

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 kini 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 kini. Dari hasil pengujian tersebut, didapatkan bahwa perhitungan yang dilakukan oleh sistem usulan lebih akurat dibandingkan dengan sistem kini.

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 codeigniter.

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

DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
1 PENDAHULUAN	1
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
2 DASAR TEORI	5
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	12
2.2.3 Data Binding	13
2.2.4 Model-View-Controller(MVC)	14
2.3 Twitter Bootstrap	16
2.3.1 Grid System	16
2.3.2 Form Class	17
3 ANALISIS	19
3.1 Analisis Sistem Kini	19
3.1.1 Form Rekapitulasi Penilaian	19
3.1.2 Form Berita Acara Sidang Skripsi	20
3.2 Analisis Sistem Usulan	20
3.2.1 Analisis Back End	21
3.2.2 Analisis Front End	22
3.2.3 Analisis Basis Data	26
3.3 Use Case	27

4 PERANCANGAN	29
4.1 Routes	29
4.2 Controllers	29
4.3 Models	29
4.4 Perancangan Basis Data	29
4.5 Perancangan Tampilan	30
5 IMPLEMENTASI DAN PENGUJIAN	33
5.1 Implementasi	33
5.1.1 Lingkungan Implementasi dan Pengujian	33
5.1.2 Hasil Implementasi	33
5.2 Hasil Pengujian	37
5.2.1 Pengujian Fungsional	38
6 KESIMPULAN DAN SARAN	47
6.1 Kesimpulan	47
6.2 Saran	47
DAFTAR REFERENSI	49
A FORM PENILAIAN SKRIPSI	51
B THE SOURCE CODE	53
C FORM REVISI	65

DAFTAR GAMBAR

2.1	Flowchart CodeIgniter	5
2.2	Data Binding Classical Templates System	13
2.3	Data Binding pada Angular	14
2.4	Grid Option pada Bootstrap	17
2.5	Contoh Pembagian Grid Columns	17
2.6	Contoh Hasil Penggunaan Kelas Form	18
3.1	Flowchart sistem usulan	21
3.2	Gambar diagram kelas <i>file controllers</i>	22
3.3	Use case diagram	27
4.1	Perkiraan Tampilan Lembar Berita Acara Sidang Skripsi	31
4.2	Perkiraan Tampilan Lembar Rekapitulasi	32
5.1	Formulir berita acara sidang skripsi 2 terisi	34
5.2	Formulir rekapitulasi ketua tim penguji terisi	35
5.3	Formulir rekapitulasi anggota tim penguji terisi	35
5.4	Formulir rekapitulasi pembimbing terisi	36
5.5	Pengisian Checkbox	36
5.6	Pop Up untuk memastikan pengisian nilai sistem	37
5.7	Tampilan sistem pada platform Microsoft Edge	42
5.8	Tampilan sistem pada platform Mozilla Firefox	43
5.9	Tampilan menu sistem pada platform browser smartphone	44
5.10	Tampilan lembar berita acara sistem sidang skripsi pada platform browser smartphone	44
5.11	Tampilan lembar nilai berita acara sidang skripsi sistem pada platform browser smartphone	44
5.12	Tampilan lembar rekapitulasi ketua tim penguji sistem pada platform browser smartphone	45
5.13	Tampilan lembar rekapitulasi anggota tim penguji sistem pada platform browser smartphone	45
5.14	Tampilan lembar rekapitulasi pembimbing sistem pada platform browser smartphone	46
5.15	Tampilan lembar selesai pada platform browser smartphone	46
A.1	Form Penilaian Skripsi saat sidang	51
A.2	Form Rekapitulasi Penilaian Skripsi saat sidang	52
C.1	Form Revisi Skripsi	65

DAFTAR TABEL

2.1	Tabel Keterangan Basis Data	11
2.2	Tabel Gambaran Konseptual	12
3.1	Tabel masukan pada tampilan	23
3.2	Tabel model	24
3.3	Tabel view	25
4.1	Tabel Perancangan Basis Data	30
5.1	Tabel Hasil Implementasi	37
5.2	Tabel Pengujian Eksperimental 1	38
5.3	Tabel Pengujian Eksperimental 2	39
5.4	Tabel Pengujian Eksperimental 3	40
5.5	Tabel Pengujian Eksperimental 4	40
5.6	Tabel Pengujian Eksperimental 5	41
5.7	Tabel Pengujian Fungsional	42

¹

BAB 1

²

PENDAHULUAN

³ 1.1 Latar Belakang

⁴ Program Studi Teknik Informatika di Universitas Katolik Parahyangan memiliki beberapa
⁵ syarat kelulusan. Salah satunya adalah minimal SKS lulus adalah 144 sks yang terdiri dari
⁶ mata kuliah wajib, pilihan wajib, dan pilihan. Selain itu Indeks Prestasi Kumulatif (IPK)
⁷ minimum yang diperlukan adalah 2.00 dengan maksimum 14 semester. Salah satu mata
⁸ kuliah wajib yang harus ditempuh dan lulus adalah Skripsi. Skripsi pada Program Studi
⁹ Teknik Informatika di Universitas Katolik Parahyangan dibagi menjadi 2 mata kuliah yaitu
¹⁰ Skripsi 1 dan Skripsi 2. Sidang pada mata kuliah Skripsi 1 dan Skripsi 2 adalah proses
¹¹ yang harus ditempuh untuk mendapatkan nilai akhir. Sidang dilakukan setelah seluruh
¹² persyaratan pada mata kuliah Skripsi 1 atau Skripsi 2 diselesaikan. Tugas akhir ini berfokus
¹³ pada sidang untuk mata kuliah Skripsi 2.

¹⁴ Pada sidang mata kuliah Skripsi 2, penilaian dilakukan oleh ketua tim penguji dan ang-
¹⁵gota tim penguji. Penilaian pada sidang skripsi 2 bersifat manual, dimana penilai akan me-
¹⁶nuliskan nilai yang ingin diberikan serta melakukan perhitungan nilai untuk mendapatkan
¹⁷ nilai akhir mahasiswa pada lembar rekapitulasi masing-masing penilai. Lembar rekapitulasi
¹⁸ yang telah selesai dihitung kemudian akan dikumpulkan kepada ketua tim penguji untuk
¹⁹ dituliskan di lembar berita acara sidang skripsi yang kemudian akan diproses menjadi nilai
²⁰ akhir mahasiswa bersangkutan.

²¹ Sifat manual ini mengakibatkan kelalaian manusia dalam melakukan penilaian bebe-
²²ra kali tidak dapat dihindarkan. Kelalaian manusia yang biasa terjadi contohnya adalah
²³ kesalahan perhitungan nilai akhir oleh penilai, kesalahan penulisan nama penilai dan NPM
²⁴ mahasiswa yang bersangkutan, kesalahan penulisan semester atau tahun ajaran saat pe-
²⁵nilaian skripsi¹. Untuk mengatasi hal-hal tersebut, diperlukan suatu sistem yang dapat
²⁶ menanggulangi masalah pengisian, kalkulasi perhitungan, dan juga penyimpanan skripsi.

²⁷ Menurut penjelasan di atas, maka penulis mengusulkan otomatisasi sistem dalam peni-
²⁸laian skripsi yang akan dibangun guna mengurangi kesalahan-kesalahan kecil yang dapat
²⁹berakibat fatal pada nilai mahasiswa yang bersangkutan. Berdasarkan hal tersebut diban-
³⁰gun tugas akhir otomatisasi sistem penilaian skripsi dengan cara membuat sebuah aplikasi
³¹ berbasis *web* yaitu Sistem Informasi Penilaian Sidang Skripsi2.

³² Pada tugas akhir ini, akan dibangun sebuah sistem penilaian yang menanggulangi
³³masalah-masalah tersebut, dengan cara menjadikan beberapa masukan(*input*) otomatis dan
³⁴juga melakukan eksekusi perhitungan nilai akhir sesuai bobot secara otomatis. Hal ini di-

¹berdasarkan diskusi dengan dosen pembimbing

anggap akan memudahkan penilai dalam melakukan proses penilaian skripsi, karena penilai tidak perlu lagi repot melakukan perhitungan nilai dan juga mengisi masukan-masukan yang sudah terisi secara otomatis. Terdapat banyak fungsi yang dikerjakan pada tugas akhir ini, seperti fungsi *insert* berfungsi untuk memasukkan nilai ke basis data, *update* berfungsi untuk mengubah nilai pada basis data yang ada, *delete* berfungsi untuk menghapus nilai yang ada pada basis data, dan fungsi lainnya. Tugas akhir ini berfokus pada fungsi *insert*.

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

AngularJS juga bisa diintegrasikan dengan aplikasi yang menggunakan *framework* lain, sehingga sangat berguna dalam pengembangan aplikasi berbasis *web* terutama pada pengembangan Sistem Informasi Penilaian Sidang Skripsi2 yang akan dibangun.

Selain AngularJS, sistem usulan ini juga memakai 2 *framework* pendukung yaitu CodeIgniter dan Twitter Bootstrap. CodeIgniter dipakai untuk memudahkan aliran data pada sistem usulan yang dibangun, sedangkan Twitter Bootstrap dipakai untuk mempermudah pengaturan tampilan pada sistem usulan yang akan dibangun.

1.2 Rumusan Masalah

Berikut adalah susunan permasalahan yang akan dibahas pada tugas akhir ini:

1. Bagaimana sistem penilaian skripsi 2 yang ada pada Program Studi Teknik Informatika di Universitas Katolik Parahyangan?
2. Bagaimana proses penyimpanan nilai skripsi?
3. Bagaimana AngularJS bekerja pada eksekusi perhitungan nilai akhir?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah dibuat, maka tujuan tugas akhir ini dijelaskan ke dalam poin-poin sebagai berikut:

1. Mempelajari sistem penilaian skripsi pada Program Studi Teknik Informatika di Universitas Katolik Parahyangan
2. Merancang dan mengimplementasi proses penyimpanan nilai skripsi
3. Mengimplementasi AngularJS untuk mengeksekusi perhitungan nilai akhir

1.4 Batasan Masalah

Tugas akhir ini memiliki batasan-batasan sebagai berikut:

1. Tugas akhir ini hanya dilakukan untuk formulir penilaian mata kuliah Skripsi 2

¹ 2. Tugas akhir ini hanya melakukan fungsi *insert* ke basis data

² 1.5 Metode Penelitian

³ Dalam tugas akhir ini, akan dilakukan langkah-langkah berikut:

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

¹¹ 1.6 Sistematika Penulisan

¹² Berikut adalah sistematika penulisan dari dokumen ini:

- ¹³ • Bab 1 membahas latar belakang, rumusan masalah, tujuan penulisan, batasan-batasan, serta metode yang digunakan pada tugas akhir ini.
- ¹⁴ • Bab 2 membahas teori-teori yang digunakan dalam tugas akhir ini, yaitu AngularJS, Code Igniter, dan Twitter Bootstrap.
- ¹⁵ • Bab 3 menganalisis sistem kini, beserta perubahan-perubahan yang harus dilakukan.
- ¹⁶ • Bab 4 membahas perancangan yang dilakukan sebelum mengimplementasikan integrasi yang dimaksud, mencakup protokol, basisdata, beserta antarmukanya.
- ¹⁷ • Bab 5 membahas implementasi serta pengujian dari integrasi yang telah dilakukan.
- ¹⁸ • Bab 6 membahas kesimpulan dari keseluruhan tugas akhir ini, serta saran-saran yang dapat diberikan untuk tugas akhir berikutnya.

¹

BAB 2

²

DASAR TEORI

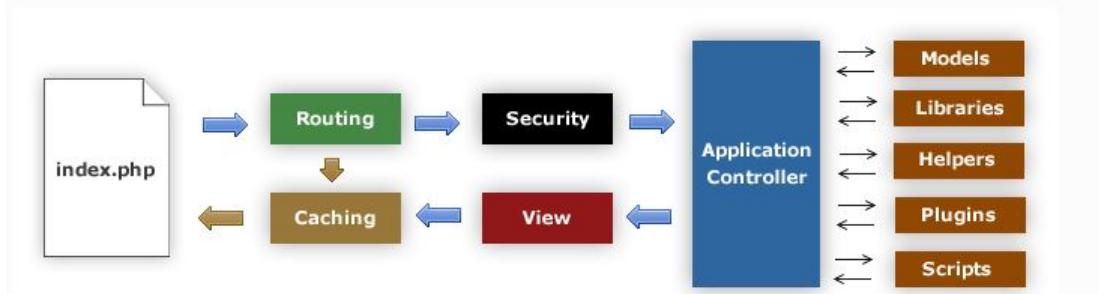
³ Pada bab ini akan dijelaskan dasar-dasar teori mengenai CodeIgniter, AngularJS, dan Twitter Bootstrap.

⁵ 2.1 CodeIgniter

⁶ 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.

¹⁴ 2.1.1 Flowchart Aplikasi CodeIgniter

¹⁵ Pada Gambar 2.1 adalah *flowchart* aliran data pada CodeIgniter.



Gambar 2.1: Flowchart CodeIgniter

¹⁶ Keterangan Gambar 2.1.

- ¹⁷ 1. Index.php berfungsi sebagai pengontrol utama, yang menginisialisasikan sumber-sumber yang diperlukan untuk menjalankan CodeIgniter.
- ¹⁹ 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 pengajaran 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.

Listing 2.1: CI Controller

```

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

```

Nama *file* pada kode Listing 2.1 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/ 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 seperti pada Listing 2.2, diperlukan tambahan pada URL pemanggilannya. Sebagai contoh, pemanggilan *method* pada kode di atas dilakukan dengan URL 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 21 URL yang lengkap untuk pemanggilan, melainkan *controller* dapat dipanggil secara otomatis dengan URL example.com saja. Namun, untuk dapat menjalankan fungsi ini, diperlukan 22 sedikit pengaturan pada file "application/config/routes.php" yaitu perubahan variabel yang 23 ditunjukkan Listing 2.3.

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 27 tanpa URL yang terspesifikasi menjalankan *controller*.

29 2.1.4 Views

30 Sebuah *views* merupakan bagian yang mengatur tampilan aplikasi yang akan ditunjukkan 31 kepada pengguna. *Views* meliputi *footer*, *header*, *sidebar*, dll. Pada CodeIgniter, *Views* 32 tidak dapat dijalankan secara langsung dari URL, tapi *views* harus dijalankan melalui file 33 *controller* yang ada. Hal ini dilakukan guna memudahkan *programmer* dan mewujudkan 34 *framework MVC* pada CodeIgniter.

35 Pembuatan Views

36 Pembuatan *file view* pada dasarnya sama seperti pembuatan *file* berbasis PHP. Berikut ini 37 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>

```

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

9 Menjalankan View

10 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 |

```

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

21 2.1.5 Models

22 *Model* merupakan *file* berbasis PHP yang didesain sebagai penghubung aplikasi dengan
 23 basis data. *Model* berfungsi menjalankan kueri-kueri sql seperti *insert*, *update*, *delete*,
 24 *select*, dll. Pada CodeIgniter terdapat fungsi *query builder* yang memudahkan *programmer*
 25 dalam membuat kueri. Berikut ini adalah contoh penggunaan *query builder* untuk kueri sql
 26 *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 |

```

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

47 Penamaan *file* pada *model* tidak memiliki aturan baku seperti pada *file controller*. *File*
 48 *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 2.7 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 Listing 2.8 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 Pada subbab ini akan dibahas mengenai koneksi pada basis data dan konfigurasi basis data pada *codeigniter*.

34 Koneksi pada Basis Data

35 Perlu diingat bahwa kelas *model* tidak menjalankan basis data secara otomatis. Untuk 36 membuat aplikasi terkoneksi dengan basis data, diperlukan beberapa tambahan kode pada 37 *file model* atau *file controller*. CodeIgniter memiliki fitur *automatically connecting* yang 38 membuat seluruh aplikasi tersambung dengan basis data pada setiap *page load*. untuk

- 1 mengaktifkan fitur ini cukup mengetikkan "database" pada variabel autoload['libraries'] di
- 2 "application/config/autoload.php" yang ditunjukkan pada Listing 2.9.

Listing 2.9: Autoload Codeigniter

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

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

7 2.1.8 Konfigurasi Basis Data

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

Listing 2.10: Array Basis Data

```
9 | $db[ 'default' ] = array(
10 |     'dsn'          => '',
11 |     'hostname'     => 'localhost',
12 |     'username'    => 'root',
13 |     'password'    => '',
14 |     'database'     => 'database_name',
15 |     'dbdriver'     => 'mysqli',
16 |     'dbprefix'     => '',
17 |     'pconnect'     => TRUE,
18 |     'db_debug'     => TRUE,
19 |     'cache_on'     => FALSE,
20 |     'cachedir'     => '',
21 |     'char_set'     => 'utf8',
22 |     'dbcollat'     => 'utf_general_ci',
23 |     'swap_pre'     => '',
24 |     'encrypt'      => FALSE,
25 |     'compress'     => FALSE,
26 |     'striction'   => FALSE,
27 |     'failover'     => array()
28 | );
```

- ¹ Berikut ini adalah keterangan dari Listing 2.10.

Tabel 2.1: Tabel Keterangan Basis Data

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, postgre, 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.

² 2.2 AngularJS

- ³ AngularJS [2] merupakan sebuah *framework* terstruktur yang digunakan untuk aplikasi ⁴ web yang bersifat dinamis. Hal tersebut memungkinkan *programmer* untuk mempergunakan ⁵ HTML sebagai *template* bahasa pemrograman dan memperluas sintaks HTML agar ⁶ dapat mengekspresikan komponen aplikasi dengan jelas dan ringkas. Sifat AngularJS yang ⁷ mengikat data dan mempunyai injeksi ketergantungan (*Dependency Injection*) yang ber- ⁸ fungsi agar suatu kelas tidak terikat dengan kelas lain, juga akan menghilangkan banyak ⁹ kode yang seharusnya dituliskan oleh *programmer*, dan semua itu terjadi pada *browser* se- ¹⁰ hingga dapat disimpulkan bahwa AngularJS merupakan pasangan yang sangat ideal bagi

¹ penggunaan teknologi server. Dalam pembuatannya, ketidakcocokkan halaman statik dan
² dinamik biasanya diselesaikan dengan pendekatan sebagai berikut:

- ³ 1. *Library*: merupakan sebuah koleksi dari berbagai macam fungsi yang berguna dalam pembuatan aplikasi *web*, contoh: JQuery.
- ⁵ 2. *Frameworks*: merupakan suatu implementasi dari sebuah aplikasi *web* yang menempatkan kode yang dituliskan secara detail. *Framework* akan berperan melakukan pemanggilan ke kode yang dituliskan *programmer* ketika aplikasi membutuhkan sesuatu yang spesifik, contoh: durandal(*light-weight javascript framework*), ember(*open-source javascript framework*), dll.

¹⁰ Dalam pembentukannya, AngularJS memiliki pendekatan yang berbeda. AngularJS
¹¹ mengajarkan *browser* sintaks baru yang disebut *directives*. Contoh *directives* adalah:

- ¹² 1. Keterikatan data di dalam {{}};
- ¹³ 2. Dukungan untuk *Form* dan *Form Validation*
- ¹⁴ 3. Pengelompokan HTML menjadi komponen - komponen yang dapat dipakai kembali.

¹⁵ 2.2.1 Gambaran Konseptual

¹⁶ Berikut ini adalah beberapa bagian-bagian terpenting dalam AngularJS.

Tabel 2.2: Tabel Gambaran Konseptual

Konsep	Deskripsi
Template	HTML dengan tambahan <i>markup</i>
Directives	Pengembangan HTML dengan atribut dan elemen yang dibuat khusus
Model	Data yang ditunjukan 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>
Filter	Mengatur nilai dari sebuah <i>expression</i> untuk ditunjukkan kepada pengguna
View	Apa yang akan dilihat oleh pengguna
Data Binding	Menyalaskan 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 dikonfigurasikan <i>injector</i>
Services	Logika bisnis independen dari <i>views</i> yang bisa dipakai kembali

¹⁷ 2.2.2 Directives

¹⁸ *Directives* merupakan penanda pada *DOM (Document Object Model) elements* (seperti attribut, nama, *comment*, dan kelas CSS) yang memberitahukan kepada *AngularJS HTML*

¹ compiler¹, sehingga dapat melampirkan perilaku yang diinginkan kepada *DOM element* (contohnya memakai *event listener*), atau bahkan mengubah *DOM element* yang dituju beserta dengan peranakannya.

⁴ AngularJS menyediakan sekumpulan *directives built-in* seperti yang akan dijelaskan
⁵ selanjutnya.

⁶ ng-Model

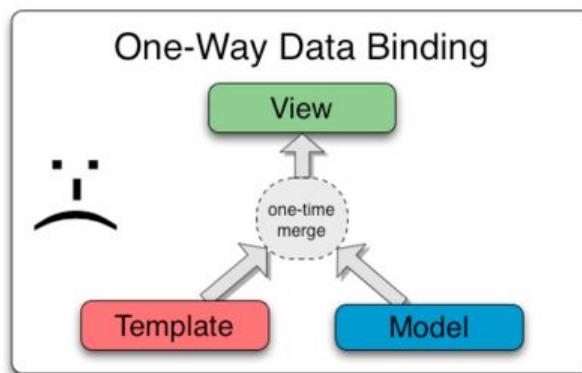
⁷ ng-Model adalah suatu *directives* yang dapat digunakan pada *form control* seperti *input*, *se-
lect*, *textarea*, dan *form control* unik lainnya untuk mengikat data ke *scope property* (`$$scope`)
⁹ dengan menggunakan *ngModelController* (kelas yang menyediakan API (*Application Pro-
gramming Interface*) untuk ng-Model)

¹¹ ng-Controller

¹² ng-Controller adalah suatu *directives* yang menambahkan suatu kelas *controller* pada ba-
¹³ gian *view* dari aplikasi angularJS. ng-Controller inilah yang merupakan kelas kunci dari
¹⁴ bagaimana AngularJS dapat menerapkan Model-View-Controller bersifat front-end (pada
¹⁵ tampilan).

¹⁶ 2.2.3 Data Binding

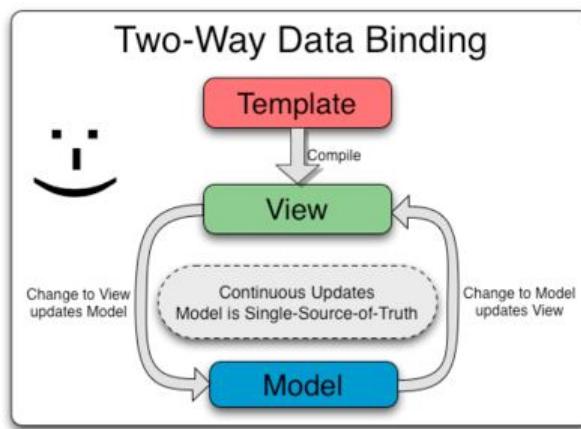
¹⁷ *Data Binding* pada AngularJS merupakan penyelarasan data antara *model* dan komponen-
¹⁸ komponen *view*. Ketika *model* berubah, maka *view* pun akan berubah, begitu juga dengan
¹⁹ sebaliknya.



Gambar 2.2: Data Binding Classical Templates System

²⁰ Pada Gambar 2.2 menjelaskan bahwa kebanyakan *data binding* adalah proses satu arah.
²¹ Hal itu dilakukan dengan menyatukan *template* dan *model* menjadi *view*. Setelah penyatu-
²² an, pergantian pada *model* tidak secara otomatis mengganti *view* yang sudah ditampilkan.

¹memperbolehkan pengembang untuk mengajarkan browser syntaks HTML baru



Gambar 2.3: Data Binding pada Angular

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

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

8 2.2.4 Model-View-Controller(MVC)

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

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

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

Listing 2.11: Model berupa variabel

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

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

Listing 2.12: Menampilkan model

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

23 *Syntax view* 2 kurung kurawal diatas disebut sebagai interlopasi (penyusupan), karena hal
 24 tersebut memasukkan konten baru ke dalam *template* yang sudah ada.
 25 Sementara kelas *controllers* berguna untuk memberitahu AngularJS tentang objek atau

1 konten primitif mana dari *model* yang akan dipakai dengan cara menetapkannya ke objek
 2 '\$scope', objek '\$scope' tersebut kemudian akan diberikan kepada *controller* seperti contoh
 3 berikut:

Listing 2.13: Controller

```
4 | function TextController($scope){  

5 |   $scope.someText = someText;  

6 | }
```

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

Listing 2.14: Penyatuan model view controller

```
8 | <html>  

9 | <body ng-controller = "TextController">  

10 |   <p>{{someText}}</p>  

11 |  

12 |   <script>  

13 |     src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.4/angular.min.js">  

14 |   </script>  

15 |  

16 |   <script>  

17 |     function TextController($scope){  

18 |       $scope.someText = 'You have started your journey.';  

19 |     }  

20 |   </script>  

21 | </body>  

22 | </html>
```

23 Hasil dari Listing 2.14 diatas adalah tulisan "You have started your journey". Walaupun
 24 cara ini dapat dilakukan dengan mudah pada aplikasi sederhana seperti contoh diatas, tetapi
 25 untuk kebanyakan aplikasi sebaiknya dibuat objek *model* untuk menyimpan *data* yang ada.
 26 Untuk itu, daripada membuat *model* seperti Listing 2.13, akan lebih baik menggunakan
 27 kode:

Listing 2.15: Penyederhanaan controller

```
28 | var message= {};  

29 | message.someText = 'You have started your journey';  

30 | function TextController($scope){  

31 |   $scope.message = message;  

32 | }
```

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

Listing 2.16: Pemanggilan variabel

```
34 | <p>{{message.someText}}</p>
```

35 Perubahan yang dilakukan diatas berfungsi untuk mencegah perilaku tidak terduga yang
 36 dapat terjadi dari *prototypal inheritance* dalam objek '\$scope'. Walaupun untuk sementara
 37 hal ini dapat berjalan dengan baik, tetapi cara yang benar dalam mendefinisikan sebuah
 38 *controller* adalah dengan menggunakan sebuah kelas yang dinamakan *module* yang menyediakan
 39 *namespace* untuk bagian lain dari aplikasi berhubungan. Perubahan tersebut akan
 40 mengubah kode-kode pada listing diatas menjadi:

Listing 2.17: Penyederhanaan MVC

```
41 | <html ng-app='myApp'>  

42 | <body ng-controller='TextController'>  

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

44 |   <script>  

45 |     src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.4/angular.min.js">  

46 |   </script>  

47 |  

48 |   <script>  

49 |     var myAppModule = angular.module('myApp', []);  

50 |  

51 |     myAppModule.controller('TextController',  

52 |       function($scope){
```

```

1 |         var someText = {};
2 |         someText.message = 'You have started your journey';
3 |         $scope.someText = someText;
4 |     });
5 | </script>
6 | </body>
7 | </html>

```

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

12 2.3 Twitter Bootstrap

13 *Twitter Bootstrap* [4] atau yang lebih dikenal dengan *Bootstrap* adalah *framework* HTML,
 14 CSS, dan JS terpopuler dalam hal pengembangan tampilan yang responsif *mobile* pertama
 15 dalam hal aplikasi berbasis *web*.

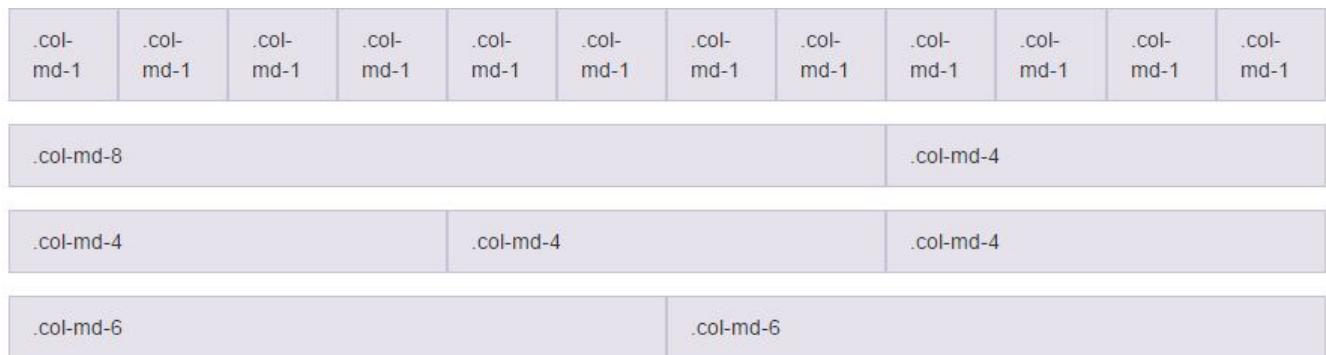
16 2.3.1 Grid System

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

- 22 1. *Rows* harus ditempatkan diantara *.container(fixed-width)* atau *.container-fluid (full-width)* untuk mendapatkan keselarasan ukuran
- 24 2. *Rows* dipergunakan untuk membuat grup kolom secara *horizontal*.
- 25 3. Konten tampilan harus berada diantara kelas *columns* atau peranakan dari kelas *columns*.
- 27 4. Kelas-kelas yang telah ditetapkan seperti ".row" dan ".col-xs-4" dapat digunakan dengan segera untuk membentuk *layout*.
- 29 5. Kelas *columns* membuat *gutters* (jarak antara kolumn konten) menggunakan kelas *padding*.
- 31 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".
- 34 7. Jika ada lebih dari 12 kolom dalam 1 baris, maka kolom yang lebih tersebut akan dipindahkan ke baris baru sebagai satu kesatuan.
- 36 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.

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

Gambar 2.4: Grid Option pada Bootstrap



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 2.18: Contoh penggunaan Twitter Bootstrap

```

7 <form>
8   <div class="form-group">
9     <label for="exampleInputEmail1">Email Address</label>
10    <input type="email" class="form-control" id="exampleInputEmail1" placeholder="Email">
11  </div>
12  <div>
13    <label for="exampleInputPassword1">Password</label>
14    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
15  </div>
16  <div class="form-group">
17    <label for="exampleInputFile">File</label>
18    <input type="file" id="exampleInputFile">
19    <p class="help-block"> Example block-level help text here.</p>
20  </div>
21  <div class="checkbox">
22    <label>
23      ...

```

```
1 |         <input type="checkbox"> Check me out
2 |     </label>
3 | </div>
4 |     <button type="submit" class="btn bt-default">Submit</button>
5 | </form>
```

EXAMPLE**Email address****Password****File input** Choose File No file chosen

Example block-level help text here.

 Check me out

Gambar 2.6: Contoh Hasil Penggunaan Kelas Form

¹

BAB 3

²

ANALISIS

³ Pada bab ini akan dibahas mengenai analisis sistem kini, analisis sistem usulan, dan use
⁴ case diagram.

⁵ 3.1 Analisis Sistem Kini

⁶ Skripsi 2 merupakan salah satu mata kuliah dengan syarat wajib lulus pada Program Studi
⁷ Teknik Informatika Universitas Katolik Parahyangan, Bandung. Untuk mendapatkan hak
⁸ sidang, mahasiswa wajib menempuh persyaratan-persyaratan yang sudah ditentukan pada
⁹ kelas matakuliah Skripsi 2, yaitu mahasiswa harus melakukan proses bimbingan (disarankan
¹⁰ 4 kali bimbingan), mahasiswa harus mengikuti seminar internal 3 kali, dan mahasiswa wajib
¹¹ menyelesaikan dokumen serta program skripsi yang dikerjakan sebelum tanggal pengum-
¹² pulan yang telah ditentukan. Jika mahasiswa tidak menyelesaikan persyaratan-persyaratan
¹³ tersebut, maka mahasiswa yang bersangkutan tidak mendapatkan hak untuk menempuh
¹⁴ sidang Skripsi 2.

¹⁵ Sistem penilaian yang dipakai saat ini pada sidang Skripsi 2 adalah sistem pencatatan
¹⁶ manual pada formulir penilaian. Formulir penilaian yang digunakan untuk sistem kini
¹⁷ terbagi menjadi 2 bagian, yaitu formulir rekapitulasi penilaian dan formulir berita acara
¹⁸ sidang skripsi. Berikut ini adalah penjelasan penggunaan kertas formulir tersebut.

¹⁹ 3.1.1 Form Rekapitulasi Penilaian

²⁰ Formulir rekapitulasi dibagi menjadi 3 bagian (Gambar A.2), yaitu:

- ²¹ 1. Lembar Rekapitulasi Penilaian Pembimbing
- ²² 2. Lembar Rekapitulasi Penilaian Ketua Tim Penguji
- ²³ 3. Lembar Rekapitulasi Penilaian Anggota Tim Penguji

²⁴ Lembaran tersebut akan diberikan kepada pembimbing, ketua tim penguji, dan anggo-
²⁵ ta tim penguji sesuai dengan keperluannya. Setelah dibagikan, penilai wajib mengisi kol-
²⁶ lom nilai yang ingin diberikan kepada mahasiswa yang bersangkutan sesuai dengan kolom
²⁷ komponen penilaian yang ada. Setelah memberikan nilai, maka penilai harus melakuk-
²⁸ an perkalian antara kolom nilai dengan kolom bobot yang akan menghasilkan kolom nilai
²⁹ akhir mahasiswa. Terakhir penilai akan menjumlahkan seluruh kolom nilai akhir yang akan
³⁰ menghasilkan total nilai akhir dari mahasiswa tersebut.

¹ Setelah semua kolom terisi, maka lembaran rekapitulasi tersebut akan dikumpulkan ke
² ketua tim penguji. Kemudian data yang telah tersedia akan disalin oleh ketua tim penguji
³ kepada formulir berita acara sidang skripsi.

⁴ 3.1.2 Form Berita Acara Sidang Skripsi

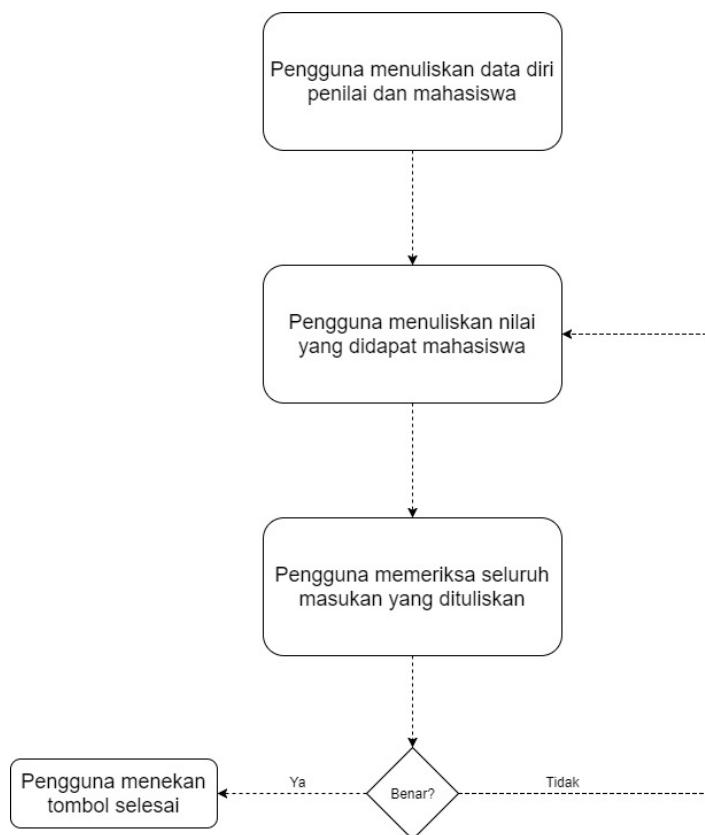
⁵ Formulir berita acara sidang skripsi (Gambar A.1) merupakan formulir yang mencakup
⁶ pengisian waktu sidang bersangkutan, data diri mahasiswa, nama dosen penguji dan pem-
⁷ bimbing, nilai akhir dari masing-masing penilai, dan nilai akhir yang diterima mahasiswa.
⁸ Seperti yang telah dibahas pada subbab sebelumnya (3.1.1) formulir berita acara sidang sk-
⁹ rripsi akan diisi oleh ketua tim penguji setelah seluruh form rekapitulasi dari masing masing
¹⁰ penilai di kumpulkan kembali kepada ketua tim penguji.

¹¹ Setelah ketua tim penguji melakukan pengisian pada masing-masing kolom dari masing-
¹² masing penilai yang bersangkutan, maka ketua tim penguji akan melakukan perkalian nilai
¹³ tersebut dengan bobot masing-masing penilai yang akan menghasilkan nilai akhir maha-
¹⁴ siswa dari masing-masing penilai. Kemudian akan dihasilkan 90% nilai akhir mahasiswa
¹⁵ untuk diberitahukan kepada mahasiswa dan diberikan kepada koordinator skripsi untuk
¹⁶ melengkapi 10% dari nilai mahasiswa berdasarkan nilai kedisiplinan. Hasil dari seluruh
¹⁷ proses tersebut adalah nilai akhir sidang skripsi 2 mahasiswa bersangkutan.

¹⁸ 3.2 Analisis Sistem Usulan

¹⁹ Masalah yang dihadapi oleh sistem kini adalah tidak adanya otomatisasi dalam pengisian
²⁰ formulir penilaian. Tidak adanya otomatisasi pada sistem kini mengakibatkan kemungkinan
²¹ dapat terjadinya *human error* (kesalahan manusia yang tidak disengaja), sehingga data yang
²² dihasilkan tidak valid.

²³ Untuk menyelesaikan masalah tersebut, penulis memakai AngularJS sebagai solusi da-
²⁴ lam otomatisasi formulir penilaian. Dengan memakai AngularJS, maka formulir penilaian
²⁵ sistem kini yang berupa kertas dapat digantikan dengan sistem penilaian berupa aplikasi
²⁶ berbasis *web*. Gambar 3.1 adalah *flowchart* dari aplikasi berbasis *web* yang akan dibangun.



Gambar 3.1: Flowchart sistem usulan

1 Sistem usulan yang akan dibuat adalah sebuah sistem berbasis *web* menggunakan bahasa
 2 pemrograman PHP yang dapat melakukan otomatisasi perhitungan nilai akhir mahasiswa.
 3 Otomatisasi tersebut dilakukan dengan mengintegrasikan bahasa pemrograman AngularJS
 4 ke dalam bahasa PHP sehingga menghasilkan sistem penilaian yang bersifat Single Page
 5 Application. Sedangkan tampilan pada sistem usulan akan dibuat semirip mungkin dengan
 6 tampilan yang ada pada sistem kini, sehingga pengguna mudah untuk mempelajari sistem
 7 usulan yang dibuat.

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

10 3.2.1 Analisis Back End

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

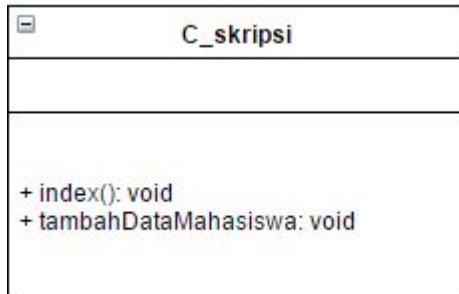
15 Model

16 Pada bagian ini akan dijelaskan tentang penggunaan *model* pada *codeigniter*. *Model* mem-
 17 punyai fungsi untuk membuat sambungan dari aplikasi ke basis data. Pada penelitian ini
 18 hanya memiliki 1 kelas diberi nama "Skripsi_model" dengan format *extends CI_Model* yang
 19 berfungsi untuk mengaktifkan fitur dari model milik *codeigniter*. Kelas model sendiri memi-
 20 liki sebuah method yang dinamakan "InsertDataMahasiswa" dengan parameter \$tablename

untuk masukan nama tabel dan \$data untuk masukan data yang diperoleh dari *controller*. Method tersebut berfungsi untuk melakukan fungsi *insert* ke dalam basis data sistem usulan.

Controller

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 dipersingkat dan dipermudah dalam pembuatannya.



Gambar 3.2: Gambar diagram kelas *file controllers*

Berikut adalah *method-method* yang dimiliki oleh kelas *controller* Gambar 3.2:

- index()

Berfungsi untuk memilih *file view* yang akan dipakai pada sistem usulan.

- 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.

Helper

Helper merupakan salah satu fungsi dari codeigniter yang berguna untuk mempermudah pembuatan sistem usulan. Sistem usulan menggunakan dua buah helper yaitu *helper url* agar sistem dapat memproses kode berupa *url* untuk merujuk suatu halaman yang dituju, serta *helper file* agar sistem dapat mengaktifkan fungsi-fungsi *path* untuk merujuk *file* yang dituju.

3.2.2 Analisis Front End

Analisis tahap *front-end* merupakan analisis pada bagian yang akan ditampilkan kepada pengguna. Pada sistem usulan, analisis *front-end* akan berfokus pada penggunaan AngularJS adalah otomatisasi perhitungan nilai akhir mahasiswa.

Tahap analisis *front-end* adalah tahap seleksi masukan yang akan ditampilkan dan diisi oleh pengguna sistem. Diskusi dengan dosen pembimbing menghasilkan 28 buah masukan pada Tabel 3.1 yang akan ditampilkan pada tampilan sistem usulan.

Tabel 3.1: Tabel masukan pada tampilan

No	Masukan
1	Tahun Sidang
2	Semester Sidang
3	NPM
4	Nama Mahasiswa
5	Judul Skripsi
6	Nama Pembimbing
7	Nama Pembimbing Pendamping
8	Nama Ketua Tim Penguji
9	Nama Anggota Tim Penguji
10	Nilai Ketua Tim Penguji
11	Nilai Anggota Tim Penguji
12	Nilai Pembimbing
13	Nilai Koordinator Skripsi
14	Nilai Koordinator Skripsi
15	Nilai Tata Tulis Laporan Anggota
16	Nilai Kelengkapan Materi Anggota
17	Nilai Penguasaan Materi Anggota
18	Nilai Presentasi Anggota
19	Nilai Pencapaian Tujuan Anggota
20	Nilai Tata Tulis Laporan Ketua
21	Nilai Kelengkapan Materi Ketua
22	Nilai Penguasaan Materi Ketua
23	Nilai Presentasi Ketua
24	Nilai Pencapaian Tujuan Ketua
25	Nilai Tata Tulis Laporan Pembimbing
26	Nilai Kelengkapan Materi Pembimbing
27	Nilai Penguasaan Materi Pembimbing
28	Proses Bimbingan

1 Berdasarkan masukan pada Tabel 3.1, dibuatlah *Model*, *View*, *Controller* bagian *front-end*
 2 yang akan dipakai oleh AngularJS sebagai cara otomatisasi sistem. Masukan-masukan
 3 yang dibuat akan diberikan variabel ng-model sebagai tempat penyimpanan masukan untuk
 4 diberikan pada *view* dari AngularJS sehingga otomatisasi dapat terjadi.

5 *View* pada AngularJS akan memanggil masukan pada *model* dan melakukan otomatisasi,
 6 baik perhitungan atau pengisian otomatis. Sementara *controller* dibuat pada awal kode
 7 sistem, agar *view* dan *model* sistem dapat terhubung satu dengan yang lain.

8 Berikut ini adalah nama-nama dari *model*, *view*, dan *controller* AngularJS yang dipakai
 9 untuk otomatisasi pada Sistem Penilaian Sidang Skripsi 2.

10 • *Controller*: "DefaultValue" = Controller yang dipakai di seluruh sistem

11 • *Model*:

12

Tabel 3.2: Tabel model

Nama Model	Tujuan
tahun	Masukan Tahun
n_npm	Masukan NPM
namaPembimbing	Masukan nama pembimbing
namaKetuaTimPenguji	Masukan nama ketua tim penguji
namaAnggotaTimPenguji	Masukan nama anggota tim penguji
nilai_koordinator	Masukan nilai Koordinator skripsi
koordinator.value	Bobot nilai koordinator skripsi
nilai_TTLaporanK	Masukan nilai tata tulis laporan milik ketua tim penguji
TTLaporanK.value	Bobot nilai tata tulis laporan milik ketua tim penguji
nilai_KMateriK	Masukan nilai kelengkapan materi milik ketua tim penguji
KMateriK.value	Bobot nilai kelengkapan materi milik ketua tim penguji
nilai_PMateriK	Masukan nilai penguasaan materi milik ketua tim penguji
PMateriK.value	Bobot nilai penguasaan materi milik ketua tim penguji
nilai_presentasiK	Masukan nilai presentasi ketua tim penguji
presentasiK.value	Bobot nilai presentasi ketua tim penguji
nilai_PTujuanK	Masukan nilai pencapaian tujuan milik ketua tim penguji
PTujuanK.value	Bobot nilai pencapaian tujuan milik ketua tim penguji
nilai_TTLaporanA	Masukan nilai tata tulis laporan milik anggota tim penguji
TTLaporanA.value	Bobot nilai tata tulis laporan milik anggota tim penguji
nilai_KMateriA	Masukan nilai kelengkapan materi milik anggota tim penguji
KMateriA.value	Bobot nilai kelengkapan materi milik anggota tim penguji
nilai_PMateriA	Masukan nilai penguasaan materi milik anggota tim penguji
PMateriA.value	Bobot nilai penguasaan materi milik anggota tim penguji
nilai_presentasiA	Masukan nilai presentasi milik anggota tim penguji
presentasiA.value	Bobot nilai presentasi milik anggota tim penguji
nilai_PTujuanA	Masukan nilai pencapaian tujuan milik anggota tim penguji
PTujuanA.value	Bobot nilai pencapaian tujuan milik anggota tim penguji
nilai_TTLaporanP	Masukan nilai tata tulis laporan milik pembimbing
TTLaporanP.value	Bobot nilai tata tulis laporan milik pembimbing
nilai_KMateriP	Masukan nilai kelengkapan materi milik pembimbing
KMateriP.value	Bobot nilai kelengkapan materi milik pembimbing
nilai_PMateriP	Masukan nilai penguasaan materi milik pembimbing
PMateriP.value	Bobot nilai penguasaan materi milik pembimbing
nilai_PBimbinganP	Masukan nilai proses bimbingan milik pembimbing
PBimbinganP.value	Bobot ilai proses bimbingan milik pembimbing

1 • View:

Tabel 3.3: Tabel view

Otomatisasi view	Keterangan
Tahun+1	mengambil masukan model "tahun" ditambah 1
NPM	mengambil masukan model "npm" dan menampilkan isinya di lembaran rekapitulasi
Nilai ketua tim penguji	mengambil total nilai akhir dari lembar rekapitulasi ketua tim penguji
Nilai anggota tim penguji	mengambil total nilai akhir dari lembar rekapitulasi anggota tim penguji
Nilai pembimbing	mengambil total nilai akhir dari lembar rekapitulasi pembimbing
Nilai akhir ketua tim penguji	mengambil nilai ketua tim penguji dan dikalikan dengan bobot nilai ketua tim penguji
Nilai akhir anggota tim penguji	mengambil nilai anggota tim penguji dan dikalikan dengan bobot anggota ketua tim penguji
Nilai akhir pembimbing	mengambil nilai pembimbing dan dikalikan dengan bobot nilai pembimbing
Nilai akhir koordinator skripsi	mengambil nilai koordinator skripsi dan dikalikan dengan bobot nilai koordinator skripsi
Nilai akhir tata tulis laporan ketua tim penguji	mengambil nilai tata tulis laporan ketua tim penguji dan dikalikan dengan bobot nilai tata tulis laporan ketua tim penguji
Nilai akhir kelengkapan materi ketua tim penguji	mengambil nilai kelengkapan materi ketua tim penguji dan dikalikan dengan bobot nilai kelengkapan materi ketua tim penguji
Nilai akhir penguasaan materi ketua tim penguji	mengambil nilai penguasaan materi ketua tim penguji dan dikalikan dengan bobot nilai penguasaan materi ketua tim penguji
Nilai akhir presentasi ketua tim penguji	mengambil nilai presentasi ketua tim penguji dan dikalikan dengan bobot nilai presentasi ketua tim penguji
Nilai akhir pencapaian tujuan ketua tim penguji	mengambil nilai pencapaian tujuan ketua tim penguji dan dikalikan dengan bobot nilai pencapaian tujuan ketua tim penguji
Total nilai akhir ketua tim penguji	menjumlahkan seluruh nilai akhir ketua tim penguji pada lembar rekapitulasi ketua tim penguji
Nilai akhir tata tulis laporan anggota tim penguji	mengambil nilai tata tulis laporan anggota tim penguji dan dikalikan dengan bobot nilai tata tulis laporan anggota tim penguji
Nilai akhir kelengkapan materi anggota tim penguji	mengambil nilai kelengkapan materi anggota tim penguji dan dikalikan dengan bobot nilai kelengkapan materi anggota tim penguji
Nilai akhir penguasaan materi anggota tim penguji	mengambil nilai penguasaan materi anggota tim penguji dan dikalikan dengan bobot nilai penguasaan materi anggota tim penguji
Nilai akhir presentasi anggota tim penguji	mengambil nilai presentasi anggota tim penguji dan dikalikan dengan bobot nilai presentasi anggota tim penguji
Nilai akhir pencapaian tujuan anggota tim penguji	mengambil nilai pencapaian tujuan anggota tim penguji dan dikalikan dengan bobot nilai pencapaian tujuan anggota tim penguji
Total nilai akhir anggota tim penguji	menjumlahkan seluruh nilai akhir anggota tim penguji pada lembar rekapitulasi anggota tim penguji
Nilai akhir tata tulis laporan pembimbing	mengambil nilai tata tulis laporan pembimbing dan dikalikan dengan bobot nilai tata tulis laporan pembimbing
Nilai akhir kelengkapan materi pembimbing	mengambil nilai kelengkapan materi pembimbing dan dikalikan dengan bobot nilai kelengkapan materi pembimbing

Nilai akhir penguasaan materi pembimbing	mengambil nilai penguasaan materi pembimbing dan dikalikan dengan bobot nilai penguasaan materi pembimbing
Nilai akhir proses bimbingan pembimbing	mengambil nilai proses bimbingan pembimbing dan dikalikan dengan bobot nilai proses bimbingan pembimbing
Total nilai akhir pembimbing	menjumlahkan seluruh nilai akhir pembimbing pada lembar rekapitulasi pembimbing
Persetujuan ketua tim penguji	menampilkan model namaKetuaTimPenguji
Persetujuan anggota tim penguji	menampilkan model namaAnggotaTimPenguji
Persetujuan pembimbing	menampilkan model namaPembimbing

1 3.2.3 Analisis Basis Data

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

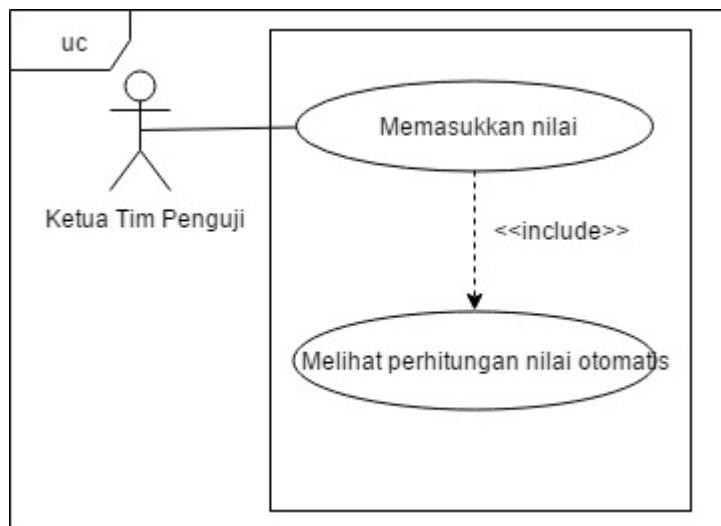
5 Berdasarkan analisa dari contoh form penilaian skripsi yang ada (Gambar A.1 dan
 6 Gambar A.2), dapat disimpulkan bahwa penilaian skripsi membutuhkan data-data sebagai
 7 berikut:

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

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

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

17 3.3 Use Case



Gambar 3.3: Use case diagram

- 18 1. Skenario memasukkan nilai
 19 Deskripsi: Kegiatan memasukkan nilai ke dalam kotak input yang ada.
 20 Aktor: Ketua tim pengaji
 21 Prakondisi: -
 22 Skenario:

- 1 • Pengguna memilih tempat/kolom yang sudah tersedia di tampilan
- 2 • Pengguna memasukkan nilai yang diinginkan pada tempat/kolom yang telah dipilih.

4 2. Skenario melakukan perhitungan otomatis

5 Deskripsi: Melakukan perhitungan secara otomatis pada tampilan

6 Aktor: -

7 Prakondisi: Nilai sudah dimasukkan

8 Skenario:

- 9 • Pengguna mengisi kolom nilai yang sudah disediakan
- 10 • Dengan ng-model, sistem mengambil nilai dari tempat/kolom yang sudah diisi dan melakukan perhitungan
- 11 • Sistem menampilkan hasil perhitungan ke dalam kolom yang disediakan untuk hasil perhitungan.

¹

BAB 4

²

PERANCANGAN

³ Pada bab ini akan dijelaskan mengenai perancangan aplikasi yang dibangun meliputi per-
⁴ ancangan kelas, *routes*, *controllers*, *models*, perancangan antarmuka.

⁵ 4.1 Routes

⁶ *Routes* merupakan bagian dari *codeigniter* untuk melakukan pemetaan terhadap lokasi *file*
⁷ *controllers* dari aplikasi. Di dalam *routes* terdapat fungsi-fungsi khusus yang diperlukan
⁸ untuk menjalankan aplikasi, seperti melakukan pemanggilan *file controller* yang digunakan
⁹ sebagai controller dasar pada aplikasi.

¹⁰ 4.2 Controllers

¹¹ *Controller* terdiri dari sebuah kelas yang dinamakan "C_Skripsi". Keseluruhan aktivitas
¹² dari sistem informasi penilaian skripsi diatur oleh kelas ini. Controller adalah *file* yang
¹³ mengatur hubungan antara *file model* dan *file view* pada sistem usulan yang akan dibangun.
¹⁴ Fungsi untuk mengubah *data* masukan yang akan diproses oleh *file model* juga diatur di
¹⁵ dalam *file controller* ini.

¹⁶ 4.3 Models

¹⁷ *Models* mempunyai fungsi menghubungkan *views* dan *controllers* pada basis data. Pada
¹⁸ penggunaan *codeigniter*, *model* dibuat dengan sangat sederhana. Pada sistem usulan yang
¹⁹ dibangun, *file model* berfungsi untuk melakukan fungsi *insert data* dari *file view* kepada
²⁰ basis data. Parameter *insert data* yaitu data masukan didapatkan dari fungsi yang terdapat
²¹ pada *file controller*.

²² 4.4 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	float
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	float

4.5 Perancangan Tampilan

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

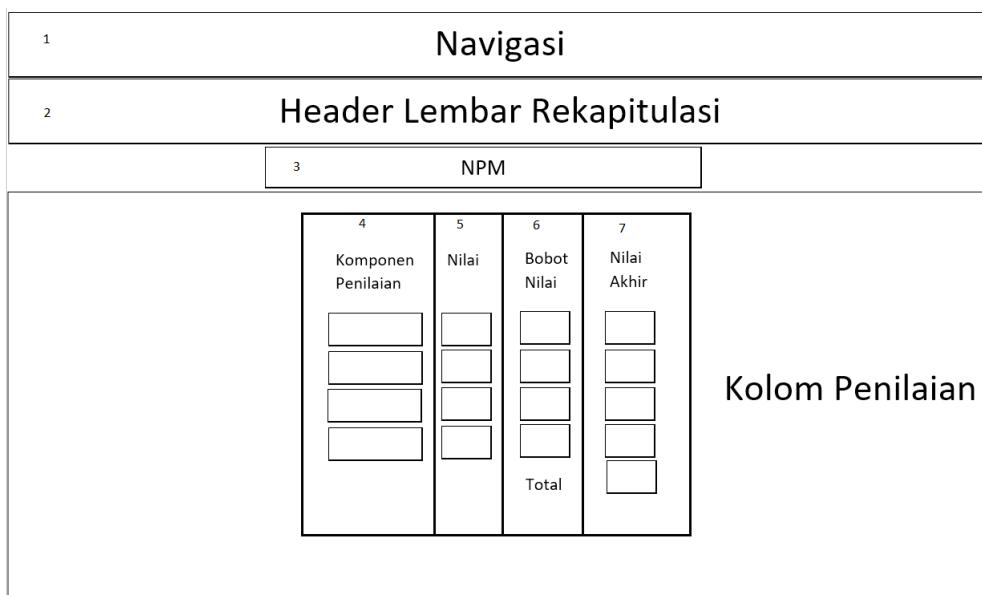
Perbedaan yang akan ditampilkan adalah dengan adanya otomatisasi penghitungan nilai sesuai dengan bobot yang diberikan kepada penilai. Hal ini akan memberikan kemudahan penilai untuk melakukan penilaian.

Gambar 4.1 adalah bayangan awal tampilan untuk sistem informasi penilaian skripsi lembar berita acara sidang skripsi.

1	Navigasi																																		
2	Berita Acara Sidang Skripsi																																		
	Semester	Tahun																																	
NPM Nama Judul Skripsi	3 Data diri mahasiswa																																		
Nama Pembimbing Nama Pembimbing Pendamping Nama Ketua Tim Penguji Nama Anggota Tim Penguji	4 Data diri penilai																																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Kategori Nilai</th> <th style="text-align: center;">Nilai</th> <th style="text-align: center;">Bobot Nilai</th> <th style="text-align: center;">Nilai Akhir</th> </tr> </thead> <tbody> <tr><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td><td><input type="text"/></td></tr> <tr><td>5</td><td>6</td><td>Total</td><td>8</td></tr> </tbody> </table> <p style="text-align: right;">Kolom Penilaian</p>				Kategori Nilai	Nilai	Bobot Nilai	Nilai Akhir	<input type="text"/>	5	6	Total	8																							
Kategori Nilai	Nilai	Bobot Nilai	Nilai Akhir																																
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>																																
5	6	Total	8																																

Gambar 4.1: Perkiraan Tampilan Lembar Berita Acara Sidang Skripsi

- 1 Keterangan Gambar 4.1.
- 2 1. Tempat navigasi untuk berpindah halaman
- 3 2. Judul formulir penilaian
- 4 3. Data diri mahasiswa (nama, NPM, judul skripsi)
- 5 4. Data diri penilai (nama ketua tim penguji, anggota tim penguji, pembimbing, dan pembimbing pendamping)
- 6 5. Penanda kategori penilaian
- 7 6. Kotak masukan nilai per kategori (otomatis)
- 8 7. Bobot nilai yang dimiliki kategori (*default*)
- 9 8. Nilai hasil perhitungan masukan dengan bobot (otomatis)
- 10 11 Gambar 4.2 adalah bayangan awal tampilan untuk sistem informasi penilaian skripsi lembar rekapitulasi ketua tim penguji, anggota tim penguji, dan pembimbing.
- 12 12



Gambar 4.2: Perkiraan Tampilan Lembar Rekapitulasi

- ¹ Keterangan Gambar 4.2.
- ² 1. Tempat navigasi untuk berpindah halaman
- ³ 2. Judul formulir penilaian
- ⁴ 3. NPM mahasiswa (otomatis)
- ⁵ 4. Penanda kategori penilaian
- ⁶ 5. Kotak masukan nilai per kategori (masukan)
- ⁷ 6. Bobot nilai yang dimiliki kategori (*default*)
- ⁸ 7. Nilai hasil perhitungan masukan dengan bobot (otomatis)

BAB 5

IMPLEMENTASI DAN PENGUJIAN

³ Pada bagian ini merupakan rincian atau penjelasan lanjut mengenai lingkungan implementasi perangkat keras maupun perangkat lunak sistem informasi penilaian sidang skripsi
⁴ 2. Bagian terakhir akan membahas tentang pengujian yang telah dilakukan pada sistem
⁵ informasi.
⁶

5.1 Implementasi

⁸ Pada bagian ini akan dijabarkan lingkungan pengembangan sistem informasi dan pengujian.

5.1.1 Lingkungan Implementasi dan Pengujian

¹⁰ Implementasi dilakukan dengan menggunakan sebuah laptop. Berikut adalah spesifikasi
¹¹ laptop yang digunakan:

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

5.1.2 Hasil Implementasi

²⁰ Hasil implementasi dari penelitian ini adalah sebuah sistem informasi berbasis web yang
²¹ menggunakan *codeigniter*, *AngularJS*, dan *Twitter Bootstrap* sebagai dasar pembuatan.
²² Aplikasi dapat diakses melalui jaringan *global* dengan URL <http://sipskripsi.com>. Sis-
²³ tem informasi terdiri dari bagian-bagian sebagai berikut:

- ²⁴ • Bagian formulir berita acara sidang skripsi
²⁵ Bagian ini adalah halaman yang bersangkutan dalam pengisian data diri mahasiswa
²⁶ yang bersangkutan, sekaligus sebagai halaman akhir yang menyimpulkan perhitungan
²⁷ 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.

No	Pembimbing/Penguji	Nilai	Bobot(%)	Nilai Akhir
1	Ketua Tim Penguji	73.6	35	25.7
2	Anggota Tim Penguji	77.1	35	26.9
3	Pembimbing	75.2	20	15.0
4	Koordinator Skripsi	100	10	10
	Total		100	77.7
		7	8	9

Gambar 5.1: Formulir berita acara sidang skripsi 2 terisi

3 Keterangan Gambar 5.1.

- 4 1. Tempat navigasi untuk berpindah halaman
- 5 2. Pengisian semester sidang (Genap/Ganjil), dan tahun sidang (default 2016 / otomatis +1)
- 6 3. NPM mahasiswa (harus tepat 10 angka)
- 7 4. Nama mahasiswa
- 8 5. Judul skripsi
- 9 6. Data diri penilai (nama ketua tim penguji, anggota tim penguji, pembimbing, dan pembimbing pendamping)
- 10 7. Kotak masukan nilai per kategori (otomatis diambil dari rekapitulasi)
- 11 8. Bobot nilai yang dimiliki kategori (*default*)
- 12 9. Nilai hasil perhitungan nilai dengan bobot (otomatis)
- 13 • Bagian formulir rekapitulasi penilaian sidang skripsi 2.
 Bagian ini adalah halaman yang bersangkutan dalam menampung nilai-nilai yang diberikan oleh ketua tim penguji, anggota tim penguji, dan pembimbing pada mahasiswa.

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	70	15	10.5
Kelengkapan Materi	70	10	7
Penggunaan Materi	73	30	21.9
Presentasi	78	15	11.7
Pencapaian Tujuan	75	30	22.5
Total	228	100	73.6

Gambar 5.2: Formulir rekapitulasi ketua tim penguji terisi

1 Keterangan Gambar 5.2.

- 2 1. Tempat navigasi untuk berpindah halaman
- 3 2. NPM mahasiswa (otomatis diambil dari berita acara)
- 4 3. Kotak masukan nilai per kategori (masukan)
- 5 4. Bobot nilai yang dimiliki kategori (*default*)
- 6 5. Nilai hasil perhitungan nilai dengan bobot (otomatis)

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	78	15	11.7
Kelengkapan Materi	78	10	7.8
Penggunaan Materi	79	30	23.7
Presentasi	76	15	11.4
Pencapaian Tujuan	75	30	22.5
Total	228	100	77.1

Gambar 5.3: Formulir rekapitulasi anggota tim penguji terisi

7 Keterangan Gambar 5.3.

- 8 1. Tempat navigasi untuk berpindah halaman
- 9 2. NPM mahasiswa (otomatis diambil dari berita acara)
- 10 3. Kotak masukan nilai per kategori (masukan)
- 11 4. Bobot nilai yang dimiliki kategori (*default*)
- 12 5. Nilai hasil perhitungan nilai dengan bobot (otomatis)

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	70	20	14
Kelengkapan Materi	78	20	15.6
Penguasaan Materi	77	30	23.1
Proses Bimbingan	75	30	22.5
Total	225	100	75.2

Gambar 5.4: Formulir rekapitulasi pembimbing terisi

1 Keterangan Gambar 5.2.

- 2 1. Tempat navigasi untuk berpindah halaman
- 3 2. NPM mahasiswa (otomatis diambil dari berita acara)
- 4 3. Kotak masukan nilai per kategori (masukan)
- 5 4. Bobot nilai yang dimiliki kategori (*default*)
- 6 5. Nilai hasil perhitungan nilai dengan bobot (otomatis)

7 • Bagian selesai.

8 Bagian ini adalah bagian terakhir dari sistem informasi. Ketika formulir sudah selesai
9 diisi, diperlukan persetujuan dengan cara menandai checkbox dari 3 pihak penilai ya-
10 itu ketua tim penguji, anggota tim penguji, dan pembimbing. Setelah selesai, dengan
11 menekan tombol selesai pada bagian ini, akan terdapat sebuah pop up untuk memas-
12 tikan bahwa pengisian sistem telah selesai. Jika "OK" d tekan, maka *data* yang telah
13 terisi akan dimasukkan ke dalam *database*.

Ditetapkan di Bandung, Jum'at, 16 Juni 2017

Dengan persetujuan:

Ketua tim penguji (Chandra Wijaya)

Anggota tim penguji (Mariskha Tri Adithia)

Pembimbing (Pascal Alfadian)

Gambar 5.5: Pengisian Checkbox



Gambar 5.6: Pop Up untuk memastikan pengisian nilai sistem

Tabel 5.1: Tabel Hasil Implementasi

Jenis Data	Input Nilai
id	32
tahun	2017
semester	2
npm	2012730017
nama	Billy Yanuar
judul	Sistem Informasi Penilaian Sidang Skripsi 2 dengan AngularJS
namaPembimbing	Pascal Alfadian
namaPembimbingPendamping	
namaKetuaTimPenguji	Chandra Wijaya
namaAnggotaTimPenguji	Mariskha Tri Adithia
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	77.785

¹ 5.2 Hasil Pengujian

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

5.2.1 Pengujian Fungsional

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 kini yang bekerja secara manual dan dengan sistem usulan menggunakan laptop. Sehubungan dengan sifat kerasahasiaan *data* pengujian, maka dengan persetujuan pembimbing pengujian eksperimental dilakukan dengan merahasiakan identitas mahasiswa yang berhubungan.

Pada saat melakukan pengujian eksperimental, sistem kini 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 kini 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 kini yang berfokus pada perhitungan penilaian mahasiswa. Dengan menggunakan sistem usulan, perhitungan nilai dibuktikan lebih akurat dibandingkan dengan sistem kini. Berikut ini adalah hasil dari pengujian eksperimental.

Tabel 5.2: Tabel Pengujian Eksperimental 1

Jenis Data	Input Nilai
id	26
tahun	2016
semester	1
npm	2012730001
nama	A
judul	Tidak Dicatat
namaPembimbing	Mariskha A
namaPembimbingPendamping	-
namaKetuaTimPenguji	Pascal A
namaAnggotaTimPenguji	Joanna H
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	83

Tabel 5.3: Tabel Pengujian Eksperimental 2

Jenis Data	Input Nilai
id	27
tahun	2016
semester	1
npm	2012730002
nama	B
judul	Tidak Dicatat
namaPembimbing	Luciana A
namaPembimbingPendamping	-
namaKetuaTimPenguji	Cecilian E N
namaAnggotaTimPenguji	Vania
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 5.4: Tabel Pengujian Eksperimental 3

Jenis Data	Input Nilai
id	28
tahun	2016
semester	1
npm	2012730003
nama	C
judul	Tidak Dicatat
namaPembimbing	Anoeng
namaPembimbingPendamping	-
namaKetuaTimPenguji	Bagoes
namaAnggotaTimPenguji	Pascal A
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	76

Tabel 5.5: Tabel Pengujian Eksperimental 4

Jenis Data	Input Nilai
id	29
tahun	2016
semester	1
npm	2012730004
nama	DE
judul	Tidak Dicatat
namaPembimbing	Pascal A
namaPembimbingPendamping	-
namaKetuaTimPenguji	Gede Karya
namaAnggotaTimPenguji	Rosa De Lima
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	78

Tabel 5.6: Tabel Pengujian Eksperimental 5

Jenis Data	Input Nilai
id	30
tahun	2016
semester	1
npm	2012730004
nama	E
judul	Tidak Dicatat
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

1 Tabel 5.7 adalah hasil pengujian sistem usulan pada berbagai macam *platform*, yang
 2 bertujuan untuk mengetahui apakah sistem informasi dapat menjalankan seluruh fungsi-
 3 fungsi yang dimiliki dengan baik ketika dijalankan pada *platform* yang berbeda.

Tabel 5.7: Tabel Pengujian Fungsional

No	Aksi Pengguna	Reaksi yang diharapkan	Keterangan
1	Pengguna menjalankan sistem informasi pada browser Microsoft Edge	Halaman ditampilkan dengan perbedaan tampilan pada tombol pengubah angka serta checklist	Fungsi berjalan sesuai
2	Pengguna menjalankan sistem informasi pada browser Mozilla Firefox	Halaman ditampilkan dengan perbedaan tampilan pada tombol pengubah angka serta checklist	Fungsi berjalan sesuai
3	Pengguna menjalankan sistem informasi pada mobile device berupa smartphone	Halaman ditampilkan dengan perbedaan pada tampilan	Fungsi berjalan sesuai

Gambar 5.7: Tampilan sistem pada platform Microsoft Edge

Berita Acara Sidang Skripsi

Semester: [Ganjil] 2016 [2017]

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

NPM:	Name:
Judul:	dengan pembimbing dan pengaji:
Pembimbing:	
Pembimbing Pendamping:	
Ketua Tim Pengaji:	
Anggota Tim Pengaji:	

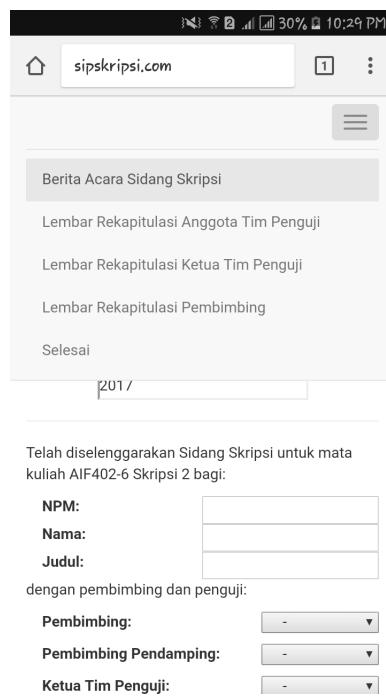
Rekapitulasi nilai Sidang Skripsi 2 yang diberikan oleh pembimbing, pengaji & koordinator skripsi:

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

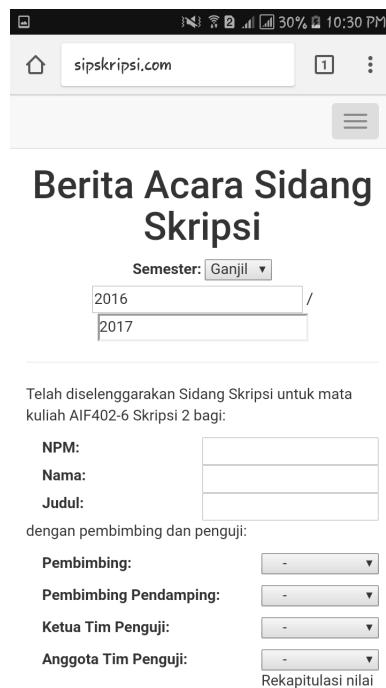
Ditetapkan di Bandung, Minggu, 18 Juni 2017
Dengan persetujuan:
 Ketua tim pengaji ()
 Anggota tim pengaji ()
 Pembimbing ()

[Selesai](#)

Gambar 5.8: Tampilan sistem pada platform Mozilla Firefox



Gambar 5.9: Tampilan menu sistem pada platform browser smartphone



Gambar 5.10: Tampilan lembar berita acara sistem sidang skripsi pada platform browser smartphone

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

Gambar 5.11: Tampilan lembar nilai berita acara sidang skripsi sistem pada platform bro-wser smartphone

Rekapitulasi Penilaian Skripsi 2 (Ketua Tim Penguji)

NPM:

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

Gambar 5.12: Tampilan lembar rekapitulasi ketua tim penguji sistem pada platform browser smartphone

Rekapitulasi Penilaian Skripsi 2 (Anggota Tim Penguji)

NPM:

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

Gambar 5.13: Tampilan lembar rekapitulasi anggota tim penguji sistem pada platform browser smartphone

The screenshot shows a mobile browser interface with a header bar at the top displaying signal strength, battery level (31%), and time (10:30 PM). The main content area has a title 'Rekapitulasi Penilaian Skripsi 2 (Pembimbing)' and a text input field labeled 'NPM:'. Below this is a table with columns 'Komponen Penilaian', 'Nilai', 'Bobot(%)', and 'Nilai Akhir'.

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

Gambar 5.14: Tampilan lembar rekapitulasi pembimbing sistem pada platform browser smartphone

The screenshot shows a mobile browser interface with a header bar at the top displaying signal strength, battery level (31%), and time (10:30 PM). The main content area displays a message: 'Ditetapkan di Bandung, Selasa, 13 Juni 2017' followed by 'Dengan persetujuan:' and three checkboxes: 'Ketua tim penguji', 'Anggota tim penguji', and 'Pembimbing'. A 'Selesai' button is located at the bottom right.

Gambar 5.15: Tampilan lembar selesai pada platform browser smartphone

¹

BAB 6

²

KESIMPULAN DAN SARAN

³

6.1 Kesimpulan

⁴ Berdasarkan hasil penelitian yang dilakukan, didapatkan kesimpulan-kesimpulan sebagai
⁵ berikut:

- ⁶ 1. Penilaian skripsi terutama pada skripsi 2 masih menggunakan sistem manual, yaitu
⁷ penilai mengisi dengan menuliskan nilai dan menghitung nilai akhir dengan alat hitung
⁸ masing-masing pada lembar penilaian yang diberikan.
- ⁹ 2. Proses penyimpanan Sistem Informasi Penilaian Sidang Skripsi 2 dilakukan dengan
¹⁰ menyimpan data diri mahasiswa, nama penilai, nilai bobot, dan nilai akhir yang telah
¹¹ dihitung secara otomatis oleh AngularJS.
- ¹² 3. AngularJS bekerja dengan mengambil nilai input yang diperlukan dan melakukan
¹³ perhitungan tanpa diperlukannya pergantian *page* pada sistem penilaian, sehingga
¹⁴ *single page application* dapat terlaksana dengan maksimal pada sistem penilaian.

¹⁵

6.2 Saran

¹⁶ Berdasarkan pengujian yang dilakukan, berikut adalah beberapa saran untuk pengembangan:

- ¹⁷ • Menambahkan sistem manajemen nilai skripsi untuk melakukan fungsi *select*, *update*,
¹⁸ dan *delete* karena Sistem Penilaian Sidang Skripsi 2 yang dibuat hanya menangani
¹⁹ fungsi *insert* ke *database*.

1

DAFTAR REFERENSI

- 2 [1] Ellis, R. (2014) Codeigniter user guide. http://www.codeigniter.com/user_guide/index.html. 31 Januari 2017.
- 3
- 4 [2] Google (2010) Guide to angularjs documentation. <https://docs.angularjs.org/guide>. 30 Desember 2016.
- 5
- 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.html>. 23 Desember 2016.
- 8

1

LAMPIRAN A

2

FORM PENILAIAN SKRIPSI

- 3 Gambar A.1 adalah lembaran penilaian Skripsi yang di pakai di Program Studi Teknik
4 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%	
	Total		100%	
Ditetapkan di Bandung, <input type="text"/> <input type="text"/> 20				
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Ketua Tim Penguji	Anggota Tim Penguji	Pembimbing**	Koordinator Skripsi	
<small>Petunjuk pengisian : 1. * = coret yang tidak perlu , ** = salah satu pembimbing saja, jika pembimbing utama hadir maka harus pembimbing utama 2. Jika minimal satu penguji dan/atau seluruh pembimbing tidak hadir, maka sidang dibatalkan dan dijadwalkan ulang, tetapi berita acara harus tetap dilaporkan kepada koordinator Skripsi atau kepada pimpinan jurusan.</small>				

Gambar A.1: Form Penilaian Skripsi saat sidang

- ¹ Gambar A.2 adalah lembaran rekapitulasi penilaian Skripsi yang di pakai di Program
² Studi Teknik Informatika Universitas Katolik Parahyangan:

Rekapitulasi Penilaian SKRIPSI 2 (PEMBIMBING)			
NPM Mahasiswa: _____			
Komponen Penilaian	Nilai	Bobot	Nilai Akhir
Tata Tulis Laporan		20%	
Kelengkapan Materi		20%	
Penguasaan Materi		30%	
Proses Bimbingan		30%	
	Total		
		Tgl: ___ / ___ / 20	
		Ttd:	
		Nama:	

Rekapitulasi Penilaian SKRIPSI 2 (KETUA TIM PENGUJI)			
NPM Mahasiswa: _____			
Komponen Penilaian	Nilai	Bobot	Nilai Akhir
Tata Tulis Laporan		15%	
Kelengkapan Materi		10%	
Penguasaan Materi		30%	
Presentasi		15%	
Pencapaian Tujuan		30%	
	Total		
		Tgl: ___ / ___ / 20	
		Ttd:	
		Nama:	

Rekapitulasi Penilaian SKRIPSI 2 (ANGGOTA TIM PENGUJI)			
NPM Mahasiswa: _____			
Komponen Penilaian	Nilai	Bobot	Nilai Akhir
Tata Tulis Laporan		15%	
Kelengkapan Materi		10%	
Penguasaan Materi		30%	
Presentasi		15%	
Pencapaian Tujuan		30%	
	Total		
		Tgl: ___ / ___ / 20	
		Ttd:	
		Nama:	

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         $stahun = $_POST['tahun'];
36         $npm = $_POST['npm'];
37         $nama = $_POST['nama'];
38         $judul = $_POST['judul'];
39         $namaPembimbing = $_POST['namaPembimbing'];
40         $namaPembimbingPendamping = $_POST['namaPembimbingPendamping'];
41         $namaKetuaTimPengaji = $_POST['namaKetuaTimPengaji'];
42         $namaAnggotaTimPengaji = $_POST['namaAnggotaTimPengaji'];
43         $sbobotKetuaTimPengaji = $_POST['sbobotKetuaTimPengaji'];
44         $sbobotAnggotaTimPengaji = $_POST['sbobotAnggotaTimPengaji'];
45         $sbobotPembimbing = $_POST['sbobotPembimbing'];
46         $nilaiKoordinatorSkripsi = $_POST['nilaiKoordinatorSkripsi'];
47         $sbobotKoordinatorSkripsi = $_POST['sbobotKoordinatorSkripsi'];
48         $sbobotTataTulisLaporanAnggota = $_POST['sbobotTataTulisLaporanAnggota'];
49         $sbobotKelengkapanMateriAnggota = $_POST['sbobotKelengkapanMateriAnggota'];
50         $sbobotPenggunaanMateriAnggota = $_POST['sbobotPenggunaanMateriAnggota'];
51         $sbobotPresentasiAnggota = $_POST['sbobotPresentasiAnggota'];
52         $sbobotPencapaianTujuanAnggota = $_POST['sbobotPencapaianTujuanAnggota'];
53         $sbobotTataTulisLaporanKetua = $_POST['sbobotTataTulisLaporanKetua'];
54         $sbobotKelengkapanMateriKetua = $_POST['sbobotKelengkapanMateriKetua'];
55         $sbobotPenggunaanMateriKetua = $_POST['sbobotPenggunaanMateriKetua'];
56         $sbobotPresentasiKetua = $_POST['sbobotPresentasiKetua'];
57         $sbobotPencapaianTujuanKetua = $_POST['sbobotPencapaianTujuanKetua'];
58         $sbobotTataTulisLaporanPembimbing = $_POST['sbobotTataTulisLaporanPembimbing'];
59         $sbobotKelengkapanMateriPembimbing = $_POST['sbobotKelengkapanMateriPembimbing'];
60         $sbobotPenggunaanMateriPembimbing = $_POST['sbobotPenggunaanMateriPembimbing'];
61         $prosesBimbinganPembimbing = $_POST['prosesBimbinganPembimbing'];
62         $nilaiAkhirMahasiswa = $_POST['nilaiAkhirMahasiswa'];
63         $data_insert = array(
64             'semester' => $semester,
65             'tahun' => $stahun,
66             'npm' => $npm,
67             'name' => $nama,
68             'judul' => $judul,
69             'namaPembimbing' => $namaPembimbing,
70             'namaPembimbingPendamping' => $namaPembimbingPendamping,
71             'namaKetuaTimPengaji' => $namaKetuaTimPengaji,
72             'namaAnggotaTimPengaji' => $namaAnggotaTimPengaji,
73             'sbobotKetuaTimPengaji' => $sbobotKetuaTimPengaji,
74             'sbobotAnggotaTimPengaji' => $sbobotAnggotaTimPengaji,
75             'sbobotPembimbing' => $sbobotPembimbing,
76             'nilaiKoordinatorSkripsi' => $nilaiKoordinatorSkripsi,
77             'sbobotKoordinatorSkripsi' => $sbobotKoordinatorSkripsi,
78             'sbobotTataTulisLaporanAnggota' => $sbobotTataTulisLaporanAnggota,
```

```

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' . ?>" ng-controller="DefaultValue
69           " onsubmit="return confirm('Apakah anda yakin ?');">
70
71
72         <!-- Collect the nav links, forms, and other content for toggling -->
73         <div class="collapse navbar-collapse navbar-ex1-collapse">
74           <ul class="nav navbar-nav">
75             <!-- Hidden li included to remove active class from about link when
76                 scrolled up past about section -->
77             <li>
78               <a class="page-scroll" href="#page-top">Berita Acara Sidang
79                   Skripsi </a>
80             </li>
81             <li>
82               <a class="page-scroll" href="#rekKetua">Lembar Rekapitulasi Ketua
83                   Tim Pengudi </a>
84             </li>
85             <li>
86               <a class="page-scroll" href="#rekAnggota">Lembar Rekapitulasi
87                   Anggota Tim Pengudi </a>
88             </li>
89             <li>
90               <a class="page-scroll" href="#rekPembimbing">Lembar Rekapitulasi
91                   Pembimbing </a>
92             </li>
93             <li>
94               <a class="page-scroll" href="#selesai">Selesai </a>
95             </li>
96           </ul>
97         </div>
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" name="tahun"/>
27              /
28              <input id="tahun_1" type="number" max="9999" value="{{tahun+1}}" disabled="disabled" />
29
30
31            </p>
32          </div>
33        </div>
34      </div>
35    </div>
36
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                minlength="10" id="npm" class="inline-form col-md-3 col-xs-6"
50                ng-model="n_npm" name="npm" required />
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        </div>
66
67        <p> dengan pembimbing dan penguji:</p>
68
69
70        <div id="pengenalPembimbing">
71          <p>
72            <label class="col-md-3 col-xs-8" for="pembimbing">Pembimbing:</label>
73            <!-- 6 -->
74            <select class="col-md-9 col-xs-4" id="pembimbing" name="namaPembimbing" ng-model="namaPembimbing">
75              <option value="Cecilia Esti Nugraheni">Cecilia Esti Nugraheni</option>
76              <option value="Veronica Sri Moertini">Veronica Sri Moertini</option>
77              <option value="Rosa De Lima">Rosa De Lima</option>
78              <option value="Thomas Anung Basuki">Thomas Anung Basuki</option>
79              <option value="Lionov">Lionov</option>
80              <option value="Mariskha Tri Adithia">Mariskha Tri Adithia</option>
81              <option value="Luciana Abednego">Luciana Abednego</option>
82              <option value="Elisati Hulu">Elisati Hulu</option>
83              <option value="Chandra Wijaya">Chandra Wijaya</option>
84              <option value="Gede Karya">Gede Karya</option>
85              <option value="Nico Saputro">Nico Saputro</option>
86              <option value="Joanna Helga">Joanna Helga</option>
87              <option value="Pascal Alfadian">Pascal Alfadian</option>
88              <option value="Husnul Hakim">Husnul Hakim</option>
89              <option value="Vania Natali">Vania Natali</option>
90              <option value="Aditya Bagoes Saputra">Aditya Bagoes Saputra</option>
91              <option value="Claudio Fransiscus">Claudio Fransiscus</option>
92              <option value="Natalia">Natalia</option>
93            </select>
94
95          </p>
96
97        </div>
98        <br/>
99
100
101
102
103

```

```

1 <div id="pengenalPembimbingPendamping">
2   <p>
3     <label class="col-md-3 col-xs-8" for="pembimbing2">Pembimbing
4       Pendamping:</label>
5     <!-- 7 -->
6     <select class="col-md-9 col-xs-4" id="pembimbing2" name="
7       namaPembimbingPendamping" ng-model="namaPembimbingPendamping">
8       <option value="Cecilia_Esti_Nugraheni">Cecilia Esti Nugraheni
9         </option>
10      <option value="Veronica_Sri_Moertini">Veronica Sri Moertini</
11        option>
12      <option value="Rosa_De_Lima">Rosa De Lima</option>
13      <option value="Thomas_Anung_Basuki">Thomas Anung Basuki</
14        option>
15      <option value="Lionov">Lionov</option>
16      <option value="Mariskha_Tri_Adithia">Mariskha Tri Adithia</
17        option>
18      <option value="Luciana_Abednego">Luciana Abednego</option>
19      <option value="Elisati_Hulu">Elisati Hulu</option>
20      <option value="Chandra_Wijaya">Chandra Wijaya</option>
21      <option value="Gede_Karya">Gede Karya</option>
22      <option value="Nico_Saputro">Nico Saputro</option>
23      <option value="Joanna_Helga">Joanna Helga</option>
24      <option value="Pascal_Alfdian">Pascal Alfdian</option>
25      <option value="Husnul_Hakim">Husnul Hakim</option>
26      <option value="Vania_Natali">Vania Natali</option>
27      <option value="Aditya_Bagoes_Saputra">Aditya Bagoes Saputra</
28        option>
29      <option value="Claudio_Franciscus">Claudio Franciscus</option>
30      <option value="Natalia">Natalia</option>
31    </select>
32  </p>
33 </div>
34 <br/>
35 <div id="pengenalKetua">
36   <p>
37     <label class="col-md-3 col-xs-8" for="ketua">Ketua Tim Penguji:</label
38       >
39     <!-- 8 -->
40     <select class="col-md-9 col-xs-4" id="ketua" name="
41       namaKetuaTimPenguji" ng-model="namaKetuaTimPenguji">
42       <option value="Cecilia_Esti_Nugraheni">Cecilia Esti Nugraheni
43         </option>
44       <option value="Veronica_Sri_Moertini">Veronica Sri Moertini</
45         option>
46       <option value="Rosa_De_Lima">Rosa De Lima</option>
47       <option value="Thomas_Anung_Basuki">Thomas Anung Basuki</
48         option>
49       <option value="Lionov">Lionov</option>
50       <option value="Mariskha_Tri_Adithia">Mariskha Tri Adithia</
51         option>
52       <option value="Luciana_Abednego">Luciana Abednego</option>
53       <option value="Elisati_Hulu">Elisati Hulu</option>
54       <option value="Chandra_Wijaya">Chandra Wijaya</option>
55       <option value="Gede_Karya">Gede Karya</option>
56       <option value="Nico_Saputro">Nico Saputro</option>
57       <option value="Joanna_Helga">Joanna Helga</option>
58       <option value="Pascal_Alfdian">Pascal Alfdian</option>
59       <option value="Husnul_Hakim">Husnul Hakim</option>
60       <option value="Vania_Natali">Vania Natali</option>
61       <option value="Aditya_Bagoes_Saputra">Aditya Bagoes Saputra</
62         option>
63       <option value="Claudio_Franciscus">Claudio Franciscus</option>
64       <option value="Natalia">Natalia</option>
65     </select>
66   </p>
67 </div>
68 <br/>
69 <div id="pengenalAnggota">
70   <p>
71     <label class="col-md-3 col-xs-8" for="anggota">Anggota Tim Penguji:</
72       label>
73     <!-- 9 -->
74     <select class="col-md-9 col-xs-4" id="anggota" name="
75       namaAnggotaTimPenguji" ng-model="namaAnggotaTimPenguji">
76       <option value="Cecilia_Esti_Nugraheni">Cecilia Esti Nugraheni
77         </option>
78       <option value="Veronica_Sri_Moertini">Veronica Sri Moertini</
79         option>
80       <option value="Rosa_De_Lima">Rosa De Lima</option>
81       <option value="Thomas_Anung_Basuki">Thomas Anung Basuki</
82         option>
83       <option value="Lionov">Lionov</option>
84       <option value="Mariskha_Tri_Adithia">Mariskha Tri Adithia</
85         option>
86       <option value="Luciana_Abednego">Luciana Abednego</option>
87       <option value="Elisati_Hulu">Elisati Hulu</option>
88       <option value="Chandra_Wijaya">Chandra Wijaya</option>
89       <option value="Gede_Karya">Gede Karya</option>
90       <option value="Nico_Saputro">Nico Saputro</option>
91       <option value="Joanna_Helga">Joanna Helga</option>
92       <option value="Pascal_Alfdian">Pascal Alfdian</option>
93       <option value="Husnul_Hakim">Husnul Hakim</option>
94       <option value="Vania_Natali">Vania Natali</option>
95       <option value="Aditya_Bagoes_Saputra">Aditya Bagoes Saputra</
96         option>
97       <option value="Claudio_Franciscus">Claudio Franciscus</option>
98       <option value="Natalia">Natalia</option>
99     </select>
100   </p>
101 </div>
102 <br/>
103
```

```

1 | <p>Rekapitulasi nilai Sidang Skripsi 2 yang diberikan oleh pembimbing,  

2 | penguji & koordinator skripsi:</p>
3 | <table class="col-md-8 col-xs-12 col-md-offset-4 col-md-pull-2 table-  

4 | responsive">
5 |   <tr>
6 |     <th>No</th>
7 |     <th>Pembimbing/Penguji</th>
8 |     <th>Nilai</th>
9 |     <th>Bobot(%)</th>
10 |    <th>Nilai Akhir</th>
11 |  </tr>
12 |  <tr>
13 |    <td>1</td>
14 |    <td><label for="nKetua">Ketua Tim Penguji</label></td>
15 |    <td><input type="number" id="nKetua" max="100" ng-model="<br/>
16 |      nilai_ketua" class="form-nilai" value="{{nilai_TTLaporanK*  

17 |      TTLaporanK.value/100+nilai_KMateriK*KMateriK.value/100  

18 |      +nilai_PMateriK*PMateriK.value/100+nilai_PresentasiK*  

19 |      presentasiK.value/100+nilai_PTujuanK*PTujuanK.value/100}}" disabled="disabled" /></td>
20 |    <!-- 10 -->
21 |    <td><input type="number" ng-model="ketua.value" ng-init="ketua.  

22 |      value=.35" min="0" max="100" class="form-nilai" name="<br/>
23 |      bobotKetuaTimPenguji" readonly="readonly" /></td>
24 |    <td><input type="number" value="{{(nilai_TTLaporanK*  

25 |      TTLaporanK.value/100+nilai_KMateriK*KMateriK.value/100+  

26 |      nilai_PMateriK*PMateriK.value/100+nilai_PresentasiK*  

27 |      presentasiK.value/100+nilai_PTujuanK*PTujuanK.value/100)}  

28 |      *ketua.value/100}}" ng-model="total_ketua" class="form-  

29 |      nilai" disabled="disabled" /></td>
30 |
31 |  </tr>
32 |  <tr>
33 |    <td>2</td>
34 |    <td><label for="nAnggota">Anggota Tim Penguji</label></td>
35 |    <td><input id="nAnggota" type="number" max="100" ng-model="<br/>
36 |      nilai_anggota" class="form-nilai" value="{{nilai_TTLaporanA*  

37 |      TTLaporanA.value/100+nilai_KMateriA*KMateriA.value/100  

38 |      +nilai_PMateriA*PMateriA.value/100+nilai_PresentasiA*  

39 |      presentasiA.value/100+nilai_PTujuanA*PTujuanA.value/100}}" disabled="disabled" /></td>
40 |    <!-- 11 -->
41 |    <td><input type="number" ng-model="anggota.value" ng-init="anggota.  

42 |      value=.35" min="0" max="100" class="form-nilai" name="<br/>
43 |      bobotAnggotaTimPenguji" readonly="readonly" /></td>
44 |    <td><input type="number" value="{{(nilai_TTLaporanA*  

45 |      TTLaporanA.value/100+nilai_KMateriA*KMateriA.value/100+  

46 |      nilai_PMateriA*PMateriA.value/100+nilai_PresentasiA*  

47 |      presentasiA.value/100+nilai_PTujuanA*PTujuanA.value/100)}  

48 |      *anggota.value/100}}" ng-model="total_anggota" class="<br/>
49 |      form-nilai" disabled="disabled" /></td>
50 |
51 |  </tr>
52 |  <tr>
53 |    <td>3</td>
54 |    <td><label for="nPembimbing">Pembimbing</label></td>
55 |    <td><input id="nPembimbing" type="number" max="100" ng-model="<br/>
56 |      nilai_pembimbing" class="form-nilai" min=0 value="{{<br/>
57 |      nilai_TTLaporanP*  

58 |      TTLaporanP.value/100+nilai_KMateriP*  

59 |      KMateriP.value/100+nilai_PMateriP*  

60 |      PMateriP.value/100+nilai_PBimbinganP*  

61 |      PBimbinganP.value/100}}" disabled="disabled" /></td>
62 |    <!-- 12 -->
63 |    <td><input type="number" ng-model="pembimbing.value" ng-init="<br/>
64 |      pembimbing.value=.20" min="0" max="100" class="form-nilai" name="<br/>
65 |      bobotPembimbing" readonly="readonly" /></td>
66 |    <td><input type="number" value="{{(nilai_TTLaporanP*  

67 |      TTLaporanP.value/100+nilai_KMateriP*  

68 |      KMateriP.value/100+nilai_PMateriP*  

69 |      PMateriP.value/100+nilai_PBimbinganP*  

70 |      PBimbinganP.value/100)}  

71 |      *pembimbing.value/100}}" ng-model="total_pembimbing" class="form-nilai" disabled="disabled" /></td>
72 |
73 |  </tr>
74 |  <tr>
75 |    <td>4</td>
76 |    <td><label for="nKoordinator">Koordinator Skripsi</label></td>
77 |    <!-- 13 -->
78 |    <td><input id="nKoordinator" type="number" max="100" ng-model="<br/>
79 |      nilai_koordinator" class="form-nilai" min=0 name="<br/>
80 |      nilaiKoordinatorSkripsi"/></td>
81 |    <!-- 14 -->
82 |    <td><input type="number" ng-model="koordinator.value" ng-init="<br/>
83 |      koordinator.value=.10" min="0" max="100" class="form-nilai" name="<br/>
84 |      bobotKoordinatorSkripsi" readonly="readonly" /></td>
85 |    <td><input type="number" value={{nilai_koordinator*koordinator.  

86 |      value/100}} ng-model="total_koordinator" class="form-nilai" disabled="disabled" /></td>
87 |
88 |  </tr>
89 |  <tr>
90 |    <td colspan="2"><label for="nTotal">Total</label></td>
91 |    <td><input type="number" id="nTotal" max="100" disabled="disabled" value={{ketua.value+anggota.value+pembimbing.value+<br/>
92 |      koordinator.value}} class="form-nilai"/></td>
93 |    <!-- 29 -->
94 |    <td><input type="number" name="nilaiAkhirMahasiswa" value="{{<br/>
95 |      nilai_TTLaporanK*  

96 |      TTLaporanK.value/100+nilai_KMateriK*  

97 |      KMateriK.value/100+nilai_PMateriK*  

98 |      PMateriK.value/100+nilai_PPresentasiK*  

99 |      PPresentasiK.value/100+nilai_PTujuanK*  

100 |      PTujuanK.value/100}*ketua.value/100+<br/>
101 |      nilai_TTLaporanA*  

102 |      TTLaporanA.value/100+nilai_KMateriA*  

103 |      KMateriA.value/100+nilai_PMateriA*  

104 |      PMateriA.value/100+nilai_PPresentasiA*  

105 |      PPresentasiA.value/100+nilai_PTujuanA*  

106 |      PTujuanA.value/100+anggota.value/100+<br/>
107 |      nilai_TTLaporanP*  

108 |      TTLaporanP.value/100+nilai_KMateriP*  

109 |      KMateriP.value/100+nilai_PMateriP*  

110 |      PMateriP.value/100+nilai_PPresentasiP*  

111 |      PPresentasiP.value/100}}" /></td>

```

```

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

```

```

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

```

```

1      <td><input type="number" disabled="disabled" value="{{  

2          nilai_PresentasiA * presentasiA.value / 100}}" ng-model="  

3          total_PresentasiA" class="form-nilai"/></td>  

4      </tr>  

5      <tr>  

6          <td><label for="nPTujuanA">Pencapaian Tujuan</label></td>  

7          <td><input type="number" id="nPTujuanA" max="100" ng-model="  

8              nilai_PTujuanA" class="form-nilai"/></td>  

9          <!-- 19 -->  

10         <td><input type="number" name="bobotPencapaianTujuanAnggota"  

11             ng-model="PTujuanA.value" ng-init="PTujuanA.value = 30"  

12             min="0" max="100" class="form-nilai" readonly="readonly"  

13             /></td>  

14         <td><input type="number" disabled="disabled" value="{{  

15             nilai_PTujuanA * PTujuanA.value / 100}}" ng-model="  

16             total_PTujuanA" class="form-nilai"/></td>  

17     </tr>  

18     <tr>  

19         <td colspan="2" ><label for="nTotalBobotA">Total</label></td>  

20         <td><input type="number" id="nTotalBobotA" max="100" disabled="  

21             disabled" value="{{TTLaporanA.value + KMateriA.value +  

22                 PMateriA.value + presentasiA.value + PTujuanA.value}}" class="  

23                 form-nilai"/></td>  

24         <td><input type="number" id="nTotalAnggota" ng-model="  

25             nTotalAnggota" max="100" class="form-nilai" value="{{  

26                 nilai_TTLaporanA * TTLaporanA.value / 100 + nilai_KMateriA  

27                     * KMateriA.value / 100 + nilai_PMateriA * PMateriA.value /  

28                         100 + nilai_PresentasiA * presentasiA.value / 100 +  

29                         nilai_PTujuanA * PTujuanA.value / 100}}" disabled="  

30                         disabled" /></td>  

31     </tr>  

32 </table>  

33 </div>  

34 </div>  

35 </div>  

36 </div>  

37 </section>  

38 <!-- Rekapitulasi Pembimbing -->  

39 <section id="rekPembimbing" class="rekPembimbing-section">  

40     <!-- Page Heading -->  

41     <div class="container">  

42         <div class="row">  

43             <div class="col-lg-6 col-lg-offset-3">  

44                 <div class="page-header">  

45                     <h1>  

46                         Rekapitulasi Penilaian Skripsi 2 (Pembimbing)  

47                     </h1>  

48                     <div class="semester">  

49                         <p>  

50                             <label for="npmP">NPM:</label><input id="npmP" maxlength="10"  

51                                 value="{{n npm}}" disabled="disabled" />  

52                         </p>  

53                     </div>  

54                 </div>  

55             </div>  

56         </div>  

57     <!-- Isi Rekapitulasi Pembimbing -->  

58     <div class="row">  

59         <div class="col-lg-12">  

60             <table class="col-md-8 col-xs-12 col-md-offset-4 col-md-pull-2 table-  

61                 responsive">  

62                 <thead>  

63                     <th>Komponen Penilaian</th>  

64                     <th>Nilai</th>  

65                     <th>Bobot (%)</th>  

66                     <th>Nilai Akhir</th>  

67                 </thead>  

68                 <tbody>  

69                     <tr>  

70                         <td><label for="nTTLaporanP">Tata Tulis Laporan</label></td>  

71                         <td><input type="number" id="nTTLaporanP" max="100" ng-model="  

72                             nilai_TTLaporanP" class="form-nilai"/></td>  

73                         <!-- 25 -->  

74                         <td><input type="number" name="bobotTataTulisLaporanPembimbing"  

75                             ng-model="TTLaporanP.value" ng-init="TTLaporanP.value = 20"  

76                             min="0" max="100" class="form-nilai" readonly="  

77                             readonly" /></td>  

78                         <td><input type="number" disabled="disabled" value="{{  

79                             nilai_TTLaporanP * TTLaporanP.value / 100}}" ng-model="  

80                             total_TTLaporanP" class="form-nilai"/></td>  

81                     </tr>  

82                     <tr>  

83                         <td><label for="nKMateriP">Kelengkapan Materi</label></td>  

84                         <td><input type="number" id="nKMateriP" max="100" ng-model="  

85                             nilai_KMateriP" class="form-nilai"/></td>  

86                         <!-- 26 -->  

87                         <td><input type="number" name="bobotKelengkapanMateriPembimbing"  

88                             ng-model="KMateriP.value" ng-init="KMateriP.value = 20"  

89                             min="0" max="100" class="form-nilai" readonly="  

90                             readonly" /></td>  

91                         <td><input type="number" disabled="disabled" value="{{  

92                             nilai_KMateriP * KMateriP.value / 100}}" ng-model="  

93                             total_KMateriP" class="form-nilai"/></td>  

94                     </tr>  

95                     <tr>  

96                         <td><label for="nPMPembimbing">Penguasaan Materi</label></td>  

97                         <td><input type="number" id="nPMPembimbing" max="100" ng-model="  

98                             nilai_PMateriP" class="form-nilai"/></td>  

99                         <!-- 27 -->  

100                        <td><input type="number" name="bobotPenguasaanMateriPembimbing"  

101                            ng-model="PMateriP.value" ng-init="PMateriP.value = 30"  

102                            min="0" max="100" class="form-nilai" readonly="readonly"  

103                            /></td>

```

```

1      <td><input type="number" disabled="disabled" value="{{  

2          nilai_PMateriP*PMateriP.value/100}}" ng-model="<br>  

3          total_PMateriP" class="form-nilai"/></td>  

4      </tr>  

5      <tr>  

6          <td><label for="nPBimbinganP">Proses Bimbingan</label></td>  

7          <td><input type="number" id="nPBimbinganP" max="100" ng-model="<br>  

8              nilai_PBimbinganP" class="form-nilai"/></td>  

9              <!-- 28 -->  

10             <td><input type="number" name="prosesBimbinganPembimbing" ng-  

11                 model="PBimbinganP.value" ng-init="PBimbinganP.value=.30"  

12                 min="0" max="100" class="form-nilai" readonly="readonly" /></td>  

13             <td><input type="number" disabled="disabled" value="{{  

14                 nilai_PBimbinganP*PBimbinganP.value/100}}" ng-model="<br>  

15                 total_PBimbinganP" class="form-nilai"/></td>  

16             </tr>  

17             <tr>  

18                 <td colspan="2" ><label for="nTotalBobotP">Total</label></td>  

19                 <td><input type="number" id="nTotalBobotP" max="100" disabled="<br>  

20                     disabled" value="{{TTLaporanP.value+KMateriP.value+<br>  

21                         PMateriP.value+PBimbinganP.value}}" class="form-nilai"/></td>  

22                     <td><input type="number" id="nTotalPembimbing" ng-model="<br>  

23                         nTotalPembimbing" max="100" value="{{nilai_TTLaporanP*  

24                             TTLaporanP.value/100+nilai_KMateriP*KMateriP.value/100+<br>  

25                             nilai_PMateriP*PMateriP.value/100+nilai_PBimbinganP.value/100}}" class="form-  

26                             -nilai" disabled="disabled" /></td>  

27             </tr>  

28         </table>  

29     </div>  

30     </div>  

31     </div>  

32     </div>  

33     </div>  

34     </div>  

35     </div>  

36     <!-- Selesai -->  

37     <section id="selesai" class="rekPembimbing-section">  

38         <!-- Page Heading -->  

39         <div class="container">  

40             <div class="row">  

41                 <div class="col-lg-6.col-lg-offset-3">  

42                     <div class="page-header">  

43                         <div class="tanggal">  

44                             <p>  

45                                 Ditetapkan di Bandung, <span id="date"></span>  

46                             <br>  

47                             </p>  

48                         <p>  

49                             Dengan persetujuan:  

50                             <p><input id="ketuaTimPengajiCheck" type="checkbox" value="1" required><label for="ketuaTimPengajiCheck">Ketua tim pengaji (<span ng-bind="namaKetuaTimPengaji"></span>)</label></p>  

51                             <p><input id="anggotaTimPengajiCheck" type="checkbox" value="1" required><label for="anggotaTimPengajiCheck">Anggota tim pengaji (<span ng-bind="namaAnggotaTimPengaji"></span>)</label></p>  

52                             <p><input id="pembimbingCheck" type="checkbox" value="1" required><label for="pembimbingCheck"> Pembimbing (<span ng-bind="namaPembimbing"></span>)</label></p>  

53                             <p><input class="submitBtn" type="submit" name="submit" value="Selesai" ></p>  

54                         </p>  

55                         <script>  

56                             var months = [ 'Januari', 'Februari', 'Maret', 'April', 'Mei', 'Juni', 'Juli', 'Agustus', 'September', 'Oktober', 'November', 'Desember' ];  

57                             var myDays = [ 'Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu' ];  

58                             var date = new Date();  

59                             var day = date.getDate();  

60                             var month = date.getMonth();  

61                             var thisDay = date.getDay();  

62                             thisDay = myDays[thisDay];  

63                             var yy = date.getYear();  

64                             var year = (yy < 1000) ? yy + 1900 : yy;  

65                         <script>  

66                             newdate = thisDay + ',' + day + ',' + months[month] + ',' + year;  

67                             document.getElementById("date").innerHTML = newdate;  

68                         <script>  

69                             newyear = parseInt(year);  

70                             document.getElementById("tahun").value = newyear;  

71                         </script>  

72                     </div>  

73                 </div>  

74             </div>  

75         </div>  

76         </div>  

77         </div>  

78         </div>  

79         </div>  

80         <!-- Set Default Value pada ng-model type number -->  

81         <script>  

82             angular.module('penilaian', [])  

83                 .controller('DefaultValue', ['$scope', function ($scope) {  

84                     }]);  

85                 </script>  

86                 <!-- jQuery -->  

87                 <script src="public/js/jquery.min.js"></script>

```

```

1      <!-- Bootstrap Core JavaScript -->
2      <script src="public/js/bootstrap/bootstrap.min.js"></script>
3
4      <!-- Scrolling Nav JavaScript -->
5      <script src="public/js/jquery.easing.min.js"></script>
6      <script src="public/js/scrolling-nav.js"></script>
7
8      </body>
9
10 </html>
```

Listing B.3: skripsi_model.php

```

11 <?php
12 defined('BASEPATH') OR exit('No direct script access allowed');
13
14 class Skripsi_model extends CI_Model {
15
16     public function insertDataMahasiswa($tableName, $data){
17         $res = $this->db->insert($tableName, $data);
18     }
19
20     public function getAllMahasiswa(){
21         $query = $this->db->get('beritaacaraSidangSkripsi');
22         return $query->result_array();
23     }
24 }
```

Listing B.4: scrolling-nav.css

```

25 /*!
26 * Start Bootstrap - Scrolling Nav HTML Template (http://startbootstrap.com)
27 * Code licensed under the Apache License v2.0.
28 * For details, see http://www.apache.org/licenses/LICENSE-2.0.
29 */
30
31 body {
32     width: 100%;
33     height: 100%;
34 }
35
36 html {
37     width: 100%;
38     height: 100%;
39 }
40
41 @media(min-width:767px) {
42     .navbar {
43         padding: 20px 0;
44         -webkit-transition: background .5s ease-in-out, padding .5s ease-in-out;
45         -moz-transition: background .5s ease-in-out, padding .5s ease-in-out;
46         transition: background .5s ease-in-out, padding .5s ease-in-out;
47     }
48
49     .top-nav-collapse {
50         padding: 0;
51     }
52 }
53
54 /* Demo Sections - You can use these as guides or delete them - the scroller will work with any
55    sort of height, fixed, undefined, or percentage based.
56    The padding is very important to make sure the scrollspy picks up the right area when scrolled to.
57    Adjust the margin and padding of sections and children
58    of those sections to manage the look and feel of the site. */
59
60 .intro-section {
61     height: 100%;
62     padding-top: 80px;
63     background: #fff;
64 }
```

Listing B.5: custom.css

```

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

```
1 |     background: #fff ;
2 | }
3 | input[disabled="disabled"]{
4 |     background-color: #fff ;
5 | }
```

Listing B.6: scrolling-nav.js

```
6 //jQuery to collapse the navbar on scroll
7 $(window).scroll(function() {
8     if ($.navbar").offset().top > 50) {
9         $(".navbar-fixed-top").addClass("top-nav-collapse");
10    } else {
11        $(".navbar-fixed-top").removeClass("top-nav-collapse");
12    }
13 });
14
15 //jQuery for page scrolling feature - requires jQuery Easing plugin
16 $(function() {
17     $('a.page-scroll').bind('click', function(event) {
18         var $anchor = $(this);
19         $('html, body').stop().animate({
20             scrollTop: $($anchor.attr('href')).offset().top
21         }, 1500, 'easeInOutExpo');
22         event.preventDefault();
23     });
24 });


```


LAMPIRAN C

FORM REVISI

- ³ Gambar C.1 adalah lembaran yang digunakan sebagai penanda revisi yang harus dilakukan pada skripsi Sistem Penilaian Sidang Skripsi2 dengan AngularJS:

1 - hal yang harus diperbaiki :

- Periksa kesalahan input (NPM), tambahkan 3 checkbox persetujuan ketiga dozen / buat form peserta dan tanda tangan ✓ [✓] masuk
- Tambahkan 3 checkbox persetujuan ketiga dozen / buat form peserta dan tanda tangan ✓ [✓] masuk
- Spelling & Grammar diperbaiki ✓ [✓] int
- ~~Abstrak~~ algoritma bisa dimasukkan ke bab 4 ✓ [✓]
- Abstrak : tambahkan rumusan masalah di abstrak. ✓ [✓]
- Logika di penulisan dilihat lagi dan diperbaiki. ✓ [✓]
- Bab 23 tidak beri pengantar ✓ [✓]
- Kode program beri caption (seperti gambar) ✓ [✓]
- CI : Buat Jelaskan catatan permasalahan model, view ditulis. ✓ [✓]
- Helper yang digunakan dijelaskan di bab 3. ✓ [✓]
- Istilah yg digunakan dimengerti & dijelaskan (DOM, ngrok, etc) ✓ [✓]
- Hal 23 - 27. ✓ [✓]
- Diganti tabel ✓ [✓]
- Jangan pakai notasi Angular JS ✓ [✓]
- Buat referensi ke gambar. Gambar diberi nomer setiap elemen. ✓ [✓]
- Bab 4: Perancangan basis data, diperbaiki (ada nilai int) ✓ [✓]
- Perancangan antarmuka: jangan seperti screenshot, ada menu, NPM Nama ✓ [✓]
- Pengujian ditulis 5-5 nya, dan hasilnya masing? ✓ [✓]
- Pengujian pada berbagai jenis platform. ✓ [✓]
- Pemrograman. ✓ [✓]
- Kesimpulan 2 : tulislah terkait program yg dibuat. ✓ [✓]
- Bicaraan lagi daftar referensi terkait URL
Tanggal pembahasan ✓ [✓]

Gambar C.1: Form Revisi Skripsi