SKRIPSI

PORTING PHP MENJADI JAVA/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/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/PLAY FRAMEWORK (STUDI KASUS KIRI *DASHBOARD SERVER SIDE*)

TOMMY ADHITYA THE

NPM: 2012730031

Bandung, 17 September 2015

Menyetujui,

Pembimbing Utama	Pembimbing Pendamping
Pascal Alfadian, M.Com.	**«pembimbing pendamping/2»**
Ketua Tim Penguji	**Anggota Tim Penguji**
«penguji 1»	**«penguji 2»**

Mengetahui,

Ketua Program Studi

Mariskha Tri Aditia, PDEng

PERNYATAAN

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

PORTING PHP MENJADI JAVA/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 17 September 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 consectetuer.

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, September 2015

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

KIRI (http://kiri.travel/) merupakan situs web untuk membantu pengguna menemukan rute transportasi umum ke tempat tujuannya. Dengan memasukkan lokasi awal serta lokasi tujuan pengguna tersebut, situs web KIRI akan memberikan langkah-langkah (contoh: berjalan sejauh berapa meter, menggunakan angkot, dll) tercepat untuk sampai ke lokasi tujuan. Situs web KIRI telah hadir di berbagai kota seperti Bandung, Depok, Jakarta, Surabaya dan Malang.

KIRI *Dashboard* (https://dev.kiri.travel/bukitjarian/) adalah bagian dari situs web KIRI. KIRI *Dashboard* berfungsi sebagai pengatur proses CRUD (*Create, Read, Update* dan *Delete*) daftar rute yang terdapat dalam *database* situs web KIRI. KIRI *Dashboard Server Side* menggunakan bahasa PHP dalam pembuatannya. Bahasa PHP kurang cocok untuk proyek besar seperti *dashboard*. Salah satu penyebab bahasa PHP kurang cocok adalah karena tidak ada deklarasi dan tipe variabel dalam penggunaan bahasa PHP.

Java merupakan bahasa pemrograman yang umum digunakan oleh banyak orang. Selain umum digunakan, Java juga merupakan bahasa pemrograman yang lebih terstruktur dibandingkan PHP. Adanya deklarasi dan tipe variabel pada Java membuat setiap variabel memiliki kegunaan yang lebih jelas dan mudah dimengerti. Play Framework merupakan *framework* yang membantu implementasi Java dalam pembuatan suatu situs web. Play Framework juga cocok untuk proyek skala besar karena arsitekturnya sudah menggunakan konsep MVC(*Model View Controller*).

Berdasarkan ditemukannya kekurangan-kekurangan pada KIRI *Dashboard Server Side* seperti yang telah dijelaskan, maka solusi untuk mengatasi kekurangan tersebut adalah dibuatnya skripsi ini, yaitu "*Porting* PHP menjadi Java/Play Framework (Studi Kasus KIRI *Dashboard Server Side*)".

1.2 Rumusan Masalah

Berikut adalah susunan permasalahan yang akan dibahas pada penelitian ini:

1. Bagaimana isi kode KIRI *Dashboard Server Side* dan apa saja kekurangan yang ada di dalamnya?

2. Bagaimana cara melakukan *porting* bahasa PHP menjadi Java/Play Framework kode KIRI *Dashboard Server Side* tanpa mengurangi fungsi-fungsi utama yang dimiliki?

1.3 Tujuan

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

1. Mengetahui isi kode KIRI *Dashboard Server Side* dan apa saja kekurangan yang ada di dalamnya.
2. Mengetahui cara melakukan *porting* bahasa PHP menjadi Java/Play Framework kode KIRI *Dashboard Server Side* tanpa mengurangi fungsi-fungsi utama yang dimiliki.

1.4 Batasan Masalah

Skripsi ini dibuat berdasarkan batasan-batasan sebagai berikut:

1. Java/Play Framework yang digunakan selama penulisan skripsi ini adalah versi 2.4.3.
2. *Porting* Kode KIRI *Dashboard Server Side* yang dilakukan adalah berdasarkan versi terbaru dari Github dengan *username*: "pascalalfadian". (https://github.com/pascalalfadian/TirtayasaGH)

1.5 Metode Penelitian

Berikut adalah metode penelitian yang digunakan dalam pembuatan skripsi ini:

1. Melakukan studi literatur mengenai kode KIRI *Dashboard Server Side*, MySQL Spatial Extensions dan Java/Play Framework.
2. Menganalisis teori-teori untuk membangun KIRI *Dashboard Server Side* dalam bahasa Java/Play Framework.
3. Merancang KIRI *Dashboard Server Side* dalam bahasa Java/Play Framework.
4. Melakukan *porting* kode situs web KIRI *Dashboard Server Side* menjadi Java/Play Framework.
5. Melakukan pengujian terhadap fitur-fitur yang sudah dibuat.
6. Menulis dokumen skripsi.

1.6 Sistematika Penulisan

Setiap bab dalam penulisan ini memiliki sistematika yang dijelaskan ke dalam poin-poin sebagai berikut:

1. Bab 1: Pendahuluan.
2. Bab 2: Dasar Teori.
3. Bab 3: Analisis.

BAB 2

THE DESCRIPTION OF A SET OF TRAJECTORIES AND ITS MEDIAN TRAJECTORY

2.1 Set of Trajectories

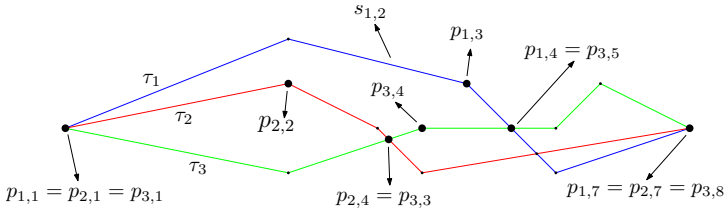
In this thesis, we only consider the spatial component of the trajectory. Therefore, we represent a trajectory as a polygonal line built by a series of points and connected by line segments.

Let $T := \{\tau_1, \tau_2, ..., \tau_m\}$ be the input set of m trajectories for which we want to compute its median trajectory τ^M. We define each trajectory in T as a list of at most n points, $\tau_i := (p_{i,1}, ..., p_{i,k})$ 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 $s_{i,j} := (\overline{p_{i,j}, p_{i,j+1}})$.

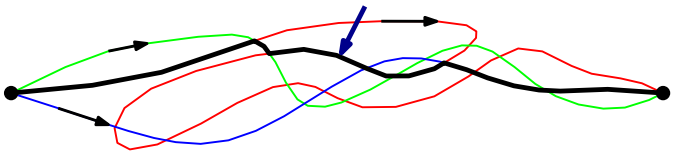
P is the set of all points in T, $P := \{p_{i,j} \mid i \in \{1...m\}, j \in \{1...n\}\}$ and S is the set of all segments in T, $S := \{s_{i,j} \mid i \in \{1...m\}, j \in \{1...n-1\}\}$. All trajectories in T share the same start and end points ($p_{1,1} = p_{2,1} = ... = p_{m,1}$ and $p_{1,k_1} =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

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).

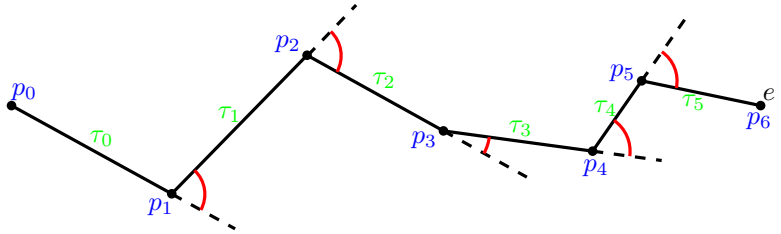
Let n' be the number of points in a trajectory, including their intersection points with other trajectories. In the worst case, $n' = mn^2$. In the rest of this thesis, we define n as a number of points in a trajectory, inclusive with its intersection points with other trajectories. Note that the number 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 τ^M with respect to the input set of trajectories T:

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

Table 2.1 shows how this information is kept in Γ.

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

	v_{start}	$\mathcal{S}_1$	v_{end}
τ_1	1	12	20
τ_2	1		20
τ_3	1	9	20
τ_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}
τ_1	1	5	12	20
τ_2	1	8		20
τ_3	1	2/8/17	9	20
τ_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}
τ_1	1	12	5	20
τ_2	1		8	20
τ_3	1	9	2/8/17	20
τ_4	1			20

The final placement of table Γ after simplification:

Tabel 2.4: Final Γ

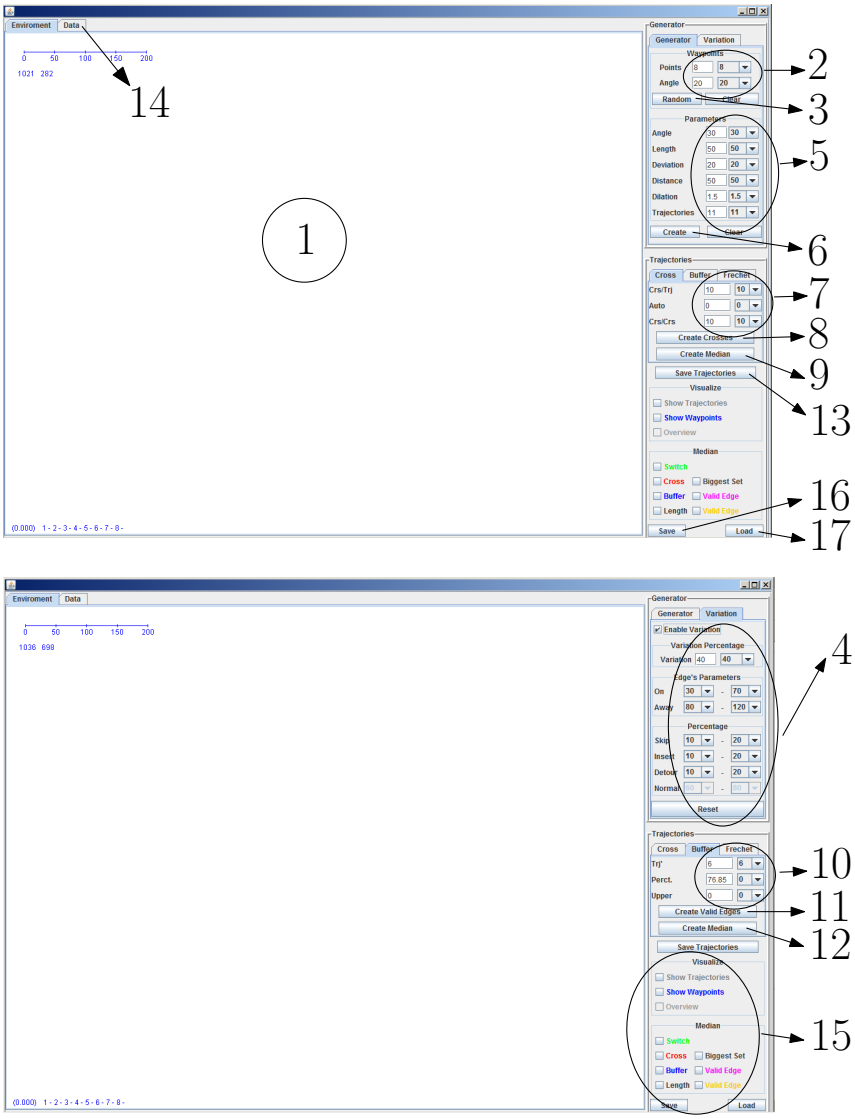
	v_{start}	$\mathcal{S}_2$	$\mathcal{S}_1$	v_{end}
τ_1	1	5	12	20
τ_2	1	8		20
τ_3	1	8	9	20
τ_4	1			20

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