

SKRIPSI

**SISTEM PENILAIAN SIDANG SKRIPSI 2 DENGAN
ANGULARJS**



BILLY YANUAR

NPM: 2012730017

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

«tahun»

UNDERGRADUATE THESIS

«JUDUL BAHASA INGGRIS»



BILLY YANUAR

NPM: 2012730017

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
«tahun»**

LEMBAR PENGESAHAN

**SISTEM PENILAIAN SIDANG SKRIPSI 2 DENGAN
ANGULARJS**

BILLY YANUAR

NPM: 2012730017

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

**«pembimbing utama/1»
Ketua Tim Penguji**

**«pembimbing pendamping/2»
Anggota Tim Penguji**

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Thomas Anung Basuki, Ph.D.

PERNYATAAN

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

SISTEM PENILAIAN SIDANG SKRIPSI 2 DENGAN ANGULARJS

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» «tahun»

Meterai

Billy Yanuar
NPM: 2012730017

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia» Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris» Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Bandung, «bulan» «tahun»

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xix
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	2
1.6 Sistematika Penulisan	3
2 DASAR TEORI	5
2.1 CodeIgniter	5
2.1.1 Keistimewaan CI	5
2.1.2 Model, View, dan Controller(MVC) pada CI	5
2.1.3 Struktur CI	6
2.2 AngularJS	7
2.2.1 Keistimewaan AngularJS	7
2.2.2 Komponen AngularJS	8
2.3 Twitter Bootstrap	9
2.3.1 Struktur direktori Bootstrap	10
2.3.2 Keuntungan menggunakan Bootstrap	10
3 ANALISIS	11
3.1 Analisis Data Penilaian Skripsi	11
3.2 Analisis Tampilan Sistem Informasi Penilaian Skripsi	11
4 PENUTUP	13
4.1 Kesimpulan	13
DAFTAR REFERENSI	15
A FORM PENILAIAN SKRIPSI	17
B THE SOURCE CODE	19

DAFTAR GAMBAR

3.1	Perbandingan Tampilan [?]	11
3.2	Perkiraan Tampilan [?]	12
A.1	Form Penilaian Skripsi saat sidang	17
A.2	Form Rekapitulasi Penilaian Skripsi saat sidang	18

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Program Studi Teknik Informatika di Universitas Katolik Parahyangan memiliki beberapa syarat kelulusan antara lain minimal sks yang lulus 144 yang terdiri dari matakuliah wajib dan pilihan, indeks prestasi minimum adalah 2.00 dengan maksimum 14 semester. Salah satu matakuliah wajib yang harus ditempuh dan lulus adalah skripsi. Skripsi di Program Studi Teknik Informatika di Universitas Katolik Parahyangan dibagi menjadi 2 matakuliah yaitu skripsi 1 dan skripsi 2.

Sistem penilaian sidang skripsi 2 pada Program Studi Teknik Informatika di Universitas Katolik Parahyangan masih bersifat manual dimana penilai mengisi data-data mahasiswa memberikan nilai untuk mahasiswa pada saat sidang dan juga melakukan penghitungan bobot nilai total.

Sifat manual ini mengakibatkan kelalaian manusia dalam melakukan penilaian pun 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. Selain itu, penyimpanan nilai skripsi pun tergolong sulit karena tidak langsung dibarengi dengan nilai dan npm mahasiswa yang mengerjakan. Untuk mengatasi hal-hal tersebut, diperlukan suatu sistem yang dapat menanggulangi masalah pengisian, kalkulasi perhitungan, dan juga penyimpanan skripsi.

Menurut penjelasan di atas, maka otomatisasi sistem dalam penilaian skripsi penulis mengusulkan oleh Universitas 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 Skripsi.

Pada penelitian ini, akan dibuat sebuah sistem penilaian yang menanggulangi masalah-masalah tersebut dengan cara membuat beberapa masukan dijadikan otomatis dan juga melakukan eksekusi perhitungan nilai akhir sesuai bobot secara otomatis. Hal ini dianggap akan memudahkan penilai dalam proses penilaian skripsi, karena penilai tidak perlu lagi repot menghitung dan juga mengisi hal-hal yang sudah terisi secara otomatis.

Dalam penelitian ini saya memakai framework AngularJS yang dimiliki oleh perusahaan *Google*. AngularJS merupakan salah satu framework yang paling sering digunakan untuk membuat sebuah aplikasi berbasis web dengan konsep *Single Page Application (SPA)*. *Single Page Application* merupakan aplikasi berbasis web yang memungkinkan sebuah halaman HTML memiliki konten - konten

¹berdasarkan diskusi dengan dosen pembimbing

yang dapat digunakan di halaman tersebut tanpa perlu berganti ke halaman lain.

AngularJS juga bisa diintegrasikan dengan aplikasi yang menggunakan framework lain, sehingga sangat berguna dalam pengerjaan aplikasi berbasis web yang sangat luas cakupannya.

1.2 Rumusan Masalah

Berikut adalah susunan permasalahan yang akan dibahas pada penelitian ini:

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

1.3 Tujuan

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

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

1.4 Batasan Masalah

Penelitian ini memiliki batasan-batasan seperti berikut:

1. Penelitian ini hanya dilakukan untuk form penilaian matakuliah skripsi 2

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 kini, 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

2.1 CodeIgniter

CodeIgniter merupakan sebuah *framework* pemrograman *web* dengan menggunakan bahasa php. *CodeIgniter* (CI) akan membantu mengurangi jumlah pengerjaan kode yang harus diketik, mempermudah pembacaan dan pembaharuan *script*, membantu penyusunan struktur yang jelas dan rinci, mendisiplinkan diri dalam pengerjaan *coding* dan memperkuat hasil tanpa disadari.

2.1.1 Keistimewaan CI

CodeIgniter memiliki beberapa keistimewaan antara lain:

1. Menghemat waktu pengembangan
2. *Reuse of code* (pemakaian kembali kode yang sudah ada)
3. Adanya bantuan dari komunitas CI

2.1.2 Model, View, dan Controller(MVC) pada CI

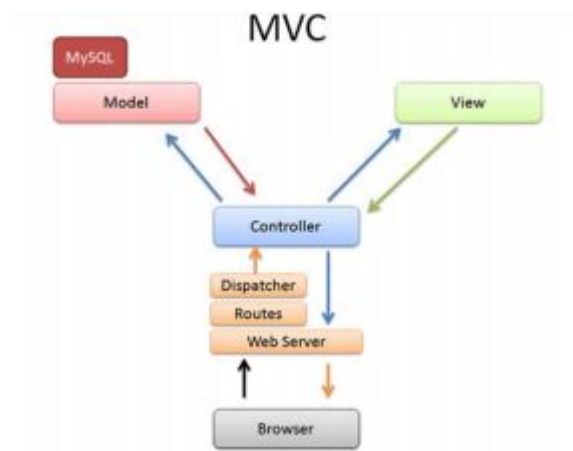
Konsep MVC adalah konsep pemisahan antara logic dengan tampilan dan database. Manfaat konsep ini adalah, membuat coding logic lebih simple, karena sudah di pisah dengan code untuk tampilan dan membuat programmer dapat bekerja secara terpisah dengan designer. CodeIgniter sendiri dibangun menggunakan konsep *Model-View-Controller development pattern*. Manfaat konsep MVC ini adalah memudahkan logika dalam *coding* aplikasi tersebut dan memudahkan pengembangan sistem yang sudah ada.

- *Model* = *Model* berhubungan dengan data dan interaksi ke *database* atau *webservice*. *Model* juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, *file* XML maupun *webservice*. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian *Model* biasanya akan berhubungan dengan perintah-perintah *query* SQL.
- *View* = *View* berhubungan dengan segala sesuatu yang akan ditampilkan ke *end-user*. Bisa berupa halaman web, rss, javascript dan lain-lain. Di dalam *view* hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman

website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. *View* hanya dikhususkan untuk menampilkan data-data hasil dari *model* dan *controller*

- *Controller* = *Controller* bertindak sebagai penghubung data dan *view*. Di dalam *Controller* inilah terdapat kelas-kelas dan fungsi-fungsi yang memproses permintaan dari *View* ke dalam struktur data di dalam *Model*. *Controller* juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada *model*. Tugas *controller* adalah menyediakan berbagai variabel yang akan ditampilkan di *view*, memanggil *model* untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/*error*, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap *input*.

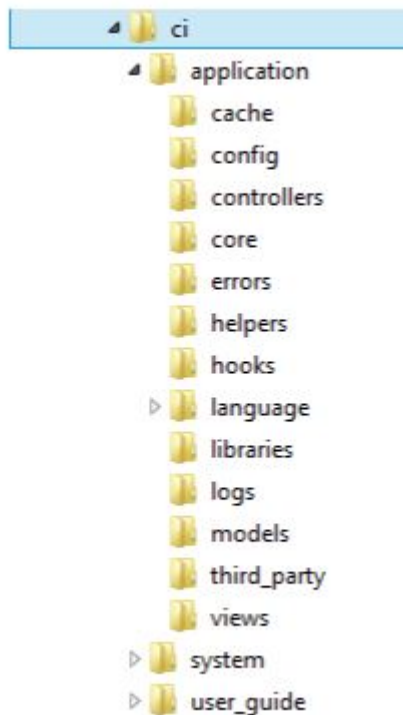
Berikut ini adalah alur proses konsep MVC:



CI menerapkan pola MVC yang flexible, karena model dapat tidak di gunakan. Anda dapat hanya menggunakan Controller dan View saja dalam menggunakan CI tanpa Model. Jika anda tidak memerlukan pemisahan di dalam struktur data dan database atau menganggap penggunaan model hanya menambah kompleks aplikasi dengan keuntungan yang kurang sebanding, maka anda dapat tidak menggunakan model.

2.1.3 Struktur CI

CI adalah sebuah php framework yang berupa kumpulan folder dan file php, java script,css,txt dan file berbasis web lainnya dengan setting tertentu untuk menggunakannya dan menyediakan library dan helper yang dapat di manfaatkan di dalam pemrograman php. CI di jalankan under web dan harus dengan web server. Program CI cukup di letakkan di bawah folder directory web server anda. Berikut adalah struktur file CI :



2.2 AngularJS

AngularJS merupakan *framework javascript* berbasis *open-source* yang dirilis oleh Google pada tahun 2009. AngularJS sendiri merupakan jawaban dari banyak tantangan pemakaian web yang memerlukan pengaplikasian suatu fungsi tanpa berganti halaman. Jika anda merujuk pada situs resmi AngularJS yaitu <http://angularjs.org> maka anda akan dapat membaca sebuah tagline "HTML Enhanced for Web Apps". Tag line tersebut mengartikan bahwa pemakaian angularJS merupakan pemakaian HTML yang telah ditingkatkan fungsinya untuk membangun sebuah aplikasi dalam web.

HTML merupakan alat yang digunakan untuk membangun web statik sehingga membutuhkan bantuan dari alat lain untuk membuat sebuah aplikasi web pada HTML ini. Oleh sebab minat dan banyaknya permintaan dari developer web untuk membuat aplikasi web dengan mudah, maka Google meresmikan AngularJS pada tahun 2009 yang lalu.

AngularJS bukan merupakan pustaka (library) javascript melainkan sebuah framework yang solid untuk membangun web app, seperti framework javascript pada umumnya AngularJS mengadopsi konsep MVC (Model, View, Controller), meskipun menggunakan implementasi yang berbeda dengan konsep asli MVC.

2.2.1 Keistimewaan AngularJS

Keistimewaan penggunaan AngularJS sangatlah banyak. Berikut beberapa diantaranya:

Two way data-binding

Two way data-binding merupakan mekanisme sinkronisasi otomatis antara Controller dan View. Gampangnya, ketika ada perubahan pada Model yang berasal dari View, Angular secara otomatis

membuat perubahan pada Controller. Begitu pula sebaliknya. Hal ini terjadi secara otomatis, jadi kita tidak perlu menuliskan kode secara manual untuk mencapai mekanisme ini.

Mengajari browsers dengan sintak HTML baru

HTML5 menawarkan sejumlah elemen baru semisal `<video>`, `<section>`, `<article>`, dsb. Nah dengan AngularJS, Kita bahkan dapat menambahkan lebih banyak lagi elemen-elemen baru yang akan dimengerti oleh browser, misal `<draggable>` membuat elemen bisa didrag, `<zipppy>` membuat elemen semisal akordion, atau bahkan menggunakan bahasa indonesia seperti `<sembunyikan>` jika diklik akan menyembunyikan elemen (contoh saja, pada praktik gunakanlah bahasa inggris sebagai bahasa internasional). Fungsi ini disebut dengan istilah Directive. Kitalah yang bertanggungjawab membuat Directive tersebut bisa ditafsirkan oleh browser dengan menuliskan kode pada deklarasi Directive itu sendiri. Atau dengan kata lain, kita mengajari browser sintak HTML baru. Bahkan tidak terbatas pada elemen, kita bisa membuat Directive menggunakan attribute, HTML comment atau class.

HTML Template

Template yang digunakan AngularJS hanyalah HTML biasa dengan penambahan ekspresi (expression), sehingga kita tidak perlu menggunakan template engine khusus.

Dependency Injection (DI)

Dependency Injection memungkinkan developer menulis beberapa komponen kode yang terpisah satu sama lain. Ketika memerlukan salah satu komponen, developer dapat memanggil komponen yang dibutuhkan tersebut dan dapat menggunakan fungsi yang tersedia. Fitur ini memudahkan developer dalam membuat komponen yang dapat dipakai berulang kali (reusable component)

2.2.2 Komponen AngularJS

Model

Dalam pola MVC, Model merepresentasikan suatu set data yang digunakan oleh Controller dan View. Model dapat mendeteksi perubahan data dan memberikan notifikasi perubahan tersebut ke Controller dan View. Pada implementasi pasif, notifikasi perubahan dapat diabaikan. Untuk membuat Model di beberapa framework selain AngularJS diperlukan konstruktor khusus. Sedangkan Model pada AngularJS tidak memiliki konstruktor tersendiri dan tidak memerlukan inheritance dari Object Class tertentu. Model tidak memerlukan setter atau getter method khusus. Model bisa berupa primitive, object hash, atau full object. Dengan kata lain Model hanyalah javascript object biasa.

Scope

Scope merupakan perekat (glue) atau perantara antara Controller dengan View. Masing-masing controller memiliki scope atau lingkup sendiri.

Controller

Controller merupakan kode dibalik View. Kode pemrosesan dan logika ditaruh pada controller yang akan menghasilkan Model untuk ditampilkan pada View.

View

View adalah apa yang terlihat oleh pengguna. Dimulai dari sebuah template kemudian digabungkan dengan Model lalu browser melakukan proses rendering dan hasilnya ditampilkan ke pengguna. Template yang digunakan hanyalah sintak HTML (bukan HTML diselingi dengan markup khusus seperti pada template engine pada umumnya).

Expression

Expression merupakan kode snippet yang dapat kita tulis pada View. Expression berkaitan dengan mekanisme binding pada AngularJS, formatnya adalah sebagai berikut `{{ expression }}` Contoh :

1. `"{{ 1+2 }}"` , akan menampilkan angka 3 ke pengguna.
2. `{{ user.name }}` , akan menampilkan nilai properti 'name' dari model 'user'
3. `{{ 1000 | currency }}` , akan menampilkan angka 1000 dalam format mata uang (currency), keyword setelah tanda pipa (|) merupakan filter.

Directive

Directive merupakan cara untuk membuat sintak HTML baru yang akan dimengerti oleh browser. Directive dapat berupa elemen, attribute, HTML comment atau Class. Angular telah menyediakan beberapa directive bawaan yang penting dalam pengembangan web app. Beberapa directive bawaan Angular diantaranya ng-controller, ng-model, ng-repeat, ng-click dll. Kita dapat pula membuat custom directive dengan perilaku (behavior) tertentu seperti yang telah dijelaskan pada pembahasan Apa yang membuat AngularJS istimewa.

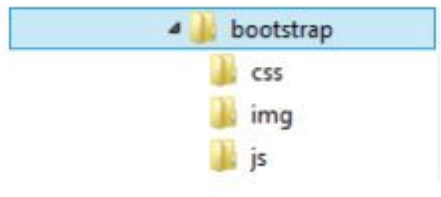
2.3 Twitter Bootstrap

Twitter Bootstrap adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, website yang dibuat dengan alat bantu ini memiliki tampilan halaman yang sama / mirip dengan tampilan halaman *Twitter* atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan.

Twitter Bootstrap dibangun dengan teknologi HTML dan CSS yang dapat membuat layout halaman website, tabel, tombol, form, navigasi, dan komponen lainnya dalam sebuah website hanya dengan memanggil fungsi CSS (class) dalam berkas HTML yang telah didefinisikan. Selain itu juga terdapat komponen-komponen lainnya yang dibangun menggunakan JavaScript.

2.3.1 Struktur direktori Bootstrap

Berikut adalah gambar struktur direktori dari *Bootstrap*:



2.3.2 Keuntungan menggunakan Bootstrap

Penggunaan *Twitter Bootstrap* tentu saja memiliki banyak keuntungan, diantaranya adalah:

- Memudahkan dalam mendesain website
- Responsif (mendukung segala macam layar dan *device*)
- Adanya dokumentasi yang cukup lengkap
- Mempunyai tampak yang elegan

BAB 3

ANALISIS

3.1 Analisis Data Penilaian Skripsi

Berdasarkan analisa dari contoh form penilaian skripsi yang ada, dapat disimpulkan bahwa penilaian skripsi membutuhkan data-data sebagai berikut:

- Semester
- Tahun ajaran
- NPM mahasiswa
- Nama mahasiswa
- Judul skripsi
- Pembimbing utama/tunggal
- Pembimbing pendamping(tidak harus)
- Ketua tim penguji
- Anggota tim penguji
- dan bobot masing-masing penilai

3.2 Analisis Tampilan Sistem Informasi Penilaian Skripsi

Tampilan pada sistem informasi penilaian skripsi haruslah dibuat semirip mungkin dengan form penilaian skripsi yang sudah ada.

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.

Berikut adalah bayangan awal tampilan untuk sistem informasi penilaian skripsi:

Gambar 3.1: Perbandingan Tampilan [?]

Gambar 3.2: Perkiraan Tampilan [?]

BAB 4

PENUTUP

4.1 Kesimpulan


Dari hasil karya ilmiah di atas, maka penulis dapat menyimpulkan bahwa penelitian membuat aplikasi sistem informasi penilaian skripsi dengan menggunakan AngularJS ini dapat mengotomatisasi perhitungan pada nilai yang diberikan oleh penilai kepada mahasiswa yang menjalankan skripsi. Dengan pengkombinasian HTML dan AngularJS, maka sistem informasi yang dihasilkan diprediksikan mempunyai tampilan yang baik dan mudah dimengerti, sekaligus mempunyai otomatisasi penghitungan dan pengisian data seperti tahun ajaran dan semester, yang dapat meminimalisir dibuatnya kelalaian penilai. Hal ini juga diprediksikan dapat memudahkan penilai dalam memberikan nilai pada mahasiswa tersebut.

DAFTAR REFERENSI

LAMPIRAN A

FORM PENILAIAN SKRIPSI

Berikut adalah lembaran penilaian Skripsi yang di pakai di Program Studi Teknik Informatika Universitas Katolik Parahyangan [A.1](#):



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

Ketua Tim Penguji	Anggota Tim Penguji	Pembimbing**	Koordinator Skripsi

Petunjuk pengisian :

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

Gambar A.1: Form Penilaian Skripsi saat sidang

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

Rekapitulasi Penilaian SKRIPSI 2 (PEMBIMBING)

NPM Mahasiswa:

Komponen Penilaian	Nilai	Bobot	Nilai Akhir
Tata Tulis Laporan		20%	
Kelengkapan Materi		20%	
Penguasaan Materi		30%	
Proses Bimbingan		30%	
		Total	

Tgl: / / 20
 Ttd:

 Nama:

Rekapitulasi Penilaian SKRIPSI 2 (KETUA TIM PENGUJI)

NPM Mahasiswa:

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

Tgl: / / 20
 Ttd:

 Nama:

Rekapitulasi Penilaian SKRIPSI 2 (ANGGOTA TIM PENGUJI)

NPM Mahasiswa:

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

Tgl: / / 20
 Ttd:

 Nama:

Gambar A.2: Form Rekapitulasi Penilaian Skripsi saat sidang

LAMPIRAN B

THE SOURCE CODE

Listing B.1: MyFurSet.java

```

1
2 import java.util.ArrayList;
3 import java.util.Collections;
4 import java.util.HashSet;
5
6 /**
7  *
8  * @author Lionov
9  */
10
11 //class for set of vertices close to furthest edge
12 public class MyFurSet {
13     protected int id; //id of the set
14     protected MyEdge FurthestEdge; //the furthest edge
15     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
16     protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each
17         trajectory
18     protected ArrayList<Integer> closeID; //store the ID of all vertices
19     protected ArrayList<Double> closeDist; //store the distance of all vertices
20     protected int totaltrj; //total trajectories in the set
21
22     /**
23      * Constructor
24      * @param id : id of the set
25      * @param totaltrj : total number of trajectories in the set
26      * @param FurthestEdge : the furthest edge
27      */
28     public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
29         this.id = id;
30         this.totaltrj = totaltrj;
31         this.FurthestEdge = FurthestEdge;
32         set = new HashSet<MyVertex>();
33         ordered = new ArrayList<ArrayList<Integer>>();
34         for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
35         closeID = new ArrayList<Integer>(totaltrj);
36         closeDist = new ArrayList<Double>(totaltrj);
37         for (int i = 0; i < totaltrj; i++) {
38             closeID.add(-1);
39             closeDist.add(Double.MAX_VALUE);
40         }
41     }
42
43     /**
44      * set a vertex into the set
45      * @param v : vertex to be added to the set
46      */
47     public void add(MyVertex v) {
48         set.add(v);
49     }
50
51     /**
52      * check whether vertex v is a member of the set
53      * @param v : vertex to be checked
54      * @return true if v is a member of the set , false otherwise
55      */
56     public boolean contains(MyVertex v) {
57         return this.set.contains(v);
58     }
59 }

```