

**SKRIPSI**

**SISTEM PENILAIAN SIDANG SKRIPSI 2 DENGAN  
ANGULARJS**



**BILLY YANUAR**

**NPM: 2012730017**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2017**



**UNDERGRADUATE THESIS**

**THE THESIS 2 ASSESSMENT DEFENSE SYSTEM**



**BILLY YANUAR**

**NPM: 2012730017**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND  
SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2017**



## ABSTRAK

Mata kuliah Skripsi 2 merupakan salah satu syarat wajib dalam proses pembelajaran yang dilakukan di Program Studi Teknik Informatika Universitas Katolik Parahyangan, Bandung. Sistem ini yang diterapkan untuk melakukan penilaian pada sidang Skripsi 2 masih menggunakan kertas penilaian, dimana penilai akan menuliskan nilai yang diinginkan lalu melakukan perhitungan dengan alat hitung masing-masing untuk mendapatkan nilai akhir mahasiswa. Untuk itu, dibuatlah Sistem Penilaian Sidang Skripsi 2 yang berupa situs *web* yang membantu perhitungan dan penyimpanan nilai sidang mata kuliah Skripsi 2 tersebut. Sistem Penilaian Sidang Skripsi 2 dibuat menggunakan bahasa PHP yang diintegrasikan dengan AngularJS sehingga dapat melakukan otomatisasi perhitungan nilai akhir mahasiswa.

AngularJS merupakan *framework* yang dipakai dalam pembangunan *single page application*. Kemudahan pemakaian AngularJS merupakan salah satu nilai tambah tersendiri. AngularJS memiliki konsep MVC (*Model, View, Controller*) yang bekerja secara *front-end* (di bagian *layout web*) sehingga dapat diintegrasikan oleh *framework* lain, seperti *codeigniter*.

Pengujian Sistem Penilaian Sidang Skripsi 2 dilakukan dengan membandingkan hasil perhitungan nilai akhir mahasiswa dari 5 sidang skripsi 2 yang berlangsung pada Semester Ganjil 2016/2017. Pengujian tersebut dilakukan dengan cara membandingkan hasil perhitungan dari Sistem Penilaian Sidang Skripsi 2 dan dari sistem ini. Dari hasil pengujian tersebut, didapatkan bahwa perhitungan yang dilakukan oleh sistem usulan lebih akurat dibandingkan dengan sistem ini.

**Kata-kata kunci:** Sistem Penilaian Skripsi 2, PHP, AngularJS



## ABSTRACT

One of the graduate condition in Informatics Engineering Parahyangan Catholic University, Bandung is Thesis 2 courses. The current system applied for assessment in defense of the Thesis 2 is still using scoring paper, where the assessor will write down the desired value and the perform the calculation with the calculator to get the final grade. Therefore, The Thesis 2 Assessment Defense System was created in a website that helped the calculation and store the final grade of Thesis 2 defense. The Thesis 2 Assessment Defense System was built in PHP programming language which is integrated with AngularJS so it can perform automatic calculation of the final grade.

AngularJS is a framework that used in single page application. Ease of use is one of its special advantage. AngularJS's concept is MVC (Model, View, Controller) that works on front-end (in layout web) so it can be integrated with other framework such as co-deigniter.

The Thesis 2 Assessment Defense System testing is done by comparing the calculation result of The Thesis 2 Assessment Defense System and the current system used. From the test result, we obtained that the calculation of The Thesis 2 Assessment Defense System is more accurate than the current system used.

**Keywords:** The Thesis 2 Assessment Defense System, PHP, AngularJS





# DAFTAR ISI

<b>DAFTAR ISI</b>	<b>ix</b>
<b>DAFTAR GAMBAR</b>	<b>xi</b>
<b>DAFTAR TABEL</b>	<b>xii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metode Penelitian . . . . .	3
1.6 Sistematika Penulisan . . . . .	3
<b>2 DASAR TEORI</b>	<b>5</b>
2.1 CodeIgniter . . . . .	5
2.1.1 Flowchart Aplikasi CodeIgniter . . . . .	5
2.1.2 Model-View-Controller . . . . .	6
2.1.3 Controller . . . . .	6
2.1.4 Views . . . . .	7
2.1.5 Models . . . . .	8
2.1.6 Helper . . . . .	9
2.1.7 Basis data . . . . .	9
2.1.8 Konfigurasi Basis Data . . . . .	10
2.2 AngularJS . . . . .	11
2.2.1 Gambaran Konseptual . . . . .	12
2.2.2 Directives . . . . .	13
2.2.3 Data Binding . . . . .	14
2.2.4 Model-View-Controller(MVC) . . . . .	15
2.3 Twitter Bootstrap . . . . .	16
2.3.1 Grid System . . . . .	16
2.3.2 Form Class . . . . .	18
<b>3 ANALISIS</b>	<b>19</b>
3.1 Analisis Sistem Kini . . . . .	19
3.1.1 Form Rekapitulasi Penilaian . . . . .	19
3.1.2 Form Berita Acara Sidang Skripsi . . . . .	19
3.2 Analisis Sistem Usulan . . . . .	20
3.2.1 Analisis Back End . . . . .	20
3.2.2 Analisis Front End . . . . .	22
3.2.3 Analisis Basis Data . . . . .	28
3.3 Use Case . . . . .	29

<b>4</b>	<b>PERANCANGAN</b>	<b>31</b>
4.1	Perancangan Kelas . . . . .	31
4.2	Routes . . . . .	31
4.3	Controllers . . . . .	31
4.4	Models . . . . .	32
4.5	Perancangan Basis Data . . . . .	32
4.6	Perancangan Tampilan . . . . .	33
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>35</b>
5.1	Implementasi . . . . .	35
5.1.1	Lingkungan Implementasi dan Pengujian . . . . .	35
5.1.2	Hasil Implementasi . . . . .	35
5.2	Hasil Pengujian . . . . .	39
5.2.1	Pengujian Eksperimental . . . . .	39
5.2.2	Pengujian Fungsional . . . . .	39
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>41</b>
6.1	Kesimpulan . . . . .	41
6.2	Saran . . . . .	41
	<b>DAFTAR REFERENSI</b>	<b>43</b>
<b>A</b>	<b>FORM PENILAIAN SKRIPSI</b>	<b>45</b>
<b>B</b>	<b>THE SOURCE CODE</b>	<b>47</b>

## DAFTAR GAMBAR

2.1	Flowchart CodeIgniter . . . . .	5
2.2	Data Binding Classical Templates System . . . . .	14
2.3	Data Binding pada Angular . . . . .	14
2.4	Grid Option pada Bootstrap . . . . .	17
2.5	Contoh Pembagian Grid Columns . . . . .	18
2.6	Contoh Hasil Penggunaan Kelas Form . . . . .	18
3.1	Use case diagram . . . . .	29
4.1	Gambar diagram kelas <i>file controllers</i> . . . . .	32
4.2	Perkiraan Tampilan . . . . .	34
5.1	Formulir berita acara sidang skripsi 2 terisi . . . . .	36
5.2	Formulir rekapitulasi ketua tim penguji terisi . . . . .	36
5.3	Formulir rekapitulasi anggota tim penguji terisi . . . . .	37
5.4	Formulir rekapitulasi pembimbing terisi . . . . .	37
5.5	Ketika tombol selesai di klik . . . . .	37
A.1	Form Penilaian Skripsi saat sidang . . . . .	45
A.2	Form Rekapitulasi Penilaian Skripsi saat sidang . . . . .	46

## DAFTAR TABEL

4.1	Tabel Perancangan Basis Data . . . . .	33
5.1	Tabel Hasil Pengujian . . . . .	38
5.2	Tabel Pengujian Eksperimental . . . . .	40
5.3	Tabel Pengujian Fungsional . . . . .	40

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Program Studi Teknik Informatika di Universitas Katolik Parahyangan memiliki beberapa syarat kelulusan. Salah satunya adalah minimal SKS lulus adalah 144 sks yang terdiri dari mata kuliah wajib, pilihan wajib, dan pilihan. Selain itu Indeks Prestasi Kumulatif (IPK) minimum yang diperlukan adalah 2.00 dengan maksimum 14 semester. Salah satu mata kuliah wajib yang harus ditempuh dan lulus adalah Skripsi. Skripsi pada Program Studi Teknik Informatika di Universitas Katolik Parahyangan dibagi menjadi 2 mata kuliah yaitu Skripsi 1 dan Skripsi 2. Penelitian ini berfokus pada sidang untuk mata kuliah Skripsi 2.

Pada sidang mata kuliah Skripsi 2, penilaian dilakukan dengan 2 lembar kertas penilaian yaitu lembar berita acara sidang skripsi (Gambar A.1) dan lembar rekapitulasi (Gambar A.2). Lembar rekapitulasi merupakan lembar penilai menuliskan nilai yang ingin diberikan kepada mahasiswa, sedangkan lembar berita acara sidang skripsi adalah lembar untuk mengumpulkan nilai akhir masing-masing penilai yang diambil dari lembar rekapitulasi untuk dihitung menjadi nilai akhir mahasiswa. Penilaian pada sidang skripsi 2 bersifat manual, dimana penilai akan menuliskan nilai yang ingin diberikan serta melakukan perhitungan nilai untuk mendapatkan nilai akhir mahasiswa pada lembar rekapitulasi masing-masing penilai. Lembar rekapitulasi yang telah selesai dihitung kemudian akan di kumpulkan kepada ketua tim penguji untuk dituliskan di lembar berita acara sidang skripsi yang kemudian akan diproses menjadi nilai akhir mahasiswa bersangkutan.

Sifat manual ini mengakibatkan kelalaian manusia dalam melakukan penilaian beberapa kali tidak dapat dihindarkan. Kelalaian manusia yang biasa terjadi contohnya adalah kesalahan perhitungan nilai akhir oleh penilai, kesalahan penulisan nama dan NPM mahasiswa yang bersangkutan, kesalahan penulisan semester atau tahun ajaran saat penilaian skripsi<sup>1</sup>. Untuk mengatasi hal-hal tersebut, diperlukan suatu sistem yang dapat menanggulangi masalah pengisian, kalkulasi perhitungan, dan juga penyimpanan skripsi.

Menurut penjelasan di atas, maka penulis mengusulkan otomatisasi sistem dalam penilaian skripsi yang akan dibuat guna mengurangi kesalahan-kesalahan kecil yang dapat berakibat fatal pada nilai mahasiswa yang bersangkutan. Berdasarkan hal tersebut dibuatlah penelitian otomatisasi sistem penilaian skripsi dengan cara membuat sebuah aplikasi berbasis *web* yaitu Sistem Informasi Penilaian Sidang Skripsi2.

Pada penelitian ini, akan dibuat sebuah sistem penilaian yang menanggulangi masalah-masalah tersebut, dengan cara menjadikan beberapa masukan(*input*) otomatis dan juga

---

<sup>1</sup>berdasarkan diskusi dengan dosen pembimbing

1 melakukan eksekusi perhitungan nilai akhir sesuai bobot secara otomatis. Hal ini dianggap  
2 akan memudahkan penilai dalam melakukan proses penilaian skripsi, karena penilai tidak  
3 perlu lagi repot melakukan perhitungan nilai dan juga mengisi masukan-masukan yang  
4 sudah terisi secara otomatis.

5 Dalam penelitian ini saya memakai *framework* AngularJS yang dimiliki oleh perusahaan  
6 *Google*. AngularJS merupakan salah satu *framework* yang paling sering digunakan untuk  
7 membuat sebuah aplikasi berbasis *web* dengan konsep *Single Page Application (SPA)*. *Single*  
8 *Page Application* merupakan aplikasi berbasis *web* yang memungkinkan sebuah halaman  
9 HTML memiliki konten-konten yang dapat digunakan di halaman tersebut tanpa perlu  
10 berganti ke halaman lain.

11 AngularJS juga bisa diintegrasikan dengan aplikasi yang menggunakan *framework* lain,  
12 sehingga sangat berguna dalam pengerjaan aplikasi berbasis *web* terutama pada pengerjaan  
13 Sistem Informasi Penilaian Sidang Skripsi2 yang akan dibuat.

14 Selain AngularJS, sistem usulan ini juga memakai 2 *framework* pendukung yaitu Code-  
15 Igniter dan Twitter Bootstrap. CodeIgniter dipakai untuk memudahkan aliran data pada  
16 sistem usulan yang dibuat, sedangkan Twitter Bootstrap dipakai untuk mempermudah pe-  
17 ngaturan tampilan pada sistem usulan yang akan dibuat.

## 18 1.2 Rumusan Masalah

19 Berikut adalah susunan permasalahan yang akan dibahas pada penelitian ini:

- 20 1. Bagaimana sistem penilaian skripsi 2 yang ada pada Program Studi Teknik Informa-  
21 tika di Universitas Katolik Parahyangan?
- 22 2. Bagaimana proses penyimpanan nilai skripsi?
- 23 3. Bagaimana AngularJS bekerja pada eksekusi perhitungan nilai akhir?

## 24 1.3 Tujuan

25 Berdasarkan rumusan masalah yang telah dibuat, maka tujuan penelitian ini dijelaskan ke  
26 dalam poin-poin sebagai berikut:

- 27 1. Mempelajari sistem penilaian skripsi pada Program Studi Teknik Informatika di Uni-  
28 versitas Katolik Parahyangan
- 29 2. Merancang dan mengimplementasi proses penyimpanan nilai skripsi
- 30 3. Mengimplementasi AngularJS untuk mengeksekusi perhitungan nilai akhir

## 31 1.4 Batasan Masalah

32 Penelitian ini memiliki batasan-batasan sebagai berikut:

- 33 1. Penelitian ini hanya dilakukan untuk *form* penilaian mata kuliah Skripsi 2
- 34 2. Penelitian ini hanya melakukan fungsi *insert* ke basis data

## 1.5 Metode Penelitian

Dalam penelitian ini, akan dilakukan langkah-langkah berikut:

1. Melakukan studi terhadap CodeIgniter, Twitter Bootstrap, dan AngularJS sebagai *framework* yang akan dipakai.
2. Melakukan perancangan untuk implementasi integrasi sistem tersebut.
3. Melakukan implementasi dari rancangan yang sudah dilakukan.
4. Melakukan pengujian pada saat sidang skripsi2 sehingga penilai dapat menguji hasil implementasi tersebut.
5. Menganalisa dan menarik kesimpulan atas hasil penelitian yang telah dilaksanakan.

## 1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan dari dokumen ini:

- Bab 1 membahas latar belakang, rumusan masalah, tujuan penulisan, batasan-batasan, serta metode yang digunakan pada penelitian ini.
- Bab 2 membahas teori-teori yang digunakan dalam penelitian ini, yaitu AngularJS, Code Igniter, dan Twitter Bootstrap.
- Bab 3 menganalisis sistem ini, beserta perubahan-perubahan yang harus dilakukan.
- Bab 4 membahas perancangan yang dilakukan sebelum mengimplementasikan integrasi yang dimaksud, mencakup protokol, basisdata, beserta antarmukanya.
- Bab 5 membahas implementasi serta pengujian dari integrasi yang telah dilakukan.
- Bab 6 membahas kesimpulan dari keseluruhan penelitian ini, serta saran-saran yang dapat diberikan untuk penelitian berikutnya.





## BAB 2

### DASAR TEORI

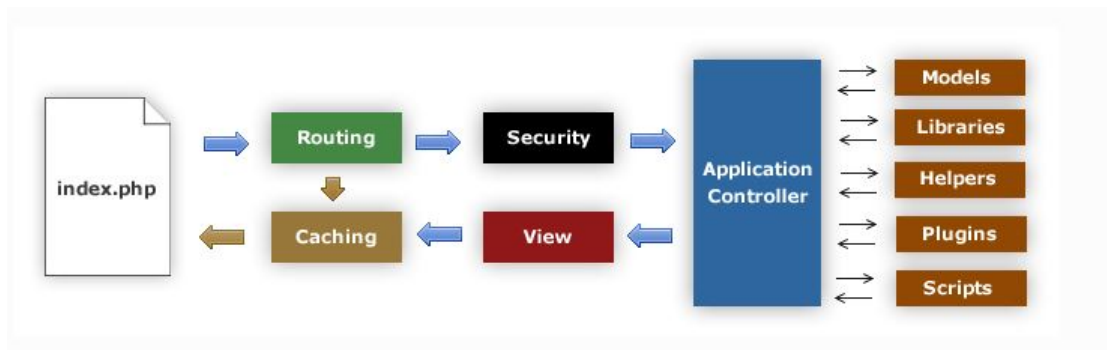
Pada bab ini akan dijelaskan dasar-dasar teori mengenai CodeIgniter, AngularJS, dan Twitter Bootstrap.

#### 2.1 CodeIgniter

CodeIgniter[1] merupakan sebuah *framework* bagi *programmer* yang ingin membuat sebuah *web* dengan menggunakan bahasa PHP. Penggunaan CodeIgniter sendiri mempunyai tujuan mempercepat pengembangan proyek-proyek bersangkutan jika dibandingkan dengan menuliskan kode dari awal. Tujuan tersebut diwujudkan dengan tersedianya *library* berisi *task* yang biasa dibutuhkan dalam pengembangan *program*, dibarengi dengan antarmuka yang sederhana serta struktur logika khusus untuk mengakses *library* tersebut. Dengan begitu dapat disimpulkan bahwa CodeIgniter membuat pemrogram fokus pada kreativitas pembuatan *program* dengan meminimalkan jumlah kode yang dituliskan.

##### 2.1.1 Flowchart Aplikasi CodeIgniter

Pada Gambar 2.1 adalah *flowchart* aliran data pada CodeIgniter.



Gambar 2.1: Flowchart CodeIgniter

Keterangan Gambar 2.1

1. Index.php berfungsi sebagai pengontrol utama, yang menginisialisasikan sumber-sumber yang diperlukan untuk menjalankan CodeIgniter.
2. *Routing* / *Router* akan memeriksa permintaan HTTP untuk menentukan apa yang harus dilakukan selanjutnya

3. Jika terdapat *cache*, maka *cache* tersebut akan dikirim langsung ke *browser* dengan menjalankan sistem eksekusi *normal*.
4. HTTP *request* dan data yang diserahkan oleh *user* akan disaring terlebih dahulu oleh bagian keamanan(*security*) dari CodeIgniter, yang dijalankan sebelum *controller* dari aplikasi diisi.
5. *Application Controller* akan mengambil isi dari *model*, *libraries*, *helpers*, *plugins*, *scripts*, dan sumber lain yang diperlukan untuk menjalankan perintah-perintah spesifik.
6. Kemudian *View* akan diterjemahkan dari *Application Controller* dan dikirim ke *web browser* untuk kemudian ditampilkan. Jika pada *view* akhir terdapat *file cache*, maka *view* tersebut akan terlebih dahulu dilakukan fungsi *cached* sehingga permintaan berikutnya dapat dilayani.

### 2.1.2 Model-View-Controller

CodeIgniter menggunakan dasar pola pengembangan *Model-View-Controller*(MVC). Pola pengembangan MVC ini merupakan suatu pendekatan yang memisahkan antara pengerjaan logika dan tampilan dari aplikasi.

MVC sendiri terdiri dari 3 bagian, yaitu:

1. *Model* merepresentasikan struktur data. Secara khusus, *model* merupakan kelas yang membantu menangani kueri-kueri sql seperti *insert*, *update*, dan *delete* pada basis data.
2. *View* merepresentasikan informasi yang ditunjukkan kepada pengguna. Sebuah *view* umumnya berbentuk *web page*, tetapi dalam CodeIgniter *view* bisa berbentuk *header*, *footer*, dan berbagai jenis *page* lainnya.
3. *Controller* berfungsi sebagai perantara antara *Model*, *View*, dan sumber daya lain yang diperlukan untuk memproses HTTP *request* dan menghasilkan halaman *web*.

### 2.1.3 Controller

*Controller* merupakan sebuah kelas sederhana dengan penerapan seperti URL. Seperti kelas pada umumnya, ketika nama kelas dari *controller* dan nama kelas dari *file controller* tersebut cocok, maka kelas dapat dijalankan dengan baik. Nama kelas suatu *controller* dikatakan sah jika diawali dengan huruf besar. Untuk lebih jelasnya, perhatikan kode di bawah ini:

Listing 2.1: CI Controller

```

32 <?php
33 class Blog extends CI_Controller{
34
35     public function index{
36         echo 'Hello World';
37     }
38 }
```

Nama *file* pada kode listing di atas haruslah "Blog.php" dengan B besar dan disimpan pada *application/controllers* sehingga URL dapat berjalan dengan baik.

## 1 Method

2 *Method* merupakan nama fungsi dari suatu kelas. Nama *method* pada Listing 2.1 adalah "index()". *Method* bernama "index" akan selalu dijalankan jika tidak ada arahan ke metode pada URL. Cara lain untuk menjalankan *method* pada kode subbab 2.1.3 adalah [example.com/index.php/blog/index/](http://example.com/index.php/blog/index/) dimana bagian terakhir adalah nama *method* yang ingin dijalankan.

Listing 2.2: CI Controller Method

```
7 <?php
8 class Products extends CI_Controller{
9
10     public function shoes($sandals, $id){
11         echo $sandals;
12         echo $id;
13     }
14 }
```

15 Jika *method* yang dituju memiliki parameter, diperlukan tambahan pada URL pemanggilannya. Sebagai contoh, pemanggilan *method* pada kode di atas dilakukan dengan URL [example.com/index.php/products/shoes/sandals/123](http://example.com/index.php/products/shoes/sandals/123) dimana "sandals" dan "123" merupakan isi dari *parameter 1* dan *2* dari *method* "shoes".

## 19 Mendefinisikan Controller Default

20 CodeIgniter dapat menjalankan *default controller* sehingga tidak diperlukannya penulisan URL yang lengkap untuk pemanggilan, melainkan *controller* dapat dipanggil secara otomatis dengan URL [example.com](http://example.com) saja. Namun, untuk dapat menjalankan fungsi ini, diperlukan sedikit pengaturan pada file "application/config/routes.php" yaitu perubahan variabel yang ditunjukkan kode di bawah ini:

Listing 2.3: Default Controller

```
25 $route['default_controller'] = 'blog';
```

26 "blog" merupakan nama *file controller* yang telah dibuat pada direktori "application/controllers/". Setelah pengaturan tersebut, maka pengguna bisa menjalankan aplikasi tanpa URL yang terspesifikasi menjalankan *controller*.

## 29 2.1.4 Views

30 Sebuah *views* merupakan bagian yang mengatur tampilan aplikasi yang akan ditunjukkan kepada pengguna. *Views* meliputi *footer*, *header*, *sidebar*, dll. Pada CodeIgniter, *Views* tidak dapat dijalankan secara langsung dari URL, tapi *views* harus dijalankan melalui file *controller* yang ada. Hal ini dilakukan guna memudahkan *programmer* dan mewujudkan *framework MVC* pada CodeIgniter.

## 35 Pembuatan Views

36 Pembuatan *file view* pada dasarnya sama seperti pembuatan *file* berbasis PHP. Berikut ini merupakan salah satu contoh kode sebuah *file view* sederhana.

Listing 2.4: File View

```
38 <html>
39 <head>
40     <title> My Blog </title>
```

```

1 </head>
2 <body>
3   <h1> Welcome to my Blog</h1>
4 </body>
5 </html>

```

Setelah selesai membuat *file view* yang diinginkan, maka penyimpanan *file* tersebut harus diletakkan di direktori "application/views/". Nama file pada view tidak diatur huruf besar maupun huruf kecil. Pada Listing 2.4, file disimpan dengan nama "blogview".

## Menjalankan View

Menjalankan *view* pada CodeIgniter dilakukan di *file controller*. Listing di bawah ini menunjukkan kode yang harus ditulis di dalam *method controller*.

Listing 2.5: Pemanggilan View

```

12 <?php
13 class Blog extends CI_Controller{
14
15     public function index(){
16         $this->load->view('blogview');
17     }
18 }

```

Listing 2.5 merupakan listing untuk memanggil view yang dibuat pada Listing 2.4, dimana "blogview" merupakan nama file view yang diinginkan.

### 2.1.5 Models

*Model* merupakan *file* berbasis PHP yang didesain sebagai penghubung aplikasi dengan basis data. *Model* berfungsi menjalankan kueri-kueri sql seperti *insert*, *update*, *delete*, *select*, dll. Pada CodeIgniter terdapat fungsi *query builder* yang memudahkan *programmer* dalam membuat kueri. Berikut ini adalah contoh penggunaan *query builder* untuk kueri sql *insert* dan *update*.

Listing 2.6: Query Builder Insert dan Update

```

27 public function insert_entry(){
28     $this->title = $_POST['title'];
29     $this->content = $_POST['content'];
30     $this->date = time();
31
32     $this->db->insert('entries', $this);
33 }
34
35 public function update_entry(){
36     $this->title = $_POST['title'];
37     $this->content = $_POST['content'];
38     $this->date = time();
39
40     $this->db->update('entries', $this, array('id') => $_POST['id']);
41 }

```

Pada Listing 2.6, merupakan contoh penggunaan Query Builder dari Codeigniter. Pada method insert terdapat "\$this->title = \$\_POST['title'];" yang berarti bahwa variabel "title" akan mengambil masukan dari *form view* yang memiliki variable "name = title". Setelah itu variabel akan dimasukkan ke Query Builder pada baris terakhir dari *method insert* dan *update*.

Penamaan *file* pada *model* tidak memiliki aturan baku seperti pada *file controller*. *File model* pada Listing 2.6 disimpan dengan nama "blog.php".

## 1 Menjalankan Model

2 Sama seperti menjalankan *file view*, *model* pun tidak bisa dijalankan secara langsung menggunakan URL. Untuk menjalankan *model* perlu dilakukan pemanggilan pada *controller*.

Listing 2.7: Pemanggilan File Model

```

4 | class Blog_controller extends CI_Controller {
5 |
6 |     public function blog() {
7 |         $this->load->model('blog');
8 |         $data['query'] = $this->blog->get_last_ten_entries();
9 |         $this->load->view('blog', $data);
10 |     }
11 | }

```

12 Listing ?? di atas menunjukkan bahwa *file controller* melakukan pemanggilan *model* yang diikuti dengan inisialisasi *array data* dari basis data yang dimasukkan ke pemanggilan *view*, dimana "blog" pada baris 4 adalah nama *file model*, dan "blog" pada baris 6 adalah nama dari *file view* yang diinginkan.

### 16 2.1.6 Helper

17 *Helper* merupakan kelas yang membantu *programmer* dalam menjalankan *task*. CodeIgniter memiliki banyak kelas *helper*, seperti *URL Helper* yang membantu dalam membuat *link*, *Form Helper* yang membantu dalam pembuatan elemen-elemen di dalam form, *Text Helper* yang membantu dalam menjalankan berbagai *text formatting routines*, *Cookies Helper* yang membantu dalam mengatur dan membaca *cookies* yang ada, dll. *Helper* pada CodeIgniter umumnya ada pada direktori "application/helpers directory" atau "system/helpers".

## 23 Menjalankan Helper

24 Cara menjalankan *helper* pada CodeIgniter cukup dengan menambahkan kode di dalam *file Helper* atau *view*.

Listing 2.8: "Kode Helper"

```

26 | $this->load->helper('name');

```

27 Penulisan "name" pada kode diatas diisi dengan *part helper* yang diinginkan. Contoh jika pada aplikasi perlu *URL Helper* maka "name" diganti dengan "url". *Helper* juga dapat dijalankan secara otomatis dengan cara mengisi variable 'helper' pada *file autoload* yang berada di direktori "application/config/autoload.php".

### 31 2.1.7 Basis data

#### 32 Menyambungkan ke Basis Data

33 Perlu diingat bahwa kelas *model* tidak menjalankan basis data secara otomatis. Untuk membuat aplikasi terkoneksi dengan basis data, diperlukan beberapa tambahan kode pada *file model* atau *file controller*. CodeIgniter memiliki fitur *automatically connecting* yang membuat seluruh aplikasi tersambung dengan basis data pada setiap *page load*. untuk mengaktifkan fitur ini cukup mengetikkan "database" pada variabel `autoload['libraries']` di "application/config/autoload.php" seperti kode dibawah ini.

```

39 | $autoload['libraries'] = array('database');

```

- 1 Selain *autoload*, CodeIgniter juga mendukung koneksi ke basis data dengan cara manual,  
2 dengan cara menambahkan "\$this->load->database();" pada *method* atau kelas basis data  
3 ingin dijalankan.

#### 4 2.1.8 Konfigurasi Basis Data

- 5 Konfigurasi basis data pada CodeIgniter disimpan dengan cara *multi-dimensional array*.

Listing 2.9: Array Basis Data

```
6 $db['default'] = array(  
7     'dsn'           => '',  
8     'hostname'      => 'localhost',  
9     'username'       => 'root',  
10    'password'        => '',  
11    'database'        => 'database_name',  
12    'dbdriver'        => 'mysqli',  
13    'dbprefix'        => '',  
14    'pconnect'        => TRUE,  
15    'db_debug'        => TRUE,  
16    'cache_on'         => FALSE,  
17    'cachedir'        => '',  
18    'char_set'        => 'utf8',  
19    'dbcollat'        => 'utf_general_ci',  
20    'swap_pre'        => '',  
21    'encrypt'         => FALSE,  
22    'compress'        => FALSE,  
23    'strict'          => FALSE,  
24    'failover'        => array()  
25 );
```

1 Keterangan Listing 2.9:

2

Nama Konfigurasi	Deskripsi
dsn	membuat koneksi string ( <i>an all-in-one configuration sequence</i> )
hostname	nama host dari server basis data yang dipakai (umumnya bernama "localhost").
username	username yang dipakai untuk menyambungkan basis data
password	password yang cocok dengan username yang dipakai untuk menyambungkan basis data
database	nama basis data yang ingin di sambungkan
dbdriver	tipe basis data (mysqli, postgres, odbc, dll). Perlu ditulis dengan huruf kecil secara spesifik.
dbprefix	dbprefix tidak harus terisi, berguna untuk menambahkan awalan nama tabel pada saat dijalankan <i>query builder</i> .
pconnect	berisi TRUE atau FALSE untuk perlunya koneksi yang tetap
db_debug	berisi TRUE atau FALSE untuk perlunya menampilkan error dari basis data
cache_on	berisi TRUE atau FALSE untuk diperbolehkannya database query caching
cachedir	server path yang mutlak untuk direktori database query cache
char_set	set karakter yang digunakan untuk komunikasi dengan basis data
dbcollat	pemeriksaan karakter yang digunakan dalam berkomunikasi dengan basis data (hanya dipakai di driver 'mysqli' dan 'mysql').
swap_pre	sebuah tabel default yang harus bertukar dengan dbprefix.
schema	skema basis data yang nilai defaultnya adalah 'public'. Digunakan untuk driver PostgreSQL and ODBC.
encrypt	berisi TRUE atau FALSE perlu tidaknya memakai koneksi yang ter-enkripsi.
compress	perlu tidaknya memakai client compression (hanya untuk MYSQL)
stricton	berisi TRUE atau FALSE untuk perlu tidaknya memakai koneksi "Strict Mode"
port	nomor port dari basis data. Untuk menggunakannya diperlukan penambahan di config array database.

3

## 4 2.2 AngularJS

5 AngularJS[2] merupakan sebuah *framework* terstruktur yang digunakan untuk aplikasi *web*  
 6 yang bersifat dinamis. Hal tersebut memungkinkan *programmer* untuk mempergunakan  
 7 HTML sebagai *template* bahasa pemrograman dan memperluas sintaks HTML agar dapat

1 mengekspresikan komponen aplikasi dengan jelas dan ringkas. Sifat AngularJS yang meng-  
2 ikat data dan mempunyai injeksi ketergantungan (*Dependency Injection*) yang berfungsi  
3 agar suatu kelas tidak terikat dengan kelas lain, juga akan menghilangkan banyak kode  
4 yang seharusnya dituliskan oleh *programmer*, dan semua itu terjadi pada *browser* sehingga  
5 dapat disimpulkan bahwa AngularJS merupakan pasangan yang sangat ideal bagi penggu-  
6 naan teknologi server. Dalam pembuatannya, ketidakcocokkan halaman statik dan dinamik  
7 biasanya diselesaikan dengan pendekatan sebagai berikut:

- 8 1. *Library*: merupakan sebuah koleksi dari berbagai macam fungsi yang berguna dalam  
9 pembuatan aplikasi *web*, contoh: JQuery.
- 10 2. *Frameworks*: merupakan suatu implementasi dari sebuah aplikasi *web* yang menem-  
11 patkan kode yang dituliskan secara detail. *Framework* akan berperan melakukan  
12 pemanggilan ke kode yang dituliskan *programmer* ketika aplikasi membutuhkan sesu-  
13 atu yang spesifik, contoh: durandal(*light-weight javascript framework*), ember(*open-*  
14 *source javascript framework*), dll.

15 Dalam pembentukannya, AngularJS memiliki pendekatan yang berbeda. AngularJS  
16 mengajarkan *browser* sintaks baru yang disebut *directives*. Contoh *directives* adalah:

- 17 1. Keterikatan data di dalam `{{}}`;
- 18 2. Dukungan untuk *Form* dan *Form Validation*
- 19 3. Pengelompokkan HTML menjadi komponen - komponen yang dapat dipakai kembali.

### 20 2.2.1 Gambaran Konseptual

21 Berikut ini adalah beberapa bagian-bagian terpenting dalam AngularJS.



1

Konsep	Deskripsi
Template	HTML dengan tambahan <i>markup</i>
Directives	Pengembangan HTML dengan atribut dan elemen yang dibuat khusus
Model	Data yang ditunjukkan kepada pengguna pada tampilan dan bagaimana pengguna berinteraksi
Scope	Konteks dimana model disimpan sehingga <i>controller</i> , <i>directives</i> dan <i>expression</i> dapat mengaksesnya
Expression	Mengakses variabel dan fungsi dari <i>scope</i>
Compiler	Menguraikan <i>template</i> , <i>directives</i> , dan <i>expression</i>
2 Filter	Mengatur nilai dari sebuah <i>expression</i> untuk di tunjukkan kepada pengguna
View	Apa yang akan dilihat oleh pengguna
Data Binding	Menyelaraskan data yang ada pada <i>model</i> dan <i>view</i>
Controller	Mengatur logika dibalik tampilan
Dependency Injection	Membuat dan menyambungkan objek dan fungsi
Injector	Tempat penyimpanan <i>dependency injection</i>
Module	Tempat penyimpanan untuk bagian-bagian yang berbeda dalam sebuah aplikasi, yang mencakup: <i>controllers</i> , <i>services</i> , <i>filters</i> , <i>directives</i> yang mengkonfigurasi <i>injector</i>
Services	Logika bisnis independen dari <i>views</i> yang bisa dipakai kembali

### 3 2.2.2 Directives

4 *Directives* merupakan penanda pada *DOM (Document Object Mode) elements* (seperti atribut, nama, *comment*, dan kelas CSS) yang memberitahukan kepada *AngularJS HTML compiler*<sup>1</sup>, sehingga dapat melampirkan perilaku yang diinginkan kepada *DOM element* (contohnya memakai *event listener*), atau bahkan mengubah *DOM element* yang dituju beserta dengan peranannya.

9 AngularJS menyediakan sekumpulan *directives built-in* seperti yang akan dijelaskan selanjutnya.

#### 11 ng-Model

12 ng-Model adalah suatu *directives* yang dapat digunakan pada *form control* seperti *input*, *select*, *textarea*, dan *form control* unik lainnya untuk mengikat data ke *scope property (\$scope)* dengan menggunakan *ngModelController* (kelas yang menyediakan API (*Application Programming Interface*) untuk ng-Model)

#### 16 ng-Controller

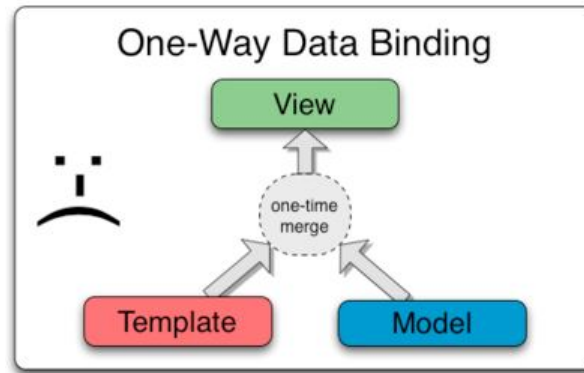
17 ng-Controller adalah suatu *directives* yang menambahkan suatu kelas *controller* pada bagian *view* dari aplikasi angularJS. ng-Controller inilah yang merupakan kelas kunci dari

<sup>1</sup>memperbolehkan pengembang untuk mengajarkan browser sintaks HTML baru

- 1 bagaimana AngularJS dapat menerapkan Model-View-Controller bersifat front-end (pada  
2 tampilan).

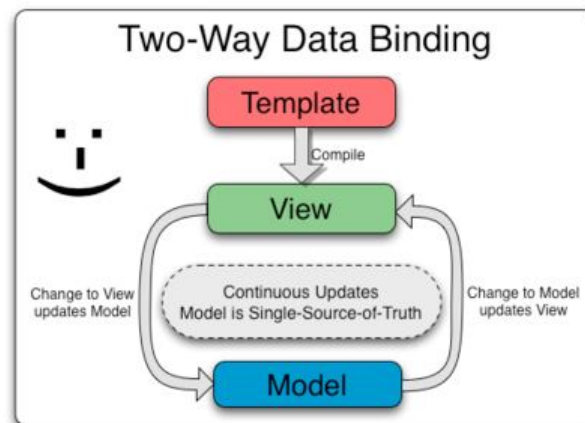
### 3 2.2.3 Data Binding

- 4 *Data Binding* pada AngularJS merupakan penyelarasan data antara *model* dan komponen-  
5 komponen *view*. Ketika *model* berubah, maka *view* pun akan berubah, begitu juga dengan  
6 sebaliknya.



Gambar 2.2: Data Binding Classical Templates System

- 7 Pada gambar 2.2 menjelaskan bahwa kebanyakan *data binding* adalah proses satu arah.  
8 Hal itu dilakukan dengan menyatukan *template* dan *model* menjadi *view*. Setelah penyatu-  
9 an, pergantian pada *model* tidak secara otomatis mengganti *view* yang sudah ditampilkan.



Gambar 2.3: Data Binding pada Angular

- 10 Pada gambar 2.3 menjelaskan perbedaan yang diberikan oleh pelaksanaan *data binding*  
11 pada AngularJS. Pertama, *template* akan di *compile* pada *browser*. Hasil dari *compile*  
12 tersebut adalah *live view*. Pada tahap ini perubahan yang terjadi di *view* akan disampaikan  
13 kepada *model*, dan perubahan yang terjadi pada *model* akan mengubah *view*.

- 14 Karena *view* merupakan proyeksi dari *model*, menyebabkan *controller* benar-benar ter-  
15 pisahkan dari *view* tanpa disadari. Hal ini mempermudah pengujian *controller*, karena  
16 terisolasi tanpa adanya *view* dan DOM (*browser dependency*).

## 2.2.4 Model-View-Controller(MVC)

AngularJS[3] juga merupakan salah satu *framework* yang menggunakan *Model-View-Controller* sebagai patokan desain aplikasi. Walaupun AngularJS mempunyai banyak fleksibilitas dalam membangun aplikasi, tetapi akan ada beberapa hal yang selalu dijumpai dalam mendesain sebuah aplikasi, diantaranya:

- Sebuah *model* selalu menampung data yang merepresentasikan keadaan aplikasi.
- *Views* yang menyajikan data tersebut.
- *Controller* yang akan selalu mengatur hubungan antara *model* dan *views*.

Model dibuat dengan menggunakan atribut berupa objek atau konten-konten primitif yang dapat menyimpan data. Berikut adalah salah satu contoh praktis dalam pembuatan *model*:

Listing 2.10: Model berupa variabel

```
var someText = 'You have started your journey.'
```

Setelah itu untuk menampilkannya maka perlu dibuat *view* dari data *model* "someText" diatas dengan cara:

Listing 2.11: Menampilkan model

```
<p> {{someText}} </p>
```

*Syntax view* 2 kurung kurawal diatas disebut sebagai interlopasi (penyusupan), karena hal tersebut memasukkan konten baru ke dalam *template* yang sudah ada.

Sementara kelas *controllers* berguna untuk memberitahu AngularJS tentang objek atau konten primitif mana dari *model* yang akan dipakai dengan cara menetapkannya ke objek '\$scope', objek '\$scope' tersebut kemudian akan diberikan kepada *controller* seperti contoh berikut:

Listing 2.12: Controller

```
function TextController($scope){
    $scope.someText = someText;
}
```

Berikut ini adalah contoh penggabungan fungsi *model*, *view*, dan *controller*:

Listing 2.13: Penyatuan model view controller

```
<html>
<body ng-controller = "TextController">
  <p>{{someText}}</p>
  <script>
    src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.4/angular.min.js">
  </script>
  <script>
    function TextController($scope){
      $scope.someText = 'You have started your journey.';
    }
  </script>
</body>
</html>
```

Hasil dari Listing 2.13 diatas adalah tulisan "You have started your journey". Walaupun cara ini dapat dilakukan dengan mudah pada aplikasi sederhana seperti contoh diatas, tetapi untuk kebanyakan aplikasi sebaiknya dibuat objek *model* untuk menyimpan *data* yang ada. Untuk itu, daripada membuat *model* seperti Listing 2.12, akan lebih baik menggunakan kode:

Listing 2.14: Penyederhanaan controller

```

1 | var message= {};
2 | message.someText = 'You have started your journey';
3 | function TextController($scope){
4 |     $scope.message = message;
5 | }

```

6 Yang kemudian akan dipanggil di *template* dengan kode:

Listing 2.15: Pemanggilan variabel

```

7 | <p>{{message.someText}}</p>

```

8 Perubahan yang dilakukan diatas berfungsi untuk mencegah perilaku tidak terduga yang  
 9 dapat terjadi dari *prototypal inheritance* dalam objek '\$scope'. Walaupun untuk sementara  
 10 hal ini dapat berjalan dengan baik, tetapi cara yang benar dalam mendefinisikan sebuah  
 11 *controller* adalah dengan menggunakan sebuah kelas yang dinamakan *module* yang menye-  
 12 diakan *namespace* untuk bagian lain dari aplikasi berhubungan. Perubahan tersebut akan  
 13 mengubah kode-kode pada listing diatas menjadi:

Listing 2.16: Penyederhanaan MVC

```

14 | <html ng-app='myApp'>
15 | <body ng-controller='TextController'>
16 |     <p>{{someText.message}}</p>
17 |     <script>
18 |         src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.4/angular.min.js">
19 |     </script>
20 |
21 |     <script>
22 |         var myAppModule = angular.module('myApp',[]);
23 |
24 |         myAppModule.controller('TextController',
25 |             function($scope){
26 |                 var someText = {};
27 |                 someText.message = 'You have started your journey';
28 |                 $scope.someText = someText;
29 |             });
30 |     </script>
31 | </body>
32 | </html>

```

33 Pada versi di atas, aplikasi memberi tahu elemen ng-app tentang nama dari modul yang  
 34 dipakai di baris ke-9. Setelah itu pada baris ke-11 sampai baris ke-16 dilakukan pemanggilan  
 35 objek AngularJS untuk membuat sebuah modul bernama myApp dan memberikan fungsi  
 36 dari *controller* untuk memanggil fungsi *controller* dari modul.

## 37 2.3 Twitter Bootstrap

38 *Twitter Bootstrap*<sup>[4]</sup> atau yang lebih dikenal dengan *Bootstrap* adalah *framework* HTML,  
 39 CSS, dan JS terpopuler dalam hal pengembangan tampilan yang responsif *mobile* pertama  
 40 dalam hal aplikasi berbasis *web*.

### 41 2.3.1 Grid System

42 *Bootstrap* merupakan responsif *mobile* pertama yang mempunyai sistem skala (*grid system*).  
 43 Sistem skala tersebut membagi layar perangkat menjadi 12 kolom yang berukuran sama,  
 44 dimana besar ukuran masing-masing kolom mengikuti besar layar perangkat. Ketika layar  
 45 semakin besar, maka ukuran masing-masing kolom pun akan semakin besar, begitu juga  
 46 sebaliknya. Cara sistem skala *Bootstrap* bekerja adalah:

- 47 1. *Rows* harus ditempatkan diantara *.container(fixed-width)* atau *.container-fluid (full-*  
 48 *width)* untuk mendapatkan keselarasan ukuran

- 1     2. *Rows* dipergunakan untuk membuat grup kolom secara *horizontal*.
- 2     3. Konten tampilan harus berada diantara kelas *columns* atau peranakan dari kelas *columns*.
- 3
- 4     4. Kelas-kelas yang telah ditetapkan seperti ".row" dan ".col-xs-4" dapat digunakan dengan segera untuk membentuk *layout*.
- 5
- 6     5. Kelas *columns* membuat *gutters* (jarak antara kolom konten) menggunakan kelas *padding*.
- 7
- 8     6. *Grid columns* dibuat dengan menyesuaikan ke-12 kolom yang sudah disediakan. Contohnya jika ingin membuat 3 kolom sama rata, maka diperlukan 3 buah kelas ".col-xs-4".
- 9
- 10
- 11    7. Jika ada lebih dari 12 kolom dalam 1 baris, maka kolom yang lebih tersebut akan dipindahkan ke baris baru sebagai satu kesatuan.
- 12
- 13    8. Kelas *grid* mempunyai fungsi untuk menyesuaikan ukuran sesuai dengan patokan ukuran yang sudah diberikan oleh *bootstrap* atau lebih besar dari angka patokan yang ada. Oleh karena itu ketika sebuah kelas ".col-md-\*" tidak memiliki kelas yang lebih besar darinya seperti kelas ".col-lg-\*", maka kelas md akan mengambil alih pada saat aplikasi dijalankan di ukuran perangkat yang lebih besar.
- 14
- 15
- 16
- 17

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
<b>Grid behavior</b>	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
<b>Container width</b>	None (auto)	750px	970px	1170px
<b>Class prefix</b>	.col-xs-	.col-sm-	.col-md-	.col-lg-
<b># of columns</b>	12			
<b>Column width</b>	Auto	~62px	~81px	~97px
<b>Gutter width</b>	30px (15px on each side of a column)			
<b>Nestable</b>	Yes			
<b>Offsets</b>	Yes			
<b>Column ordering</b>	Yes			

Gambar 2.4: Grid Option pada Bootstrap

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Gambar 2.5: Contoh Pembagian Grid Columns

### 2.3.2 Form Class

Masing-masing form akan memiliki bentuk otomatis yang diatur secara global. Dengan memakai kelas ".form-control", pengaturan ukuran dari kelas <input>, <textarea>, dan <select> akan otomatis memiliki variabel *width* 100% secara *default*. Untuk mendapatkan jarak *spacing* yang maksimal, *Bootstrap* memiliki kelas ".form-group" yang membungkus kelas *form* menjadi grup-grup.

Listing 2.17: Contoh penggunaan Twitter Bootstrap

```

7 <form>
8   <div class="form-group">
9     <label for="exampleInputEmail">Email Address</label>
10    <input type="email" class="form-control" id="exampleInputEmail" placeholder="Email">
11  </div>
12  <div>
13    <label for="exampleInputPassword1">Password</label>
14    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="
15      Password">
16  </div>
17  <div class="form-group">
18    <label for="exampleInputFile">File</label>
19    <input type="file" id="exampleInputFile">
20    <p class="help-block"> Example block-level help text here.</p>
21  </div>
22  <div class="checkbox">
23    <label>
24      <input type="checkbox"> Check me out
25    </label>
26  </div>
27  <button type="submit" class="btn btn-default">Submit</button>
28 </form>

```

#### EXAMPLE

##### Email address

##### Password

##### File input

 No file chosen

Example block-level help text here.

☐ Check me out

Gambar 2.6: Contoh Hasil Penggunaan Kelas Form

## BAB 3

## ANALISIS

### 3.1 Analisis Sistem Kini

Analisis sistem kini akan menjelaskan bagaimana sistem penilaian sidang skripsi yang telah ada dan dipakai saat ini di Universitas Katolik Parahyangan Jurusan Teknik Informatika. *Form* yang digunakan untuk sistem kini terbagi menjadi 2 bagian, yaitu *form* rekapitulasi penilaian dan *form* berita acara sidang skripsi. Berikut ini adalah penjelasan penggunaan kertas *form* tersebut.

#### 3.1.1 Form Rekapitulasi Penilaian

*Form* rekapitulasi dibagi menjadi 3 bagian (Gambar A.2), yaitu:

1. Lembar Rekapitulasi Penilaian Pembimbing
2. Lembar Rekapitulasi Penilaian Ketua Tim Penguji
3. Lembar Rekapitulasi Penilaian Anggota Tim Penguji

Lembaran tersebut akan diberikan kepada pembimbing, ketua tim penguji, dan anggota tim penguji sesuai dengan keperluannya. Setelah dibagikan, penilai wajib mengisi kolom nilai yang ingin diberikan kepada mahasiswa yang bersangkutan sesuai dengan kolom komponen penilaian yang ada. Setelah memberikan nilai, maka penilai harus melakukan perkalian antara kolom nilai dengan kolom bobot yang akan menghasilkan kolom nilai akhir mahasiswa. Terakhir penilai akan menjumlahkan seluruh kolom nilai akhir yang akan menghasilkan total nilai akhir dari mahasiswa tersebut.

Setelah semua kolom terisi, maka lembaran rekapitulasi tersebut akan dikumpulkan ke ketua tim penguji. Kemudian data yang telah tersedia akan disalin oleh ketua tim penguji kepada *form* berita acara sidang skripsi.

#### 3.1.2 Form Berita Acara Sidang Skripsi

*Form* berita acara sidang skripsi (Gambar A.1) merupakan *form* yang mencakup pengisian waktu sidang bersangkutan, data diri mahasiswa, nama dosen penguji dan pembimbing, nilai akhir dari masing-masing penilai, dan nilai akhir yang diterima mahasiswa. Seperti yang telah dibahas pada subbab sebelumnya (3.1.1) *form* berita acara sidang skripsi akan diisi oleh ketua tim penguji setelah seluruh *form* rekapitulasi dari masing-masing penilai dikumpulkan kembali kepada ketua tim penguji.

Setelah ketua tim penguji melakukan pengisian pada masing-masing kolom dari masing-masing penilai yang bersangkutan, maka ketua tim penguji akan melakukan perkalian nilai tersebut dengan bobot masing-masing penilai yang akan menghasilkan nilai akhir mahasiswa dari masing-masing penilai. Kemudian akan dihasilkan 90% nilai akhir mahasiswa untuk diberitahukan kepada mahasiswa dan diberikan kepada koordinator skripsi untuk melengkapi 10% dari nilai mahasiswa berdasarkan nilai kedisiplinan. Hasil dari seluruh proses tersebut adalah nilai akhir sidang skripsi 2 mahasiswa bersangkutan.

## 3.2 Analisis Sistem Usulan

Analisis sistem usulan dibagi menjadi beberapa tahap, yaitu analisis *back end*, analisis *front end*, dan analisis basis data. Berikut ini penjelasannya:

### 3.2.1 Analisis Back End

Analisis tahap *back end* merupakan analisis pada lapisan data akses dan kode-kode yang bekerja secara tidak terlihat pada suatu aplikasi. Pada sistem informasi penilaian sidang skripsi 2, analisis tahap ini membahas tentang pembuatan kode *model*, *view*, *controller* dari *codeigniter*. Berikut ini adalah penjelasan lengkapnya:

#### Model

Pada bagian ini akan dijelaskan tentang penggunaan *model* pada *codeigniter*. *Model* mempunyai fungsi untuk membuat sambungan dari aplikasi ke basis data. Pada *codeigniter* pemanggilan *model* dilakukan pada file *controller* dengan menggunakan fungsi khusus *codeigniter* yaitu:

```
$data = $this->skripsi_model->getAllMahasiswa();
```

Kode di atas merupakan fungsi dari *codeigniter* yang melakukan pemanggilan terhadap *file model* yang akan dipakai. Pada kasus sistem usulan, nama *file model* yang digunakan adalah "skripsi\_model". *Model* sendiri berisi kode-kode sebagai berikut:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Skripsi_model extends CI_Model {

    public function insertDataMahasiswa($tableName, $data){
        $res = $this->db->insert($tableName, $data);
    }
}
```

Berikut adalah *method* yang dimiliki oleh kelas *model*:

- public function insertDataMahasiswa(\$tablename, \$data)

Berfungsi untuk melakukan fungsi *insert* pada basis data.

Parameter:

- tablename merepresentasikan nama tabel basis data.
- data merepresentasikan data dari controller yang sudah diubah dan ingin dimasukkan kedalam basis data.



## 1 Controller

2 Pada bagian ini akan dijelaskan tentang kode dan kegunaannya pada kelas *controller*. *Controller* merupakan kelas yang mengatur hubungan antara kelas *model* dan *view* pada *codeigniter*. Dengan memanfaatkan fungsi-fungsi dari *codeigniter*, maka kelas *controller* dapat  
 4 dipersingkat dan dipermudah dalam pembuatannya. Berikut ini adalah kode pada kelas  
 6 *C\_Skripsi*:

```

7  <?php
8  defined('BASEPATH') OR exit('No direct script access allowed');
9
10 class C_skripsi extends CI_Controller {
11
12  /**
13   * Index Page for this controller.
14   *
15   * Maps to the following URL
16   *      http://example.com/index.php/welcome
17   * - or -
18   *      http://example.com/index.php/welcome/index
19   * - or -
20   * Since this controller is set as the default controller in
21   * config/routes.php, it's displayed at http://example.com/
22   *
23   * So any other public methods not prefixed with an underscore will
24   * map to /index.php/welcome/<method_name>
25   * @see https://codeigniter.com/user_guide/general/urls.html
26   */
27   public function index()
28   {
29       $this->load->view('skripsi');
30   }
31   //Check database
32   public function view_cekMahasiswa(){
33       $data = $this->skripsi_model->getAllMahasiswa();
34       $this->load->view('cek_mahasiswa', array('data' => $data));
35   }
36
37   public function tambahDataMahasiswa(){
38       $semester = $_POST['semester'];
39       $tahun = $_POST['tahun'];
40       $npm = $_POST['npm'];
41       $nama = $_POST['nama'];
42       $judul = $_POST['judul'];
43       $namaPembimbing = $_POST['namaPembimbing'];
44       $namaPembimbingPendamping = $_POST['namaPembimbingPendamping'];
45       $namaKetuaTimPenguji = $_POST['namaKetuaTimPenguji'];
46       $namaAnggotaTimPenguji = $_POST['namaAnggotaTimPenguji'];
47       $bobotKetuaTimPenguji = $_POST['bobotKetuaTimPenguji'];
48       $bobotAnggotaTimPenguji = $_POST['bobotAnggotaTimPenguji'];
49       $bobotPembimbing = $_POST['bobotPembimbing'];
50       $nilaiKoordinatorSkripsi = $_POST['nilaiKoordinatorSkripsi'];
51       $bobotKoordinatorSkripsi = $_POST['bobotKoordinatorSkripsi'];
52       $bobotTataTulisLaporanAnggota = $_POST['bobotTataTulisLaporanAnggota'];
53       $bobotKelengkapanMateriAnggota = $_POST['bobotKelengkapanMateriAnggota'];
54       $bobotPenguasaanMateriAnggota = $_POST['bobotPenguasaanMateriAnggota'];
55       $bobotPresentasiAnggota = $_POST['bobotPresentasiAnggota'];
56       $bobotPencapaianTujuanAnggota = $_POST['bobotPencapaianTujuanAnggota'];
57       $bobotTataTulisLaporanKetua = $_POST['bobotTataTulisLaporanKetua'];
58       $bobotKelengkapanMateriKetua = $_POST['bobotKelengkapanMateriKetua'];
59       $bobotPenguasaanMateriKetua = $_POST['bobotPenguasaanMateriKetua'];
60       $bobotPresentasiKetua = $_POST['bobotPresentasiKetua'];
61       $bobotPencapaianTujuanKetua = $_POST['bobotPencapaianTujuanKetua'];
62       $bobotTataTulisLaporanPembimbing = $_POST['bobotTataTulisLaporanPembimbing'];
63       $bobotKelengkapanMateriPembimbing = $_POST['bobotKelengkapanMateriPembimbing'];
64       $bobotPenguasaanMateriPembimbing = $_POST['bobotPenguasaanMateriPembimbing'];
65       $prosesBimbinganPembimbing = $_POST['prosesBimbinganPembimbing'];
66       $nilaiAkhirMahasiswa = $_POST['nilaiAkhirMahasiswa'];
67       $data_insert = array(
68           'semester' => $semester,
69           'tahun' => $tahun,
70           'npm' => $npm,
71           'nama' => $nama,
72           'judul' => $judul,
73           'namaPembimbing' => $namaPembimbing,
74           'namaPembimbingPendamping' => $namaPembimbingPendamping,
75           'namaKetuaTimPenguji' => $namaKetuaTimPenguji,
76           'namaAnggotaTimPenguji' => $namaAnggotaTimPenguji,
77           'bobotKetuaTimPenguji' => $bobotKetuaTimPenguji,
78           'bobotAnggotaTimPenguji' => $bobotAnggotaTimPenguji,
79           'bobotPembimbing' => $bobotPembimbing,

```

```

1      'nilaiKoordinatorSkripsi' => $nilaiKoordinatorSkripsi ,
2      'bobotKoordinatorSkripsi' => $bobotKoordinatorSkripsi ,
3      'bobotTataTulisLaporanAnggota' => $bobotTataTulisLaporanAnggota ,
4      'bobotKelengkapanMateriAnggota' => $bobotKelengkapanMateriAnggota ,
5      'bobotPenguasaanMateriAnggota' => $bobotPenguasaanMateriAnggota ,
6      'bobotPresentasiAnggota' => $bobotPresentasiAnggota ,
7      'bobotPencapaianTujuanAnggota' => $bobotPencapaianTujuanAnggota ,
8      'bobotTataTulisLaporanKetua' => $bobotTataTulisLaporanKetua ,
9      'bobotKelengkapanMateriKetua' => $bobotKelengkapanMateriKetua ,
10     'bobotPenguasaanMateriKetua' => $bobotPenguasaanMateriKetua ,
11     'bobotPresentasiKetua' => $bobotPresentasiKetua ,
12     'bobotPencapaianTujuanKetua' => $bobotPencapaianTujuanKetua ,
13     'bobotTataTulisLaporanPembimbing' => $bobotTataTulisLaporanPembimbing ,
14     'bobotKelengkapanMateriPembimbing' => $bobotKelengkapanMateriPembimbing ,
15     'bobotPenguasaanMateriPembimbing' => $bobotPenguasaanMateriPembimbing ,
16     'prosesBimbinganPembimbing' => $prosesBimbinganPembimbing ,
17     'nilaiAkhirMahasiswa' => $nilaiAkhirMahasiswa ,
18
19     );
20     $res = $this->skripsi_model->insertDataMahasiswa('beritaacarasidangskripsi',
21         $data_insert);
22     redirect(base_url(), 'refresh');
23 }
24
25 }

```

Berikut adalah *method-method* yang dimiliki oleh kelas *controller*:

- view\_cekMahasiswa

Berfungsi untuk memilih *file view* dan *model* yang akan dipakai pada sistem informasi.

- tambahDataMahasiswa

Berfungsi untuk mengambil data yang telah terisi dari *view* sistem dan mengubahnya menjadi *compatible* sehingga dapat diproses kedalam *method* insertDataMahasiswa pada kelas *model* yang kemudian akan diproses ke dalam bahasa sql.

### 3.2.2 Analisis Front End

Pada subbab ini akan dijelaskan bagaimana pembuatan dan fungsi otomatisasi dari AngularJS di sistem usulan. Berikut ini adalah contoh proses otomatisasi pada sistem usulan:

```

36 <body ng-app="penilaian">
37   <form ng-controller="DefaultValue">
38
39   </form>
40
41   <script>
42     angular.module('penilaian', [])
43     .controller('DefaultValue', ['$scope', function ($scope) {
44
45     }]);
46   </script>
47 </body>

```

Contoh diatas adalah inisialisasi dari AngularJS dengan menggunakan fungsi ng-app dan ng-controller yang mengatur keseluruhan fungsi otomatisasi pada sistem usulan. Pada baris pertama dilakukan inisialisasi ng-app yang berfungsi menginisialisasi nama app yang digunakan pada sistem. Setelah ng-app diinisialisasi, baru sistem usulan dapat menggunakan fungsi-fungsi AngularJS seperti menginisialisasi ng-controller pada baris ke-2 dengan nama "Default Value".

Agar *controller* dapat berfungsi, perlu dilakukan pemanggilan terhadap ng-controller dengan memanfaatkan fungsi "angular.module". Baris ke-7 bekerja dengan *parameter* ng-app ("penilaian"). Selanjutnya diikuti dengan sebuah *array*([]) kosong yang merupakan tempat yang menunjukkan *list modules* diperlukan oleh ng-app ('penilaian'). Setelah melakuk-

an inialisasi modul, maka kita dapat memanggil fungsi-fungsi daripada AngularJS untuk dijalankan, seperti *controller("DefaultValue")* ke dalam aplikasi AngularJS.

```

3 | <tr>
4 |   <td><label for="nTTLaporanK">Tata Tulis Laporan</label></td>
5 |   <td><input type="number" id="nTTLaporanK" max="100" ng-model="nilai_TTLaporanK" class="form-
6 |     nilai"/></td>
7 |   <!-- 20 -->
8 |   <td><input type="number" name="bobotTataTulisLaporanKetua" ng-model="TTLaporanK.value" ng-init
9 |     ="TTLaporanK.value = 15" min="0" max="100" class="form-nilai" readonly="readonly" /></td>
10 |   <td><input type="number" disabled="disabled" value="{{nilai_TTLaporanK * TTLaporanK.value /
11 |     100}}" ng-model="total_TTLaporanK" class="form-nilai"/></td>
12 | </tr>

```

Contoh diatas diambil dari kode *file view* untuk mengatur otomatisasi pada kolom tata tulis laporan milik ketua tim penguji. Pada baris ke-3 dan baris ke-5 adalah contoh kode diatas merupakan contoh inialisasi fungsi ng-model dari AngularJS, sementara baris ke-6 dilakukan perhitungan otomatis dari ng-model baris ke-3 dikalikan dengan ng-model baris ke-5 dan hasilnya ditampung di nilai *value* yang kemudian akan muncul ke layar *user* secara otomatis.

Berikut ini adalah nama-nama dari *model*, *view*, dan *controller* AngularJS yang dipakai pada Sistem Penilaian Sidang Skripsi 2:

- *Controller*: "DefaultValue" = Controller yang dipakai di seluruh sistem
- *Model*:
  - "tahun" = untuk melakukan otomatisasi tahun+1
  - "n\_npm" = untuk melakukan pengisian otomatis npm mahasiswa pada semua lembaran penilaian
  - "nilai\_ketua" = untuk menyimpan hasil perolehan total nilai dari lembar rekapitulasi ketua tim penguji
  - "ketua.value" = untuk menyimpan bobot ketua tim penguji terhadap nilai akhir mahasiswa
  - "total\_ketua" = untuk menyimpan perolehan nilai akhir dari ketua tim penguji
  - "nilai\_anggota" = untuk menyimpan hasil perolehan total nilai dari lembar rekapitulasi anggota tim penguji
  - "anggota.value" = untuk menyimpan bobot anggota tim penguji terhadap nilai akhir mahasiswa
  - "total\_anggota" = untuk menyimpan perolehan nilai akhir dari anggota tim penguji
  - "nilai\_pembimbing" = untuk menyimpan hasil perolehan total nilai dari lembar rekapitulasi pembimbing
  - "pembimbing.value" = untuk menyimpan bobot pembimbing terhadap nilai akhir mahasiswa
  - "total\_pembimbing" = untuk menyimpan perolehan nilai akhir dari pembimbing
  - "nilai\_koordinator" = untuk menyimpan hasil perolehan total nilai dari lembar rekapitulasi koordinator
  - "koodinator.value" = untuk menyimpan bobot koordinator terhadap nilai akhir mahasiswa

- 1       – "total\_koordinator" = untuk menyimpan perolehan nilai akhir dari koodinator
- 2       – "nilai\_TTLaporanK" = untuk menyimpan nilai tata tulis laporan pada lembar
- 3       rekapitulasi ketua tim penguji
- 4       – "TTLaporanK.value" = untuk menyimpan bobot nilai tata tulis laporan pada
- 5       lembar rekapitulasi ketua tim penguji
- 6       – "total\_TTLaporanK" = untuk menyimpan nilai akhir tata tulis laporan pada
- 7       lembar rekapitulasi ketua tim penguji
- 8       – "nilai\_KMateriK" = untuk menyimpan nilai kelengkapan materi pada lembar
- 9       rekapitulasi ketua tim penguji
- 10      – "KMateriK.value" = untuk menyimpan bobot nilai kelengkapan materi pada lem-
- 11      bar rekapitulasi ketua tim penguji
- 12      – "total\_KMateriK" = untuk menyimpan bobot nilai akhir kelengkapan materi
- 13      pada lembar rekapitulasi ketua tim penguji
- 14      – "nilai\_PMateriK" = untuk menyimpan nilai penguasaan materi pada lembar
- 15      rekapitulasi ketua tim penguji
- 16      – "PMateriK.value" = untuk menyimpan bobot nilai penguasaan materi pada lem-
- 17      bar rekapitulasi ketua tim penguji
- 18      – "total\_PMateriK" = untuk menyimpan bobot nilai akhir penguasaan materi
- 19      pada lembar rekapitulasi ketua tim penguji
- 20      – "nilai\_presentasiK" = untuk menyimpan nilai presentasi pada lembar rekapitu-
- 21      lasi ketua tim penguji
- 22      – "presentasiK.value" = untuk menyimpan bobot nilai presentasi pada lembar re-
- 23      kapitulasi ketua tim penguji
- 24      – "total\_presentasiK" = untuk menyimpan bobot nilai akhir presentasi pada lem-
- 25      bar rekapitulasi ketua tim penguji
- 26      – "nilai\_PMateriK" = untuk menyimpan nilai penguasaan materi pada lembar
- 27      rekapitulasi ketua tim penguji
- 28      – "PMateriK.value" = untuk menyimpan bobot nilai penguasaan materi pada lem-
- 29      bar rekapitulasi ketua tim penguji
- 30      – "total\_PMateriK" = untuk menyimpan bobot nilai akhir penguasaan materi
- 31      pada lembar rekapitulasi ketua tim penguji
- 32      – "nTotalKetua" = untuk menyimpan perhitungan nilai keseluruhan ketua tim
- 33      penguji yang akan dimasukkan ke lembar berita acara sidang skripsi
- 34      – "nilai\_TTLaporanA" = untuk menyimpan nilai tata tulis laporan pada lembar
- 35      rekapitulasi anggota tim penguji
- 36      – "TTLaporanA.value" = untuk menyimpan bobot nilai tata tulis laporan pada
- 37      lembar rekapitulasi anggota tim penguji
- 38      – "total\_TTLaporanA" = untuk menyimpan nilai akhir tata tulis laporan pada
- 39      lembar rekapitulasi anggota tim penguji

- 1       – "nilai\_KMateriA" = untuk menyimpan nilai kelengkapan materi pada lembar
- 2       rekapitulasi anggota tim penguji
- 3       – "KMateriA.value" = untuk menyimpan bobot nilai kelengkapan materi pada lem-
- 4       bar rekapitulasi anggota tim penguji
- 5       – "total\_KMateriA" = untuk menyimpan bobot nilai akhir kelengkapan materi
- 6       pada lembar rekapitulasi anggota tim penguji
- 7       – "nilai\_PMateriA" = untuk menyimpan nilai penguasaan materi pada lembar
- 8       rekapitulasi anggota tim penguji
- 9       – "PMateriA.value" = untuk menyimpan bobot nilai penguasaan materi pada lem-
- 10      bar rekapitulasi anggota tim penguji
- 11      – "total\_PMateriA" = untuk menyimpan bobot nilai akhir penguasaan materi pa-
- 12      da lembar rekapitulasi anggota tim penguji
- 13      – "nilai\_presentasiA" = untuk menyimpan nilai presentasi pada lembar rekapitu-
- 14      lasi anggota tim penguji
- 15      – "presentasiA.value" = untuk menyimpan bobot nilai presentasi pada lembar re-
- 16      kapitulasi anggota tim penguji
- 17      – "total\_presentasiA" = untuk menyimpan bobot nilai akhir presentasi pada lem-
- 18      bar rekapitulasi anggota tim penguji
- 19      – "nilai\_PMateriA" = untuk menyimpan nilai penguasaan materi pada lembar
- 20      rekapitulasi anggota tim penguji
- 21      – "PMateriA.value" = untuk menyimpan bobot nilai penguasaan materi pada lem-
- 22      bar rekapitulasi anggota tim penguji
- 23      – "total\_PMateriA" = untuk menyimpan bobot nilai akhir penguasaan materi pa-
- 24      da lembar rekapitulasi anggota tim penguji
- 25      – "nTotalAnggota" = untuk menyimpan perhitungan nilai keseluruhan anggota tim
- 26      penguji yang akan dimasukkan ke lembar berita acara sidang skripsi
- 27      – "nilai\_TTLaporanP" = untuk menyimpan nilai tata tulis laporan pada lembar
- 28      rekapitulasi pembimbing tim penguji
- 29      – "TTLaporanP.value" = untuk menyimpan bobot nilai tata tulis laporan pada
- 30      lembar rekapitulasi pembimbing
- 31      – "total\_TTLaporanP" = untuk menyimpan nilai akhir tata tulis laporan pada
- 32      lembar rekapitulasi pembimbing
- 33      – "nilai\_KMateriP" = untuk menyimpan nilai kelengkapan materi pada lembar
- 34      rekapitulasi pembimbing
- 35      – "KMateriP.value" = untuk menyimpan bobot nilai kelengkapan materi pada lem-
- 36      bar rekapitulasi pembimbing
- 37      – "total\_KMateriP" = untuk menyimpan bobot nilai akhir kelengkapan materi
- 38      pada lembar rekapitulasi pembimbing
- 39      – "nilai\_PMateriP" = untuk menyimpan nilai penguasaan materi pada lembar
- 40      rekapitulasi pembimbing

- 1       – "PMateriP.value" = untuk menyimpan bobot nilai penguasaan materi pada lem-
- 2       bar rekapitulasi pembimbing
- 3       – "total\_PMateriP" = untuk menyimpan bobot nilai akhir penguasaan materi pa-
- 4       da lembar rekapitulasi pembimbing
- 5       – "nilai\_PBimbinganP" = untuk menyimpan nilai proses bimbingan pada lembar
- 6       rekapitulasi pembimbing
- 7       – "PBimbinganP.value" = untuk menyimpan bobot nilai proses bimbingan pada
- 8       lembar rekapitulasi pembimbing
- 9       – "total\_PBimbinganP" = untuk menyimpan bobot nilai akhir proses bimbingan
- 10      pada lembar rekapitulasi pembimbing
- 11      – "nTotalPembimbing" = untuk menyimpan perhitungan nilai keseluruhan pem-
- 12      bimbing yang akan dimasukkan ke lembar berita acara sidang skripsi

13      • View:

- 14      – "{{tahun+1}}" = menampilkan hasil dari model "tahun" ditambahkan dengan 1
- 15      – "{{ n\_npm }}" = menampilkan npm mahasiswa
- 16      – "{nilai\_TTLaporanK \* TTLaporanK.value / 100 + nilai\_KMateriK \* KMate-
- 17      riK.value / 100 + nilai\_PMateriK \* PMateriK.value / 100 + nilai\_PresentasiK
- 18      \* presentasiK.value / 100 + nilai\_PTujuanK \* PTujuanK.value / 100}" = nilai
- 19      ketua tim penguji pada lembar berita acara sidang skripsi
- 20      – "{((nilai\_TTLaporanK \* TTLaporanK.value / 100 + nilai\_KMateriK \* KMate-
- 21      riK.value / 100 + nilai\_PMateriK \* PMateriK.value / 100 + nilai\_PresentasiK
- 22      \* presentasiK.value / 100 + nilai\_PTujuanK \* PTujuanK.value / 100) \* ke-
- 23      tua.value / 100}" = menampilkan perhitungan nilai ketua dikalikan dengan
- 24      bobot ketua tim penguji pada lembar berita acara sidang skripsi
- 25      – "{nilai\_TTLaporanA \* TTLaporanA.value / 100 + nilai\_KMateriA \* KMate-
- 26      riA.value / 100 + nilai\_PMateriA \* PMateriA.value / 100 + nilai\_PresentasiA
- 27      \* presentasiA.value / 100 + nilai\_PTujuanA \* PTujuanA.value / 100}" = nilai
- 28      anggota tim penguji pada lembar berita acara sidang skripsi
- 29      – "{((nilai\_TTLaporanA \* TTLaporanA.value / 100 + nilai\_KMateriA \* KMate-
- 30      riA.value / 100 + nilai\_PMateriA \* PMateriA.value / 100 + nilai\_PresentasiA
- 31      \* presentasiA.value / 100 + nilai\_PTujuanA \* PTujuanA.value / 100) \* anggo-
- 32      ta.value / 100}" = nilai anggota dikalikan dengan bobot anggota tim penguji
- 33      pada lembar berita acara sidang skripsi
- 34      – "{nilai\_TTLaporanP \* TTLaporanP.value / 100 + nilai\_KMateriP \* KMate-
- 35      riP.value / 100 + nilai\_PMateriP \* PMateriP.value / 100 + nilai\_PBimbinganP
- 36      \* PBimbinganP.value / 100}" = nilai pembimbing pada lembar berita acara si-
- 37      dang skripsi
- 38      – "{((nilai\_TTLaporanP \* TTLaporanP.value / 100 + nilai\_KMateriP \* KMate-
- 39      riP.value / 100 + nilai\_PMateriP \* PMateriP.value / 100 + nilai\_PBimbinganP
- 40      \* PBimbinganP.value / 100) \* pembimbing.value / 100}" = nilai pembimbing
- 41      dikalikan bobot pembimbing pada lembar berita acara sidang skripsi

- 1       – " $\{ \{ \text{nilai\_koordinator} * \text{koordinator.value} / 100 \} \}$ " = nilai koordinator skripsi dika-
- 2       likan dengan bobot koordinator skripsi pada lembar berita acara sidang skripsi
- 3       – " $\{ \{ \text{ketua.value} + \text{anggota.value} + \text{pembimbing.value} + \text{koordinator.value} \} \}$ " = menam-
- 4       pilkan total bobot pada lembar berita acara sidang skripsi
- 5       – " $\{ \{ (\text{nilai\_TTLaporanK} * \text{TTLaporanK.value} / 100 + \text{nilai\_KMateriK} * \text{KMate-}$
- 6       riK.value / 100 + nilai\\_PMateriK \* PMateriK.value / 100 + nilai\\_PresentasiK
- 7       \* presentasiK.value / 100 + nilai\\_PTujuanK \* PTujuanK.value / 100) \* ke-
- 8       tua.value / 100 + (nilai\\_TTLaporanA \* TTLaporanA.value / 100 + nilai\\_KMateriA
- 9       \* KMateriA.value / 100 + nilai\\_PMateriA \* PMateriA.value / 100 + nilai\\_PresentasiA
- 10      \* presentasiA.value / 100 + nilai\\_PTujuanA \* PTujuanA.value / 100) \* anggo-
- 11      ta.value / 100 + (nilai\\_TTLaporanP \* TTLaporanP.value / 100 + nilai\\_KMateriP
- 12      \* KMateriP.value / 100 + nilai\\_PMateriP \* PMateriP.value / 100 + nilai\\_PBimbinganP
- 13      \* PBimbinganP.value / 100) \* pembimbing.value / 100 + nilai\\_koordinator \*
- 14      koordinator.value / 100 \}" = nilai akhir mahasiswa
- 15      – " $\{ \{ \text{nilai\_TTLaporanK} * \text{TTLaporanK.value} / 100 \} \}$ " = nilai tata tulis laporan
- 16      ketua tim penguji
- 17      – " $\{ \{ \text{nilai\_KMateriK} * \text{KMateriK.value} / 100 \} \}$ " = nilai kelengkapan materi ketua
- 18      tim penguji
- 19      – " $\{ \{ \text{nilai\_PMateriK} * \text{PMateriK.value} / 100 \} \}$ " = nilai penguasaan materi ketua
- 20      tim penguji
- 21      – " $\{ \{ \text{nilai\_PresentasiK} * \text{presentasiK.value} / 100 \} \}$ " = nilai presentasi ketua tim
- 22      penguji
- 23      – " $\{ \{ \text{nilai\_PTujuanK} * \text{PTujuanK.value} / 100 \} \}$ " = nilai pencapaian tujuan ketua
- 24      tim penguji
- 25      – " $\{ \{ \text{nilai\_TTLaporanK} * \text{TTLaporanK.value} / 100 + \text{nilai\_KMateriK} * \text{KMate-}$
- 26      riK.value / 100 + nilai\\_PMateriK \* PMateriK.value / 100 + nilai\\_PresentasiK
- 27      \* presentasiK.value / 100 + nilai\\_PTujuanK \* PTujuanK.value / 100 \}" = total
- 28      nilai ketua tim penguji
- 29      – " $\{ \{ \text{nilai\_TTLaporanA} * \text{TTLaporanA.value} / 100 \} \}$ " = nilai tata tulis laporan
- 30      anggota tim penguji
- 31      – " $\{ \{ \text{nilai\_KMateriA} * \text{KMateriA.value} / 100 \} \}$ " = nilai kelengkapan materi ang-
- 32      gota tim penguji
- 33      – " $\{ \{ \text{nilai\_PMateriA} * \text{PMateriA.value} / 100 \} \}$ " = nilai penguasaan materi ketua
- 34      tim penguji
- 35      – " $\{ \{ \text{nilai\_PresentasiA} * \text{presentasiA.value} / 100 \} \}$ " = nilai presentasi anggota tim
- 36      penguji
- 37      – " $\{ \{ \text{nilai\_PTujuanA} * \text{PTujuanA.value} / 100 \} \}$ " = nilai pencapaian tujuan ang-
- 38      gota tim penguji
- 39      – " $\{ \{ \text{nilai\_TTLaporanA} * \text{TTLaporanA.value} / 100 + \text{nilai\_KMateriA} * \text{KMate-}$
- 40      riA.value / 100 + nilai\\_PMateriA \* PMateriA.value / 100 + nilai\\_PresentasiA
- 41      \* presentasiA.value / 100 + nilai\\_PTujuanA \* PTujuanA.value / 100 \}" = total
- 42      nilai anggota tim penguji

- 1        –  $\{\{\text{nilai\_TTLaporanP} * \text{TTLaporanP.value} / 100\}\} = \text{nilai tata tulis laporan}$
- 2               pembimbing
- 3        –  $\{\{\text{nilai\_KMateriP} * \text{KMateriP.value} / 100\}\} = \text{nilai kelengkapan materi pem-}$
- 4               bimbing
- 5        –  $\{\{\text{nilai\_PMateriP} * \text{PMateriP.value} / 100\}\} = \text{nilai penguasaan materi pem-}$
- 6               bimbing
- 7        –  $\{\{\text{nilai\_PBimbinganP} * \text{PBimbinganP.value} / 100\}\} = \text{nilai proses bimbingan}$
- 8               pembimbing
- 9        –  $\{\{\text{nilai\_TTLaporanP} * \text{TTLaporanP.value} / 100 + \text{nilai\_KMateriP} * \text{KMate-}$
- 10              riP.value} / 100 + \text{nilai\\_PMateriP} \* \text{PMateriP.value} / 100 + \text{nilai\\_PBimbinganP}
- 11               $* \text{PBimbinganP.value} / 100\}\} = \text{nilai akhir pembimbing}$

### 12 3.2.3 Analisis Basis Data

13 Sistem Informasi Penilaian Sidang Skripsi 2 menggunakan perangkat lunak mysql sebagai  
 14 sarana penyimpanan dan pengolahan basis data. Didalam *folder* "application" terdapat  
 15 *folder* "models" yang berfungsi menghubungkan basis data dengan sistem.

16 Berdasarkan analisa dari contoh form penilaian skripsi yang ada (Gambar A.1 dan  
 17 Gambar A.2), dapat disimpulkan bahwa penilaian skripsi membutuhkan data-data sebagai  
 18 berikut:

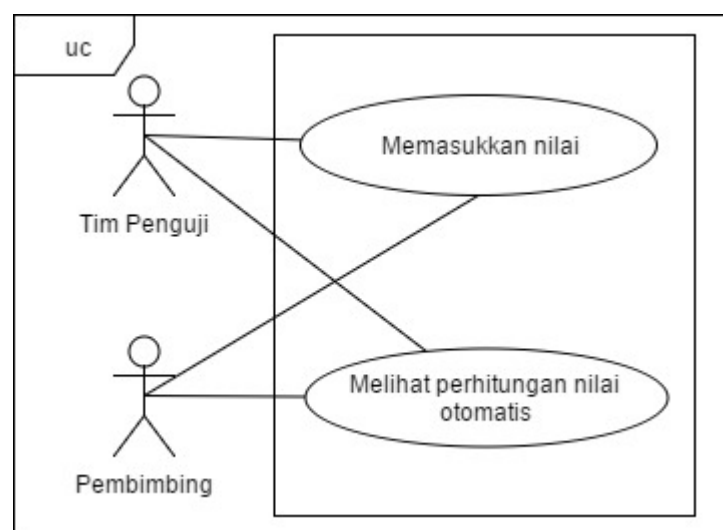
- 19 1. Semester
- 20 2. Tahun ajaran
- 21 3. NPM mahasiswa
- 22 4. Nama mahasiswa
- 23 5. Judul skripsi
- 24 6. Nama pembimbing utama/tunggal
- 25 7. Nama pembimbing pendamping(tidak harus)
- 26 8. Nama ketua tim penguji
- 27 9. Nama anggota tim penguji
- 28 10. Bobot ketua tim penguji
- 29 11. Bobot anggota tim penguji
- 30 12. Bobot pembimbing
- 31 13. Nilai koordinator skripsi
- 32 14. Bobot koordinator skripsi
- 33 15. Bobot tata tulis laporan ketua
- 34 16. Bobot kelengkapan materi ketua



- 1 17. Bobot penguasaan materi ketua
- 2 18. Bobot presentasi ketua
- 3 19. Bobot pencapaian tujuan ketua
- 4 20. Bobot tata tulis laporan anggota
- 5 21. Bobot kelengkapan materi anggota
- 6 22. Bobot penguasaan materi anggota
- 7 23. Bobot presentasi anggota
- 8 24. Bobot pencapaian tujuan anggota
- 9 25. Bobot tata tulis laporan pembimbing
- 10 26. Bobot kelengkapan materi pembimbing
- 11 27. Bobot penguasaan materi pembimbing
- 12 28. Bobot bimbingan pembimbing
- 13 29. Nilai akhir mahasiswa

14 Berdasarkan diskusi dengan dosen pembimbing, disimpulkan bahwa sistem penilaian  
 15 sidang skripsi 2 ini hanya memerlukan penyimpanan untuk bobot masing-masing penilaian  
 16 dan nilai akhir mahasiswa untuk tahap perhitungan. Hal ini dikarenakan nilai-nilai lainnya  
 17 dapat dihasilkan dengan melakukan perhitungan pada nilai akhir mahasiswa dan bobot  
 18 nilai yang diinginkan. Begitu pula dengan nilai dari masing-masing penguji.

### 19 3.3 Use Case



Gambar 3.1: Use case diagram

- 20 1. Skenario memasukkan nilai
- 21 Deskripsi: Kegiatan memasukkan nilai ke dalam kotak input yang ada.

1       Aktor: Pengguna

2       Prakondisi: -

3       Skenario:

- 4           • Pengguna memilih tempat/kolom yang sudah tersedia di tampilan
- 5           • Pengguna memasukkan nilai yang diinginkan pada tempat/kolom yang telah
- 6           dipilih.

7       2. Skenario melakukan perhitungan otomatis

8       Deskripsi: Kegiatan melakukan perhitungan secara otomatis pada tampilan

9       Aktor: Sistem

10      Prakondisi: Tempat atau kolom nilai yang ingin dihitung sudah terisi

11      Skenario:

- 12           • Pengguna mengisi kolom nilai yang sudah disediakan
- 13           • Dengan ng-model, sistem mengambil nilai dari tempat/kolom yang sudah diisi
- 14           dan melakukan perhitungan
- 15           • Sistem menampilkan hasil perhitungan ke dalam kolom yang disediakan untuk
- 16           hasil perhitungan.

## BAB 4

### PERANCANGAN

Pada bab ini akan dijelaskan mengenai perancangan aplikasi yang dibangun meliputi perancangan kelas, *routes*, *controllers*, *models*, perancangan antarmuka.

#### 4.1 Perancangan Kelas

Seperti yang sudah di jelaskan pada bab sebelumnya, untuk memodelkan sistem penilaian sidang skripsi 2 dengan menggunakan *codeigniter* membutuhkan *routes*, *controllers*, *models*, dan *views*. Hal-hal berikut akan dijelaskan pada subbab selanjutnya.

#### 4.2 Routes

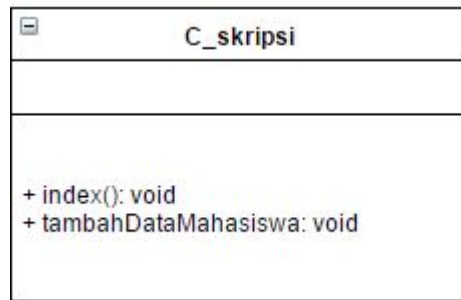
*Routes* merupakan bagian dari *codeigniter* untuk melakukan pemetaan terhadap lokasi *file controllers* dari aplikasi. Berikut adalah isi dari "config/routes":

```
$route['default_controller'] = 'C_skripsi';  
$route['404_override'] = '';  
$route['translate_url_dahses'] = FALSE;
```

Baris pertama dari kode di atas adalah nama *file controller* yang terletak di *folder controllers* yang akan diambil. Baris kedua merupakan kode untuk menangani *error* yang terjadi jika *file* yang dicari tidak ditemukan, contoh penggunaanya adalah "\$route['404\_override'] = 'errors/page\_missing;". Baris ketiga mempunyai fungsi mengganti seluruh nama *file* yang mengandung '-' menjadi '\_', contoh penggunaanya adalah: "my-controller/index" menjadi "my\_controller/index".

#### 4.3 Controllers

*Controller* terdiri dari sebuah kelas yang dinamakan "C\_Skripsi". Keseluruhan aktivitas dari sistem informasi penilaian skripsi diatur oleh kelas ini. Berikut adalah gambar kelas diagram dari *controllers*:



Gambar 4.1: Gambar diagram kelas *file controllers*

- 1 • public function `index()`
- 2 Berfungsi untuk mengarahkan pengguna ke *file views default* dari aplikasi.
- 3 • public function `tambahDataMahasiswa()`
- 4 Berfungsi untuk mengambil data dari *view* yang tersedia, untuk kemudian diolah
- 5 menjadi bahasa sql oleh *models*.

## 6 4.4 Models

7 *Models* mempunyai fungsi menghubungkan *views* dan *controllers* pada basis data. Pada  
 8 penggunaan *codeigniter*, *model* dibuat dengan sangat sederhana. Berikut adalah isi dari  
 9 kelas model:

```

10 <?php
11     defined('BASEPATH') OR exit('No direct script access allowed');
12
13     class Skripsi_model extends CI_Model {
14
15         public function insertDataMahasiswa($tableName, $data){
16             $res = $this->db->insert($tableName, $data);
17         }
18     }
  
```

- 19 • public function `insertDataMahasiswa($tablename, $data)`
- 20 Berfungsi untuk mengolah data yang sudah diolah oleh *controllers* menjadi kueri sql
- 21 *insert data*.

## 22 4.5 Perancangan Basis Data

23 Berdasarkan analisis basis data pada bab 3.2.3, maka dibuat tabel basis data.

Tabel 4.1: Tabel Perancangan Basis Data

No	Nama Tabel	Jenis Data
1	<u>id</u>	int(11)
2	tahun	year(4)
3	semester	int(1)
4	npm	varchar(10)
5	nama	varchar(256)
6	judul	varchar(256)
7	namaPembimbing	varchar(256)
8	namaPembimbingPendamping	varchar(256)
9	namaKetuaTimPenguji	varchar(256)
10	namaAnggotaTimPenguji	varchar(256)
11	bobotKetuaTimPenguji	int(2)
12	bobotAnggotaTimPenguji	int(2)
13	bobotPembimbing	int(2)
14	nilaiKoordinatorSkripsi	int(2)
15	bobotKoordinatorSkripsi	int(2)
16	bobotTataTulisLaporanAnggota	int(2)
17	bobotKelengkapanMateriAnggota	int(2)
18	bobotPenguasaanMateriAnggota	int(2)
19	bobotPresentasiAnggota	int(2)
20	bobotPencapaianTujuanAnggota	int(2)
21	bobotTataTulisLaporanKetua	int(2)
22	bobotKelengkapanMateriKetua	int(2)
23	bobotPenguasaanMateriKetua	int(2)
24	bobotPresentasiKetua	int(2)
25	bobotPencapaianTujuanKetua	int(2)
26	bobotTataTulisLaporanPembimbing	int(2)
27	bobotKelengkapanMateriPembimbing	int(2)
28	bobotPenguasaanMateriPembimbing	int(2)
29	prosesBimbinganPembimbing	int(2)
30	nilaiAkhirMahasiswa	int(2)

## 1 4.6 Perancangan Tampilan

2 Tampilan pada sistem informasi penilaian skripsi haruslah dibuat semirip mungkin dengan  
3 form penilaian skripsi yang sudah ada seperti pada lampiran gambar A.1 dan gambar A.2.

4 Perbedaan yang akan ditampilkan adalah dengan adanya otomatisasi penghitungan nilai  
5 sesuai dengan bobot yang diberikan kepada penilai. Hal ini akan memberikan kemudahan  
6 penilai untuk melakukan penilaian.

7 Gambar 4.2 adalah bayangan awal tampilan untuk sistem informasi penilaian skripsi:

**Berita Acara Sidang Skripsi**

Semester: Ganjil 2016 2017 a

Telah diselenggarakan Sidang Skripsi untuk mata kuliah AIF402-6 Skripsi 2 bagi:

Nama:  NPM:

Judul:

Dengan pembimbing dan penguji:

- Pembimbing Utama/Tunggal :
- Pembimbing Pendamping :
- Ketua Tim Penguji :
- Anggota Tim Penguji :

Rekapitulasi nilai Sidang Skripsi 2 yang diberikan oleh pembimbing, penguji, & koordinator skripsi:

No	Pembimbing/Penguji	Nilai	Bobot(%)	Nilai Akhir
1	Ketua Tim Penguji		35	
2	Anggota Tim Penguji		35	
3	Pembimbing		20	
4	Koordinator Skripsi		10	
	Total		100	

#### Rekapitulasi Penilaian Skripsi 2 (PEMBIMBING)

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan		20	
Kelengkapan Materi		20	
Penggunaan Materi		30	
Proses Bimbingan		30	
Total		100	

#### Rekapitulasi Penilaian Skripsi 2 (Ketua Tim Penguji)

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan		15	
Kelengkapan Materi		10	
Penggunaan Materi		30	
Presentasi		15	
Pencapaian Tujuan		30	
Total		100	

#### Rekapitulasi Penilaian Skripsi 2 (Anggota Tim Penguji)

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan		15	
Kelengkapan Materi		10	
Penggunaan Materi		30	
Presentasi		15	
Pencapaian Tujuan		30	
Total		100	

Gambar 4.2: Perkiraan Tampilan

## BAB 5

### IMPLEMENTASI DAN PENGUJIAN

Pada bagian ini merupakan rincian atau penjelasan lanjut mengenai lingkungan implementasi perangkat keras maupun perangkat lunak sistem informasi sidang skripsi. Bagian terakhir akan membahas tentang pengujian yang telah dilakukan pada sistem informasi.

#### 5.1 Implementasi

Pada bagian ini akan dijabarkan lingkungan pengembangan sistem informasi dan pengujian.

##### 5.1.1 Lingkungan Implementasi dan Pengujian

Implementasi dilakukan dengan menggunakan sebuah laptop. Berikut adalah spesifikasi laptop yang digunakan:

1. Processor: Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz (4 CPUs), 2.6GHz
2. RAM : 4096 MB
3. Sistem operasi : Windows 10 Home Single Language 64-bit (10.0, Build 14393)
4. Versi AngularJS : Version 1.5.2
5. Versi Codeigniter : Version 3.1.3
6. Versi TwitterBootstrap : Version 2.3.2
7. Versi Google Chrome : Version 57.0.2987.133 (64-bit)

##### 5.1.2 Hasil Implementasi

Hasil implementasi dari penelitian ini adalah sebuah sistem informasi berbasis web yang menggunakan *codeigniter*, *AngularJS*, dan *Twitter Bootstrap* sebagai dasar pembuatan. Aplikasi dapat diakses melalui jaringan *global* dengan URL <http://sipskripsi.com>. Sistem informasi terdiri dari bagian-bagian sebagai berikut:

1. Bagian formulir berita acara sidang skripsi  
Bagian ini adalah halaman yang bersangkutan dalam pengisian data diri mahasiswa yang bersangkutan, sekaligus sebagai halaman akhir yang menyimpulkan perhitungan nilai akhir mahasiswa. Kolom penilaian pada halaman ini tidak dapat diisi secara

- 1 manual kecuali kolom penilaian milik koordinator skripsi. Kolom penilaian yang lain  
2 didapatkan berdasarkan perhitungan nilai akhir masing-masing penguji.

Berita Acara Sidang Skripsi
Lembar Rekapitulasi Ketua Tim Penguji
Lembar Rekapitulasi Anggota Tim Penguji
Lembar Rekapitulasi Pembimbing
Selesai

### Berita Acara Sidang Skripsi

Semester: Genap / 2017 / 2018

---

Telah diselenggarakan Sidang Skripsi untuk mata kuliah AIF402-6 Skripsi 2 bagi:

**NPM:**  **Nama:**

**Judul:**

dengan pembimbing dan penguji:

**Pembimbing:**

**Pembimbing Pendamping:**

**Ketua Tim Penguji:**

**Anggota Tim Penguji:**

Rekapitulasi nilai Sidang Skripsi 2 yang diberikan oleh pembimbing, penguji & koordinator skripsi:

No	Pembimbing/Penguji	Nilai	Bobot(%)	Nilai Akhir
1	Ketua Tim Penguji	<input type="text" value="76.7"/>	<input type="text" value="35"/>	<input type="text" value="26.8"/>
2	Anggota Tim Penguji	<input type="text" value="77.4"/>	<input type="text" value="35"/>	<input type="text" value="27.1"/>
3	Pembimbing	<input type="text" value="77.8"/>	<input type="text" value="20"/>	<input type="text" value="15.5"/>
4	Koordinator Skripsi	<input type="text" value="100"/>	<input type="text" value="10"/>	<input type="text" value="10"/>
<b>Total</b>			<input type="text" value="100"/>	<input type="text" value="79.5"/>

Gambar 5.1: Formulir berita acara sidang skripsi 2 terisi

- 3 2. Bagian formulir rekapitulasi penilaian sidang skripsi 2.  
4 Bagian ini adalah halaman yang bersangkutan dalam menampung nilai-nilai yang  
5 diberikan oleh ketua tim penguji, anggota tim penguji, dan pembimbing pada maha-  
6 siswa.

Berita Acara Sidang Skripsi
Lembar Rekapitulasi Ketua Tim Penguji
Lembar Rekapitulasi Anggota Tim Penguji
Lembar Rekapitulasi Pembimbing
Selesai

### Rekapitulasi Penilaian Skripsi 2 (Ketua Tim Penguji)

NPM:

---

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	<input type="text" value="70"/>	<input type="text" value="15"/>	<input type="text" value="10.5"/>
Kelengkapan Materi	<input type="text" value="80"/>	<input type="text" value="10"/>	<input type="text" value="8"/>
Penguasaan Materi	<input type="text" value="75"/>	<input type="text" value="30"/>	<input type="text" value="22.5"/>
Presentasi	<input type="text" value="80"/>	<input type="text" value="15"/>	<input type="text" value="12"/>
Pencapaian Tujuan	<input type="text" value="79"/>	<input type="text" value="30"/>	<input type="text" value="23.7"/>
<b>Total</b>		<input type="text" value="100"/>	<input type="text" value="76.7"/>

Gambar 5.2: Formulir rekapitulasi ketua tim penguji terisi



[Berita Acara Sidang Skripsi](#)
[Lembar Rekapitulasi Ketua Tim Penguji](#)
[Lembar Rekapitulasi Anggota Tim Penguji](#)
[Lembar Rekapitulasi Pembimbing](#)
[Selesai](#)

### Rekapitulasi Penilaian Skripsi 2 (Anggota Tim Penguji)

NPM:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	<input type="text" value="78"/>	<input type="text" value="15"/>	<input type="text" value="11.7"/>
Kelengkapan Materi	<input type="text" value="80"/>	<input type="text" value="10"/>	<input type="text" value="8"/>
Penguasaan Materi	<input type="text" value="79"/>	<input type="text" value="30"/>	<input type="text" value="23.7"/>
Presentasi	<input type="text" value="77"/>	<input type="text" value="15"/>	<input type="text" value="11.5"/>
Pencapaian Tujuan	<input type="text" value="75"/>	<input type="text" value="30"/>	<input type="text" value="22.5"/>
Total		<input type="text" value="100"/>	<input type="text" value="77.4"/>

Gambar 5.3: Formulir rekapitulasi anggota tim penguji terisi

[Berita Acara Sidang Skripsi](#)
[Lembar Rekapitulasi Ketua Tim Penguji](#)
[Lembar Rekapitulasi Anggota Tim Penguji](#)
[Lembar Rekapitulasi Pembimbing](#)
[Selesai](#)

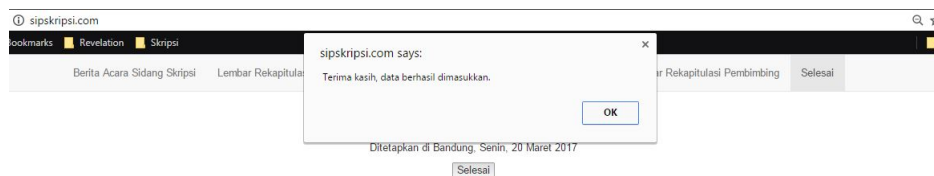
### Rekapitulasi Penilaian Skripsi 2 (Pembimbing)

NPM:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	<input type="text" value="78"/>	<input type="text" value="20"/>	<input type="text" value="15.6"/>
Kelengkapan Materi	<input type="text" value="77"/>	<input type="text" value="20"/>	<input type="text" value="15.4"/>
Penguasaan Materi	<input type="text" value="79"/>	<input type="text" value="30"/>	<input type="text" value="23.7"/>
Proses Bimbingan	<input type="text" value="77"/>	<input type="text" value="30"/>	<input type="text" value="23.1"/>
Total		<input type="text" value="100"/>	<input type="text" value="77.8"/>

Gambar 5.4: Formulir rekapitulasi pembimbing terisi

- 1 3. Bagian selesai.
- 2 Bagian ini adalah bagian terakhir dari sistem informasi. Ketika formulir sudah selesai
- 3 diisi, maka dengan menekan tombol selesai pada bagian ini, *data* yang telah terisi
- 4 akan dimasukkan ke dalam *database*.



Gambar 5.5: Ketika tombol selesai di klik

Tabel 5.1: Tabel Hasil Pengujian

Jenis Data	Input Nilai
id	32
tahun	2017
semester	2
npm	2012730017
nama	Billy Yanuar
judul	Sistem Informasi Penilaian Si- dang Skripsi 2 dengan Angu- larJS
namaPembimbing	Pascal Alfadian
namaPembimbingPendamping	-
namaKetuaTimPenguji	Ketua tim penguji
namaAnggotaTimPenguji	Anggota tim penguji
bobotKetuaTimPenguji	35
bobotAnggotaTimPenguji	35
bobotPembimbing	20
nilaiKoordinatorSkripsi	100
bobotKoordinatorSkripsi	10
bobotTataTulisLaporanAnggota	15
bobotKelengkapanMateriAnggota	10
bobotPenguasaanMateriAnggota	30
bobotPresentasiAnggota	15
bobotPencapaianTujuanKetua	30
bobotTataTulisLaporanKetua	15
bobotKelengkapanMateriKetua	10
bobotPenguasaanMateriKetua	30
bobotPresentasiKetua	15
bobotPencapaianTujuanKetua	30
bobotTataTulisLaporanPembimbing	20
bobotKelengkapanMateriPembimbing	20
bobotPenguasaanMateriPembimbing	30
prosesBimbinganPembimbing	30
nilaiAkhirMahasiswa	80

## 5.2 Hasil Pengujian

Pengujian pada sistem informasi penilaian sidang skripsi 2 merupakan pengujian bersifat fungsional, dan pengujian eksperimental. Berikut penjelasannya:

### 5.2.1 Pengujian Eksperimental

Pengujian eksperimental dilakukan dengan cara mengikuti sidang skripsi 2 yang dilakukan pada semester ganjil 2016/2017. Pada sidang yang diujikan, penilaian dilakukan dengan dua cara, yaitu dengan sistem ini yang bekerja secara manual dan dengan sistem usulan menggunakan laptop. Sehubungan dengan sifat kerahasiaan *data* pengujian, maka dengan persetujuan pembimbing pengujian eksperimental dilakukan dengan merahasiakan identitas mahasiswa yang berhubungan.

Pada saat melakukan pengujian eksperimental, sistem ini memiliki kekurangan kecerobohan manusia yang mengakibatkan kesalahan dalam perhitungan nilai baik dari lembar rekapitulasi maupun lembar berita acara sidang skripsi. Hal tersebut diketahui pada saat membandingkan nilai perhitungan nilai akhir yang didapatkan oleh mahasiswa pada sistem ini dan sistem usulan. Pada beberapa kesalahan tersebut, penguji kembali melakukan perhitungan secara manual dengan menggunakan mesin hitung berupa kalkulator pada *gadget* penguji. Setelah perhitungan dilakukan, didapatkan bahwa sistem usulan memiliki hasil yang benar.

Percobaan eksperimental menghasilkan kesimpulan sistem usulan dapat menutupi kekurangan sistem ini yang berfokus pada perhitungan penilaian mahasiswa. Dengan menggunakan sistem usulan, perhitungan nilai dibuktikan lebih akurat dibandingkan dengan sistem ini. Berikut ini adalah salah satu hasil dari pengujian eksperimental:

### 5.2.2 Pengujian Fungsional

Pengujian fungsional dilakukan untuk mengetahui apakah sistem informasi dapat menjalankan seluruh fungsi-fungsi yang dimiliki dengan baik. Hasil pengujian fungsional sistem informasi akan dijabarkan pada tabel berikut:

Tabel 5.2: Tabel Pengujian Eksperimental

Jenis Data	Input Nilai
id	30
tahun	2016
semester	1
npm	2012730004
nama	E
judul	Watermarking
namaPembimbing	Mariskha A
namaPembimbingPendamping	-
namaKetuaTimPenguji	Husnul
namaAnggotaTimPenguji	Chandra
bobotKetuaTimPenguji	35
bobotAnggotaTimPenguji	35
bobotPembimbing	20
nilaiKoordinatorSkripsi	100
bobotKoordinatorSkripsi	10
bobotTataTulisLaporanAnggota	15
bobotKelengkapanMateriAnggota	10
bobotPenguasaanMateriAnggota	30
bobotPresentasiAnggota	15
bobotPencapaianTujuanKetua	30
bobotTataTulisLaporanKetua	15
bobotKelengkapanMateriKetua	10
bobotPenguasaanMateriKetua	30
bobotPresentasiKetua	15
bobotPencapaianTujuanKetua	30
bobotTataTulisLaporanPembimbing	20
bobotKelengkapanMateriPembimbing	20
bobotPenguasaanMateriPembimbing	30
prosesBimbinganPembimbing	30
nilaiAkhirMahasiswa	84

Tabel 5.3: Tabel Pengujian Fungsional

No	Aksi Pengguna	Reaksi yang diharapkan	Keterangan
1	Pengguna menjalankan sistem informasi	Halaman berita acara ditampilkan	Reaksi sesuai
2	Pengguna memasukkan nilai	Menampilkan hasil dari perhitungan otomatis	Reaksi sesuai
3	Pengguna menekan tombol selesai	Menampilkan notifikasi data telah tersimpan	Reaksi sesuai
4	Pengguna menekan tombol ok pada notifikasi	Menampilkan kembali halaman lembar formulir berita acara awal sebelum terisi	Reaksi sesuai

## BAB 6

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, didapatkan kesimpulan-kesimpulan sebagai berikut:

1. Penilaian skripsi terutama pada skripsi 2 masih menggunakan sistem manual, yaitu penilai mengisi dengan menuliskan nilai dan menghitung nilai akhir dengan alat hitung masing-masing.
2. Proses penyimpanan nilai skripsi dilakukan oleh koordinator skripsi, setelah seluruh proses penilaian dan perhitungan nilai akhir skripsi selesai.
3. AngularJS bekerja dengan mengambil nilai input yang diperlukan dan melakukan perhitungan tanpa diperlukannya pergantian *page* pada sistem penilaian, sehingga *single page application* dapat terlaksana dengan maksimal pada sistem penilaian.

#### 6.2 Saran

Berdasarkan pengujian yang dilakukan, berikut adalah beberapa saran untuk pengembang:

- Menambahkan sistem manajemen nilai skripsi untuk melakukan fungsi *select*, *update*, dan *delete*.



## DAFTAR REFERENSI

1

- 2 [1] Ellis, R. (2014) Codeigniter user guide. [http://www.codeigniter.com/user\\_guide/](http://www.codeigniter.com/user_guide/index.html)  
3 [index.html](http://www.codeigniter.com/user_guide/index.html). 2 Mei 2017.
- 4 [2] Google (2010) Guide to angularjs documentation. [https://docs.angularjs.org/](https://docs.angularjs.org/guide)  
5 [guide](https://docs.angularjs.org/guide). 3 Mei 2017.
- 6 [3] Green, B. dan Seshadri, S. (2013) *AngularJS*. " O'Reilly Media, Inc."
- 7 [4] Mark Otto, J. (2004) Scaffolding. [http://getbootstrap.com/2.3.2/scaffolding.](http://getbootstrap.com/2.3.2/scaffolding.html)  
8 [html](http://getbootstrap.com/2.3.2/scaffolding.html). 2 Mei 2017.






# LAMPIRAN A

## FORM PENILAIAN SKRIPSI

Berikut adalah lembaran penilaian Skripsi yang di pakai di Program Studi Teknik Informatika Universitas Katolik Parahyangan:



Program Studi Teknik Informatika  
Fakultas Teknologi Informasi dan Sains  
Universitas Katolik Parahyangan

**Berita Acara Sidang Skripsi**  
Semester: Ganjil/Genap\* 20..... /20.....

Telah diselenggarakan Sidang Skripsi untuk mata kuliah AIF402-6 Skripsi 2 bagi :

NPM :  Nama:

Judul :

dengan pembimbing dan penguji :

- Pembimbing Utama/Tunggal\* :
- Pembimbing Pendamping :
- Ketua Tim Penguji :
- Anggota Tim Penguji :

Rekapitulasi nilai Sidang Skripsi 2 yang diberikan oleh pembimbing, penguji & koordinator skripsi:

No	Pembimbing/Penguji	Nilai	Bobot	Nilai Akhir
1	Ketua Tim Penguji		35%	
2	Anggota Tim Penguji		35%	
3	Pembimbing		20%	
4	Koordinator Skripsi		10%	
	<b>Total</b>		<b>100%</b>	

Ditetapkan di Bandung,   20

Ketua Tim Penguji	Anggota Tim Penguji	Pembimbing**	Koordinator Skripsi

Petunjuk pengisian :

- \* = coret yang tidak perlu, \*\* = salah satu pembimbing saja, jika pembimbing utama hadir maka harus pembimbing utama
- Jika minimal satu penguji dan/atau seluruh pembimbing tidak hadir, maka sidang harus dibatalkan dan dijadwalkan ulang, tetapi berita acara harus tetap dilaporkan kepada koordinator Skripsi atau kepada pimpinan Jurusan.

Gambar A.1: Form Penilaian Skripsi saat sidang

- 1 Berikut adalah lembaran rekapitulasi penilaian Skripsi yang di pakai di Program Studi
- 2 Teknik Informatika Universitas Katolik Parahyangan ??:

**Rekapitulasi Penilaian SKRIPSI 2 (PEMBIMBING)**

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot	Nilai Akhir	Tgl: / / 20
Tata Tulis Laporan		20%		Ttd:    Nama:
Kelengkapan Materi		20%		
Penguasaan Materi		30%		
Proses Bimbingan		30%		
<b>Total</b>				

---

**Rekapitulasi Penilaian SKRIPSI 2 (KETUA TIM PENGUJI)**

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot	Nilai Akhir	Tgl: / / 20
Tata Tulis Laporan		15%		Ttd:    Nama:
Kelengkapan Materi		10%		
Penguasaan Materi		30%		
Presentasi		15%		
Pencapaian Tujuan		30%		
<b>Total</b>				

---

**Rekapitulasi Penilaian SKRIPSI 2 (ANGGOTA TIM PENGUJI)**

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot	Nilai Akhir	Tgl: / / 20
Tata Tulis Laporan		15%		Ttd:    Nama:
Kelengkapan Materi		10%		
Penguasaan Materi		30%		
Presentasi		15%		
Pencapaian Tujuan		30%		
<b>Total</b>				

Gambar A.2: Form Rekapitulasi Penilaian Skripsi saat sidang

## LAMPIRAN B

### THE SOURCE CODE

Listing B.1: C\_skripsi.php

```
3 <?php
4 defined( 'BASEPATH' ) OR exit( 'No direct script access allowed' );
5
6 class C_skripsi extends CI_Controller {
7
8     /**
9      * Index Page for this controller.
10
11     * Maps to the following URL
12     *      http://example.com/index.php/welcome
13     * - or -
14     *      http://example.com/index.php/welcome/index
15     * - or -
16     * Since this controller is set as the default controller in
17     * config/routes.php, it's displayed at http://example.com/
18     *
19     * So any other public methods not prefixed with an underscore will
20     * map to /index.php/welcome/<method_name>
21     * @see https://codeigniter.com/user_guide/general/urls.html
22     */
23     public function index()
24     {
25         $this->load->view( 'skripsi' );
26     }
27     //Check database
28     public function view_cekMahasiswa() {
29         $data = $this->skripsi_model->getAllMahasiswa();
30         $this->load->view( 'cek_mahasiswa', array( 'data' => $data ) );
31     }
32
33     public function tambahDataMahasiswa() {
34         $semester = $_POST[ 'semester' ];
35         $tahun = $_POST[ 'tahun' ];
36         $npm = $_POST[ 'npm' ];
37         $nama = $_POST[ 'nama' ];
38         $judul = $_POST[ 'judul' ];
39         $namaPembimbing = $_POST[ 'namaPembimbing' ];
40         $namaPembimbingPendamping = $_POST[ 'namaPembimbingPendamping' ];
41         $namaKetuaTimPenguji = $_POST[ 'namaKetuaTimPenguji' ];
42         $namaAnggotaTimPenguji = $_POST[ 'namaAnggotaTimPenguji' ];
43         $bobotKetuaTimPenguji = $_POST[ 'bobotKetuaTimPenguji' ];
44         $bobotAnggotaTimPenguji = $_POST[ 'bobotAnggotaTimPenguji' ];
45         $bobotPembimbing = $_POST[ 'bobotPembimbing' ];
46         $nilaiKoordinatorSkripsi = $_POST[ 'nilaiKoordinatorSkripsi' ];
47         $bobotKoordinatorSkripsi = $_POST[ 'bobotKoordinatorSkripsi' ];
48         $bobotTataTulisLaporanAnggota = $_POST[ 'bobotTataTulisLaporanAnggota' ];
49         $bobotKelengkapanMateriAnggota = $_POST[ 'bobotKelengkapanMateriAnggota' ];
50         $bobotPenguasaanMateriAnggota = $_POST[ 'bobotPenguasaanMateriAnggota' ];
51         $bobotPresentasiAnggota = $_POST[ 'bobotPresentasiAnggota' ];
52         $bobotPencapaianTujuanAnggota = $_POST[ 'bobotPencapaianTujuanAnggota' ];
53         $bobotTataTulisLaporanKetua = $_POST[ 'bobotTataTulisLaporanKetua' ];
54         $bobotKelengkapanMateriKetua = $_POST[ 'bobotKelengkapanMateriKetua' ];
55         $bobotPenguasaanMateriKetua = $_POST[ 'bobotPenguasaanMateriKetua' ];
56         $bobotPresentasiKetua = $_POST[ 'bobotPresentasiKetua' ];
57         $bobotPencapaianTujuanKetua = $_POST[ 'bobotPencapaianTujuanKetua' ];
58         $bobotTataTulisLaporanPembimbing = $_POST[ 'bobotTataTulisLaporanPembimbing' ];
59         $bobotKelengkapanMateriPembimbing = $_POST[ 'bobotKelengkapanMateriPembimbing' ];
60         $bobotPenguasaanMateriPembimbing = $_POST[ 'bobotPenguasaanMateriPembimbing' ];
61         $prosesBimbinganPembimbing = $_POST[ 'prosesBimbinganPembimbing' ];
62         $nilaiAkhirMahasiswa = $_POST[ 'nilaiAkhirMahasiswa' ];
63         $data_insert = array(
64             'semester' => $semester,
65             'tahun' => $tahun,
66             'npm' => $npm,
67             'nama' => $nama,
68             'judul' => $judul,
69             'namaPembimbing' => $namaPembimbing,
70             'namaPembimbingPendamping' => $namaPembimbingPendamping,
71             'namaKetuaTimPenguji' => $namaKetuaTimPenguji,
72             'namaAnggotaTimPenguji' => $namaAnggotaTimPenguji,
73             'bobotKetuaTimPenguji' => $bobotKetuaTimPenguji,
74             'bobotAnggotaTimPenguji' => $bobotAnggotaTimPenguji,
75             'bobotPembimbing' => $bobotPembimbing,
76             'nilaiKoordinatorSkripsi' => $nilaiKoordinatorSkripsi,
77             'bobotKoordinatorSkripsi' => $bobotKoordinatorSkripsi,
78             'bobotTataTulisLaporanAnggota' => $bobotTataTulisLaporanAnggota,
```

```

1      'bobotKelengkapanMateriAnggota' => $bobotKelengkapanMateriAnggota,
2      'bobotPenguasaanMateriAnggota' => $bobotPenguasaanMateriAnggota,
3      'bobotPresentasiAnggota' => $bobotPresentasiAnggota,
4      'bobotPencapaianTujuanAnggota' => $bobotPencapaianTujuanAnggota,
5      'bobotTataTulisLaporanKetua' => $bobotTataTulisLaporanKetua,
6      'bobotKelengkapanMateriKetua' => $bobotKelengkapanMateriKetua,
7      'bobotPenguasaanMateriKetua' => $bobotPenguasaanMateriKetua,
8      'bobotPresentasiKetua' => $bobotPresentasiKetua,
9      'bobotPencapaianTujuanKetua' => $bobotPencapaianTujuanKetua,
10     'bobotTataTulisLaporanPembimbing' => $bobotTataTulisLaporanPembimbing,
11     'bobotKelengkapanMateriPembimbing' => $bobotKelengkapanMateriPembimbing,
12     'bobotPenguasaanMateriPembimbing' => $bobotPenguasaanMateriPembimbing,
13     'prosesBimbinganPembimbing' => $prosesBimbinganPembimbing,
14     'nilaiAkhirMahasiswa' => $nilaiAkhirMahasiswa,
15
16 );
17 $res = $this->skripsi_model->insertDataMahasiswa('beritaacarasidangskripsi',$data_insert);
18 redirect(base_url(), 'refresh');
19 }
20
21 }

```

Listing B.2: skripsi.php

```

22 <!DOCTYPE html>
23 <!--
24 To change this license header, choose License Headers in Project Properties.
25 To change this template file, choose Tools | Templates
26 and open the template in the editor.
27 -->
28 <html>
29 <head>
30 <meta charset="utf-8">
31 <meta http-equiv="X-UA-Compatible" content="IE=edge">
32 <meta name="viewport" content="width=device-width, initial-scale=1">
33 <title> Berita Acara Sidang Skripsi </title>
34
35 <!-- Bootstrap Core CSS -->
36 <link href="public/css/bootstrap/bootstrap.min.css" rel="stylesheet">
37
38 <!-- Custom Scroll Nav CSS -->
39 <link href="public/css/scrolling-nav.css" rel="stylesheet">
40
41 <!-- Custom CSS -->
42 <link href="public/css/custom.css" rel="stylesheet">
43
44 <!-- AngularJS -->
45 <script src="public/js/angularJS/angular.min.js"></script>
46
47 <!-- Mobile friendly bootstrap -->
48
49
50 </head>
51 <body ng-app="penilaian" id="page-top" data-spy="scroll" data-target=".navbar-fixed-top">
52
53
54 <!-- Navigation -->
55 <nav class="navbar navbar-default navbar-fixed-top" role="navigation">
56 <div class="container">
57 <div class="navbar-header page-scroll">
58 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target
59 =".navbar-ex1-collapse">
60 <span class="sr-only">Toggle navigation</span>
61 <span class="icon-bar"></span>
62 <span class="icon-bar"></span>
63 <span class="icon-bar"></span>
64 </button>
65 </div>
66
67 <form role="form" method="post" accept-charset="utf-8" action="<?php echo base_url
68 ()>.index.php/c_skripsi/tambahDataMahasiswa">
69 <div class="form-group">
70 <input type="text" class="form-control" value="">
71 </div>
72
73 <!-- Collect the nav links, forms, and other content for toggling -->
74 <div class="collapse navbar-collapse">
75 <ul class="nav navbar-nav">
76 <!-- Hidden li included to remove active class from about link when
77 scrolled up past about section -->
78 <li>
79 <a class="page-scroll" href="#page-top">Berita Acara Sidang
80 Skripsi</a>
81 </li>
82 <li>
83 <a class="page-scroll" href="#rekAnggota">Lembar Rekapitulasi
84 Anggota Tim Penguji</a>
85 </li>
86 <li>
87 <a class="page-scroll" href="#rekKetua">Lembar Rekapitulasi Ketua
88 Tim Penguji</a>
89 </li>
90 <li>
91 <a class="page-scroll" href="#rekPembimbing">Lembar Rekapitulasi
92 Pembimbing</a>
93 </li>
94 <li>
95 <a class="page-scroll" href="#selesai">Selesai</a>
96 </li>
97 </ul>
98 <!-- /.navbar-collapse -->
99 </div>

```

```

1      <!-- /.container -->
2      </nav>
3
4
5
6      <!-- Berita Acara Sidang Skripsi -->
7      <section id="intro" class="intro-section">
8          <!-- Page Heading -->
9          <div class="container">
10             <div class="row">
11                 <div class="col-lg-12">
12                     <div class="page-header">
13                         <h1>
14                             Berita Acara Sidang Skripsi
15                         </h1>
16
17                         <div class="semester">
18                             <p>
19                                 <label>Semester:</label>
20                                 <!-- 1 -->
21                                 <select name="semester">
22                                     <option value="1">Ganjil</option>
23                                     <option value="2">Genap</option>
24                                 </select>
25                                 <!-- 2 -->
26                                 <input id="tahun" type="number" max="9999" ng-model="tahun"
27                                     name="tahun"/>
28                                 /
29                                 <input id="tahun_1" type="number" max="9999" value="{{tahun_1+
30                                     1}}" disabled="disabled"/>
31
32                                 </p>
33                             </div>
34                         </div>
35                     </div>
36                 </div>
37             <!-- Isi -->
38             <div class="row">
39                 <div class="col-lg-12">
40
41                     <div class="form-group">
42                         Telah diselenggarakan Sidang Skripsi untuk mata kuliah AIF402-6 Skripsi 2
43                         bagi :
44
45                         <div id="pengenalMahasiswa">
46                             <p>
47                                 <!-- 3 -->
48                                 <label class="col-md-1 col-xs-6" for="npm">NPM:</label><input
49                                     maxlength="10" id="npm" class="inline-form col-md-3 col-xs-6"
50                                     ng-model="n_npm" name="npm"/>
51                                 <!-- 4 -->
52                                 <label class="col-md-1 col-xs-6" for="nama">Nama:</label><input id
53                                     ="nama" class="inline-form col-md-7 col-xs-6" name="nama"/>
54                             </p>
55                         </div>
56                         <br/>
57                         <div id="pengenalJudul">
58                             <p>
59                                 <!-- 5 -->
60                                 <label class="col-md-1 col-xs-6" for="judul">Judul:</label><input
61                                     id="judul" class="inline-form col-md-11 col-xs-6" name="judul"
62                                     />
63                             </p>
64                         </div>
65
66                         <p> dengan pembimbing dan penguji:</p>
67
68
69                         <div id="pengenalPembimbing">
70                             <p>
71                                 <label class="col-md-3 col-xs-8" for="pembimbing">Pembimbing:</label>
72                                 <!-- 6 -->
73                                 <input class="col-md-9 col-xs-4" id="pembimbing" name="namaPembimbing"
74                                     />
75                             </p>
76                         </div>
77                         <br/>
78
79                         <div id="pengenalPembimbingPendamping">
80                             <p>
81                                 <label class="col-md-3 col-xs-8" for="pembimbing2">Pembimbing
82                                     Pendamping:</label>
83                                 <!-- 7 -->
84                                 <input class="col-md-9 col-xs-4" id="pembimbing2" name="
85                                     namaPembimbingPendamping"/>
86                             </p>
87                         </div>
88                         <br/>
89
90                         <div id="pengenalKetua">
91                             <p>
92                                 <label class="col-md-3 col-xs-8" for="ketua">Ketua Tim Penguji:</label>
93                                 <!-- 8 -->
94                                 <input class="col-md-9 col-xs-4" id="ketua" name="namaKetuaTimPenguji"
95                                     />
96                             </p>
97                         </div>
98                         <br/>
99
100                         <div id="pengenalAnggota">
101                             <p>

```

```

1      <label class="col-md-3 col-xs-8" for="anggota">Anggota Tim Penguji:</
2      label>
3      <!-- 9 -->
4      <input class="col-md-9 col-xs-4" id="anggota" name="
5      namaAnggotaTimPenguji" />
6
7      </p>
8  </div>
9  <br/>
10 <p>Rekapitulasi nilai Sidang Skripsi 2 yang diberikan oleh pembimbing,
11 penguji & koordinator skripsi:</p>
12 <table class="col-md-8 col-xs-12 col-md-offset-4 col-md-pull-2 table-
13 responsive">
14 <tr>
15 <th>No</th>
16 <th>Pembimbing/Penguji</th>
17 <th>Nilai</th>
18 <th>Bobot(</th>
19 <th>Nilai Akhir</th>
20 </tr>
21 <tr>
22 <td>1</td>
23 <td><label for="nKetua">Ketua Tim Penguji</label></td>
24 <td><input type="number" id="nKetua" max="100" ng-model="
25 nilai_ketua" class="form-nilai" value="{{nilai_TTLaporanK_
26 TTLaporanK.value/_100+_nilai_KMateriK_
27 _nilai_PMateriK_
28 _presentasiK.value/_100+_nilai_PTujuanK_
29 100}}" disabled="disabled" /></td>
30 <!-- 10 -->
31 <td><input type="number" ng-model="ketua.value" ng-init="ketua.
32 value=_35" min="0" max="100" class="form-nilai" name="
33 bobotKetuaTimPenguji" readonly="readonly" /></td>
34 <td><input type="number" value="{{(nilai_TTLaporanK_
35 value/_100+_nilai_KMateriK_
36 _nilai_PMateriK_
37 _presentasiK.value/_100+_nilai_PTujuanK_
38 100)_*ketua.value/_100}}" ng-model="total_ketua" class="form
39 -nilai" disabled="disabled" /></td>
40 </tr>
41 <tr>
42 <td>2</td>
43 <td><label for="nAnggota">Anggota Tim Penguji</label></td>
44 <td><input id="nAnggota" type="number" max="100" ng-model="
45 nilai_anggota" class="form-nilai" value="{{nilai_TTLaporanA_
46 TTLaporanA.value/_100+_nilai_KMateriA_
47 _nilai_PMateriA_
48 _presentasiA.value/_100+_nilai_PTujuanA_
49 100}}" disabled="disabled" /></td>
50 <!-- 11 -->
51 <td><input type="number" ng-model="anggota.value" ng-init="anggota
52 .value=_35" min="0" max="100" class="form-nilai" name="
53 bobotAnggotaTimPenguji" readonly="readonly" /></td>
54 <td><input type="number" value="{{(nilai_TTLaporanA_
55 value/_100+_nilai_KMateriA_
56 _nilai_PMateriA_
57 _presentasiA.value/_100+_nilai_PTujuanA_
58 100)_*anggota.value/_100}}" ng-model="total_anggota" class="
59 form-nilai" disabled="disabled" /></td>
60 </tr>
61 <tr>
62 <td>3</td>
63 <td><label for="nPembimbing">Pembimbing</label></td>
64 <td><input id="nPembimbing" type="number" max="100" ng-model="
65 nilai_pembimbing" class="form-nilai" min=0 value="{{
66 nilai_TTLaporanP_
67 _nilai_KMateriP_
68 _nilai_PBimbinganP_
69 100}}" disabled="
70 disabled" /></td>
71 <!-- 12 -->
72 <td><input type="number" ng-model="pembimbing.value" ng-init="
73 pembimbing.value=_20" min="0" max="100" class="form-nilai"
74 name="bobotPembimbing" readonly="readonly" /></td>
75 <td><input type="number" value="{{(nilai_TTLaporanP_
76 value/_100+_nilai_KMateriP_
77 _nilai_PBimbinganP_
78 _nilai_KMateriP_
79 100)_*pembimbing.value/_100}}" ng-model=
80 "total_pembimbing" class="form-nilai" disabled="disabled" /></
81 td>
82 </tr>
83 <tr>
84 <td>4</td>
85 <td><label for="nKoordinator">Koordinator Skripsi</label></td>
86 <!-- 13 -->
87 <td><input id="nKoordinator" type="number" max="100" ng-model="
88 nilai_koordinator" class="form-nilai" min=0 name="
89 nilaiKoordinatorSkripsi" /></td>
90 <!-- 14 -->
91 <td><input type="number" ng-model="koordinator.value" ng-init="
92 koordinator.value=_10" min="0" max="100" class="form-nilai"
93 name="bobotKoordinatorSkripsi" readonly="readonly" /></td>
94 <td><input type="number" value="{{(nilai_koordinator*koordinator.
95 value/100}} ng-model="total_koordinator" class="form-nilai"
96 disabled="disabled" /></td>
97 </tr>
98 <tr>
99 <td></td>
100 <td colspan="2"><label for="nTotal">Total</label></td>
101 <td><input type="number" id="nTotal" max="100" disabled="disabled"
102 value="{{ketua.value+anggota.value+pembimbing.value+
103 koordinator.value}}" class="form-nilai" /></td>
104 <!-- 29 -->
105 <td><input type="number" name="nilaiAkhirMahasiswa" value=" {{(

```

```

1         nilai_TTLaporanK*TTLaporanK.value/100+nilai_KMateriK*
2         KMateriK.value/100+nilai_PMateriK*PMateriK.value/100+
3         nilai_PresentasiK*presentasiK.value/100+nilai_PTujuanK
4         *PTujuanK.value/100)*ketua.value/100+(
5         nilai_TTLaporanA*TTLaporanA.value/100+nilai_KMateriA*
6         KMateriA.value/100+nilai_PMateriA*PMateriA.value/100+
7         nilai_PresentasiA*presentasiA.value/100+nilai_PTujuanA
8         *PTujuanA.value/100)*anggota.value/100+(
9         nilai_TTLaporanP*TTLaporanP.value/100+nilai_KMateriP*
10        KMateriP.value/100+nilai_PMateriP*PMateriP.value/100+
11        nilai_PBimbinganP*PBimbinganP.value/100)*pembimbing.
12        value/100+nilai_koordinator*koordinator.value/100}}
13        class="form-nilai"/></td>
14    </tr>
15 </table>
16 </div>
17 </div>
18 </section>
19
20
21
22
23 <!-- Rekapitulasi Ketua Tim Penguji -->
24 <section id="rekKetua" class="rekKetua-section">
25     <!-- Page Heading -->
26     <div class="container">
27         <div class="row">
28             <div class="col-lg-6.col-lg-offset-3">
29                 <div class="page-header">
30                     <h1>
31                         Rekapitulasi Penilaian Skripsi 2 (Ketua Tim Penguji)
32                     </h1>
33                     <div class="semester">
34                         <p>
35                             <label for="npmK">NPM:</label><input id="npmK" maxlength="10"
36                             value="{n_npm}" disabled="disabled" />
37                         </p>
38                     </div>
39                 </div>
40             </div>
41         </div>
42         <!-- Isi Rekapitulasi Ketua Tim Penguji -->
43         <div class="row">
44             <div class="col-lg-12">
45                 <table class="col-md-8.col-xs-12.col-md-offset-4.col-md-pull-2.table-
46                 responsive">
47                     <tr>
48                         <th>Komponen Penilaian</th>
49                         <th>Nilai</th>
50                         <th>Bobot(%)</th>
51                         <th>Nilai Akhir</th>
52                     </tr>
53                     <tr>
54                         <td><label for="nTTLaporanK">Tata Tulis Laporan</label></td>
55                         <td><input type="number" id="nTTLaporanK" max="100" ng-model=
56                         nilai_TTLaporanK" class="form-nilai"/></td>
57                         <!-- 20 -->
58                         <td><input type="number" name="bobotTataTulisLaporanKetua" ng-
59                         model="TTLaporanK.value" ng-init="TTLaporanK.value/15"
60                         min="0" max="100" class="form-nilai" readonly="readonly"
61                         /></td>
62                         <td><input type="number" disabled="disabled" value="{
63                         nilai_TTLaporanK*TTLaporanK.value/100}" ng-model=
64                         total_TTLaporanK" class="form-nilai"/></td>
65                     </tr>
66                     <tr>
67                         <td><label for="nKMateriK">Kelengkapan Materi</label></td>
68                         <td><input type="number" id="nKMateriK" max="100" ng-model=
69                         nilai_KMateriK" class="form-nilai"/></td>
70                         <!-- 21 -->
71                         <td><input type="number" name="bobotKelengkapanMateriKetua" ng-
72                         model="KMateriK.value" ng-init="KMateriK.value/10" min=
73                         "0" max="100" class="form-nilai" readonly="readonly" /></
74                         td>
75                         <td><input type="number" disabled="disabled" value="{
76                         nilai_KMateriK*KMateriK.value/100}" ng-model=
77                         total_KMateriK" class="form-nilai"/></td>
78                     </tr>
79                     <tr>
80                         <td><label for="nPMateriK">Penguasaan Materi</label></td>
81                         <td><input type="number" id="nPMateriK" max="100" ng-model=
82                         nilai_PMateriK" class="form-nilai"/></td>
83                         <!-- 22 -->
84                         <td><input type="number" name="bobotPenguasaanMateriKetua" ng-
85                         model="PMateriK.value" ng-init="PMateriK.value/30" min=
86                         "0" max="100" class="form-nilai" readonly="readonly" /></td>
87                         <td><input type="number" disabled="disabled" value="{
88                         nilai_PMateriK*PMateriK.value/100}" ng-model=
89                         total_PMateriA" class="form-nilai"/></td>
90                     </tr>
91                     <tr>
92                         <td><label for="nPresentasiK">Presentasi</label></td>
93                         <td><input type="number" id="nPresentasiK" max="100" ng-model=
94                         "nilai_PresentasiK" class="form-nilai"/></td>
95                         <!-- 23 -->
96                         <td><input type="number" name="bobotPresentasiKetua" ng-model=
97                         "presentasiK.value" ng-init="presentasiK.value/10" min=
98                         "0" max="100" class="form-nilai" readonly="readonly" /></td>
99                         <td><input type="number" disabled="disabled" value="{
100                        nilai_PresentasiK*presentasiK.value/100}" ng-model=
101                        total_PresentasiK" class="form-nilai"/></td>
102                    </tr>
103                </table>

```

```

1      </tr>
2      <tr>
3          <td><label for="nPTujuanK">Pencapaian Tujuan</label></td>
4          <td><input type="number" id="nPTujuanK" max="100" ng-model="
5              nilai_PTujuanK" class="form-nilai"/></td>
6          <!-- 24 -->
7          <td><input type="number" name="bobotPencapaianTujuanKetua" ng-
8              model="PTujuanK.value" ng-init="PTujuanK.value_30" min="
9              0" max="100" class="form-nilai" readonly="readonly" /></td>
10         <td><input type="number" disabled="disabled" value="{{
11             nilai_PTujuanK*_PTujuanK.value_/100}}" ng-model="
12             total_PTujuanK" class="form-nilai"/></td>
13     </tr>
14 </tr>
15 <tr>
16     <td colspan="2"><label for="nTotalBobotK">Total</label></td>
17     <td><input type="number" id="nTotalBobotK" max="100" disabled="
18         disabled" value="{{TTLaporanK.value+KMateriK.value+
19         PMateriK.value+presentasiK.value+PTujuanK.value}}" class="
20         form-nilai"/></td>
21     <td><input type="number" id="nTotalKetua" ng-model="
22         nTotalKetua" max="100" value="{{nilai_TTLaporanK*_
23         TTLaporanK.value_/100+nilai_KMateriK*_KMateriK.value_/
24         100+nilai_PMateriK*_PMateriK.value_/100+
25         nilai_PresentasiK*_presentasiK.value_/100+
26         nilai_PTujuanK*_PTujuanK.value_/100}}" class="form-nilai
27         disabled="disabled" /></td>
28 </tr>
29 </table>
30 </div>
31 </div>
32 </div>
33 </section>
34
35 <!-- Rekapitulasi Anggota Tim Penguji -->
36 <section id="rekAnggota" class="rekAnggota-section">
37     <!-- Page Heading -->
38     <div class="container">
39         <div class="row">
40             <div class="col-lg-6.col-lg-offset-3">
41                 <div class="page-header">
42                     <h1>
43                         Rekapitulasi Penilaian Skripsi 2 (Anggota Tim Penguji)
44                     </h1>
45                     <div class="semester">
46                         <p>
47                             <label for="npmA">NPM:</label><input id="npmA" maxlength="10"
48                                 value="{{_npm}}" disabled="disabled" />
49                         </p>
50                     </div>
51                 </div>
52             </div>
53 </div>
54 <!-- Isi Rekapitulasi Anggota Tim Penguji -->
55 <div class="row">
56     <div class="col-lg-12">
57         <table class="col-md-8.col-xs-12.col-md-offset-4.col-md-pull-2.table-
58             responsive">
59             <tr>
60                 <th>Komponen Penilaian</th>
61                 <th>Nilai</th>
62                 <th>Bobot (%)</th>
63                 <th>Nilai Akhir</th>
64             </tr>
65             <tr>
66                 <td><label for="nTTLaporanA">Tata Tulis Laporan</label></td>
67                 <td><input type="number" id="nTTLaporanA" max="100" ng-model="
68                     nilai_TTLaporanA" class="form-nilai"/></td>
69                 <!-- 15 -->
70                 <td><input type="number" name="bobotTataTulisLaporanAnggota"
71                     ng-model="TTLaporanA.value" ng-init="TTLaporanA.value_15"
72                     min="0" max="100" class="form-nilai" readonly="readonly"
73                     /></td>
74                 <td><input type="number" disabled="disabled" value="{{
75                     nilai_TTLaporanA*_TTLaporanA.value_/100}}" ng-model="
76                     total_TTLaporanA" class="form-nilai"/></td>
77             </tr>
78             <tr>
79                 <td><label for="nKMateriA">Kelengkapan Materi</label></td>
80                 <td><input type="number" id="nKMateriA" max="100" ng-model="
81                     nilai_KMateriA" class="form-nilai"/></td>
82                 <!-- 16 -->
83                 <td><input type="number" name="bobotKelengkapanMateriAnggota"
84                     ng-model="KMateriA.value" ng-init="KMateriA.value_10"
85                     min="0" max="100" class="form-nilai" readonly="readonly"
86                     /></td>
87                 <td><input type="number" disabled="disabled" value="{{
88                     nilai_KMateriA*_KMateriA.value_/100}}" ng-model="
89                     total_KMateriA" class="form-nilai"/></td>
90             </tr>
91             <tr>
92                 <td><label for="nPMateriA">Penguasaan Materi</label></td>
93                 <td><input type="number" id="nPMateriA" max="100" ng-model="
94                     nilai_PMateriA" class="form-nilai"/></td>
95                 <!-- 17 -->
96                 <td><input type="number" name="bobotPenguasaanMateriAnggota"
97                     ng-model="PMateriA.value" ng-init="PMateriA.value_30"
98                     min="0" max="100" class="form-nilai" readonly="readonly"
99                     /></td>
100                 <td><input type="number" disabled="disabled" value="{{
101                     nilai_PMateriA*_PMateriA.value_/100}}" ng-model="
102                     total_PMateriA" class="form-nilai"/></td>
103             </tr>

```



```

1      <tr>
2          <td><label for="nPresentasiA">Presentasi</label></td>
3          <td><input type="number" id="nPresentasiA" max="100" ng-model=
4              "nilai_PresentasiA" class="form-nilai"/></td>
5          <td><input type="number" name="bobotPresentasiAnggota" ng-
6              model="presentasiA.value" ng-init="presentasiA.value=15"
7              min="0" max="100" class="form-nilai" readonly="readonly"
8              /></td>
9          <td><input type="number" disabled="disabled" value="{{
10              nilai_PresentasiA*presentasiA.value/100}}" ng-model="
11              total_PresentasiA" class="form-nilai"/></td>
12      </tr>
13      <tr>
14          <td><label for="nPTujuanA">Pencapaian Tujuan</label></td>
15          <td><input type="number" id="nPTujuanA" max="100" ng-model="
16              nilai_PTujuanA" class="form-nilai"/></td>
17          <td><input type="number" name="bobotPencapaianTujuanAnggota"
18              ng-model="PTujuanA.value" ng-init="PTujuanA.value=30"
19              min="0" max="100" class="form-nilai" readonly="readonly"
20              /></td>
21          <td><input type="number" disabled="disabled" value="{{
22              nilai_PTujuanA*PTujuanA.value/100}}" ng-model="
23              total_PTujuanA" class="form-nilai"/></td>
24      </tr>
25      <tr>
26          <td colspan="2"><label for="nTotalBobotA">Total</label></td>
27          <td><input type="number" id="nTotalBobotA" max="100" disabled="
28              disabled" value="{{TTLaporanA.value+KMateriA.value+
29              PMateriA.value+presentasiA.value+PTujuanA.value}}" class="
30              form-nilai"/></td>
31          <td><input type="number" id="nTotalAnggota" ng-model="
32              nTotalAnggota" max="100" class="form-nilai" value="{{
33              nilai_TTLaporanA*TTLaporanA.value/100+nilai_KMateriA
34              *KMateriA.value/100+nilai_PMateriA*PMateriA.value/
35              100+nilai_PresentasiA*presentasiA.value/100+
36              nilai_PTujuanA*PTujuanA.value/100}}" disabled="
37              disabled" /></td>
38      </tr>
39  </table>
40
41  </div>
42  </div>
43  </div>
44  </div>
45  </section>
46
47  <!-- Rekapitulasi Pembimbing -->
48  <section id="rekPembimbing" class="rekPembimbing-section">
49      <!-- Page Heading -->
50      <div class="container">
51          <div class="row">
52              <div class="col-lg-6.col-lg-offset-3">
53                  <div class="page-header">
54                      <h1>
55                          Rekapitulasi Penilaian Skripsi 2 (Pembimbing)
56                      </h1>
57                      <div class="semester">
58                          <p>
59                              <label for="npmP">NPM:</label><input id="npmP" maxlength="10"
60                              value="{{n_npm}}" disabled="disabled" />
61                          </p>
62                      </div>
63                  </div>
64              </div>
65          </div>
66          <!-- Isi Rekapitulasi Pembimbing -->
67          <div class="row">
68              <div class="col-lg-12">
69                  <table class="col-md-8.col-xs-12.col-md-offset-4.col-md-pull-2.table-
70                      responsive">
71                      <tr>
72                          <th>Komponen Penilaian</th>
73                          <th>Nilai</th>
74                          <th>Bobot(%)</th>
75                          <th>Nilai Akhir</th>
76                      </tr>
77                      <tr>
78                          <td><label for="nTTLaporanP">Tata Tulis Laporan</label></td>
79                          <td><input type="number" id="nTTLaporanP" max="100" ng-model="
80                              nilai_TTLaporanP" class="form-nilai"/></td>
81                          <td><input type="number" name="bobotTataTulisLaporanPembimbing"
82                              ng-model="TTLaporanP.value" ng-init="TTLaporanP.value=
83                              20" min="0" max="100" class="form-nilai" readonly="
84                              readonly" /></td>
85                          <td><input type="number" disabled="disabled" value="{{
86                              nilai_TTLaporanP*TTLaporanP.value/100}}" ng-model="
87                              total_TTLaporanP" class="form-nilai"/></td>
88                      </tr>
89                      <tr>
90                          <td><label for="nKMateriP">Kelengkapan Materi</label></td>
91                          <td><input type="number" id="nKMateriP" max="100" ng-model="
92                              nilai_KMateriP" class="form-nilai"/></td>
93                          <td><input type="number" name="bobotKelengkapanMateriPembimbing"
94                              ng-model="KMateriP.value" ng-init="KMateriP.value=
95                              20" min="0" max="100" class="
96                              form-nilai" readonly="readonly" /></td>
97                          <td><input type="number" disabled="disabled" value="{{
98                              nilai_KMateriP*KMateriP.value/100}}" ng-model="
99                              total_KMateriP" class="form-nilai"/></td>
100                      </tr>
101                  </table>
102              </div>
103          </div>

```

```

1      <tr>
2          <td><label for="nP MateriP">Penguasaan Materi</label></td>
3          <td><input type="number" id="nP MateriP" max="100" ng-model="
4              nilai_P MateriP" class="form-nilai"/></td>
5          <!-- 27 -->
6          <td><input type="number" name="bobotPenguasaanMateriPembimbing
7              " ng-model="PMateriP.value" ng-init="PMateriP.value=30"
8              min="0" max="100" class="form-nilai" readonly="readonly"
9              /></td>
10         <td><input type="number" disabled="disabled" value="{{
11             nilai_P MateriP * PMateriP.value / 100 }}" ng-model="
12             total_P MateriP" class="form-nilai"/></td>
13     </tr>
14     <tr>
15         <td><label for="nP BimbinganP">Proses Bimbingan</label></td>
16         <td><input type="number" id="nP BimbinganP" max="100" ng-model="
17             nilai_P BimbinganP" class="form-nilai"/></td>
18         <!-- 28 -->
19         <td><input type="number" name="prosesBimbinganPembimbing" ng-
20             model="PBimbinganP.value" ng-init="PBimbinganP.value=30"
21             min="0" max="100" class="form-nilai" readonly="readonly"
22             /></td>
23         <td><input type="number" disabled="disabled" value="{{
24             nilai_P BimbinganP * PBimbinganP.value / 100 }}" ng-model="
25             total_P BimbinganP" class="form-nilai"/></td>
26     </tr>
27     <tr>
28         <td colspan="2"><label for="nTotalBobotP">Total</label></td>
29         <td><input type="number" id="nTotalBobotP" max="100" disabled="
30             disabled" value="{{TTLaporanP.value+KMateriP.value+
31             PMateriP.value+PBimbinganP.value}}" class="form-nilai"/></td>
32         <td><input type="number" id="nTotalPembimbing" ng-model="
33             nTotalPembimbing" max="100" value="{{nilai_TTLaporanP *
34             TTLaporanP.value / 100 + nilai_KMateriP * KMateriP.value /
35             100 + nilai_P MateriP * PMateriP.value / 100 +
36             nilai_P BimbinganP * PBimbinganP.value / 100 }}" class="form-
37             nilai" disabled="disabled" /></td>
38     </tr>
39 </table>
40 </div>
41 </div>
42 </div>
43 </div>
44 </section>
45
46 <!-- Selesai -->
47 <section id="selesai" class="rekPembimbing-section">
48     <!-- Page Heading -->
49     <div class="container">
50         <div class="row">
51             <div class="col-lg-6.col-lg-offset-3">
52                 <div class="page-header">
53                     <div class="tanggal">
54                         <p>
55                             Ditetapkan di Bandung, <span id="date"></span>
56                         </p>
57                     </div>
58
59                     <script>
60                         var months = ['Januari', 'Februari', 'Maret', 'April', 'Mei', '
61                             Juni', 'Juli', 'Agustus', 'September', 'Oktober', 'November',
62                             'Desember'];
63                         var myDays = ['Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jum
64                             &#39;at', 'Sabtu'];
65                         var date = new Date();
66                         var day = date.getDate();
67                         var month = date.getMonth();
68                         var thisDay = date.getDay();
69                         thisDay = myDays[thisDay];
70                         var yy = date.getYear();
71                         var year = (yy < 1000) ? yy + 1900 : yy;
72
73                         newdate = thisDay + ',' + day + ' ' + months[month] + ' ' + year;
74                         document.getElementById("date").innerHTML = newdate;
75
76                         newyear = parseInt(year);
77                         document.getElementById("tahun").value = newyear;
78                     </script>
79                     <p><input type="submit" name="submit" value="Selesai"></p>
80                 </div>
81             </div>
82         </div>
83     </div>
84 </div>
85 </div>
86 </section>
87 </form>
88 <!-- Set Default Value pada ng-model type number -->
89 <script>
90     angular.module('penilaian', [])
91     .controller('DefaultValue', ['$scope', function ($scope) {
92
93     }]);
94 </script>
95
96 <!-- jQuery -->
97 <script src="public/js/jquery/jquery.min.js"></script>
98
99 <!-- Bootstrap Core JavaScript -->
100 <script src="public/js/bootstrap/bootstrap.min.js"></script>
101
102 <!-- Scrolling Nav JavaScript -->
103 <script src="public/js/jquery.easing.min.js"></script>

```

```

1      <script src="public/js/scrolling-nav.js"></script>
2
3      </body>
4  </html>

```

Listing B.3: skripsi\_model.php

```

5  <?php
6  defined('BASEPATH') OR exit('No direct script access allowed');
7
8  class Skripsi_model extends CI_Model {
9
10     public function insertDataMahasiswa($tableName, $data){
11         $res = $this->db->insert($tableName, $data);
12     }
13
14     public function getAllMahasiswa(){
15         $query = $this->db->get('beritaacarasidangskripsi');
16         return $query->result_array();
17     }
18 }

```

Listing B.4: scrolling-nav.css

```

19  /*
20  * Start Bootstrap - Scrolling Nav HTML Template (http://startbootstrap.com)
21  * Code licensed under the Apache License v2.0.
22  * For details, see http://www.apache.org/licenses/LICENSE-2.0.
23  */
24
25  body {
26      width: 100%;
27      height: 100%;
28  }
29
30  html {
31      width: 100%;
32      height: 100%;
33  }
34
35  @media(min-width:767px) {
36      .navbar {
37          padding: 20px 0;
38          -webkit-transition: background .5s ease-in-out, padding .5s ease-in-out;
39          -moz-transition: background .5s ease-in-out, padding .5s ease-in-out;
40          transition: background .5s ease-in-out, padding .5s ease-in-out;
41      }
42
43      .top-nav-collapse {
44          padding: 0;
45      }
46  }
47
48  /* Demo Sections - You can use these as guides or delete them - the scroller will work with any
49     sort of height, fixed, undefined, or percentage based.
50     The padding is very important to make sure the scrollspy picks up the right area when scrolled to.
51     Adjust the margin and padding of sections and children
52     of those sections to manage the look and feel of the site. */
53
54  .intro-section {
55      height: 100%;
56      padding-top: 80px;
57      background: #fff;
58  }

```

Listing B.5: custom.css

```

59  /* My Custom CSS */
60  .form-nilai{
61      width: 45px;
62  }
63
64  .page-header{
65      text-align: center;
66  }
67  tr{
68      height:30px;
69  }
70  .rekAnggota-section {
71      height: 100%;
72      padding-top: 80px;
73      background: #fff;
74  }
75
76  .rekKetua-section {
77      height: 100%;
78      padding-top: 80px;
79      background: #fff;
80  }
81
82  .rekPembimbing-section {
83      height: 100%;
84      padding-top: 80px;
85      background: #fff;
86  }
87  input[disabled="disabled"]{
88      background-color: #fff;
89  }

```

Listing B.6: scrolling-nav.js

```
1 //jQuery to collapse the navbar on scroll
2 $(window).scroll(function() {
3     if ($("#navbar").offset().top > 50) {
4         $(".navbar-fixed-top").addClass("top-nav-collapse");
5     } else {
6         $(".navbar-fixed-top").removeClass("top-nav-collapse");
7     }
8 });
9
10 //jQuery for page scrolling feature - requires jQuery Easing plugin
11 $(function() {
12     $('a.page-scroll').bind('click', function(event) {
13         var $anchor = $(this);
14         $('html, body').stop().animate({
15             scrollTop: $($anchor.attr('href')).offset().top
16         }, 1500, 'easeInOutExpo');
17         event.preventDefault();
18     });
19 });
```