

**SKRIPSI**

**PORTING PHP MENJADI JAVA DAN PLAY FRAMEWORK.  
STUDI KASUS: KIRI DASHBOARD SERVER SIDE**



**TOMMY ADHITYA THE**

**NPM: 2012730031**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2015**



**UNDERGRADUATE THESIS**

**PORTING PHP TO JAVA AND PLAY FRAMEWORK. CASE  
STUDY: KIRI DASHBOARD SERVER SIDE**



**TOMMY ADHITYA THE**

**NPM: 2012730031**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2015**



# LEMBAR PENGESAHAN

## PORTING PHP MENJADI JAVA DAN PLAY FRAMEWORK. STUDI KASUS: KIRI DASHBOARD SERVER SIDE

TOMMY ADHITYA THE

NPM: 2012730031

Bandung, 5 Juli 2015

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Pascal Alfadian, M.Com./1»

«pembimbing pendamping/2»

Ketua Tim Penguji

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:

### **PORTING PHP MENJADI JAVA DAN PLAY FRAMEWORK. STUDI KASUS: KIRI DASHBOARD SERVER SIDE**

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 5 Juli 2015

Meterai

Tommy Adhitya The  
NPM: 2012730031





## 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, Juli 2015

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
2 THE DESCRIPTION OF A SET OF TRAJECTORIES AND ITS MEDIAN TRAJECTORY	5
2.1 Set of Trajectories . . . . .	5
2.2 Properties of the Median Trajectory . . . . .	6
DAFTAR REFERENSI	9
A THE PROGRAM	11
B THE SOURCE CODE	13

## DAFTAR GAMBAR

2.1	Numbering of points and segments . . . . .	5
2.2	Possible median trajectory (in black) with backward direction (pointed by the blue arrow) . . . . .	6
2.3	Red arcs indicate the angular change at each vertex . . . . .	6
A.1	Interface of the program . . . . .	11

## DAFTAR TABEL

2.1	Table $\Gamma$ after inserting $\mathcal{S}_1$ . . . . .	7
2.2	$\mathcal{S}_2$ between $v_{start}$ and $\mathcal{S}_1$ . . . . .	7
2.3	$\mathcal{S}_2$ between $\mathcal{S}_1$ and $v_{end}$ . . . . .	7
2.4	Final $\Gamma$ . . . . .	7



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kemacetan merupakan salah satu masalah yang umumnya ditemukan di kota-kota besar. Kemacetan telah menghabiskan banyak sekali waktu produktivitas orang-orang yang tinggal di kota besar. Kemacetan dapat terjadi karena banyak faktor. Salah satu faktor kemacetan adalah karena jumlah kendaraan yang ada dalam suatu kota tidak sebanding dengan kapasitas jalan yang ada pada kota tersebut.

Bandung merupakan salah satu kota besar di mana kemacetan terjadi hampir setiap hari terutama pada saat waktu menjelang berangkat dan pulang kerja. Kemacetan terjadi dikarenakan setiap individu di Bandung umumnya memiliki jam kerja yang sama. Jam kerja yang sama membuat ledakan kendaraan terjadi di jalanan pada saat-saat tertentu. Dikarenakan kapasitas jalanan tidak dapat menampung banyaknya kendaraan yang ada maka terjadilah kemacetan.

Salah satu solusi untuk mengatasi kemacetan adalah penggunaan fasilitas angkutan umum. Angkutan umum akan mengurangi penggunaan kendaraan pribadi secara drastis. Faktanya, di Bandung orang-orang masih jarang menggunakan angkutan umum. Orang takut dan bingung menggunakan angkutan umum dikarenakan kendaraan umum seperti angkot memiliki rute jalannya masing-masing sesuai kode angkot. Terkadang perlu digunakan lebih dari 1 angkot untuk sampai ke tujuan. Bandung juga merupakan kota yang banyak dihuni oleh kaum-kaum pelajar(mahasiswa) asal luar Bandung. Otomatis banyak penghuni Bandung adalah orang-orang yang masih buta akan jalan apalagi rute angkutan umum yang ada di Bandung.

KIRI merupakan sebuah solusi untuk permasalahan kebutaan rute angkutan umum masyarakat yang tinggal di Bandung. KIRI memberikan layanan berupa pengetahuan akan rute angkutan umum yang harus ditempuh oleh seseorang untuk mencapai tujuannya dari tempat orang tersebut berasal. Dengan adanya KIRI orang-orang akan menjadi lebih berani dan siap untuk menggunakan kendaraan umum dalam bepergian baik orang asli Bandung maupun orang dari luar Bandung. Permasalahan kemacetan di Bandung diharapkan dapat berkurang atau bahkan terselesaikan dengan adanya bantuan dari KIRI.

KIRI terbagi ke dalam 2 bagian penting, yaitu: *frontend* dan *dashboard*. Pada bagian *frontend* pada KIRI merupakan bagian *user interface* dan juga sekaligus merupakan halaman utama KIRI, yang dapat diakses melalui alamat <http://kiri.travel>. Pada bagian *dashboard* pada KIRI merupakan bagian yang digunakan untuk tim *developer* untuk memperbaiki ataupun melakukan pengembangan sistem KIRI, yang dapat diakses melalui alamat <https://dev.kiri.travel/bukitjarian>.

KIRI menggunakan bahasa pemrograman PHP untuk membuat bagian *dashboard server side*.

KIRI merupakan proyek besar yang akan terus berkembang seiring dengan perkembangan waktu dikarenakan kebutuhan masyarakat akan rute angkutan umum akan selalu ada dan terus berkembang. Bahasa PHP kurang cocok untuk proyek skala besar seperti KIRI. Tidak ada deklarasi dan tipe variabel yang jelas pada bahasa PHP. Setiap programmer PHP memiliki ciri khasnya sendiri dalam membangun program. Untuk setiap programmer baru yang ingin mengembangkan sistem perlu mempelajari sistem secara detail terlebih dahulu agar mampu mengerti konsep desain program yang ada. Java merupakan bahasa yang umum dan lebih terstruktur. Play Framework membantu implementasi Java untuk pembuatan *website*. Play Framework menggunakan prinsip MVC sehingga setiap programmer sudah langsung mengetahui desain program yang ada. Play Framework membuat setiap programmer memiliki konsep dan prinsip desain program yang sama antara yang satu dan yang lainnya, sehingga mempermudah programmer untuk melakukan pengembangan sistem.

Berdasarkan permasalahan-permasalahan yang telah peneliti jelaskan, peneliti tertarik untuk melakukan perombakan sistem bagian *dashboard* KIRI(<https://dev.kiri.travel/bukitjarian>). Peneliti tertarik untuk melakukan perombakan bahasa yang digunakan, dari yang semula menggunakan bahasa PHP menjadi menggunakan Java dengan menggunakan Play Framework. Untuk itu dibuatlah penelitian “Porting PHP menjadi Java dan Play Framework. Studi Kasus: KIRI Dashboard Server Side” guna menyelesaikan masalah-masalah yang ada.

## 1.2 Rumusan Masalah

Berikut adalah susunan permasalahan yang akan dibahas pada penelitian ini:

1. Bagaimana rancangan dan cara kerja sistem *dashboard server side* KIRI versi lama dengan menggunakan bahasa pemrograman PHP?
2. Apa saja kelebihan dan kekurangan rancangan sistem *dashboard server side* KIRI versi lama dengan menggunakan bahasa pemrograman PHP?
3. Bagaimana rancangan dan cara kerja sistem *dashboard server side* KIRI versi baru dengan menggunakan bahasa pemrograman Java dengan memanfaatkan Play Framework?
4. Apa saja kelebihan dan kekurangan rancangan sistem *dashboard server side* KIRI versi baru dengan menggunakan bahasa pemrograman Java dengan memanfaatkan Play Framework?

## 1.3 Tujuan

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

1. Mengetahui rancangan dan cara kerja sistem *dashboard server side* KIRI versi lama dengan menggunakan bahasa pemrograman PHP,
2. Mengetahui kelebihan dan kekurangan rancangan sistem *dashboard server side* KIRI versi lama dengan menggunakan bahasa pemrograman PHP,

- 
3. Mengetahui rancangan dan cara kerja sistem *dashboard server side* KIRI versi baru dengan menggunakan bahasa pemrograman Java dengan memanfaatkan Play Framework,
  4. Mengetahui kelebihan dan kekurangan rancangan sistem *dashboard server side* KIRI versi baru dengan menggunakan bahasa pemrograman Java dengan memanfaatkan Play Framework.





## BAB 2

# THE DESCRIPTION OF A SET OF TRAJECTORIES AND ITS MEDIAN TRAJECTORY

### 2.1 Set of Trajectories

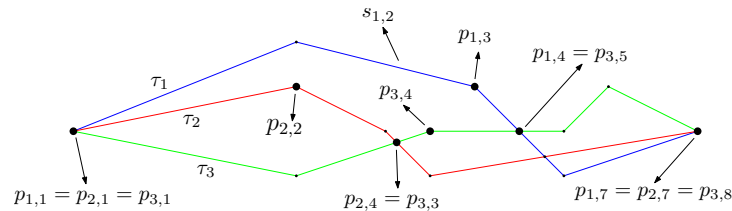
In this thesis, we only consider the spatial component of the trajectory. Therefore, we represent a trajectory as a polygonal line built by a series of points and connected by line segments.

Let  $T := \{\tau_1, \tau_2, \dots, \tau_m\}$  be the input set of  $m$  trajectories for which we want to compute its median trajectory  $\tau^M$ . We define each trajectory in  $T$  as a list of at most  $n$  points,  $\tau_i := (p_{i,1}, \dots, p_{i,k})$  where  $1 \leq i \leq m$  and  $2 \leq k \leq n$ . Note that the number of points for each trajectory can be different. Every two consecutive points  $p_{i,j}$  and  $p_{i,j+1}$  ( $1 \leq j \leq k-1$ ) are connected by a segment  $s_{i,j} := (\overline{p_{i,j}, p_{i,j+1}})$ .

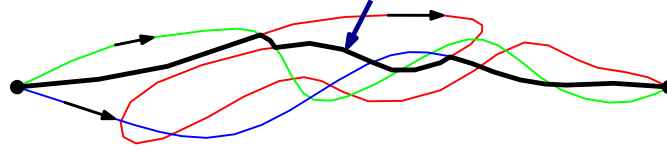
$P$  is the set of all points in  $T$ ,  $P := \{p_{i,j} \mid i \in \{1 \dots m\}, j \in \{1 \dots n\}\}$  and  $S$  is the set of all segments in  $T$ ,  $S := \{s_{i,j} \mid i \in \{1 \dots m\}, j \in \{1 \dots n-1\}\}$ . All trajectories in  $T$  share the same start and end points ( $p_{1,1} = p_{2,1} = \dots = p_{m,1}$  and  $p_{1,k_1} = p_{2,k_2} = \dots = p_{m,k_m}$  where  $\{k_1, \dots, k_m\} \in \{1, \dots, n\}$ ).

Trajectories can intersect with other trajectories in other points than their start and end points. These intersection points are also included in the list of points that define the trajectory. When two segments intersect each other, then both segments will be split into two parts and all four segments share one intersection point as one of their endpoints (see Figure 2.1).

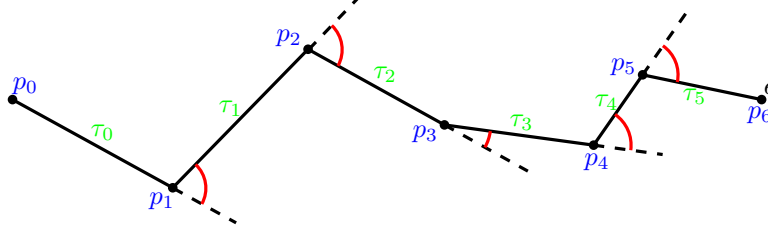
Let  $n'$  be the number of points in a trajectory, including their intersection points with other trajectories. In the worst case,  $n' = mn^2$ . In the rest of this thesis, we define  $n$  as a number of points in a trajectory, inclusive with its intersection points with other trajectories. Note that the number of segments for each trajectory is linear to the number of points, because trajectory with  $n$  points has  $n-1$  segments.



Gambar 2.1: Numbering of points and segments



Gambar 2.2: Possible median trajectory (in black) with backward direction (indicated by the blue arrow)



Gambar 2.3: Red arcs indicate the angular change at each vertex

## 2.2 Properties of the Median Trajectory

We define several properties for the median trajectory  $\tau^M$  with respect to the input set of trajectories  $T$ :

- $\tau^M$  is a directed polygonal line from start point to end point and should be similar to most trajectories in  $T$ .
- It must be built only using points and segments which are parts of trajectories in the input set.
- The usage of segments should follow the direction of them. Therefore, it is not allowed to use a segment such that the direction of  $\tau^M$  is opposite to the direction of that segment in a trajectory (see Figure 2.2, indicated by the dark blue arrow).
- The length of the median trajectory should be relatively the same as the average length of all trajectories in the input set.
- The total angular change should also be similar to the average of total angular change of all trajectories in the input set. The total angular change of a trajectory is the sum of all angular changes at every vertex in that trajectory (see Figure 2.3).
- The number of vertices and edges of  $\tau^M$  should be about the same with the average of the number of vertices and edges from all trajectories in the input set.

Using the definition of the input set of trajectories defined in the previous section, we define a median trajectory  $\tau^M$  as a sequence of points from  $T$ ,  $\tau^M := (p_{i_1, j_1}, p_{i_2, j_2}, \dots, p_{i_k, j_k})$  where  $\{i_1, i_2, \dots, i_k\} \in \{1 \dots m\}$  and  $\{j_1, j_2, \dots, j_k\} \in \{1 \dots n\}$ , or defined as a sequence of segments:  $\tau^M := (s_{i_1, j_1}, s_{i_2, j_2}, \dots, s_{i_k, j_k})$  where  $\{i_1, i_2, \dots, i_k\} \in \{1 \dots m\}$  and  $\{j_1, j_2, \dots, j_k\} \in \{1 \dots n - 1\}$ . Note that  $\tau^M$  and all trajectories in  $T$  share the same start point and end point.

Table 2.1 shows how this information is kept in  $\Gamma$ .

Tabel 2.1: Table  $\Gamma$  after inserting  $\mathcal{S}_1$ 

	$v_{start}$	$\mathcal{S}_1$	$v_{end}$
$\tau_1$	1	12	20
$\tau_2$	1		20
$\tau_3$	1	9	20
$\tau_4$	1		20

There are two possibilities of the placement of  $\mathcal{S}_2$ :

Tabel 2.2:  $\mathcal{S}_2$  between  $v_{start}$  and  $\mathcal{S}_1$ 

	$v_{start}$	$\mathcal{S}_2$	$\mathcal{S}_1$	$v_{end}$
$\tau_1$	1	5	12	20
$\tau_2$	1	8		20
$\tau_3$	1	2/8/17	9	20
$\tau_4$	1			20

Tabel 2.3:  $\mathcal{S}_2$  between  $\mathcal{S}_1$  and  $v_{end}$ 

	$v_{start}$	$\mathcal{S}_1$	$\mathcal{S}_2$	$v_{end}$
$\tau_1$	1	12	5	20
$\tau_2$	1		8	20
$\tau_3$	1	9	2/8/17	20
$\tau_4$	1			20

The final placement of table  $\Gamma$  after simplification:

Tabel 2.4: Final  $\Gamma$ 

	$v_{start}$	$\mathcal{S}_2$	$\mathcal{S}_1$	$v_{end}$
$\tau_1$	1	5	12	20
$\tau_2$	1	8		20
$\tau_3$	1	8	9	20
$\tau_4$	1			20



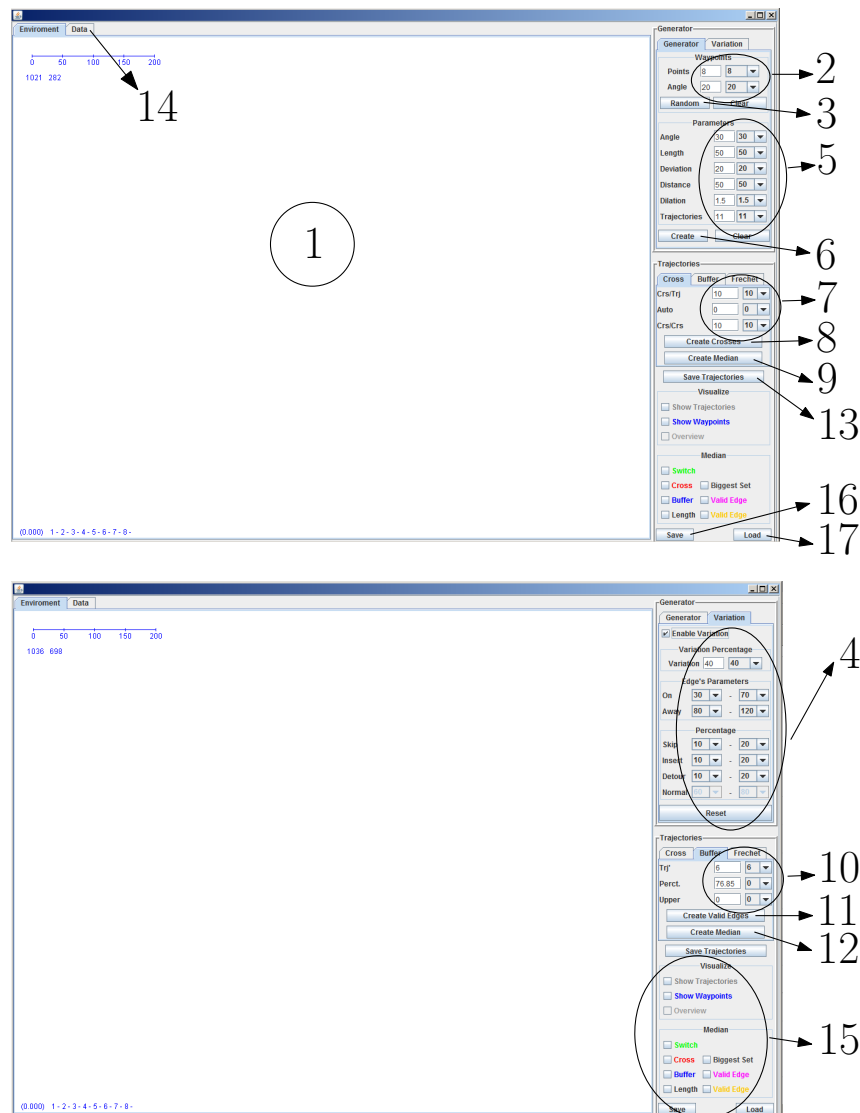
## DAFTAR REFERENSI



# 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 |     }

```