

**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

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xviii</b>
<b>DAFTAR TABEL</b>	<b>xix</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metode Penelitian . . . . .	2
1.6 Sistematika Penulisan . . . . .	3
<b>2 DASAR TEORI</b>	<b>5</b>
2.1 AngularJS . . . . .	5
2.2 Keistimewaan AngularJS . . . . .	5
2.2.1 Two way data-binding . . . . .	5
2.2.2 Mengajari browsers dengan sintak HTML baru . . . . .	6
2.2.3 HTML Template . . . . .	6
2.2.4 Dependecy Injection (DI) . . . . .	6
2.3 Key Concept AngularJS . . . . .	6
2.3.1 Model . . . . .	6
2.3.2 Scope . . . . .	6
2.3.3 Controller . . . . .	7
2.3.4 View . . . . .	7
2.3.5 Expression . . . . .	7
2.3.6 Directive . . . . .	7
<b>3 ANALISIS</b>	<b>9</b>
3.1 Analisis Data Penilaian Skripsi . . . . .	9
3.2 Analisis Tampilan Sistem Informasi Penilaian Skripsi . . . . .	9
<b>4 PENUTUP</b>	<b>11</b>
4.1 Kesimpulan . . . . .	11
<b>A THE PROGRAM</b>	<b>13</b>
<b>B THE SOURCE CODE</b>	<b>15</b>

## DAFTAR GAMBAR

3.1	Perbandingan Tampilan [?]	9
3.2	Perkiraan Tampilan [?]	10
A.1	Interface of the program	13

## 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 skripsi<sup>1</sup> berdasarkan diskusi dengan dosen pembimbing. 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* meru-

---

<sup>1</sup>b

pakan aplikasi berbasis web yang memungkinkan sebuah halaman HTML memiliki konten - konten yang dapat digunakan di halaman tersebut tanpa perlu berganti ke halaman lain.

AngularJS juga bisa di integrasikan 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 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 yagn 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 Keistimewaan AngularJS

Keistimewaan penggunaan AngularJS sangatlah banyak. Berikut beberapa diantaranya:

1. Two way data-binding
2. Mengajari browsers dengan sintak HTML baru
3. HTML Template
4. Dependency Injection (DI)

##### 2.2.1 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.

### 2.2.2 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, `<zipy>` 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.

### 2.2.3 HTML Template

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

### 2.2.4 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)

## 2.3 Key Concept AngularJS

### 2.3.1 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.

### 2.3.2 Scope

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

### 2.3.3 Controller

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

### 2.3.4 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 diselengi dengan markup khusus seperti pada template engine pada umumnya).

### 2.3.5 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.

### 2.3.6 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.



## 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.

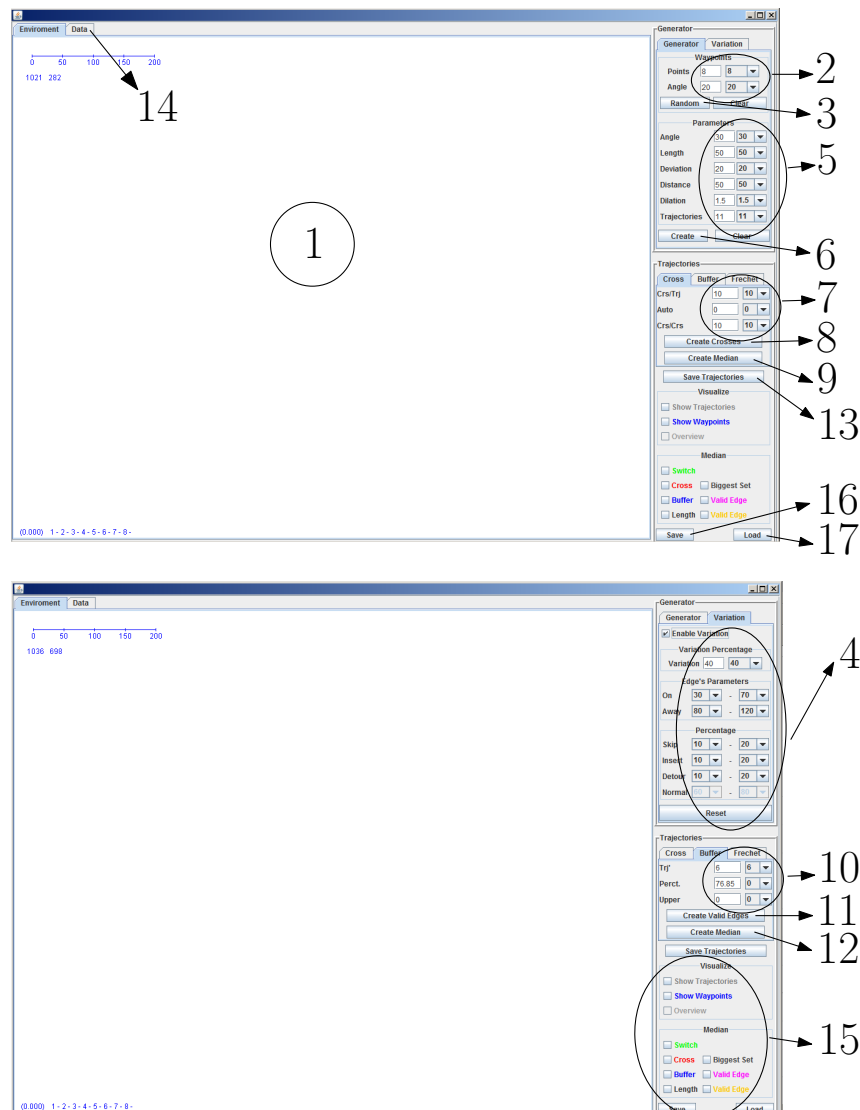




# LAMPIRAN A

## THE PROGRAM

The interface of the program is shown in Figure A.1:



Gambar A.1: Interface of the program

Step by step to compute the median trajectory using the program:

1. Create several waypoints. Click anywhere in the “Environment” area(1) or create them automatically by setting the parameters for waypoint(2) or clicking the button “Random”(3).

2. The “Variation” tab could be used to create variations by providing values needed to make them(4).
3. Create a set of trajectories by setting all parameters(5) and clicking the button “Create”(6).
4. Compute the median using the homotopic algorithm:
  - Define all parameters needed for the homotopic algorithm(7).
  - Create crosses by clicking the “Create Crosses” button(8).
  - Compute the median by clicking the “Compute Median” button(9).
5. Compute the median using the switching method and the buffer algorithm:
  - Define all parameters needed for the buffer algorithm(10).
  - Create valid edges by clicking the “Create Valid Edges”button(11).
  - Compute the median by clicking the “Compute Median”button(12).
6. Save the resulting median by clicking the “Save Trajectories” button(13). The result is saved in the computer memory and can be seen in “Data” tab(14)
7. The set of trajectories and its median trajectories will appear in the “Environment” area(1) and the user can change what to display by selecting various choices in “Visualize” and “Median” area(15).
8. To save all data to the disk, click the “Save”(16) button. A file dialog menu will appear.
9. To load data from the disk, click the “Load”(17) button.

# 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 }

```