href="#">Link </a>
15                 </li>
16                 <li class="nav-item dropdown">
17                     <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"
18                         data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
19                         Dropdown
20                     </a>
21                     <div class="dropdown-menu" aria-labelledby="navbarDropdown">
22                         <a class="dropdown-item" href="#">Action </a>
23                         <a class="dropdown-item" href="#">Another action </a>
24                         <div class="dropdown-divider"></div>
25                         <a class="dropdown-item" href="#">Something else here </a>
26                     </div>
27                 </li>
28                 <li class="nav-item">
29                     <a class="nav-link disabled" href="#">Disabled </a>
30                 </li>
31             </ul>
32             <form class="form-inline my-2 my-lg-0">
33                 <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="
                    Search">
34                 <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search </button>
35             </form>
36         </div>
37     </nav>

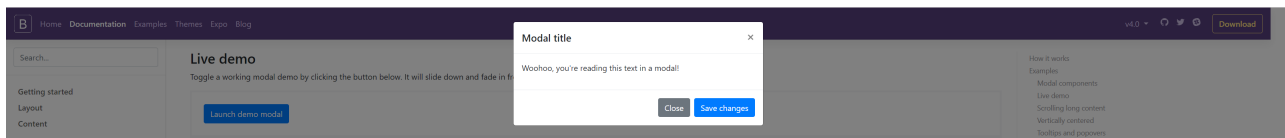
```

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 21: Modal pada Bootstrap

```

1      <!-- Button trigger modal -->
2      <button type="button" class="btn btn-primary" data-toggle="modal" data-target="#myModal">
3      Launch demo modal
4      </button>
5
6      <!-- Modal -->
7      <div class="modal fade" id="exampleModal" tabindex="-1" role="dialog"
8      aria-labelledby="exampleModalLabel" aria-hidden="true">
9          <div class="modal-dialog" role="document">
10             <div class="modal-content">
11                 <div class="modal-header">
12                     <h5 class="modal-title" id="exampleModalLabel">Modal title </h5>
13                     <button type="button" class="close" data-dismiss="modal" aria-label="Close">
14                         <span aria-hidden="true">&times;</span>
15                     </button>
16                 </div>
17                 <div class="modal-body">
18                     ...
19                 </div>
20                 <div class="modal-footer">
21                     <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
22                     <button type="button" class="btn btn-primary">Save changes</button>
23                 </div>
24             </div>
25         </div>
26     </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).

```

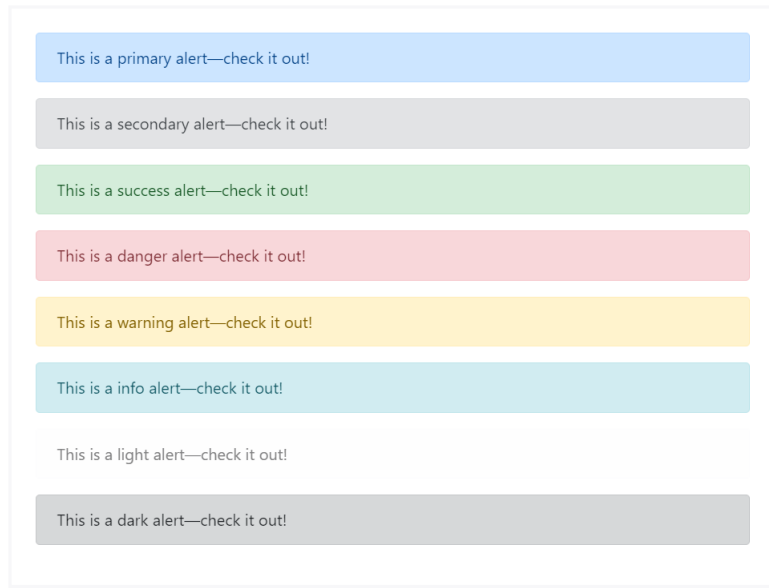
1      <i class="fa fa-coffee"></i>

```

Gambar 22: 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 23: Alert pada Bootstrap

```

1      <div class="alert alert-primary" role="alert">
2          This is a primary alert check it out!
3      </div>
4      <div class="alert alert-secondary" role="alert">
5          This is a secondary alert check it out!
6      </div>
7      <div class="alert alert-success" role="alert">
8          This is a success alert check it out!
9      </div>
10     <div class="alert alert-danger" role="alert">
11         This is a danger alert check it out!
12     </div>
13     <div class="alert alert-warning" role="alert">
14         This is a warning alert check it out!
15     </div>
16     <div class="alert alert-info" role="alert">
17         This is a info alert check it out!
18     </div>
19     <div class="alert alert-light" role="alert">
20         This is a light alert check it out!
21     </div>
22     <div class="alert alert-dark" role="alert">
23         This is a dark alert check it out!
24     </div>

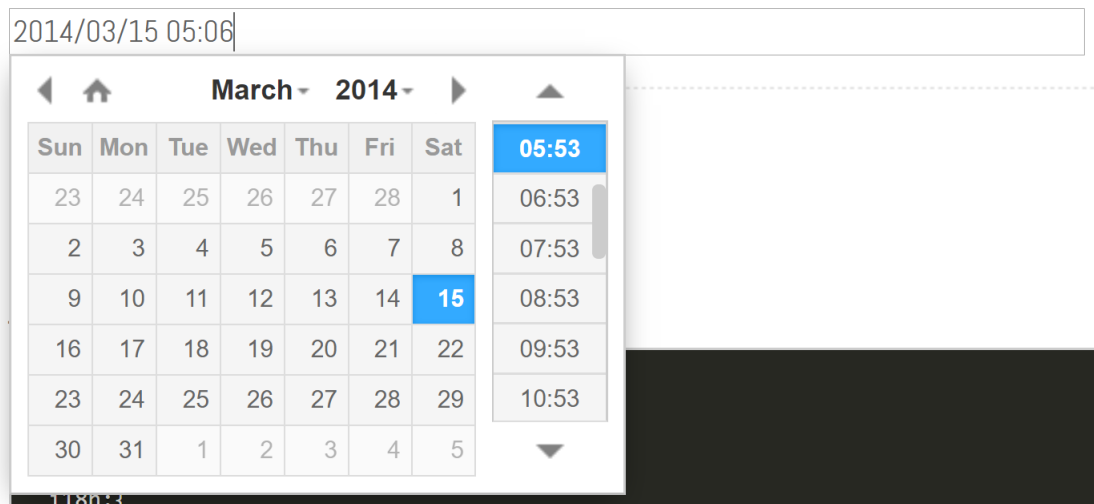
```

Plugin

DateTimePicker dengan menggunakan jQuery untuk memilih tanggal dan waktu pada bagian forms.

Inline DateTimePicker

Penggunaan plugin ini, memungkinkan *users* untuk memilih tanggal dan waktu secara bersamaan.



Gambar 24: Datetimepicker pada Bootstrap

Penggunaan nya dalam kode HTML sebagai berikut :

```
1 <input id="datetimepicker" type="text" >
```

Penggunaan dalam kode Javascript sebagai berikut :

```
1 jQuery('#datetimepicker').datetimepicker();
```

3. Membuat rancangan tampilan website dan template.

Status : Ada sejak rencana kerja skripsi.

Hasil : Belum diimplementasikan.

4. Mengimplementasikan keseluruhan rancangan tampilan dan template.

Status : Ada sejak rencana kerja skripsi.

Hasil : Sebagian besar tampilan sudah diimplementasi.

Berikut ini link perubahannya :

<https://github.com/ftisunpar/BlueTape/compare/master...hapsarilw:master>

5. Menganalisis website BlueTape.

Status : Baru dibuat saat ini.

Hasil :

Analisis Proyek BlueTape dijalankan menggunakan *framework front-end* Foundation. Secara garis besar, file-file yang berkaitan dengan Foundation seperti file javascript dan css akan dipanggil di file **BlueTape/www/application/views/templates/script_foundation.php** dan **BlueTape/www/application/views/templates/head_loggedin.php**. Kemudian penggunaan komponen Foundation terdapat pada file-file main.php yang terletak di setiap modul pada folder **BlueTape/www/application/views/** beserta kegunaan nya pada website.

Analisis Frontend Library

Foundation

Dalam proyek BlueTape, file Foundation tersimpan di folder js dan css. Foundation yang digunakan pada proyek ini adalah versi 6.1.2. Detail komponen yang digunakan adalah sebagai berikut :

(a) `.row`

- Untuk membuat konten yang terletak di dalam satu baris untuk setiap halaman website.
- Untuk memisahkan baris dari sekumpulan field dalam sebuah form.

(b) `.column`

- Membuat kolom untuk menampung konten
- Membuat kolom pada field dalam form

(c) `<table>`

- Seluruh tabel dalam proyek BlueTape memiliki format yang akan menyesuaikan posisinya dengan menampilkan data nya secara bertumpuk, sehingga dibutuhkan tag `<table>` dan kelas `.stack`. Tabel akan terdiri dari satu tag `<thead>` dan `<tbody>`.
- Bagian thead terdiri dari satu tag `<tr>` dan beberapa tag `<th>` yang membuat tulisan di dalam sel bersifat bold. Bagian tbody terdiri dari satu tag `<tr>` dan beberapa tag `<td>`.

Tabel yang menggunakan kelas ini sebagai berikut :

Tabel	Modul	Keterangan
Daftar Jadwal	Entri Jadwal Dosen	
Permohonan Perubahan Kuliah	Perubahan Kuliah Manage	
Detail Permohonan	Perubahan Kuliah Manage	Modal dari Aksi Lihat
Histori Permohonan	Perubahan Kuliah Request	
Permintaan Transkrip	Transkrip Manage	
Detail Permohonan	Transkrip Manage	Modal dari Aksi Lihat
Histori Permohonan	Transkrip Request	Modal dari Aksi Lihat

(d) Kelas `.callout` : Border untuk menampung setiap konten di website BlueTape seperti form dan tabel.(e) Kelas `.large-*`, `.medium-*` :

- Komponen medium-12 digunakan dalam callout, sehingga apabila layar berukuran sedang, callout akan memiliki lebar 12 grid.
- Sedangkan komponen large-* digunakan untuk mengatur lebar suatu input field dalam satu form. Lebar field berbeda - beda tergantung presentase suatu input field dalam satu baris.

(f) Tag `<form>` :

Ada dua jenis metode yang digunakan dalam form :

- POST : Digunakan untuk memasukan input user yang disertai oleh aksi yang memanggil suatu method tertentu dari controller. Dalam BlueTape form post digunakan untuk membuat request transkrip, setiap aksi untuk konfirmasi permohonan transkrip, permohonan baru perubahan kuliah.
- GET : Digunakan untuk mencari permintaan transkrip berdasarkan NPM.

Atribut yang digunakan dalam website BlueTape :

- `aria-label` : digunakan untuk memberi keterangan pada input field dalam form.
- `aria-hidden` : digunakan ketika ingin mengambil data dari database atau data dari akun yang teregeristrasi.
- `aria-selected` : digunakan apabila input terdiri dari beberapa pilihan dan menggunakan empa

(g) Reveal Modal digunakan untuk menampilkan data tertentu berdasarkan aksi yang dipilih. Terdapat empat aksi yang menggunakan modal yaitu modal lihat, setuju, tolak dan hapus.

- Modal memiliki kelas `.reveal` dan atribut `data-reveal`.
- Setiap modal akan memiliki tombol hapus. Penggunaan kelas `.close-button` dan atribut `data-close` akan diaplikasikan.

Template Flash Message

File ini terletak di **BlueTape**/`www/application/views/templates/flashmessage.php`, disini akan diletakkan tampilan alerts yang terdiri dari dua jenis :

Anda berhasil logout



Silahkan login di aplikasi BlueTape, dengan menekan tombol di bawah ini.
Gunakan akun `@unpar.ac.id` atau `@student.unpar.ac.id`.

Login dengan Google

[Petunjuk Penggunaan](#)

Gambar 25: Alert mengenai 'info' pada BlueTape

Mohon login terlebih dahulu.



Silahkan login di aplikasi BlueTape, dengan menekan tombol di bawah ini.
Gunakan akun `@unpar.ac.id` atau `@student.unpar.ac.id`.

Login dengan Google

[Petunjuk Penggunaan](#)

Gambar 26: Alert mengenai 'error' pada BlueTape

- Alert 'error' : Penggunaan kelas `.error` akan membuat komponen *alert* berwarna merah.
- Alert 'info' : Penggunaan kelas `.alert` akan membuat komponen *alert* berwarna biru.

Template Head Logged In

File ini terletak di **BlueTape**/www/application/views/templates/head_loggedin.php. Disini seluruh file css dari foundation dan plugin akan dimuat.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?><head>
    <meta charset="utf-8" />
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title><?= $this->config->item('module-names')[$currentModule] ?></title>
    <link rel="stylesheet" href="/public/css/foundation.css" />
    <link rel="stylesheet" href="/public/css/foundation-icons.css" />
    <link rel="stylesheet" href="/public/css/app.css" />
    <link rel="stylesheet" href="/public/lib/xdan-datetimepicker/jquery.datetimepicker.min.css" />
</head>
```

Gambar 27: Kode template untuk file head_loggedin.php

Template Script Foundation

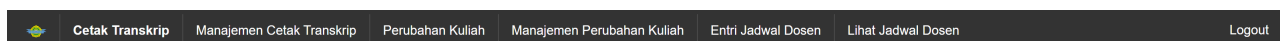
File ini terletak di **BlueTape**/www/application/views/templates/script_loggedin.php. Disini seluruh file js dari foundation, jQuery dan plugin akan dimuat.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?><script src="/public/js/vendor/jquery.min.js"></script>
<script src="/public/js/vendor/what-input.min.js"></script>
<script src="/public/js/foundation.min.js"></script>
<script src="/public/js/app.js"></script>
<script src="/public/lib/xdan-datetimepicker/jquery.datetimepicker.full.min.js"></script>
```

Gambar 28: Kode template untuk file script_foundation.php

5.1 Template Top Bar Logged In

File ini terletak di **BlueTape**/www/application/views/templates/topbar_loggedin.php.



Gambar 29: Tampilan navbar dengan foundation

Antarmuka Cetak Transkrip

Antarmuka diaplikasikan pada file **BlueTape**/www/application/views/TranskripRequest/main.php. Isi dari halaman antarmuka cetak transkrip terdiri dari dua bagian yaitu :

- Permohonan Baru** : Sistem akan memberikan dua tampilan untuk bagian ini, dengan kondisi sebagai berikut:
 - Sistem akan menampilkan form pengajuan transkrip, apabila mahasiswa belum pernah mengajukan permohonan atau pengajuan sebelumnya dikonfirmasi staf TU, maka mahasiswa dapat mengajukan permohonan baru.
 - Sistem akan menampilkan informasi "Anda tidak bisa meminta cetak karena ada permintaan lain yang belum selesai", apabila mahasiswa memiliki pengajuan permohonan transkrip yang belum dikonfirmasi staf TU.
- Histori Permohonan** : Tabel untuk menampilkan informasi permohonan transkrip seorang mahasiswa. Status, tanggal pembuatan, tipe transkrip, tanggal cetak keterangan dan aksi.

Desain antarmuka sebagai berikut :

Konten 'Permohonan Baru' dan 'Histori Permohonan' akan diletakkan pada satu *row* yang memiliki kolom sebesar 12 grid pada layar medium dengan menggunakan komponen `medium-12 column`, untuk setiap konten nya akan dipisahkan oleh panel yang disebut dengan *callout*.

Untuk konten Permohonan Baru :

- Bagian isi akan memiliki dua tampilan yaitu berbentuk form atau notifikasi yang berbentuk paragraf. Sehingga dibutuhkan komponen `<form>` dan `<p>`.
- Form terdiri dari dua baris yang dipisahkan oleh komponen `row`.
- Untuk field pada baris pertama akan memiliki kolom dengan panjang 4 grid. Sehingga dibutuhkan kelas `large-4 column`. Jenis input yang digunakan bertipe email dan text. Selain itu untuk setiap input akan disertakan satu label sehingga menggunakan komponen `<label>`.
- Untuk field pada baris kedua memiliki dua jenis lebar grid yaitu 4 grid dan 8 grid. Sehingga membutuhkan kelas `large-4 column`, `large-8 column`. Untuk field pada baris ketiga, tombol "Kirim Permohonan" memiliki *background color* berwarna biru sehingga akan digunakan kelas `button`.

Untuk konten "Histori Permohonan" :

- Pada kolom "Status" akan memiliki tiga jenis bentuk alert dengan *background* warna hijau, abu-abu dan merah. Sehingga dibutuhkan kelas `warning`, `success`, `alert`.
- Aksi memiliki satu ikon "lihat" berwarna biru yang akan menampilkan sebuah modal. Sehingga dibutuhkan kelas ikon `fi-eye`.

The screenshot shows the 'Permohonan Baru' form with fields for 'Yang memohon' (Email: amihapsahapsa@gmail.com), 'NPM' (Hapsari Laksmi Wijayanti), and 'Tipe Transkrip' (DPS Bahasa Indonesia (Seluruh Semester)). A 'Kirim Permohonan' button is present. Below the form is the 'Histori Permohonan' table:

ID	Status	Tanggal Permohonan	Tipe Transkrip	Tanggal Jawab/Cetak	Keterangan	Aksi
#3	TERCEKSA	Kamis, 2 Mei 2019	DPS_ID	Minggu, 27 Oktober 2019	tercetak	

Gambar 30: Antarmuka Cetak Transkrip bagian 1

The screenshot shows the 'Permohonan Baru' form with a message: 'Anda tidak bisa meminta cetak karena ada permintaan lain yang belum selesai.' Below the form is the 'Histori Permohonan' table:

ID	Status	Tanggal Permohonan	Tipe Transkrip	Tanggal Jawab/Cetak	Keterangan	Aksi
#3	TUNGGU	Kamis, 2 Mei 2019	DPS_ID			

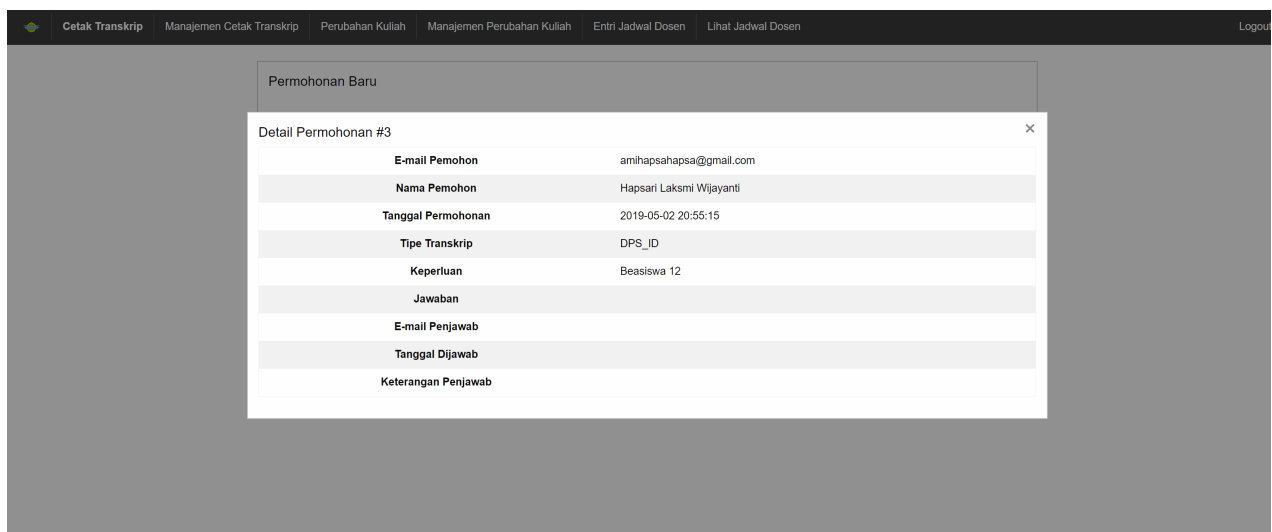
Gambar 31: Antarmuka Cetak Transkrip bagian 2

Detail dari form permohonan baru, semua field akan disertai sebuah label untuk menjelaskan fungsi setiap field sehingga membutuhkan tag `<label>` :

- Yang memohon : Berisi email UNPAR mahasiswa, otomatis terisi saat login melalui gmail. Sehingga field tidak bisa diisi atau *disabled* dibutuhkan atribut boolean `readonly`

- **NPM** : Berisi NPM mahasiswa yang ter-*generate* secara otomatis. Sehingga field tidak bisa diisi atau *disabled* dibutuhkan atribut boolean `readonly` dan `<input>` dengan tipe text.
- **Nama** : Nama mahasiswa yang tergenerate secara otomatis. Sehingga field tidak bisa diisi atau *disabled* dibutuhkan atribut boolean `readonly` dan `<input>` dengan tipe text.
- **Tipe Transkrip** : Terdiri dari tiga pilihan yaitu DPS Bahasa Indonesia(Seluruh Semester), DPS Bahasa Inggris(Seluruh Semester), LHS (Semester Terakhir). Wajib diisi. Tag `<select>` dan `<option>` yang *value* nya diambil dari variabel `$type`.
- **Keperluan** : Keterangan keperluan dibuat nya transkrip, wajib diisi mahasiswa. Menggunakan tag `<input>` dengan tipe text.

Apabila ada form yang belum diisi maka akan terdapat *warning* untuk *field* yang kosong. Berikut ini apabila mahasiswa menekan tombol aksi lihat 👁 :

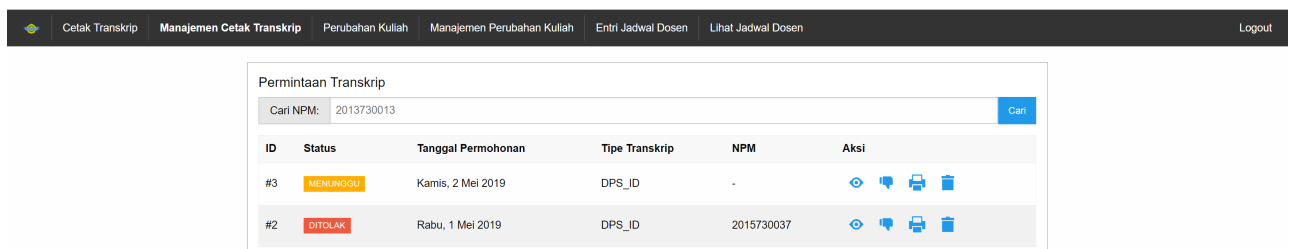


Gambar 32: Modal Lihat Cetak Transkrip

Disini aksi 'lihat' akan menampilkan sebuah modal yang berisi sebuah tabel bergaris yang menyimpan informasi detil permohonan, baik detil informasi dari mahasiswa maupun konfirmasi dari staf Tata Usaha. Sehingga dibutuhkan kelas `.reveal` dan atribut `data-reveal`.

Antarmuka Manajemen Cetak Transkrip

Antarmuka diaplikasikan pada file `BlueTape/www/application/views/TranskripManage/main.php`.



Gambar 33: Tampilan Manajemen Cetak Transkrip

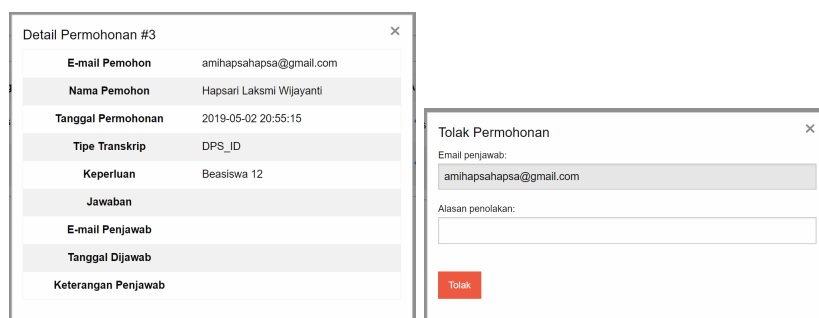
Tampilan Manajemen Cetak Transkrip berisi sebuah tabel permintaan transkrip yang terdiri dari daftar permintaan transkrip dan form pencarian transkrip berdasarkan NPM.

Detail penjelasan untuk field 'Status' dan 'Aksi' :

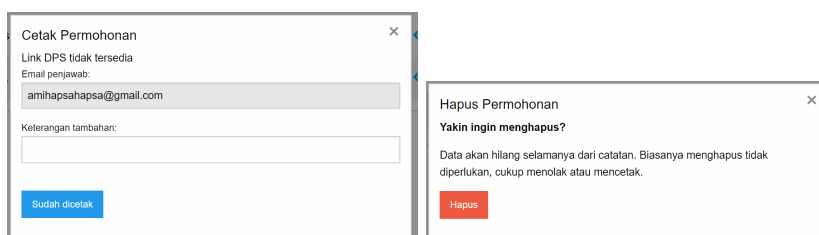
- **Status** : Output terdiri dari tiga jenis label yaitu 'MENUNGGU'(berwarna kuning), 'DITOLAK'(berwarna merah) dan 'TERCETAK'(berwarna hijau). Sehingga dibutuhkan kelas `secondary`, `success`, `alert`.
- **Aksi** : Terdiri dari empat ikon font-awesome yaitu `fi-eye`, `fi-dislike`, `fi-print`, `fi-trash`, yang akan menampilkan modal berisi informasi yang sesuai dengan perintah.

Modal akan menggunakan kelas `reveal` dan atribut `data-reveal` Detail penjelasan untuk modal :

- Modal Lihat : Terdiri dari sebuah `table` yang menampilkan data permintaan transkrip. Ikon menggunakan kelas `fi-eye` dan menerapkan atribut `data-open` yang berisi method hapus menuju ID tertentu.
- Modal Tolak : Terdiri dari sebuah `form` yang memiliki method `POST` yang memanggil sebuah method `"/TranskripManage/answer"`. Terdapat tiga tipe input yang digunakan yaitu `hidden`, `text`, `submit`. Pada input text untuk label Alasan Penolakan, menggunakan kelas `input-group-field`. Lalu untuk input bertipe `submit` menggunakan kelas `alert-button` untuk membuat button berwarna merah. Ikon menggunakan kelas `fi-dislike` dan menerapkan atribut `data-open` yang berisi method tolak menuju ID tertentu.
- Modal Print : Terdiri dari sebuah form yang terdiri dari label, field dan tombol berwarna biru sehingga membutuhkan kelas `input-group-field`. Ikon menggunakan kelas `fi-print` dan menerapkan atribut `data-open` yang berisi method cetak menuju ID tertentu.
- Modal Hapus : Terdiri dari sebuah form yang terdiri dari paragraf yang bersifat bold, beberapa input dan tombol berwarna merah sehingga membutuhkan kelas `<p>`, ``, `<input>`. Ikon menggunakan kelas `fi-trash` dan menerapkan atribut `data-open` yang berisi method hapus menuju ID tertentu.



Gambar 34: Tampilan Modal untuk aksi 'Lihat' dan 'Tolak'



Gambar 35: Tampilan Modal untuk aksi 'Print' dan 'Hapus'

Antarmuka Perubahan Kuliah

Antarmuka diaplikasikan pada file **BlueTape/wwww/application/views/PerubahanKuliahRequest/main.php**.

ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Tanggal Jawab	Keterangan	Aksi
#1	DITOLAK	Rabu, 1 Mei 2019	AIF123	Ditladakan	Rabu, 1 Mei 2019	tidak bisa	👁

Gambar 36: Tampilan Perubahan Kuliah

Modul Perubahan Kuliah terdiri dari dua tabel yaitu :

Permohonan Baru

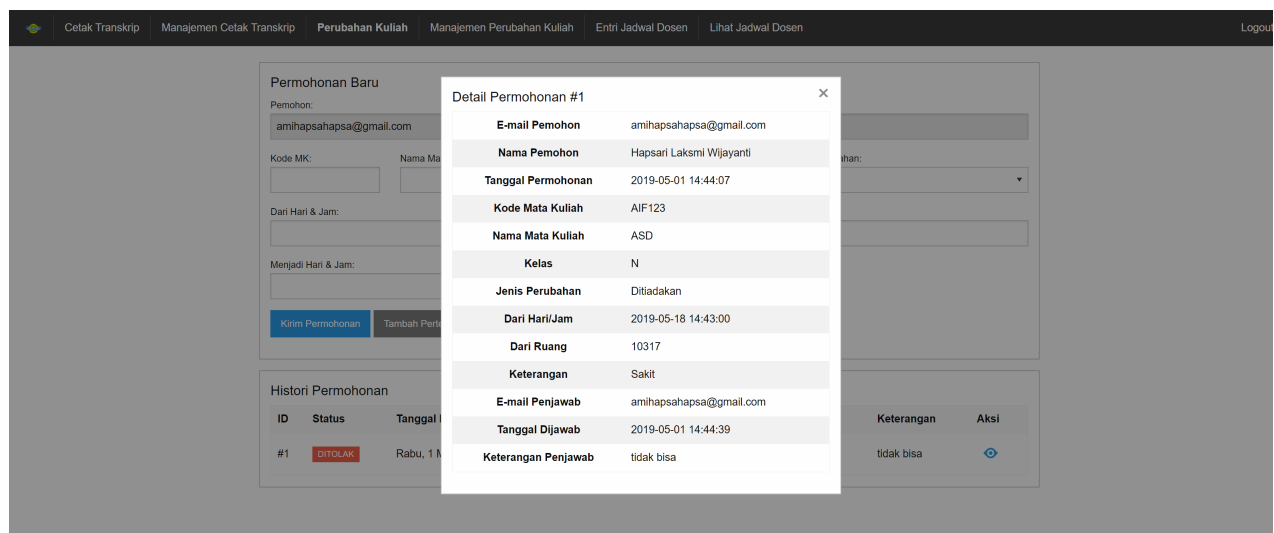
Permohonan baru diletakkan dalam sebuah row selebar 12 grid dan dikelilingi oleh sebuah border, membutuhkan kelas `row`, `.large-12`, `column` dan `callout`. Selain itu sebuah form dengan tipe POST akan ditampilkan untuk menyimpan masukan pemohon. Format dari form sebagai berikut :

Histori Pemohonan

Histori Permohonan menggunakan tabel yang bisa menyesuaikan lebarnya dengan menampilkan data secara bertumpuk sehingga menggunakan kelas `.stack`. Jenis data akan diletakkan dalam tag `colorboxmygraytexttt dalam satu baris sehingga menggunakan satu tag <tr> dan delapan tag <th>. Untuk bagian isi data menggunakan tag <tbody>.`

Nama Kolom	Pilihan	Kelas Foundation yang digunakan
Status	dikonfirmasi, menggunakan kelas <code>.success</code>	Apabila staf TU menyetujui permohonan
	ditolak, menggunakan kelas <code>.alert</code>	Apabila staf TU menolak permohonan
	ditunggu, menggunakan kelas <code>.secondary</code>	Apabila staf TU belum konfirmasi permohonan
Tanggal Permohonan		Data bertipe tanggal dengan format yang sudah ditentukan
Kode MK		Data berbentuk text
Perubahan		Data berbentuk text
Tanggal Jawab		Data bertipe tanggal
Keterangan		Data berbentuk text
Aksi		Terdapat tombol aksi 'Lihat', menggunakan font awesome dan kelas <code>fas fa-eye</code> yang akan memanggil sebuah modal sesuai id yang diinginkan user. Untuk modal, menggunakan kelas <code>reveal</code> dan atribut <code>data-reveal</code>

Ketika user menekan tombol aksi 'lihat', maka modal berisi sebuah tabel informasi data permohonan akan ditampilkan sesuai dengan ID.








Gambar 37: Modal Lihat Perubahan Kuliah

Antarmuka Manajemen Perubahan Kuliah

Antarmuka diaplikasikan pada file `BlueTape/www/application/views/PerubahanKuliahManage/main.php`.

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

ID	Status	Tanggal Permohonan	Kode MK	Perubahan	Aksi
#1	DITOLAK	Rabu, 1 Mei 2019	AIF123	Ditiadakan	    

Gambar 38: Tampilan Manajemen Perubahan Kuliah

Tabel Pemohonan Kuliah memiliki detail yang sama dengan tabel histori permohonan, namun aksi yang dilakukan terdiri dari lima perintah:

Nama Kolom	Pilihan	Keterangan
Status	dikonfirmasi, menggunakan kelas <code>success</code>	Apabila staf TU menyetujui permohonan
	ditolak, menggunakan kelas <code>alert</code>	Apabila staf TU menolak permohonan
	ditunggu, menggunakan kelas <code>secondary</code>	Apabila staf TU belum konfirmasi permohonan
Tanggal Permohonan		Data bertipe tanggal dengan format yang sudah ditentukan
Kode MK		Menampilkan data berbentuk text
Perubahan		Menampilkan data berbentuk text
Tanggal Jawab		Menampilkan data bertipe tanggal
Keterangan		Menampilkan data berbentuk text
Aksi	ikon Lihat	Ikon menggunakan font awesome dan kelas <code>fas fa-eye</code> yang akan memanggil sebuah modal sesuai id yang diinginkan user. Untuk modal, menggunakan kelas <code>reveal</code> dan atribut <code>data-reveal</code>
	ikon Print	Ikon menggunakan font awesome dan kelas <code>fas fa-print</code> yang akan memanggil sebuah modal sesuai id yang diinginkan user. Untuk modal, menggunakan kelas <code>reveal</code> dan atribut <code>data-reveal</code>
	ikon Setuju	Ikon menggunakan font awesome dan kelas <code>fa-thumbs-up</code> yang akan memanggil sebuah modal sesuai id yang diinginkan user. Untuk modal, menggunakan kelas <code>reveal</code> dan atribut <code>data-reveal</code>
	ikon Tolak	Ikon menggunakan font awesome dan kelas <code>fa-thumbs-down</code> yang akan memanggil sebuah modal sesuai id yang diinginkan user. Untuk modal, menggunakan kelas <code>reveal</code> dan atribut <code>data-reveal</code>
	ikon Hapus	Ikon menggunakan font awesome dan kelas <code>fas fa-trash</code> yang akan memanggil sebuah modal sesuai id yang diinginkan user. Untuk modal, menggunakan kelas <code>reveal</code> dan atribut <code>data-reveal</code>

The image shows two screenshots. On the left is a modal titled 'Detail Permohonan #1' with a close button (X). It contains a table with the following data:

E-mail Pemohon	amhaphaspa@gmail.com
Nama Pemohon	Hapsari Laksmi Wijayanti
Tanggal Permohonan	2019-05-01 14:44:07
Kode Mata Kuliah	AF123
Nama Mata Kuliah	ASD
Kelas	N
Jenis Perubahan	Ditajukan
Dari Harijam	2019-05-15 14:43:00
Dari Ruang	10317
Keterangan	Sakit
E-mail Penjawab	amhaphaspa@gmail.com
Tanggal Dijawab	2019-05-01 14:44:39
Keterangan Penjawab	tidak bisa

On the right is a page titled 'PENGUMUMAN' with a subtitle 'Jadwal Kuliah'. It contains a table with the following data:

Mata Kuliah	: ASD
Kode MK	: AF123
Kelas	: N
Hari	: Sabtu, 18 Mei 2019
Jam	: 14:43
Ruang	: 10317
Keterangan	: Sakit
Disetujui	: 27-05-2019
Administri FTN	

Gambar 39: Modal aksi Lihat dan Print Manajemen Perubahan Kuliah

The image shows two modals side-by-side. The left modal is titled 'Konfirmasi Permohonan' and has a close button (X). It contains a table with the following data:

E-mail pemohon	amhaphaspa@gmail.com
Permohonan	

The right modal is titled 'Tolak Permohonan' and has a close button (X). It contains a table with the following data:

E-mail pemohon	amhaphaspa@gmail.com
Alasan penolakan	

Gambar 40: Modal aksi Setuju dan Tolak Manajemen Perubahan Kuliah

The image shows a modal titled 'Hapus Permohonan' with a close button (X). It contains a table with the following data:

E-mail pemohon	amhaphaspa@gmail.com
Permohonan	

Gambar 41: Modal Hapus Manajemen Perubahan Kuliah

Berikut ini penjelasan masing - masing modal.

- Modal Lihat : Terdiri dari sebuah tabel yang menampilkan data tabel permohonan. Ikon menggunakan kelas `fi-eye` dan menerapkan atribut `data-open` yang berisi method hapus menuju ID tertentu.
- Modal Print : Terdiri dari sebuah form yang terdiri dari label, field dan tombol berwarna biru sehingga membutuhkan kelas `input-group-field`. Ikon menggunakan kelas `fi-print` dan menerapkan atribut `data-open` yang berisi method cetak menuju ID tertentu.
- Modal Setuju : Terdiri dari sebuah `table` yang menampilkan data permintaan transkrip. Ikon menggunakan kelas `fi-eye` dan menerapkan atribut `data-open` yang berisi method hapus menuju ID tertentu.
- Modal Tolak : Terdiri dari sebuah `form` yang memiliki method `POST` yang memanggil sebuah method `"/TranskripManage/answer"`. Terdapat tiga tipe input yang digunakan yaitu `hidden`, `text`, `submit`. Pada input `text` untuk label Alasan Penolakan, menggunakan kelas `input-group-field`. Lalu untuk input bertipe `submit` menggunakan kelas `alert-button` untuk membuat button berwarna merah. Ikon menggunakan kelas `fi-dislike` dan menerapkan atribut `data-open` yang berisi method tolak menuju ID tertentu.
- Modal Hapus : Terdiri dari sebuah form yang terdiri dari paragraf yang bersifat bold, beberapa input dan tombol berwarna merah sehingga membutuhkan kelas `<p>`, `` dan `<input>`. Ikon menggunakan kelas `fi-trash` dan menerapkan atribut `data-open` yang berisi method hapus menuju ID tertentu.

Antarmuka Entri Jadwal Dosen

Antarmuka diaplikasikan pada file `BlueTape/www/application/views/EntriJadwalDosen/main.php`.

Gambar 42: Modal Print Manajemen Perubahan Kuliah

Detail mengenai tabel Tambah Jadwal :

- **Hari** : Terdiri dari nama hari dari senin sampai jumat.
- **Durasi** : Terdiri dari rentang jam kelas berlangsung dari 1 jam hingga 9 jam.
- **Label** : Field bertipe text.
- **Jam Mulai** : Terdiri dari jam dari rentang 07:00 sampai 16:00.
- **Jenis** : Terdiri dari tiga macam pilihan
 - (a) Konsultasi : Memiliki background berwarna hijau.
 - (b) Terjadwal : Memiliki background berwarna biru.
 - (c) Kelas : Memiliki background putih.

Desain antarmuka sebagai berikut :

Konten "Tambah Jadwal" dan "Daftar Jadwal" akan diletakkan pada satu *row* yang memiliki kolom sebesar 12 grid pada layar medium dengan menggunakan komponen `large-12 column`, untuk setiap konten nya akan dipisahkan oleh panel yang disebut dengan `callout`.

Untuk konten Tambah Jadwal :

- **Hari** : Penggunaan tag `<select>` dan `<option>`.
- **Jam Mulai** : Penggunaan tag `<select>` dan `<option>`.
- **Durasi** : Penggunaan tag `large-4 columns`

Tabel Daftar jadwal akan *retrieve* data jadwal dari dosen yang dibuat. Terdiri dari rentang waktu dan hari. Jadwal yang terlihat pada tabel ini bisa diedit dan dihapus. Menggunakan kelas `large-12 column`, `.callout` dan `table-scroll`. Dibagian bawah tabel akan terlihat tanggal jadwal tersebut di update dan memiliki dua tombol :

- **Delete All** : Menghapus semua jadwal yang sudah dibuat, tombol berwarna merah dengan menerapkan kelas `.alert`.
- **Export ke XLS** : Secara otomatis akan membuat file excel dan mendownload di device secara lokal. Tombol berwarna biru dengan menerapkan kelas `button`.

Setiap data yang ditampilkan bisa diedit atau hapus, apabila ditekan maka sebuah modal akan muncul.

Gambar 43:

Modal merupakan sebuah form dengan metode "POST". Untuk field Hari, Jam Mulai, Durasi, Jenis dan label memiliki input bertipe `select`. Lalu terdapat dua button Save dan Submit yang masing - masing menggunakan kelas tag `<button>` dan `.alert`.

Antarmuka Lihat Jadwal Dosen

Antarmuka diaplikasikan pada file `BlueTape/www/application/views/LihatJadwalDosen/main.php`.

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

Hapsari Laksmi Wigyantri						
	Senin	Selasa	Rabu	Kamis	Jumat	
7-8						
8-9						
9-10						
10-11		ADBO				
11-12						
12-13						
13-14						
14-15						
15-16						
16-17						

Terakhir diupdate pada : 01 Mei 2019

[Ekspor ke XLS](#)

Gambar 44: Struktur File Zurb Foundation

Tabel Jadwal Dosen terdiri dari list nama dosen yang dibuat dalam bentuk tabs sehingga menggunakan kelas `tabs-panel`. Apabila dipilih sebuah tabs maka tabel akan menampilkan data jadwal dosen. Terdiri dari rentang waktu dan hari. Dibagian bawah tabel terdapat tanggal kapan terakhir jadwal dibuat dan tombol "Ekspor ke XLS".

6. Menulis dokumen skripsi.

Status : Masih dalam pengerjaan.

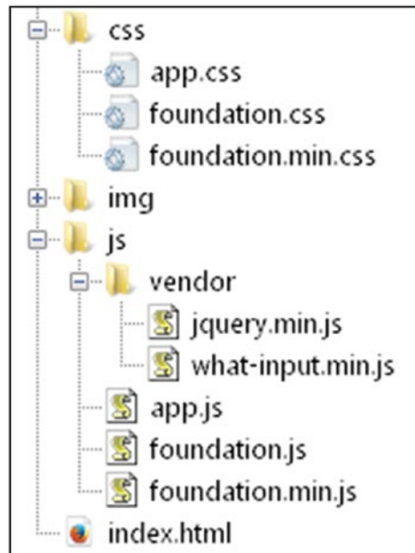
Hasil :

Detail Pengerjaan Skripsi sebagai berikut :

- Bab 1 Pendahuluan : Dijelaskan pada Latar Belakang
- Bab 2 Landasan Teori : Dijelaskan pada poin 1 dan 2.
- Bab 3 Analisis : Dijelaskan pada poin nomor 5.

Berikut ini landasan teori untuk framework Foundation :

Struktur File



Gambar 45: 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 yaitu *foundation.min.css* atau versi yang tidak dikompresi *foundation.css*. Lalu seluruh modifikasi *stylesheets* ditempatkan didalam folder ini agar lebih terstruktur.
- Folder **img** tempat meletakkan semua gambar untuk proyek web.
- Folder **js** terdiri dari semua file Javascript.

3

Sistem Grid pada Foundation

Penggunaan grid pada Foundation dapat dilakukan dengan menambahkan sebuah elemen dengan sebuah kelas `.row` sehingga akan membuat blok horizontal yang berisi kolom vertikal. Kemudian kelas `.column` akan ditambahkan pada baris tersebut, lalu masing-masing kolom ditentukan kelasnya dengan tiga pilihan yaitu `.small-#`, `.medium-#` dan `.large-#`.

```

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

```

³Foundation 6 (2017) Foundation v6.3.1 Documentation. ZurbInc. 100 W RinconAve, Campbell, CA 95008, USA.

```

10     </div>
11     <div class="row">
12         <div class="columns small-6 large-2"><!-- ... --></div>
13         <div class="columns small-6 large-8"><!-- ... --></div>
14         <div class="columns small-12 large-2"><!-- ... --></div>
15     </div>
16     <div class="row">
17         <div class="columns small-3"><!-- ... --></div>
18         <div class="columns small-9"><!-- ... --></div>
19     </div>
20     <div class="row">
21         <div class="columns large-4"><!-- ... --></div>
22         <div class="columns large-8"><!-- ... --></div>
23     </div>
24     <div class="row">
25         <div class="columns small-6 large-5"><!-- ... --></div>
26         <div class="columns small-6 large-7"><!-- ... --></div>
27     </div>
28     <div class="row">
29         <div class="columns large-6"><!-- ... --></div>
30         <div class="columns large-6"><!-- ... --></div>
31     </div>

```

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

Gambar 46: Grid pada Zurb Foundation

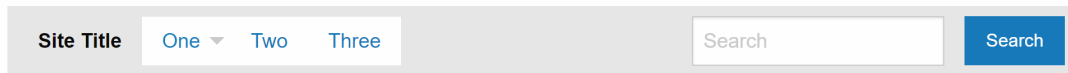
Top Bar

Top bar digunakan untuk membuat menu responsif dengan memberikan ID unik pada menu, Lalu judul menu akan menggunakan kelas `.title-bar` dan atribut `data-responsive-toggle`. Lalu untuk *trigger* toggle, maka dapat menggunakan elemen `data-toggle` dengan menyertakan ID unik terhadap menu yang ingin dialihkan. Secara default, judul bar akan terlihat pada layar kecil, dan menu akan di *hide*. Pada layar medium, judul bar akan menghilang, dan menu selalu terlihat dengan menerapkan atribut `data-hide-for`.

```

1     <div class="title-bar" data-responsive-toggle="responsive-menu" data-hide-for="medium">
2         <button class="menu-icon" type="button" data-toggle="responsive-menu"></button>
3         <div class="title-bar-title">Menu</div>
4     </div>
5
6     <div class="top-bar" id="responsive-menu">
7         <div class="top-bar-left">
8             <ul class="dropdown menu" data-dropdown-menu>
9                 <li class="menu-text">Site Title</li>
10                <li class="has-submenu">
11                    <a href="#0">One</a>
12                    <ul class="submenu menu vertical" data-submenu>
13                        <li><a href="#0">One</a></li>
14                        <li><a href="#0">Two</a></li>
15                        <li><a href="#0">Three</a></li>
16                    </ul>
17                </li>
18                <li><a href="#0">Two</a></li>
19                <li><a href="#0">Three</a></li>
20            </ul>
21        </div>
22        <div class="top-bar-right">
23            <ul class="menu">
24                <li><input type="search" placeholder="Search"></li>
25                <li><button type="button" class="button">Search</button></li>
26            </ul>
27        </div>
28    </div>

```



Gambar 47: Top Bar pada Zurb Foundation

Button

Basic button dapat digunakan untuk banyak tujuan, sehingga penting untuk *developer* menggunakan tag yang tepat. Berikut ini penjelasan penggunaan *Basic button* dalam Foundation

- Penggunaan tag `<a>` digunakan apabila tombol memiliki *link* ke halaman lain, atau *link* menuju ke halaman itu sendiri. Penggunaan links tidak membutuhkan JavaScript.
- Penggunaan tag `<button>` jika tombol melakukan tindakan yang mengubah sesuatu pada halaman seperti proses *delete* atau *save*. Elemen `<button>` akan membutuhkan JavaScript agar proses tersebut berfungsi.

Warna pada button dapat diterapkan untuk memperlihatkan fungsi yang sesuai dengan aksi yang digunakan.

```

1 <a class="button primary" href="#">Primary</a>
2 <a class="button secondary" href="#">Secondary</a>
3 <a class="button success" href="#">Success</a>
4 <a class="button alert" href="#">Alert</a>
5 <a class="button warning" href="#">Warning</a>

```



Gambar 48: Coloring Button pada Foundation

Tabel

Tabel dalam Foundation akan menjadikan proses penampilan data bersifat responsif dan memiliki tata letak yang bisa disesuaikan oleh kebutuhan *developer*.

```

1 <table>
2   <thead>
3     <tr>
4       <th width="200">Table Header</th>
5       <th>Table Header</th>
6       <th width="150">Table Header</th>
7       <th width="150">Table Header</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>Content Goes Here</td>
13      <td>This is longer content Donec id elit non mi porta gravida at eget metus.</td>
14      <td>Content Goes Here</td>
15      <td>Content Goes Here</td>
16    </tr>
17    <tr>
18      <td>Content Goes Here</td>
19      <td>This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.</td>
20      <td>Content Goes Here</td>
21      <td>Content Goes Here</td>
22    </tr>
23    <tr>
24      <td>Content Goes Here</td>
25      <td>This is longer Content Goes Here Donec id elit non mi porta gravida at eget metus.</td>
26      <td>Content Goes Here</td>
27      <td>Content Goes Here</td>
28    </tr>
29  </tbody>
30 </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 49: Basic Table pada Foundation

Hover State

Hover State diaplikasikan menggunakan kelas `.hover` untuk sedikit membedakan baris terpilih dalam tabel dengan baris-baris lainnya dengan cara menggelapkan baris terpilih.

```
1 <table class="hover">
2
3 </table>
```

Forms

Forms pada Foundation dibuat dengan kombinasi standar dari elemen `form`, serta *grid rows* dan *columns* atau *cells*.

Text Inputs

Kode berikut ini akan membuat sebuah *text field* yang bisa diterapkan untuk *field* : `text`, `datetime`, `date`, `email`, `month`, `search`, `time`.

```
1 <form>
2   <div class="grid-container">
3     <div class="grid-x grid-padding-x">
4       <div class="medium-6 cell">
5         <label>Input Label
6         <input type="text" placeholder=".medium-6.cell">
7       </label>
8     </div>
9     <div class="medium-6 cell">
10      <label>Input Label
11      <input type="text" placeholder=".medium-6.cell">
12    </label>
13  </div>
14 </div>
15 </form>
```

Input Label

`.medium-6.cell`

Input Label

`.medium-6.cell`

Gambar 50: Text Input pada Foundation

Select Menus

Penggunaan `<select>` digunakan apabila *developer* menginginkan banyak pilihan data dalam satu menu.

```
1 <label>Select Menu
2   <select>
3     <option value="husker">Husker</option>
4     <option value="starbuck">Starbuck</option>
5     <option value="hotdog">Hot Dog</option>
6     <option value="apollo">Apollo</option>
7   </select>
8 </label>
```

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.

```

1      <ul class="tabs" data-tabs id="tab_component">
2          <li class="tabs-title"><a href="#pub1">Section 1</a></li>
3          <li class="tabs-title is-active"><a href="#pub2">Section 2</a></li>
4          <li class="tabs-title"><a href="#pub3">Section 3</a></li>
5          <li class="tabs-title"><a href="#pub4">Section 4</a></li>
6      </ul>
7      <div class="tabs-content" data-tabs-content="tab_component">
8          <div class="tabs-panel" id="pub1">
9              <p>Far far away, behind the word mountains, far from the countries
10                 Vokalia and Consonantia, there live the blind texts.</p>
11          </div>
12          <div class="tabs-panel is-active" id="pub2">
13              <p>Separated they live in Bookmarksgrove right at the coast of the
14                 Semantics, a large language ocean. </p>
15          </div>
16          <div class="tabs-panel" id="pub3">
17              <p>A small river named Duden flows by their place and supplies it with
18                 the necessary regalia.</p>
19          </div>
20          <div class="tabs-panel" id="pub4">
21              <p>It is a paradisematic country, in which roasted parts of sentences
22                 fly into your mouth. </p>
23          </div>
24      </div>

```



Gambar 51: Grid pada Zurb Foundation

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*.

```

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

```

6 Pencapaian Rencana Kerja

Langkah-langkah kerja yang berhasil diselesaikan dalam Skripsi 1 ini adalah sebagai berikut:

- Mempelajari framework PHP Codeigniter.
- Mempelajari framework front-end Bootstrap 4 beserta plugin-plugin yang tersedia.
- Mengimplementasikan sebagian besar framework front-end Bootstrap 4 beserta plugin-plugin yang tersedia.
- Menulis dokumen skripsi.

Hapsari Laksmi W

Menyetujui,

Nama: Pascal Alfadian Nugroho
Pembimbing Tunggal