

SKRIPSI

MIGRASI ZURB FOUNDATION KE BOOTSTRAP 4



Hapsari Laksmi W

NPM: 2015730037

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN

«tahun»

UNDERGRADUATE THESIS

MIGRATION FROM ZURB FOUNDATION TO BOOTSTRAP 4



Hapsari Laksmi W

NPM: 2015730037

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY

«tahun»

LEMBAR PENGESAHAN

MIGRASI ZURB FOUNDATION KE BOOTSTRAP 4

Hapsari Laksmi W

NPM: 2015730037

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

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

MIGRASI ZURB FOUNDATION KE BOOTSTRAP 4

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

Hapsari Laksmi W
NPM: 2015730037

ABSTRAK

Proses administrasi suatu universitas sebelum menggunakan teknologi memiliki beberapa hambatan. Di Fakultas Teknologi Informasi dan Sains misalnya, mahasiswa kesulitan untuk mengajukan permohonan cetak transkrip apabila berada jauh di daerah Universitas atau dosen tidak bisa merubah jadwal kuliah secara online apabila suatu kelas ditiadakan. Sehingga secara tidak langsung menghambat proses administrasi dalam FTIS selain itu karena harus dilakukan secara manual, kertas akan lebih banyak digunakan untuk proses pencatatan data sehingga tidak ramah lingkungan selain itu proses manual juga memiliki presentase kesalahan yang lebih banyak dibanding dengan menggunakan proses digital.

Sehingga Fakultas Teknologi Informasi dan Sains di Universitas Katolik Parahyangan mengembangkan aplikasi BlueTape yang dikembangkan dari tahun 2015 dengan FrameWork CodeIgniter dan Zurb Foundation sebagai salah satu cara untuk mengurangi pekerjaan paper-based menjadi paperless. Aplikasi ini dapat digunakan untuk melakukan beberapa proses administrasi seperti manajemen cetak transkrip, manajemen jadwal dosen dan manajemen perubahan kuliah.

Skripsi ini ditunjukan untuk migrasi framework front-end sebelumnya yaitu Zurb Foundation menuju ke Bootstrap.

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: «paperless, BlueTape, Bootstrap, Foundation, CodeIgniter, mahasiswa, dosen, »

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 Foundation 6	10
2.3.1 Struktur File	10
2.3.2 Sistem Grid pada Foundation	11
2.3.3 <i>Navigation</i> dan <i>Media Attributes</i>	12
2.3.4 Komponen	13
2.4 Bootstrap 4	17
2.4.1 Sistem Grid Bootstrap	17
2.4.2 Konten	18
2.4.3 Komponen	21
2.5 Plugin	28
3 ANALISIS	29
3.1 Analisis Frontend Library	29
3.1.1 Foundation	29

3.1.2	Template Flash Message	30
3.1.3	Template Head Logged In	31
3.1.4	Template Script Foundation	32
3.1.5	Template Top Bar Logged In	32
3.1.6	Antarmuka Cetak Transkrip	32
3.1.7	Antarmuka Manajemen Cetak Transkrip	34
3.1.8	Antarmuka Perubahan Kuliah	36
3.1.9	Permohonan Baru	36
3.1.10	Histori Pemohonan	36
3.1.11	Antarmuka Manajemen Perubahan Kuliah	37
3.1.12	Antarmuka Entri Jadwal Dosen	39
3.1.13	Antarmuka Lihat Jadwal Dosen	41
DAFTAR REFERENSI		43
A KODE PROGRAM		45
B HASIL EKSPERIMEN		47

DAFTAR GAMBAR

2.1	Flow Chart Aplikasi CodeIgniter	6
2.2	Struktur File Zurb Foundation	10
2.3	Grid pada Zurb Foundation	12
2.4	<i>Basic Navigation Menu</i> pada Foundation	12
2.5	Menu <i>align to right in Foundation</i>	12
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>	13
2.9	Basic Button pada Foundation	14
2.10	Coloring Button pada Foundation	14
2.11	Basic Table pada Foundation	15
2.12	Text Input pada Foundation	16
2.13	Grid pada Zurb Foundation	17
2.14	Grid pada Bootstrap	17
2.15	Pilihan kelas grid pada Bootstrap	18
2.16	Tabel default pada Bootstrap	19
2.17	Tabel default pada Bootstrap	19
2.18	Tabel default pada Bootstrap	20
2.19	<i>Menyelaraskan gambar ke kanan dan kiri pada bootstrap</i>	21
2.20	Forms Basic pada Bootstrap	22
2.21	Forms Basic pada Bootstrap	22
2.22	Disabled Basic pada Bootstrap	23
2.23	Button pada Bootstrap	23
2.24	Tombol <i>dropdown</i> pada Bootstrap	24
2.25	Badge pada Bootstrap	24
2.26	Card pada Bootstrap	25
2.27	Navigation Bar pada Bootstrap	25
2.28	Modal pada Bootstrap	26
2.29	Ikon <i>Coffee</i> pada Font Awesome	27
2.30	Alert pada Bootstrap	27
2.31	Datetimepicker pada Bootstrap	28
3.1	Alert mengenai 'info' pada BlueTape	31
3.2	Alert mengenai 'error' pada BlueTape	31
3.3	Kode template untuk file head_loggedin.php	32
3.4	Kode template untuk file script_foundation.php	32
3.5	Tampilan navbar dengan foundation	32
3.6	Antarmuka Cetak Transkrip bagian 1	33
3.7	Antarmuka Cetak Transkrip bagian 2	33
3.8	Modal Lihat Cetak Transkrip	34
3.9	Tampilan Manajemen Cetak Transkrip	34
3.10	Tampilan Modal untuk aksi 'Lihat' dan 'Tolak'	35

3.11	Tampilan Modal untuk aksi 'Print' dan 'Hapus'	35
3.12	Tampilan Perubahan Kuliah	36
3.13	Modal Lihat Perubahan Kuliah	37
3.14	Tampilan Manajemen Perubahan Kuliah	38
3.15	Modal aksi Lihat dan Print Manajemen Perubahan Kuliah	39
3.16	Modal aksi Setuju dan Tolak Manajemen Perubahan Kuliah	39
3.17	Modal Hapus Manajemen Perubahan Kuliah	39
3.18	Modal Print Manajemen Perubahan Kuliah	40
3.19	41
3.20	Struktur File Zurb Foundation	41
B.1	Hasil 1	47
B.2	Hasil 2	47
B.3	Hasil 3	47
B.4	Hasil 4	47

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 **Foundation**.

Foundation adalah kerangka kerja atau *Framework* untuk semua perangkat, media, dan semua aksesibilitas. Foundation adalah bagian dari *front-end framework* yang responsif dan memiliki kemudahan untuk merancang situs web, aplikasi, dan email. Sehingga akan terlihat lebih menarik saat dilihat dari perangkat mana pun. Foundation bersifat semantik, mudah dibaca, fleksibel, dan sepenuhnya *customizable*. [1].

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. Bootstrap memiliki fitur grid 12-kolom dan *container* selebar 940px yang kuat. Salah satu yang menarik adalah *build tool* di situs web Bootstrap, di mana *developer* dapat menyesuaikan pembangunan sesuai dengan kebutuhan, seperti memilih fitur CSS dan JavaScript yang ingin disertakan dalam situs. [2]

Pada skripsi ini akan diubah 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 mengimplementasikan *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, *template* entri jadwal dosen dan *template* lihat jadwal dosen dengan *framework Bootstrap 4*.
2. Mengimplementasikan *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 6 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.

Bab Perancangan antarmuka

Bab 4 berisi mengenai isi 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>. [3]

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 tombol "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 excel (.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.

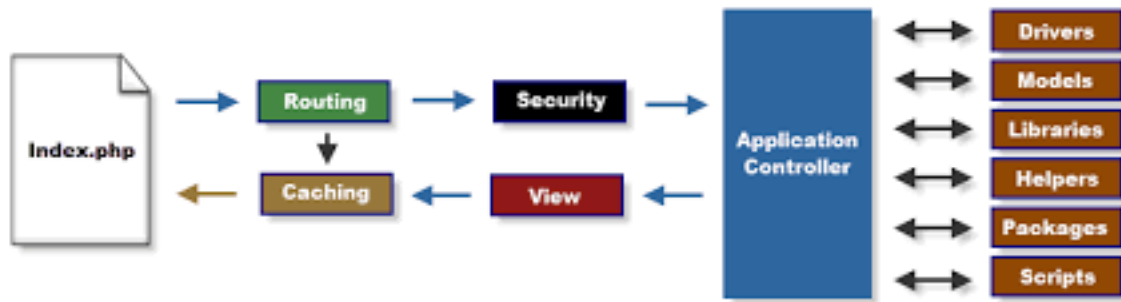
Beberapa tombol yang tersedia untuk setiap permohonan :

-  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 SI Akad.
-  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. **Cache** : Apabila terdapat *cache*, maka *cache* tersebut akan dikirimkan langsung ke browser, dengan melewati sistem eksekusi normal.
4. **Security** : Sebelum *controller* dimuat, *HTTP request* dan *user* mana pun yang mengirimkan data diseleksi dahulu untuk keamanan.
5. **Controller** : Terdiri dari *model*, *core libraries*, *helpers*, dan *resources* yang dibutuhkan untuk proses *request* tertentu.
6. **View** : Tampilan yang telah selesai dirubah kemudian dikirim ke *web browser* untuk dilihat. Jika *caching* diaktifkan, tampilan di *cache* terlebih dahulu sehingga pada permintaan selanjutnya dapat dilayani.[4]

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.

Anatomi Model

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

```
<?php
class Model_name extends CI_Model {

}
```

```
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:

```
$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 konektivitas database melalui parameter ketiga:

```
$config[ 'hostname' ] = 'localhost';
$config[ 'username' ] = 'myusername';
$config[ 'password' ] = 'mypassword';
$config[ 'database' ] = 'mydatabase';
$config[ 'dbdriver' ] = 'mysqli';
$config[ 'dbprefix' ] = '';
$config[ 'pconnect' ] = FALSE;
$config[ 'db_debug' ] = TRUE;

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

2.2.4 View

View adalah informasi yang disajikan kepada pengguna. Tampilan atau *View* biasanya akan menjadi halaman web, tetapi dalam CodeIgniter, tampilan juga bisa berupa fragmen halaman seperti header atau footer.

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:

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

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

Loading sebuah View

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

```
$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 :

```
$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 sebuah halaman yang terdiri dari **header**, **menu**, **content** dan **footer**.

```
<?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 Direktori

Untuk menyimpan didalam sub direktori maka dapat 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 berbentuk array atau objek, sehingga 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 dimasukkan 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 *developer* dapat menggunakan 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 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:

```
<?php  
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.

```
<?php  
class Blog extends CI_Controller {  
  
    public function index()  
    {  
        echo 'Hello World'  
    }  
}
```

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()`.

```
<?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. Namun ada cara lain untuk menampilkan pesan "Hello World" yang dapat dilakukan dengan:

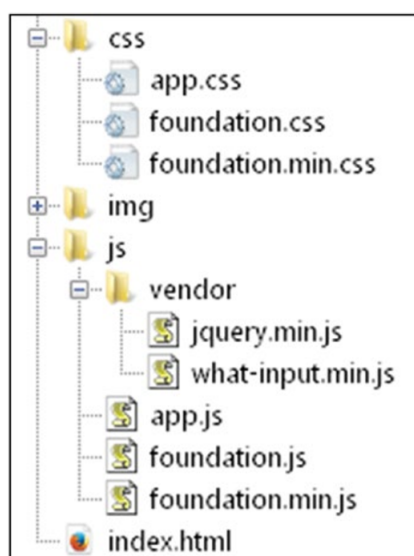
```
example.com/index.php/blog/index/
```

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

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

2.3 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 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.

[1]

2.3.2 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-#**.

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

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

Gambar 2.3: Grid pada Zurb Foundation

2.3.3 *Navigation dan Media Attributes*

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

Basic Menu

Menu terdiri dari sebuah `` yang diisi oleh beberapa tag ``. Secara default, menu akan berorientasi horizontal.

Berikut ini contoh penggunaan kode navigasi pada menu:

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

One Two Three Four

Gambar 2.4: *Basic Navigation Menu* pada Foundation

Item Alignment

Secara default, setiap item dalam menu akan berlajur ke arah kiri. Menu dapat diubah lajurnya ke arah kanan dengan menggunakan kelas `.align-right` atau ke arah tengah dengan menambahkan kelas `.align-center` didalam kelas `.menu`.

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

One Two Three Four

Gambar 2.5: Menu *align to right in Foundation*

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


Gambar 2.6: Menu *align to center* in Foundation

Active State

Kelas `.is-active` dapat ditambahkan ke dalam tag `` untuk membuat menu terpilih yang aktif terlihat saat di klik.

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


Gambar 2.7: Menu *active state menu* in Foundation

Text

Karena *padding* untuk setiap item menu menggunakan tag `<a>`, maka apabila sebuah item yang hanya berisi teks, maka teks tersebut tidak selaras dengan item menu lainnya. Sehingga untuk menyiasatinya, dapat menggunakan kelas `.menu-text` yang dituliskan dalam tag `` dengan menyertakan sebuah 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>
```


Gambar 2.8: Menu *active state menu* in Foundation

2.3.4 Komponen

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.

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

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



Gambar 2.9: Basic Button pada Foundation

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

```
<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

Tabel

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

```
<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 metus.</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 eget metus.</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 eget metus.</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

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

```

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

```

Striped

Secara default, tabel akan memiliki baris yang bergaris. Ada beberapa pilihan kelas untuk mengubah desain tabelnya.

- Kelas `.unstriped` dapat digunakan untuk menghapus garis-garis atau dengan mengubah `$table-is-striped` ke `false` untuk menghapus semua strip pada seluruh tabel.
- Kelas `.striped` untuk menambahkan strip pada tabel.

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`, `date`, `datetime`, `datetime-local`, `email`, `month`, `number`, `password`, `search`, `tel`, `time`, `url`, dan `week`.

```

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

```

```

        <input type="text" placeholder=".medium-6.cell">
      </label>
    </div>
  </div>
</div>
</form>

```



Gambar 2.12: Text Input pada Foundation

Select Menus Penggunaan `select` menus digunakan apabila *developer* menginginkan banyak pilihan data 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>

```

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 regelialia.</p>
  </div>
  <div class="tabs-panel" id="pub4">
    <p>...</p>
  </div>
</div>

```

```

</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*.

```

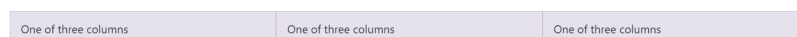
<div class="reveal" id="exampleModal1" data-reveal>
  <h1>Awesome. I Have It.</h1>
  <p class="lead">Your couch. It is mine.</p>
  <p>I'm a cool paragraph that lives inside of an even cooler modal. Wins!</p>
  <button class="close-button" data-close-aria-label="Close modal" type="button">
    <span aria-hidden="true">&times;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



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 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 2.15: Pilihan kelas grid pada Bootstrap

2.4.2 Konten

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 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>

```

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 2.17: Tabel default pada Bootstrap

```

<table class="table table-bordered">
  <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 colspan="2">Larry the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</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 2.18: Tabel default pada Bootstrap

```

<table class="table table-striped">
  <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 colspan="2">Larry the Bird</td>
                <td>@twitter</td>

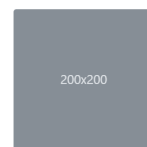
            </tr>
        </tbody>
    </table>

```

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 2.19: 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 Bootstrap form with the following elements:

- Email address:** A text 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 the values "2", "3", "4", and "5" selected.
- Example textarea:** A text area with 3 rows and 1 column.

Gambar 2.20: 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:** A text input field with the placeholder text "City".
- State:** A text input field with the placeholder text "State".
- Zip:** A text input field with the placeholder text "Zip".

Gambar 2.21: 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>
```

```

</div>
<div class="col">
  <input type="text" class="form-control" placeholder="Zip">
</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 2.22: 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>

```

Button

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



Gambar 2.23: 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>

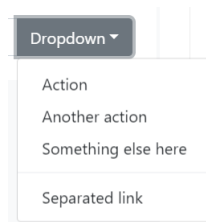
```

```
<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 2.24: Tombol *dropdown* pada Bootstrap

Badge

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

```
<a href="#" class="badge badge-primary">Primary</a>
<a href="#" class="badge badge-secondary">Secondary</a>
<a href="#" class="badge badge-success">Success</a>
<a href="#" class="badge badge-danger">Danger</a>
<a href="#" class="badge badge-warning">Warning</a>
<a href="#" class="badge badge-info">Info</a>
<a href="#" class="badge badge-light">Light</a>
<a href="#" class="badge badge-dark">Dark</a>
```



Gambar 2.25: Badge pada Bootstrap

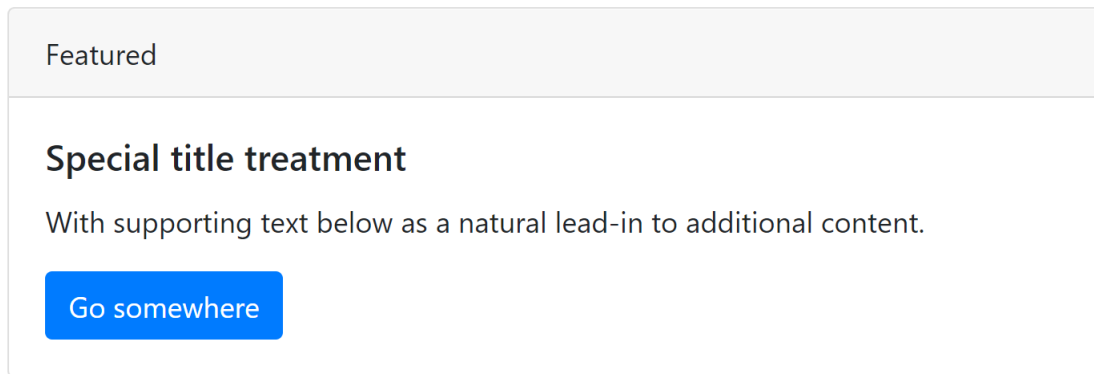
Card

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

```

<div class="card">
  <div class="card-header">
    Featured
  </div>
  <div class="card-body">
    <h5 class="card-title">Special title treatment</h5>
    <p class="card-text">With supporting text below as
      a natural lead-in to additional content.</p>
    <a href="#" class="btn btn-primary" card>Go somewhere</a>
  </div>
</div>

```



Gambar 2.26: 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 2.27: 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>

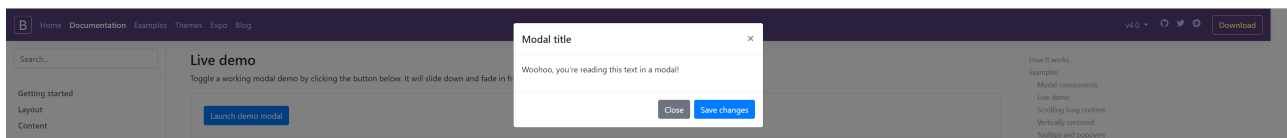
```

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 2.28: 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 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 2.29: 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 2.30: Alert pada Bootstrap

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

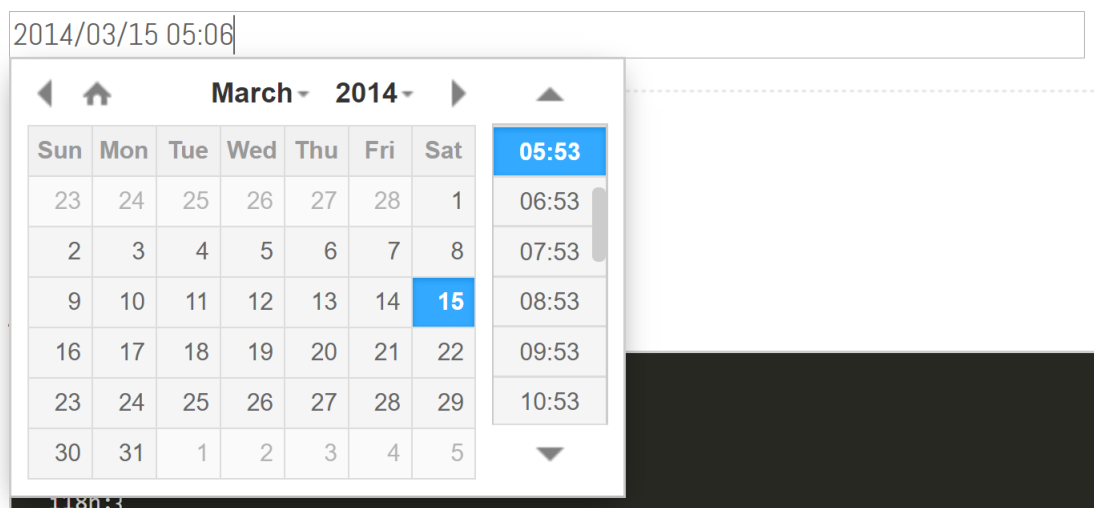
```

2.5 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 2.31: Datetimepicker pada Bootstrap

Penggunaan nya dalam kode HTML sebagai berikut :

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

Penggunaan dalam kode Javascript sebagai berikut :

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

BAB 3

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 kegunaannya pada website.

3.1 Analisis Frontend Library

3.1.1 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 :

1. **.row**
 - Untuk membuat konten yang terletak di dalam satu baris untuk setiap halaman website.
 - Untuk memisahkan baris dari sekumpulan field dalam sebuah form.
2. **.column**
 - Membuat kolom untuk menampung konten
 - Membuat kolom pada field dalam form
3. **<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

4. Kelas `.callout` : Border untuk menampung setiap konten di website BlueTape seperti form dan tabel.

5. 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.

6. 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 :

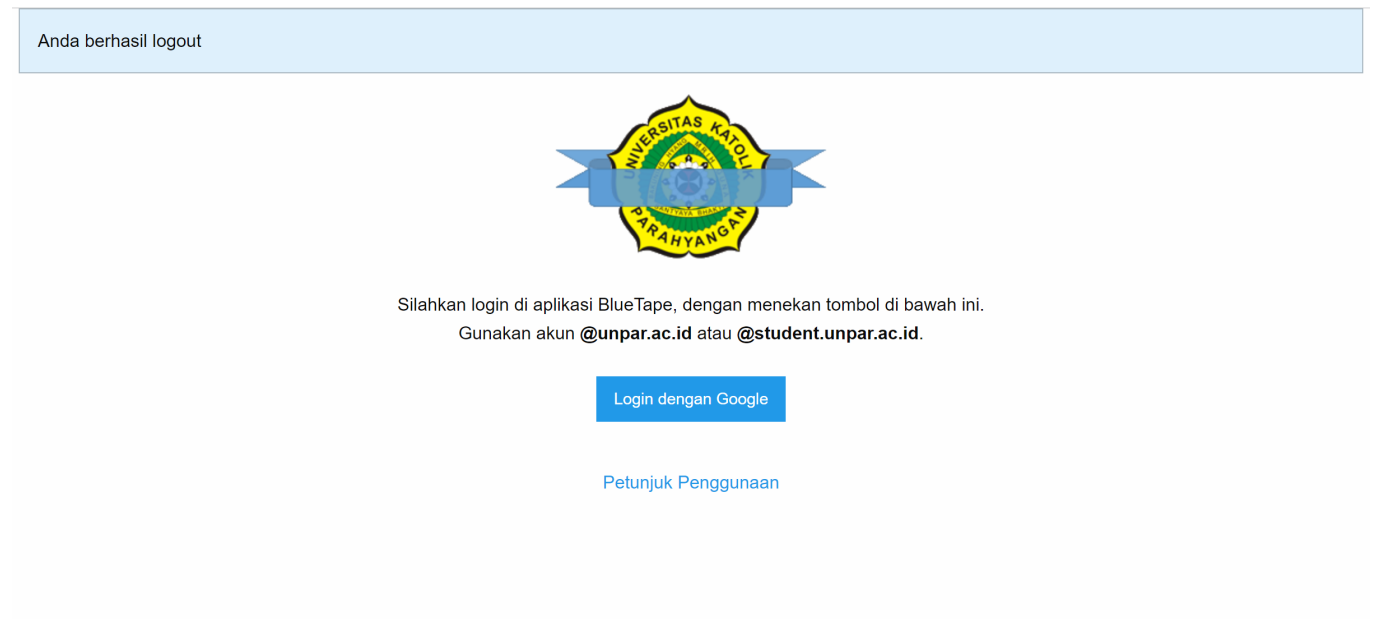
- `aria-label` : digunakan untuk memberi keterangan pada input field dalam form.
- `aria-hidden` : digunakan ketika ingin mengambil data dari database atau data dari akun yang teregistrasi.
- `aria-selected` : digunakan apabila input terdiri dari beberapa pilihan dan menggunakan empat

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

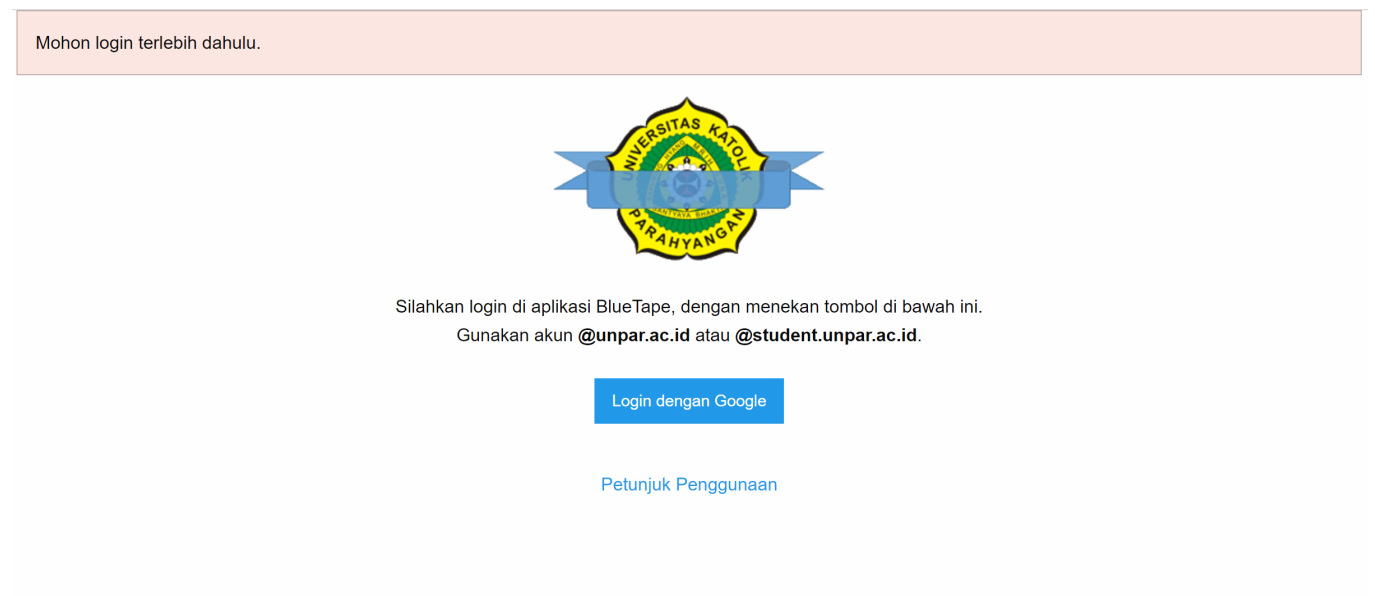
- Modal memiliki kelas `.reveal` dan atribut `data-reveal`.
- Setiap modal akan memiliki tombol hapus. Penggunaan kelas `.close-button` dan atribut `data-close` akan diaplikasikan.

3.1.2 Template Flash Message

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



Gambar 3.1: Alert mengenai 'info' pada BlueTape



Gambar 3.2: Alert mengenai 'error' pada BlueTape

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

3.1.3 Template Head Logged In

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

```

<?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 3.3: Kode template untuk file head_loggedin.php

3.1.4 Template Script Foundation

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

```

<?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 3.4: Kode template untuk file script *foundation.php*

3.1.5 Template Top Bar Logged In

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



Gambar 3.5: Tampilan navbar dengan foundation

3.1.6 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 :

1. **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.
2. **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" :

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

The screenshot shows the 'Permohonan Baru' (New Request) form and the 'Histori Permohonan' (Request History) table. The form includes fields for 'Yang memohon' (Email: amihapsahapsa@gmail.com), 'NPM' (NPM: -), 'Nama' (Name: Hapsari Laksmi Wijayanti), 'Tipe Transkrip' (DPS Bahasa Indonesia (Seluruh Semester)), and 'Keperluan' (Purpose). A 'Kirim Permohonan' button is present. The 'Histori Permohonan' table has columns: ID, Status, Tanggal Permohonan, Tipe Transkrip, Tanggal Jawab/Cetak, Keterangan, and Aksi. The first row shows ID #3, Status TERSELESAI (green), Tanggal Permohonan Kamis, 2 Mei 2019, Tipe Transkrip DPS_ID, Tanggal Jawab/Cetak Minggu, 27 Oktober 2019, Keterangan teroetak, and Aksi with an eye icon.

Gambar 3.6: Antarmuka Cetak Transkrip bagian 1

The screenshot shows the 'Permohonan Baru' (New Request) form and the 'Histori Permohonan' (Request History) table. The form includes a message: 'Anda tidak bisa meminta cetak karena ada permintaan lain yang belum selesai.' (You cannot request printing because there are other requests that are not yet completed). The 'Histori Permohonan' table has columns: ID, Status, Tanggal Permohonan, Tipe Transkrip, Tanggal Jawab/Cetak, Keterangan, and Aksi. The first row shows ID #3, Status TUNGGU (grey), Tanggal Permohonan Kamis, 2 Mei 2019, Tipe Transkrip DPS_ID, and Aksi with an eye icon.

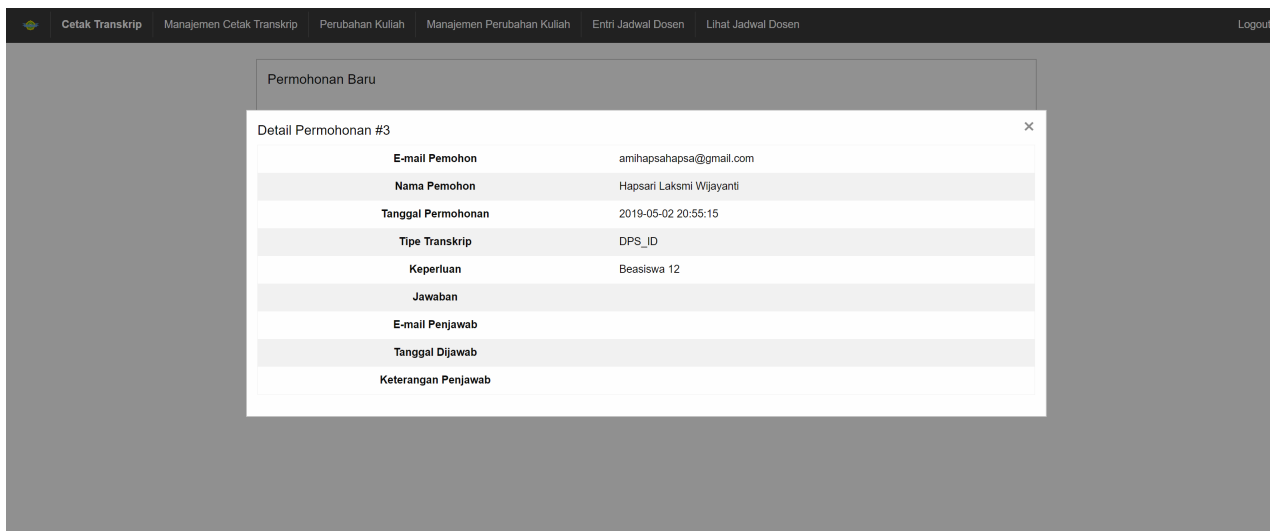
Gambar 3.7: Antarmuka Cetak Transkrip bagian 2

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 📖 :

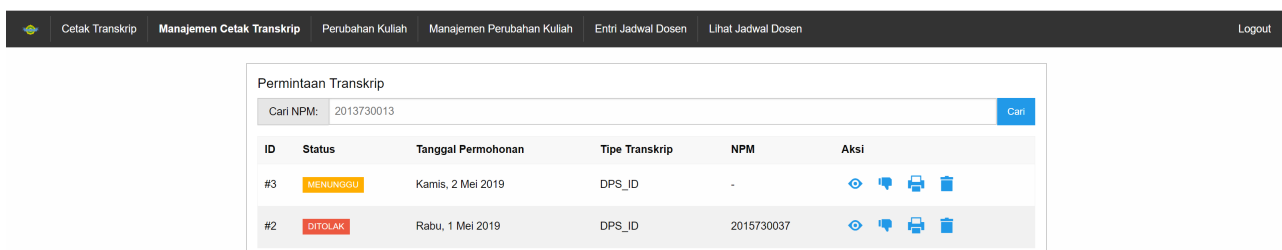


Gambar 3.8: Modal Lihat Cetak Transkrip

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`.

3.1.7 Antarmuka Manajemen Cetak Transkrip

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



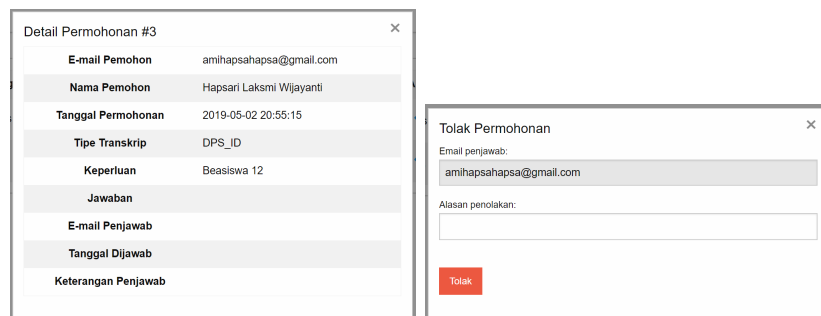
Gambar 3.9: Tampilan Manajemen Cetak Transkrip

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' :

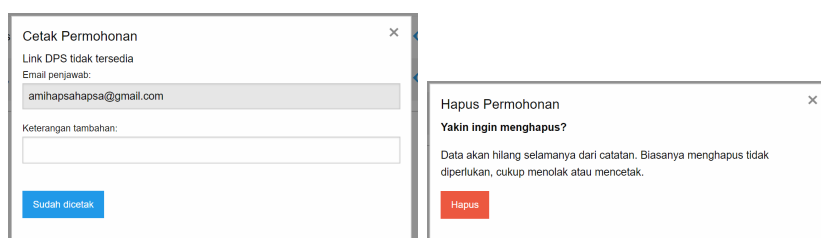
- **Status** : Output terdiri dari tiga jenis label yaitu 'MENUNGGU'(berwarna kuning), 'DI-TOLAK' (berwarna merah) dan 'TERCETAK'(berwarna hijau). Sehingga dibutuhkan kelas `secondary|`, `success|`, `alert|`.
- **Aksi** : Terdiri dari empat ikon font-awesome yaitu `fi-eye`, `fi-dislike`, `fi-print`, `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 :

1. 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.
2. 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.
3. 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.
4. 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.



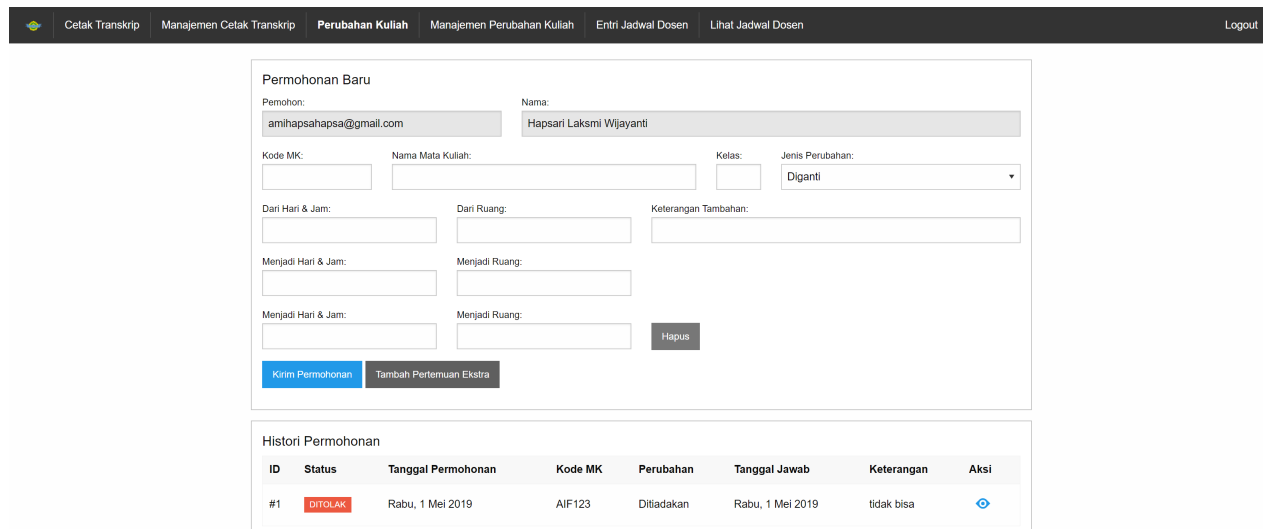
Gambar 3.10: Tampilan Modal untuk aksi 'Lihat' dan 'Tolak'



Gambar 3.11: Tampilan Modal untuk aksi 'Print' dan 'Hapus'

3.1.8 Antarmuka Perubahan Kuliah

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



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

Gambar 3.12: Tampilan Perubahan Kuliah

Modul Perubahan Kuliah terdiri dari dua tabel yaitu :

3.1.9 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 :

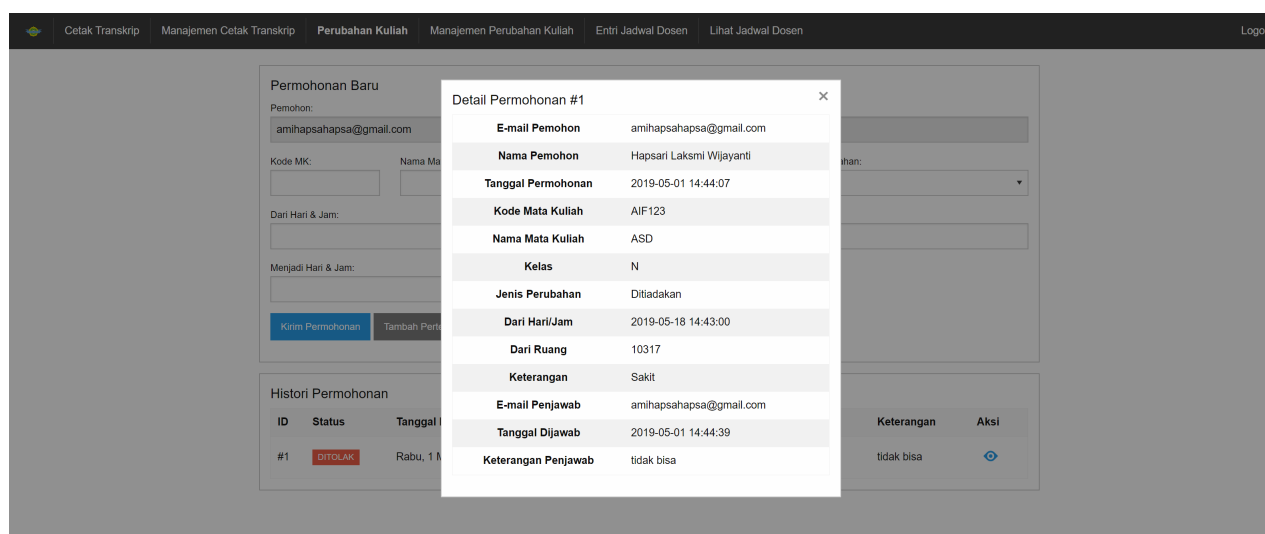
3.1.10 Histori Pemohonan

Histori Permohonan menggunakan tabel yang bisa menyesuaikan lebarnya dengan menampilkan data secara bertumpuk sehingga menggunakan kelas `.stack`. Jenis data akan diletakkan dalam tag

`colorboxmygraytexttt<thead>` 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 3.13: Modal Lihat Perubahan Kuliah

3.1.11 Antarmuka Manajemen Perubahan Kuliah

Antarmuka diaplikasikan pada file `BlueTape/www/application/views/PerubahanKuliah-
Manage/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 3.14: 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>

Detail Permohonan #1

E-mail Pemohon	amhapahepa@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	Ditindakan
Dari Hari/Kam	2019-05-18 14:43:00
Dari Ruang	10317
Keterangan	Sakit
E-mail Penjawab	amhapahepa@gmail.com
Tanggal Dijawab	2019-05-01 14:44:28
Keterangan Penjawab	tidak bisa

PENGUMUMAN Jadwal Kuliah

Mata Kuliah	ASD
Kode MK	AF123
Kelas	N
nama	
Hari	Sabtu, 18 Mei 2019
Jam	14:43
Ruang	10317
alamat	
Keterangan	Sakit
Buat tanggal	27-04-2019
Administri FTIS	

Gambar 3.15: Modal aksi Lihat dan Print Manajemen Perubahan Kuliah

Konfirmasi Permohonan

E-mail pemohon: amhapahepa@gmail.com

Keterangan:

Tolak Permohonan

E-mail pemohon: amhapahepa@gmail.com

Alasan penolakan:

Gambar 3.16: Modal aksi Setuju dan Tolak Manajemen Perubahan Kuliah

Hapus Permohonan

Yakin ingin menghapus?

Data akan hilang selamanya dari sistem. Beresnya menghapus tidak dapat dikembalikan, tidak dapat dihapus atau direstore.

Gambar 3.17: Modal Hapus Manajemen Perubahan Kuliah

Berikut ini penjelasan masing - masing modal.

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.

3.1.12 Antarmuka Entri Jadwal Dosen

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

Tambah Jadwal

Hari: Durasi: Label:

Jam Mulai: Jenis:

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

Gambar 3.18: 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
 1. Konsultasi : Memiliki background berwarna hijau.
 2. Terjadwal : Memiliki background berwarna biru.
 3. 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.



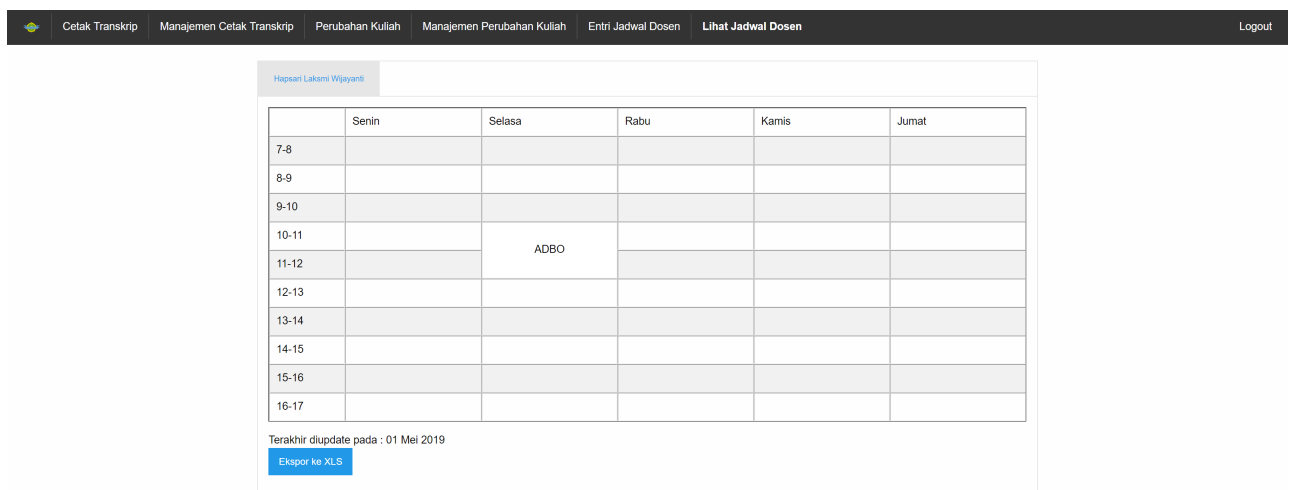
The image shows a modal form titled "Edit Jadwal" with a close button (X) in the top right corner. The form contains five dropdown menus: "Hari" (set to Selasa), "Jam Mulai" (set to 10:00), "Durasi" (set to 2 jam), "Jenis" (set to Kelas), and "Label".

Gambar 3.19:

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`.

3.1.13 Antarmuka Lihat Jadwal Dosen

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



The screenshot shows the "Lihat Jadwal Dosen" interface. At the top, there is a navigation bar with tabs: "Cetak Transkrip", "Manajemen Cetak Transkrip", "Perubahan Kulliah", "Manajemen Perubahan Kulliah", "Entri Jadwal Dosen", and "Lihat Jadwal Dosen". Below the navigation bar, there is a sidebar with a tab labeled "Hapsari Laksmi Wijayanti". The main content area displays a table of class schedules. The table has columns for days of the week (Senin, Selasa, Rabu, Kamis, Jumat) and rows for time slots (7-8, 8-9, 9-10, 10-11, 11-12, 12-13, 13-14, 14-15, 15-16, 16-17). The cell for 10-11 on Tuesday contains the text "ADBO". Below the table, there is a message "Terakhir diupdate pada : 01 Mei 2019" and a blue button labeled "Ekspor ke XLS".

Gambar 3.20: 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".

DAFTAR REFERENSI

- [1] Foundation 6 (2017) *Foundation v6.3.1 Documentation*. Zurb Inc. 100 W Rincon Ave, Campbell, CA 95008, USA.
- [2] Bootstrap 4 (2019) *Bootstrap v4.3.1 Documentation*. Bootstrap Core Team. 77 Massachusetts Avenue Cambridge, MA 02139, USA.
- [3] BlueTape (2019) *UserGuide BlueTape*. Fakultas Teknologi Informasi dan Sains (FTIS) Universitas Katolik Parahyangan. Bandung, Jawa Barat.
- [4] Codeigniter 3 (2017) *Codeigniter 3.1.4 Documentation*. British Columbia Institute of Technology. 5112 Bobby Hicks Hwy, Gray, TN 37615, USA.

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf

```

Listing A.2: MyCode.java

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }

```


LAMPIRAN B

HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4