

SKRIPSI

**SISTEM PENILAIAN SIDANG SKRIPSI 2 DENGAN
ANGULARJS**



BILLY YANUAR

NPM: 2012730017

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

UNDERGRADUATE THESIS

THE THESIS 2 ASSESSMENT DEFENSE SYSTEM



BILLY YANUAR

NPM: 2012730017

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

ABSTRAK

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

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

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

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

ABSTRACT

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

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

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

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

DAFTAR ISI

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	13
2.2.3 Data Binding	14
2.2.4 Model-View-Controller(MVC)	15
2.3 Twitter Bootstrap	16
2.3.1 Grid System	16
2.3.2 Form Class	18
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	20
3.2.2 Analisis Front End	21
3.2.3 Analisis Basis Data	25
3.3 Use Case	26

4	PERANCANGAN	29
4.1	Perancangan Kelas	29
4.2	Routes	29
4.3	Controllers	29
4.4	Models	31
4.5	Perancangan Basis Data	31
4.6	Perancangan Tampilan	32
5	IMPLEMENTASI DAN PENGUJIAN	35
5.1	Implementasi	35
5.1.1	Lingkungan Implementasi dan Pengujian	35
5.1.2	Hasil Implementasi	35
5.2	Hasil Pengujian	38
5.2.1	Pengujian Eksperimental	39
5.2.2	Pengujian Fungsional	44
6	KESIMPULAN DAN SARAN	51
6.1	Kesimpulan	51
6.2	Saran	51
	DAFTAR REFERENSI	53
A	FORM PENILAIAN SKRIPSI	55
B	THE SOURCE CODE	57

DAFTAR GAMBAR

2.1	Flowchart CodeIgniter	5
2.2	Data Binding Classical Templates System	14
2.3	Data Binding pada Angular	14
2.4	Grid Option pada Bootstrap	17
2.5	Contoh Pembagian Grid Columns	18
2.6	Contoh Hasil Penggunaan Kelas Form	18
3.1	Gambar diagram kelas <i>file controllers</i>	21
3.2	Use case diagram	26
4.1	Perkiraan Tampilan Lembar Berita Acara Sidang Skripsi	32
4.2	Perkiraan Tampilan Lembar Rekapitulasi	33
5.1	Formulir berita acara sidang skripsi 2 terisi	36
5.2	Formulir rekapitulasi ketua tim penguji terisi	36
5.3	Formulir rekapitulasi anggota tim penguji terisi	37
5.4	Formulir rekapitulasi pembimbing terisi	37
5.5	Pengisian Checkbox	37
5.6	Pop Up untuk memastikan pengisian nilai sistem	38
5.7	Tampilan sistem pada platform Microsoft Edge	46
5.8	Tampilan sistem pada platform Mozilla Firefox	47
5.9	Tampilan menu sistem pada platform browser smartphone	48
5.10	Tampilan lembar berita acara sistem sidang skripsi pada platform browser smartphone	48
5.11	Tampilan lembar nilai berita acara sidang skripsi sistem pada platform browser smartphone	48
5.12	Tampilan lembar rekapitulasi ketua tim penguji sistem pada platform browser smartphone	49
5.13	Tampilan lembar rekapitulasi anggota tim penguji sistem pada platform browser smartphone	49
5.14	Tampilan lembar rekapitulasi pembimbing sistem pada platform browser smartphone	50
A.1	Form Penilaian Skripsi saat sidang	55
A.2	Form Rekapitulasi Penilaian Skripsi saat sidang	56

DAFTAR TABEL

3.1	Tabel masukan pada tampilan	22
3.2	Tabel model	23
3.3	Tabel view	24
4.1	Tabel Perancangan Basis Data	31
5.1	Tabel Hasil Implementasi	38
5.2	Tabel Pengujian Eksperimental 1	40
5.3	Tabel Pengujian Eksperimental 2	41
5.4	Tabel Pengujian Eksperimental 3	42
5.5	Tabel Pengujian Eksperimental 4	43
5.6	Tabel Pengujian Eksperimental 5	44
5.7	Tabel Pengujian Fungsional	45

BAB 1

PENDAHULUAN

1.1 Latar Belakang

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

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

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

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

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

¹berdasarkan diskusi dengan dosen pembimbing

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

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

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

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

18 1.2 Rumusan Masalah

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

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

24 1.3 Tujuan

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

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

31 1.4 Batasan Masalah

32 Penelitian ini memiliki batasan-batasan sebagai berikut:

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

1.5 Metode Penelitian

Dalam penelitian ini, akan dilakukan langkah-langkah berikut:

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

1.6 Sistematika Penulisan

Berikut adalah sistematika penulisan dari dokumen ini:

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

BAB 2

DASAR TEORI

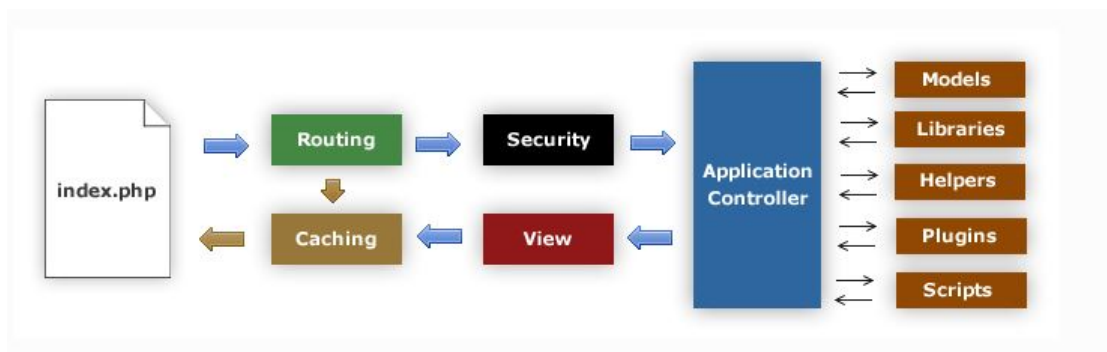
Pada bab ini akan dijelaskan dasar-dasar teori mengenai CodeIgniter, AngularJS, dan Twitter Bootstrap.

2.1 CodeIgniter

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

2.1.1 Flowchart Aplikasi CodeIgniter

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



Gambar 2.1: Flowchart CodeIgniter

Keterangan Gambar 2.1

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

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

2.1.2 Model-View-Controller

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

MVC sendiri terdiri dari 3 bagian, yaitu:

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

2.1.3 Controller

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

Listing 2.1: CI Controller

```

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

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

1 Method

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

Listing 2.2: CI Controller Method

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

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

19 Mendefinisikan Controller Default

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

Listing 2.3: Default Controller

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

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

29 2.1.4 Views

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

35 Pembuatan Views

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

Listing 2.4: File View

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

```

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

```

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

Menjalankan View

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

Listing 2.5: Pemanggilan View

```

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

```

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

2.1.5 Models

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

Listing 2.6: Query Builder Insert dan Update

```

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

```

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

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

1 Menjalankan Model

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

Listing 2.7: Pemanggilan File Model

```

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

```

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

16 2.1.6 Helper

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

23 Menjalankan Helper

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

Listing 2.8: "Kode Helper"

```

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

```

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

31 2.1.7 Basis data

32 Menyambungkan ke Basis Data

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

```

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

```

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

4 2.1.8 Konfigurasi Basis Data

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

Listing 2.9: Array Basis Data

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

1 Keterangan Listing 2.9:

2

Nama Konfigurasi	Deskripsi
dsn	membuat koneksi string (<i>an all-in-one configuration sequence</i>)
hostname	nama host dari server basis data yang dipakai (umumnya bernama "localhost").
username	username yang dipakai untuk menyambungkan basis data
password	password yang cocok dengan username yang dipakai untuk menyambungkan basis data
database	nama basis data yang ingin di sambungkan
dbdriver	tipe basis data (mysql, 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 'mysql' dan 'mysql').
swap_pre	sebuah tabel default yang harus bertukar dengan dbprefix.
schema	skema basis data yang nilai defaultnya adalah 'public'. Digunakan untuk driver PostgreSQL and ODBC.
encrypt	berisi TRUE atau FALSE perlu tidaknya memakai koneksi yang ter-enkripsi.
compress	perlu tidaknya memakai client compression (hanya untuk MYSQL)
stricton	berisi TRUE atau FALSE untuk perlu tidaknya memakai koneksi "Strict Mode"
port	nomor port dari basis data. Untuk menggunakannya diperlukan penambahan di config array database.

3

4 2.2 AngularJS

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

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

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

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

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

20 2.2.1 Gambaran Konseptual

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

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

2.2.2 Directives

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

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

ng-Model

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

ng-Controller

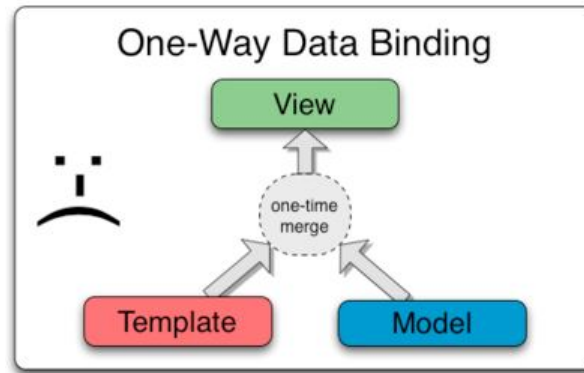
ng-Controller adalah suatu *directives* yang menambahkan suatu kelas *controller* pada bagian *view* dari aplikasi angularJS. ng-Controller inilah yang merupakan kelas kunci dari

¹memperbolehkan pengembang untuk mengajarkan browser sintaks HTML baru

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

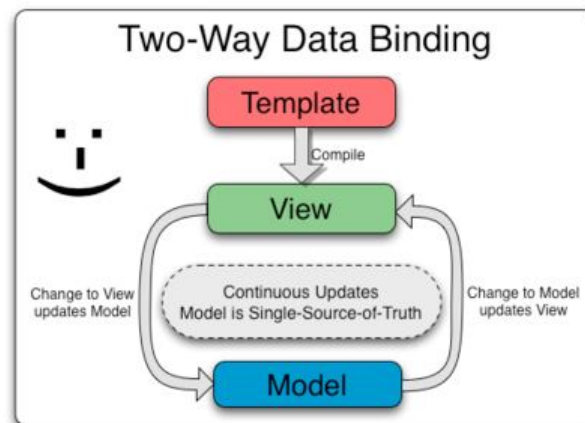
3 2.2.3 Data Binding

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



Gambar 2.2: Data Binding Classical Templates System

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



Gambar 2.3: Data Binding pada Angular

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

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

2.2.4 Model-View-Controller(MVC)

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

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

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

Listing 2.10: Model berupa variabel

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

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

Listing 2.11: Menampilkan model

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

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

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

Listing 2.12: Controller

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

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

Listing 2.13: Penyatuan model view controller

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

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

Listing 2.14: Penyederhanaan controller

```

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

```

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

Listing 2.15: Pemanggilan variabel

```

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

```

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

Listing 2.16: Penyederhanaan MVC

```

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

```

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

37 2.3 Twitter Bootstrap

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

41 2.3.1 Grid System

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

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

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

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
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

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

Gambar 2.5: Contoh Pembagian Grid Columns

2.3.2 Form Class

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

Listing 2.17: Contoh penggunaan Twitter Bootstrap

```

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

```

EXAMPLE

Email address

Password

File input

 No file chosen

Example block-level help text here.

☐ Check me out

Gambar 2.6: Contoh Hasil Penggunaan Kelas Form

BAB 3

ANALISIS

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

3.1 Analisis Sistem Kini

Skripsi 2 merupakan salah satu matakuliah dengan syarat wajib lulus pada Program Studi Teknik Informatika Universitas Katolik Parahyangan, Bandung. Untuk mendapatkan hak sidang, mahasiswa wajib menempuh persyaratan-persyaratan yang sudah ditentukan pada kelas matakuliah Skripsi 2, yaitu mahasiswa harus melakukan proses bimbingan minimal 4 kali, mahasiswa harus mengikuti seminar internal 3 kali, dan mahasiswa wajib menyelesaikan dokumen serta program skripsi yang dikerjakan sebelum tanggal pengumpulan yang telah ditentukan. Jika mahasiswa tidak menyelesaikan persyaratan-persyaratan tersebut, maka mahasiswa yang bersangkutan tidak mendapatkan hak untuk menempuh sidang skripsi 2.

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

3.1.1 Form Rekapitulasi Penilaian

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

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

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

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

3.1.2 Form Berita Acara Sidang Skripsi

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

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

3.2 Analisis Sistem Usulan

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

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

3.2.1 Analisis Back End

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

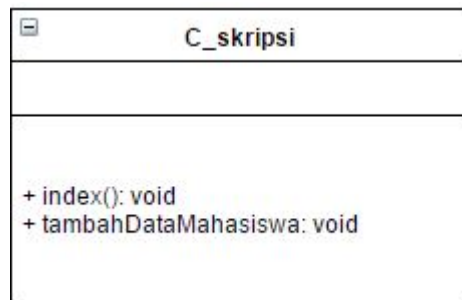
Model

Pada bagian ini akan dijelaskan tentang penggunaan *model* pada *codeigniter*. *Model* mempunyai fungsi untuk membuat sambungan dari aplikasi ke basis data. Pada penelitian ini hanya memiliki 1 kelas diberi nama "Skripsi_model" dengan format *extends CI_Model* yang berfungsi untuk mengaktifka fitur dari model milik *codeigniter*. Kelas model sendiri memiliki sebuah method yang dinamakan "InsertDataMahasiswa" dengan parameter \$tablename

1 untuk masukan nama tabel dan \$data untuk masukan data yang diperoleh dari *contro-*
 2 *ller*. Method tersebut berfungsi untuk melakukan fungsi *insert* ke dalam basis data sistem
 3 usulan.

4 Controller

5 Pada bagian ini akan dijelaskan tentang kode dan kegunaannya pada kelas *controller*. *Con-*
 6 *troller* merupakan kelas yang mengatur hubungan antara kelas *model* dan *view* pada *code-*
 7 *igniter*. Dengan memanfaatkan fungsi-fungsi dari *codeigniter*, maka kelas *controller* dapat
 8 dipersingkat dan dipermudah dalam pembuatannya.



Gambar 3.1: Gambar diagram kelas *file controllers*

9 Berikut adalah *method-method* yang dimiliki oleh kelas *controller* Gambar 3.1:

- 10 • `index()`
 11 Berfungsi untuk memilih *file view* yang akan dipakai pada sistem usulan.
- 12 • `tambahDataMahasiswa`
 13 Berfungsi untuk mengambil data yang telah terisi dari *view* sistem dan mengubahnya
 14 menjadi *compatible* sehingga dapat diproses kedalam *method* `insertDataMahasiswa`
 15 pada kelas *model* yang kemudian akan diproses ke dalam bahasa sql.

16 Helper

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

22 3.2.2 Analisis Front End

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

26 Tahap analisis *front-end* adalah tahap seleksi masukan yang akan ditampilkan dan diisi
 27 oleh pengguna sistem. Diskusi dengan dosen pembimbing menghasilkan 28 buah masukan
 28 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-*
2 *end* 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

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.

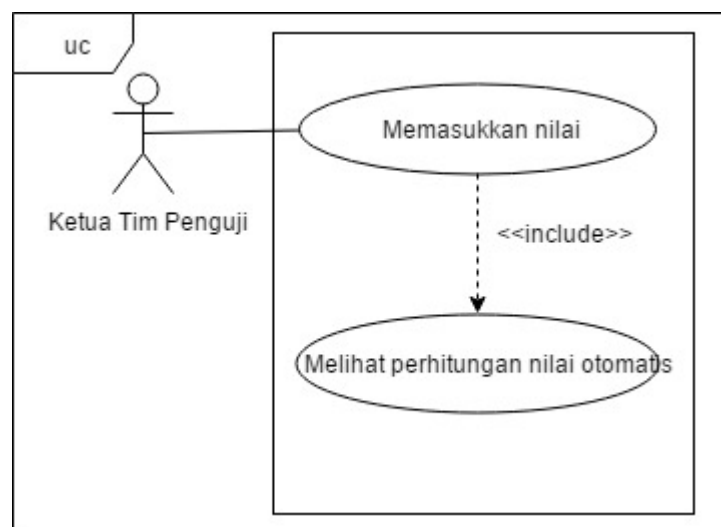
5 Berdasarkan analisa dari contoh form penilaian skripsi yang ada (Gambar A.1 dan
 6 Gambar A.2), dapat disimpulkan bahwa penilaian skripsi membutuhkan data-data sebagai
 7 berikut:

- 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 3.2: Use case diagram

- 18 1. Skenario memasukkan nilai
- 19 Deskripsi: Kegiatan memasukkan nilai ke dalam kotak input yang ada.
- 20 Aktor: Ketua tim penguji
- 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
- 3 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
- 11 dan melakukan perhitungan
- 12 • Sistem menampilkan hasil perhitungan ke dalam kolom yang disediakan untuk
- 13 hasil perhitungan.

BAB 4

PERANCANGAN

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

4.1 Perancangan Kelas

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

4.2 Routes

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

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

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

4.3 Controllers

Controller terdiri dari sebuah kelas yang dinamakan "C_Skripsi". Keseluruhan aktivitas dari sistem informasi penilaian skripsi diatur oleh kelas ini. Seperti yang telah dibahas pada Gambar 3.1, berikut ini adalah *file controller* dengan nama C_skripsi.php.

Listing 4.1: File Controller

```
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
class C_skripsi extends CI_Controller {  
  
    public function index()  
    {  
        $this->load->view('skripsi');  
    }  
  
    public function tambahDataMahasiswa(){
```

```

1         $semester = $_POST['semester'];
2         $tahun = $_POST['tahun'];
3         $npm = $_POST['npm'];
4         $nama = $_POST['nama'];
5         $judul = $_POST['judul'];
6         $namaPembimbing = $_POST['namaPembimbing'];
7         $namaPembimbingPendamping = $_POST['namaPembimbingPendamping'];
8         $namaKetuaTimPenguji = $_POST['namaKetuaTimPenguji'];
9         $namaAnggotaTimPenguji = $_POST['namaAnggotaTimPenguji'];
10        $bobotKetuaTimPenguji = $_POST['bobotKetuaTimPenguji'];
11        $bobotAnggotaTimPenguji = $_POST['bobotAnggotaTimPenguji'];
12        $bobotPembimbing = $_POST['bobotPembimbing'];
13        $nilaiKoordinatorSkripsi = $_POST['nilaiKoordinatorSkripsi'];
14        $bobotKoordinatorSkripsi = $_POST['bobotKoordinatorSkripsi'];
15        $bobotTataTulisLaporanAnggota = $_POST['bobotTataTulisLaporanAnggota'];
16        $bobotKelengkapanMateriAnggota = $_POST['bobotKelengkapanMateriAnggota'];
17        $bobotPenguasaanMateriAnggota = $_POST['bobotPenguasaanMateriAnggota'];
18        $bobotPresentasiAnggota = $_POST['bobotPresentasiAnggota'];
19        $bobotPencapaianTujuanAnggota = $_POST['bobotPencapaianTujuanAnggota'];
20        $bobotTataTulisLaporanKetua = $_POST['bobotTataTulisLaporanKetua'];
21        $bobotKelengkapanMateriKetua = $_POST['bobotKelengkapanMateriKetua'];
22        $bobotPenguasaanMateriKetua = $_POST['bobotPenguasaanMateriKetua'];
23        $bobotPresentasiKetua = $_POST['bobotPresentasiKetua'];
24        $bobotPencapaianTujuanKetua = $_POST['bobotPencapaianTujuanKetua'];
25        $bobotTataTulisLaporanPembimbing = $_POST['bobotTataTulisLaporanPembimbing'];
26        $bobotKelengkapanMateriPembimbing = $_POST['bobotKelengkapanMateriPembimbing'];
27        $bobotPenguasaanMateriPembimbing = $_POST['bobotPenguasaanMateriPembimbing'];
28        $prosesBimbinganPembimbing = $_POST['prosesBimbinganPembimbing'];
29        $nilaiAkhirMahasiswa = $_POST['nilaiAkhirMahasiswa'];
30        $data_insert = array(
31            'semester' => $semester,
32            'tahun' => $tahun,
33            'npm' => $npm,
34            'nama' => $nama,
35            'judul' => $judul,
36            'namaPembimbing' => $namaPembimbing,
37            'namaPembimbingPendamping' => $namaPembimbingPendamping,
38            'namaKetuaTimPenguji' => $namaKetuaTimPenguji,
39            'namaAnggotaTimPenguji' => $namaAnggotaTimPenguji,
40            'bobotKetuaTimPenguji' => $bobotKetuaTimPenguji,
41            'bobotAnggotaTimPenguji' => $bobotAnggotaTimPenguji,
42            'bobotPembimbing' => $bobotPembimbing,
43            'nilaiKoordinatorSkripsi' => $nilaiKoordinatorSkripsi,
44            'bobotKoordinatorSkripsi' => $bobotKoordinatorSkripsi,
45            'bobotTataTulisLaporanAnggota' => $bobotTataTulisLaporanAnggota,
46            'bobotKelengkapanMateriAnggota' => $bobotKelengkapanMateriAnggota,
47            'bobotPenguasaanMateriAnggota' => $bobotPenguasaanMateriAnggota,
48            'bobotPresentasiAnggota' => $bobotPresentasiAnggota,
49            'bobotPencapaianTujuanAnggota' => $bobotPencapaianTujuanAnggota,
50            'bobotTataTulisLaporanKetua' => $bobotTataTulisLaporanKetua,
51            'bobotKelengkapanMateriKetua' => $bobotKelengkapanMateriKetua,
52            'bobotPenguasaanMateriKetua' => $bobotPenguasaanMateriKetua,
53            'bobotPresentasiKetua' => $bobotPresentasiKetua,
54            'bobotPencapaianTujuanKetua' => $bobotPencapaianTujuanKetua,
55            'bobotTataTulisLaporanPembimbing' => $bobotTataTulisLaporanPembimbing,
56            'bobotKelengkapanMateriPembimbing' => $bobotKelengkapanMateriPembimbing,
57            'bobotPenguasaanMateriPembimbing' => $bobotPenguasaanMateriPembimbing,
58            'prosesBimbinganPembimbing' => $prosesBimbinganPembimbing,
59            'nilaiAkhirMahasiswa' => $nilaiAkhirMahasiswa,
60        );
61        $res = $this->skripsi_model->insertDataMahasiswa('beritaacarasingkripsi',
62            $data_insert);
63        redirect(base_url(), 'refresh');
64    }
65 }
66
67 }

```

Keterangan Listing 4.1.

- public function index()
Befungsi untuk mengarahkan pengguna ke *file views default* dari aplikasi.
- public function tambahDataMahasiswa()
Befungsi untuk mengambil data dari *view* yang tersedia, untuk kemudian diolah menjadi bahasa sql oleh *models*.

4.4 Models

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

```

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

```

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

Berfungsi untuk mengolah data yang sudah diolah oleh *controllers* menjadi kueri sql *insert data*.

4.5 Perancangan Basis Data

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

4.6 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.

Navigasi																											
Berita Acara Sidang Skripsi																											
	Semester	Tahun																									
<div style="display: flex; justify-content: space-between;"> NPM Data diri mahasiswa </div> <div style="display: flex; justify-content: space-between;"> Nama </div> <div style="display: flex; justify-content: space-between;"> Judul Skripsi </div>																											
<div style="display: flex; justify-content: space-between;"> Nama Pembimbing Data diri penilai </div> <div style="display: flex; justify-content: space-between;"> Nama Pembimbing Pendamping </div> <div style="display: flex; justify-content: space-between;"> Nama Ketua Tim Penguji </div> <div style="display: flex; justify-content: space-between;"> Nama Anggota Tim Penguji </div>																											
<table style="margin: auto;"> <thead> <tr> <th style="text-align: left;">Nama Penilai</th> <th style="text-align: left;">Nilai</th> <th style="text-align: left;">Bobot Nilai</th> <th style="text-align: left;">Nilai Akhir</th> </tr> </thead> <tbody> <tr><td><input style="width: 100%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td></tr> <tr><td><input style="width: 100%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td></tr> <tr><td><input style="width: 100%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td></tr> <tr><td><input style="width: 100%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td><td><input style="width: 50%;" type="text"/></td></tr> <tr> <td></td> <td></td> <td>Total</td> <td><input style="width: 50%;" type="text"/></td> </tr> </tbody> </table> <div style="margin-top: 10px; font-size: 1.2em; font-weight: bold;">Kolom Penilaian</div>				Nama Penilai	Nilai	Bobot Nilai	Nilai Akhir	<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>			Total	<input style="width: 50%;" type="text"/>
Nama Penilai	Nilai	Bobot Nilai	Nilai Akhir																								
<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>																								
<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>																								
<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>																								
<input style="width: 100%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>	<input style="width: 50%;" type="text"/>																								
		Total	<input style="width: 50%;" type="text"/>																								

Gambar 4.1: Perkiraan Tampilan Lembar Berita Acara Sidang Skripsi

Gambar 4.2 adalah bayangan awal tampilan untuk sistem informasi penilaian skripsi lembar rekapitulasi ketua tim penguji, anggota tim penguji, dan pembimbing.

Navigasi

Header Lembar Rekapitulasi

NPM

Komponen Penilaian	Nilai	Bobot Nilai	Nilai Akhir
		Total	

Kolom Penilaian

Gambar 4.2: Perkiraan Tampilan Lembar Rekapitulasi

BAB 5

IMPLEMENTASI DAN PENGUJIAN

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

5.1 Implementasi

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

5.1.1 Lingkungan Implementasi dan Pengujian

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

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

5.1.2 Hasil Implementasi

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

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

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

Berita Acara Sidang Skripsi

Semester: Ganjil 2016 / 2017

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

NPM: 2012730017 Nama: Billy Yanuar

Judul: Sistem Informasi Penilaian Sidang Skripsi 2 dengan AngularJS

dengan pembimbing dan penguji:

Pembimbing: Pascal Alfadian

Pembimbing Pendamping:

Ketua Tim Penguji: Chandra Wljaya

Anggota Tim Penguji: Mariskha Tri Adithia

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

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

Gambar 5.1: Formulir berita acara sidang skripsi 2 terisi

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

Berita Acara Sidang Skripsi

Lembar Rekapitulasi Ketua Tim Penguji

Lembar Rekapitulasi Anggota Tim Penguji

Lembar Rekapitulasi Pembimbing

Selesai

Rekapitulasi Penilaian Skripsi 2 (Ketua Tim Penguji)

NPM: 2012730017

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

Gambar 5.2: Formulir rekapitulasi ketua tim penguji terisi

Berita Acara Sidang Skripsi				Lembar Rekapitulasi Ketua Tim Penguji				Lembar Rekapitulasi Anggota Tim Penguji				Lembar Rekapitulasi Pembimbing				Selesai			
<h3>Rekapitulasi Penilaian Skripsi 2 (Anggota Tim Penguji)</h3>																			
NPM: <input type="text" value="2012730017"/>																			
Komponen Penilaian				Nilai		Bobot(%)		Nilai Akhir											
Tata Tulis Laporan				<input type="text" value="78"/>		<input type="text" value="15"/>		<input type="text" value="11.7"/>											
Kelengkapan Materi				<input type="text" value="78"/>		<input type="text" value="10"/>		<input type="text" value="7.8"/>											
Penguasaan Materi				<input type="text" value="79"/>		<input type="text" value="30"/>		<input type="text" value="23.7"/>											
Presentasi				<input type="text" value="76"/>		<input type="text" value="15"/>		<input type="text" value="11.4"/>											
Pencapaian Tujuan				<input type="text" value="75"/>		<input type="text" value="30"/>		<input type="text" value="22.5"/>											
Total						<input type="text" value="100"/>		<input type="text" value="77.1"/>											

Gambar 5.3: Formulir rekapitulasi anggota tim penguji terisi

Berita Acara Sidang Skripsi				Lembar Rekapitulasi Ketua Tim Penguji				Lembar Rekapitulasi Anggota Tim Penguji				Lembar Rekapitulasi Pembimbing				Selesai			
<h3>Rekapitulasi Penilaian Skripsi 2 (Pembimbing)</h3>																			
NPM: <input type="text" value="2012730017"/>																			
Komponen Penilaian				Nilai		Bobot(%)		Nilai Akhir											
Tata Tulis Laporan				<input type="text" value="70"/>		<input type="text" value="20"/>		<input type="text" value="14"/>											
Kelengkapan Materi				<input type="text" value="78"/>		<input type="text" value="20"/>		<input type="text" value="15.6"/>											
Penguasaan Materi				<input type="text" value="77"/>		<input type="text" value="30"/>		<input type="text" value="23.1"/>											
Proses Bimbingan				<input type="text" value="75"/>		<input type="text" value="30"/>		<input type="text" value="22.5"/>											
Total						<input type="text" value="100"/>		<input type="text" value="75.2"/>											

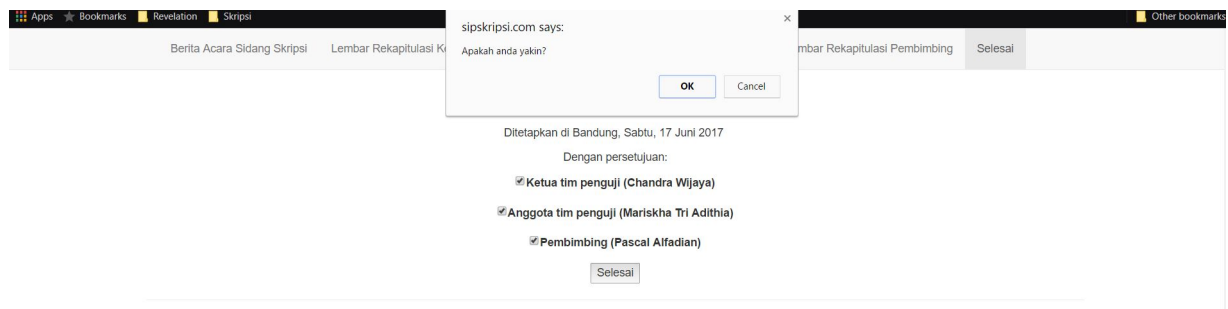
Gambar 5.4: Formulir rekapitulasi pembimbing terisi

3. Bagian selesai.

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

Berita Acara Sidang Skripsi				Lembar Rekapitulasi Ketua Tim Penguji				Lembar Rekapitulasi Anggota Tim Penguji				Lembar Rekapitulasi Pembimbing				Selesai			
Ditetapkan di Bandung, Jum'at, 16 Juni 2017 Dengan persetujuan: <input checked="" type="checkbox"/> Ketua tim penguji (Chandra Wijaya) <input checked="" type="checkbox"/> Anggota tim penguji (Mariskha Tri Adithia) <input checked="" type="checkbox"/> Pembimbing (Pascal Alfadian)																			
<input type="button" value="Selesai"/>																			

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
- 3 fungsional, dan pengujian eksperimental. Berikut penjelasannya:

1 5.2.1 Pengujian Eksperimental

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

8 Pada saat melakukan pengujian eksperimental, sistem kini memiliki kekurangan kece-
9 robohan manusia yang mengakibatkan kesalahan dalam perhitungan nilai baik dari lembar
10 rekapitulasi maupun lembar berita acara sidang skripsi. Hal tersebut diketahui pada saat
11 membandingkan nilai perhitungan nilai akhir yang didapatkan oleh mahasiswa pada sistem
12 kini dan sistem usulan. Pada beberapa kesalahan tersebut, penguji kembali melakukan per-
13 hitungan secara manual dengan menggunakan mesin hitung berupa kalkulator pada *gadget*
14 penguji. Setelah perhitungan dilakukan, didapatkan bahwa sistem usulan memiliki hasil
15 yang benar.

16 Percobaan eksperimental menghasilkan kesimpulan sistem usulan dapat menutupi ke-
17 kurangan sistem kini yang berfokus pada perhitungan penilaian mahasiswa. Dengan meng-
18 gunakan sistem usulan, perhitungan nilai dibuktikan lebih akurat dibandingkan dengan
19 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	A(Tidak Tercatat)
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	Pembangunan Robot Forex dengan Memanfaatkan Analisis Teknikal dan Fundamental
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	ASD(Tidak tercatat)
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	Converter
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	Watermarking
namaPembimbing	Mariskha A
namaPembimbingPendamping	-
namaKetuaTimPenguji	Husnul
namaAnggotaTimPenguji	Chandra
bobotKetuaTimPenguji	35
bobotAnggotaTimPenguji	35
bobotPembimbing	20
nilaiKoordinatorSkripsi	100
bobotKoordinatorSkripsi	10
bobotTataTulisLaporanAnggota	15
bobotKelengkapanMateriAnggota	10
bobotPenguasaanMateriAnggota	30
bobotPresentasiAnggota	15
bobotPencapaianTujuanKetua	30
bobotTataTulisLaporanKetua	15
bobotKelengkapanMateriKetua	10
bobotPenguasaanMateriKetua	30
bobotPresentasiKetua	15
bobotPencapaianTujuanKetua	30
bobotTataTulisLaporanPembimbing	20
bobotKelengkapanMateriPembimbing	20
bobotPenguasaanMateriPembimbing	30
prosesBimbinganPembimbing	30
nilaiAkhirMahasiswa	84

1 5.2.2 Pengujian Fungsional

- 2 Pengujian fungsional dilakukan untuk mengetahui apakah sistem informasi dapat menja-
3 lankan seluruh fungsi-fungsi yang dimiliki dengan baik ketika dijalankan pada *platform*
4 yang berbeda. Hasil pengujian fungsional sistem informasi akan dijabarkan pada Tabel 5.7
5 berikut.

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

Berita Acara Sidang Skripsi

Semester: Ganjil 2017 /

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

NPM: **Nama:**

Judul:

dengan pembimbing dan penguji:

Pembimbing: Cecilia Esti Nugraheni

Pembimbing Pendamping: Cecilia Esti Nugraheni

Ketua Tim Penguji: Cecilia Esti Nugraheni

Anggota Tim Penguji: Cecilia Esti Nugraheni

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

No	Pembimbing/Penguji	Nilai	Bobot(%)	Nilai Akhir
1	Ketua Tim Penguji	<input type="text"/>	<input type="text"/>	<input type="text"/>
2	Anggota Tim Penguji	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	Pembimbing	<input type="text"/>	<input type="text"/>	<input type="text"/>
4	Koordinator Skripsi	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Total		<input type="text"/>	<input type="text"/>

Ditetapkan di Bandung, Minggu, 18 Juni 2017

Dengan persetujuan:

☐ Ketua tim penguji ()

☐ Anggota tim penguji ()

☐ Pembimbing ()

Gambar 5.7: Tampilan sistem pada platform Microsoft Edge

Berita Acara Sidang Skripsi

Semester: / /

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

NPM: Nama:

Judul:

dengan pembimbing dan penguji:

Pembimbing:

Pembimbing Pendamping:

Ketua Tim Penguji:

Anggota Tim Penguji:

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

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

Ditetapkan di Bandung, Minggu, 18 Juni 2017

Dengan persetujuan:

☐ Ketua tim penguji ()

☐ Anggota tim penguji ()

☐ Pembimbing ()

Gambar 5.8: Tampilan sistem pada platform Mozilla Firefox

2017 /

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

NPM:

Nama:

Judul:

dengan pembimbing dan penguji:

Pembimbing:

Pembimbing Pendamping:

Ketua Tim Penguji:

Gambar 5.9: Tampilan menu sistem pada platform browser smartphone

Berita Acara Sidang Skripsi

Semester:

2016 / 2017

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

NPM:

Nama:

Judul:

dengan pembimbing dan penguji:

Pembimbing:

Pembimbing Pendamping:

Ketua Tim Penguji:


Anggota Tim Penguji:

Rekapitulasi nilai

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

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	

Gambar 5.11: Tampilan lembar nilai berita acara sidang skripsi sistem pada platform browser smartphone



Total 100

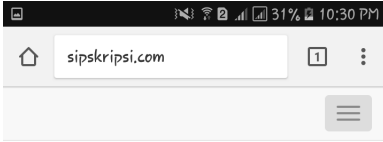
Rekapitulasi Penilaian Skripsi 2 (Ketua Tim Penguji)

NPM:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	<input type="text"/>	15	<input type="text"/>
Kelengkapan Materi	<input type="text"/>	10	<input type="text"/>
Penguasaan Materi	<input type="text"/>	30	<input type="text"/>
Presentasi	<input type="text"/>	15	<input type="text"/>
Pencapaian Tujuan	<input type="text"/>	30	<input type="text"/>
Total		100	<input type="text"/>

Rekapitulasi Penilaian Skripsi 2 (Anggota Tim Penguji)

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




sipskripsi.com

Rekapitulasi Penilaian Skripsi 2 (Anggota Tim Penguji)

NPM:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	<input type="text"/>	15	<input type="text"/>
Kelengkapan Materi	<input type="text"/>	10	<input type="text"/>
Penguasaan Materi	<input type="text"/>	30	<input type="text"/>
Presentasi	<input type="text"/>	15	<input type="text"/>
Pencapaian Tujuan	<input type="text"/>	30	<input type="text"/>
Total		100	<input type="text"/>

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



The screenshot shows a mobile browser interface with a status bar at the top displaying signal strength, Wi-Fi, 31% battery, and the time 10:30 PM. Below the browser header is a hamburger menu icon. The main content area features the title 'Rekapitulasi Penilaian Skripsi 2 (Pembimbing)' in bold black text. Below the title is a text input field labeled 'NPM:'. At the bottom is a table with four columns: 'Komponen Penilaian', 'Nilai', 'Bobot(%)', and 'Nilai Akhir'. The table contains five rows of evaluation criteria with corresponding input fields for scores and weights.

Rekapitulasi Penilaian Skripsi 2 (Pembimbing)

NPM:

Komponen Penilaian	Nilai	Bobot(%)	Nilai Akhir
Tata Tulis Laporan	<input type="text"/>	20	<input type="text"/>
Kelengkapan Materi	<input type="text"/>	20	<input type="text"/>
Penguasaan Materi	<input type="text"/>	30	<input type="text"/>
Proses Bimbingan	<input type="text"/>	30	<input type="text"/>
Total		100	<input type="text"/>

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

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

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

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

6.2 Saran

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

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

DAFTAR REFERENSI

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

2

LAMPIRAN A

3

FORM PENILAIAN SKRIPSI

4 Berikut adalah lembaran penilaian Skripsi yang di pakai di Program Studi Teknik Informa-
1 tika 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, 20

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Ketua Tim Penguji	Anggota Tim Penguji	Pembimbing**	Koordinator Skripsi

Petunjuk pengisian :
1. * = orang yang tidak hadir, ** = salah satu pembimbing atau, jika pembimbing utama tidak maka harus pembimbing utama
2. Bila minimal satu penguji dan/atau seluruh pembimbing tidak hadir, maka sidang harus dibatalkan dan dipaparkan ulang, tetapi berita acara harus tetap dilaporkan kepada koordinator Skripsi atau kepada pimpinan jurusan.

Gambar A.1: Form Penilaian Skripsi saat sidang

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

Rekapitulasi Penilaian SKRIPSI 2 (PEMBIMBING)

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot	Nilai Akhir	Tgl: / / 20
Tata Tulis Laporan		20%		<div style="border: 1px solid black; height: 100px; width: 100%;"></div>
Kelengkapan Materi		20%		
Penguasaan Materi		30%		
Proses Bimbingan		30%		
Total				<div style="border: 1px solid black; height: 20px; width: 100%;"></div>

Rekapitulasi Penilaian SKRIPSI 2 (KETUA TIM PENGUJI)

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot	Nilai Akhir	Tgl: / / 20
Tata Tulis Laporan		15%		<div style="border: 1px solid black; height: 100px; width: 100%;"></div>
Kelengkapan Materi		10%		
Penguasaan Materi		30%		
Presentasi		15%		
Pencapaian Tujuan		30%		<div style="border: 1px solid black; height: 20px; width: 100%;"></div>
Total				

Rekapitulasi Penilaian SKRIPSI 2 (ANGGOTA TIM PENGUJI)

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot	Nilai Akhir	Tgl: / / 20
Tata Tulis Laporan		15%		<div style="border: 1px solid black; height: 100px; width: 100%;"></div>
Kelengkapan Materi		10%		
Penguasaan Materi		30%		
Presentasi		15%		
Pencapaian Tujuan		30%		<div style="border: 1px solid black; height: 20px; width: 100%;"></div>
Total				

Gambar A.2: Form Rekapitulasi Penilaian Skripsi saat sidang

LAMPIRAN B

THE SOURCE CODE

Listing B.1: C_skripsi.php

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

```

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

```

Listing B.2: skripsi.php

```

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

```

```

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

```

```

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

```



```

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

```



```

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

```

```

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

```

```

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

```

Listing B.3: skripsi_model.php

```

6  <?php
7  defined('BASEPATH') OR exit('No direct script access allowed');
8
9  class Skripsi_model extends CI_Model {
10
11      public function insertDataMahasiswa($tableName, $data){
12          $res = $this->db->insert($tableName, $data);
13      }
14
15      public function getAllMahasiswa(){
16          $query = $this->db->get('beritaacarasidangskripsi');
17          return $query->result_array();
18      }
19  }

```

Listing B.4: scrolling-nav.css

```

20  /*
21  * Start Bootstrap - Scrolling Nav HTML Template (http://startbootstrap.com)
22  * Code licensed under the Apache License v2.0.
23  * For details, see http://www.apache.org/licenses/LICENSE-2.0.
24  */
25
26  body {
27      width: 100%;
28      height: 100%;
29  }
30
31  html {
32      width: 100%;
33      height: 100%;
34  }
35
36  @media(min-width:767px) {
37      .navbar {
38          padding: 20px 0;
39          -webkit-transition: background .5s ease-in-out, padding .5s ease-in-out;
40          -moz-transition: background .5s ease-in-out, padding .5s ease-in-out;
41          transition: background .5s ease-in-out, padding .5s ease-in-out;
42      }
43
44      .top-nav-collapse {
45          padding: 0;
46      }
47  }
48
49  /* Demo Sections - You can use these as guides or delete them - the scroller will work with any
50     sort of height, fixed, undefined, or percentage based.
51     The padding is very important to make sure the scrollspy picks up the right area when scrolled to.
52     Adjust the margin and padding of sections and children
53     of those sections to manage the look and feel of the site. */
54
55  .intro-section {
56      height: 100%;
57      padding-top: 80px;
58      background: #fff;
59  }

```

Listing B.5: custom.css

```

60  /* My Custom CSS */
61  .form-nilai{
62      width: 45px;
63  }
64
65  .page-header{
66      text-align: center;
67  }
68  tr{
69      height:30px;
70  }
71  .rekAnggota-section {
72      height: 100%;
73      padding-top: 80px;
74      background: #fff;
75  }
76
77  .rekKetua-section {
78      height: 100%;
79      padding-top: 80px;
80      background: #fff;
81  }
82
83  .rekPembimbing-section {
84      height: 100%;
85      padding-top: 80px;
86      background: #fff;
87  }
88  input[disabled="disabled"]{
89      background-color: #fff;
90  }

```

Listing B.6: scrolling-nav.js

```
2  //jQuery to collapse the navbar on scroll
3  $(window).scroll(function() {
4      if ($("#navbar").offset().top > 50) {
5          $(".navbar-fixed-top").addClass("top-nav-collapse");
6      } else {
7          $(".navbar-fixed-top").removeClass("top-nav-collapse");
8      }
9  });
10
11 //jQuery for page scrolling feature - requires jQuery Easing plugin
12 $(function() {
13     $('a.page-scroll').bind('click', function(event) {
14         var $anchor = $(this);
15         $('html, body').stop().animate({
16             scrollTop: $($anchor.attr('href')).offset().top
17         }, 1500, 'easeInOutExpo');
18         event.preventDefault();
19     });
1808 });
```