

## CHAPTER 2

### THEORETICAL FOUNDATION

#### 2.1 Theoretical Foundation

##### 2.1.1 Google Maps

Google Maps basically is a free web mapping service application and technology which is presented by Google. It powers many map-based applications such as Google Maps website, Google Ride Finder, Google Transit, and the other third-party websites that embed it via Google Maps API.

Google Maps is developed by Lars and Jens Rasmussen from the company Where2. However, Google took over the company and changed it into the web application Google Maps [2]. Google Maps published firstly at <http://maps.google.com> on February 8, 2005 [3]. It can run on Internet Explorer 6.0+, Mozilla Firefox 2.0+, Safari 3.1+, and Google Chrome [4].

Google Maps has three main types of viewing map which are standard maps, satellite view, and hybrid view. The standard maps is a graphic representation of streets, city names, and landmarks. The satellite view is the view from commercial satellite photos. The hybrid view is a combination of satellite imagery with and overlay of the standard maps. [5]

There are many features that are provided by Google Maps such as searching locations, finding driving direction, showing traffic overlay, etc. Google Maps represents the

world's map clearly. The map is very interactive moreover it can be dragged and zoomed easily. Furthermore, Google Maps can be embedded to any websites via Google Maps API which is an open source. One can use the Google Maps API once they registered to the Google site and getting the Google Maps API key.

Thank You for Signing Up for a Google Maps API Key!

Your key is:

ABQIAAAAVDHMZH5N910COciLu2UyTxT2yXp\_ZAY8\_ufC3CFXhHIE1N  
vwkxQTCzyfhCEHGK8SqPQZgWq7Sf0a6w

This key is good for all URLs consisting of this registered domain (and directory if applicable):

<http://localhost/>

Note: for more information on the API key system, consult <http://code.google.com/apis/maps/faq.html#keyssystem>.

Based on the research on Google Maps, the author decided to use Google Maps for building the application instead of developing a new map by using other technology. Google Maps is better, more efficient and easier to use. In this thesis, the one that will be used is only the standard maps view since it is the most appropriate view for the application.

Although Google Maps already provides the shortest path for finding driving directions and traffic overlay to show traffic information on the map, it is only applicable for certain regions and unfortunately not for Indonesia, especially Jakarta which is the main city for this thesis project. That is why the author did not use those features provided by Google Maps and build the new system to acquire the solution for this thesis.

### 2.1.2 Flex 3

Flex is a free open source framework released by Adobe Systems that can be used for developing Rich Internet Applications (RIAs) or in other words highly interactive and expressive web applications that deploy consistently on all major browsers, desktops, and operating systems. By providing a modern, standards-based language and programming model that supports common design patterns, Flex minimizes the animation metaphor problem that is faced in Flash platform. [6]

The Flex 3 SDK was released by Adobe in February 2008 under the open source Mozilla Public License. The Flex 3 has several improvements, such as supporting the Adobe desktop application for AIR. [7]

Flex is installed on top of a J2EE application server or servlet container and basically uses two main languages which are MXML and ActionScript™ 3. MXML is a declarative XML-based language that is used to describe the graphical user interface and behaviors while ActionScript™ 3, the core language of Flash player, is a powerful object-oriented programming language that is used to create the client logic [6]. The Flex SDK comes with an interactive Flex application debugger and a set component library including form controls, menu controls, media components, layout containers, data components and data binding, formatters and validators, cursor management, state management, effects, and deep linking and browser back and forward button integration [8], [9].

In order to run RIAs which are created by Flex on the browser, Adobe Flash Player 9 or newer is needed. However, to be able to run the RIAs on the desktop, the Adobe AIR™ is required. [6]

To develop Flex 3 applications, developers can use Free Adobe Flex 3 SDK or purchase the Adobe Flex Builder 3 software. Adobe Flex Builder 3 is an Eclipse based IDE (Integrated Development Environment) which includes complete Flex framework, editors, compilers, component libraries, and debuggers [6].

Flex applications will be compiled into an SWF file, the intermediate bytecode format, which runs on Flash Player. Flash Player 9 has two virtual machines called AVM1 and AVM2 for running Flash and Flex content. Flex application always run in AVM2 that compiles content so that it runs significantly faster and more efficiently than previous versions of Flash Player [9].

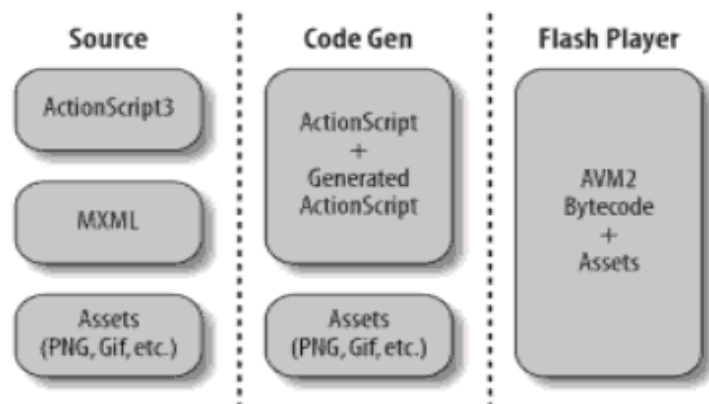


Figure 2.1 Flex Application source-compile-deploy workflow

Flex application is different with a traditional web application which uses HTML as its user interface. If most traditional applications have a data tier, business tier, and

presentation tier, Flex applications introduce client tier which enables clients to offload computation from the server, freeing up network latency and making for responsive and highly interactive user interface. Data tiers consist of database, business tiers consist of business logics to accept requests from a client or presentation tiers, query the data, and return the requested data, and the Flex applications reside on the presentation tiers. Differently with HTML, the Flex application client is stateful which means it can make changes to the view without having to make request to the server. Moreover, it is also responsive so that it can interact with business tier and return the requested data to the client. [9]

The client tier of Flex application communicates with server to send and retrieve data. There are data services in the servers whereby client can make a request by means of Remote Procedure Call (RPC). There are several types of data services. The one that is used in this thesis is the dynamic XML document generated via server-side script which is PHP page. [9] The Flex application uses the HTTPService tag to talk directly to the PHP code on the web server and the PHP code is the one that communicates to the database then returns XML [10].

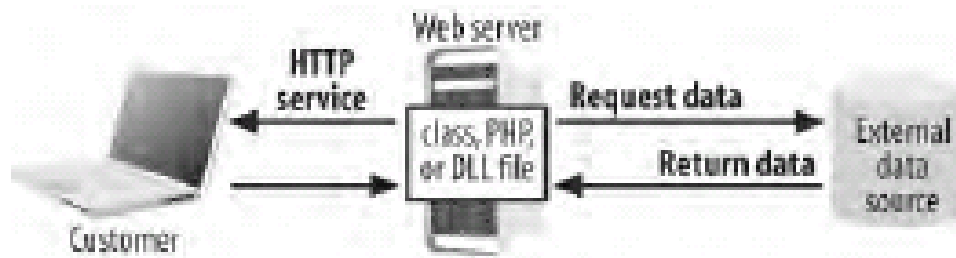


Figure 2.2 Network access from the Flex Application

### 2.1.2.1 MXML

MXML is an XML-based language to create user interface and to view portions of Flex applications that is first introduced by Macromedia in March 2004. Although Adobe Systems gives no official meaning of the acronym, it is likely that the name comes from “Macromedia eXtensible Markup Language”. [9], [11]

MXML has more set of tags and less ambiguous than HTML. MXML not only provides traditional data input controls such as TextBox, ComboBox, TextArea, and CheckBox, but also includes some advanced components to manipulate structured data such as DataGrid and Tree. It also provides navigation component such as TabNavigator and Accordion and containers to specify layout management such as HBox and VBox. Moreover, we can also create our own component by extending the MXML tags. By using MXML, we can also add animation effect to the components to enhance the clues regarding the state transitions. Additionally, MXML also adopts the CSS (Cascading Style Sheet) standard to facilitate our application by using the `<mx:style>` tag.

Besides, MXML offers three different data service components to aggregate data at the client side and to accommodate data access requirement of the service provider. They are the WebService component, the HTTPService component, and the RemoteObject component [11]. As mentioned before, the one that will be used in this thesis is the HTTPService component.

The other advantage of MXML is the easiness of binding data to the user interface codes. It presents a codeless bidirectional data-binding mechanism so that we can bind

user interface controls to data resulting from a service invocation and reversely, we can bind service parameters to values entered in user interface controls [11].

In conclusion, MXML-defined user interfaces which are rendered by Flash player are more interactive and richer than the static page-centric HTML-based web applications.

Below is the example of MXML codes that are used in the thesis related to the Google Maps application:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
<mx:Panel title="Traffic Map" width="1200" height="550">
<mx:VDividedBox width="100%" height="100%">
<mx:HDividedBox width="100%" height="100%">
<maps:Map xmlns:maps="com.google.maps.*" id="map"
mapevent_mapready="onMapReady(event)" width="850" height="450"
key="ABQIAAAAVDHMZH5N910COciLu2UyTxT2yXp_ZAY8_ufC3CFXhHIE1NvwkxQTC
zyfhCEHGK8SqPQZgWq7Sf0a6w" x="0" y="0"/>
.
.
.
</mx:VDividedBox>
</mx:Panel>
<mx:Script> ... </mx:Script>
<mx:HTTPService id="service" url="http://localhost/TrafficSolution/Test_Map/map.php"
result="onResult(event)" />
<mx:HTTPService id="dijkstraservice"
url="http://localhost/TrafficSolution/Test_Map/dijkstraProcess.php" method="post"
result="onDijkstraResult(event)">
  <mx:request xmlns="">
    <startvertex>{ startvertex.text }</startvertex>
    <endvertex>{ endvertex.text }</endvertex>
```

```
<avoidcrowd>{avoidCrowd.selected}</avoidcrowd>  
</mx:request>  
</mx:HTTPService>  
</mx:Application>
```

### 2.1.2.2 ActionScript™ 3

ActionScript is a powerful object-oriented programming language for the Adobe Flash Player run-time environment which enables interactivity, data handling, and much more in Flash content and applications. ActionScript is executed by the part of Flash player called ActionScript Virtual Machine (AVM). The compiler compile ActionScript code into bytecode format, a kind of programming language that is written and understood by computers. The bytecode is embedded in SWF files, which are carried out by the Flash Player. [12]

ActionScript 3 which was released on June 2006 is executed on Flash player 9 or later version. The uniqueness of Flash Player 9 from the previous versions is the two AVM that it has. They are AVM1 for executing ActionScript 1 and 2 codes and AVM2 for executing the ActionScript 3 codes. AVM2 significantly improves the execution performance by up to 10 times faster than the AVM1 [13].

The programming model of ActionScript 3 will be easy to recognize by the developers since it uses a fundamental knowledge of object-oriented programming. It can be used to build complex applications with large data sets and the code will be reusable since it is based on the object-oriented programming.



These are some main features of ActionScript 3:

- The new ActionScript Virtual Machine, called AVM2 which has already been explained before.
- A more modern compiler code base that is more optimized and closed to the ECMAScript (ECMA 262) standard.
- An expanded and improved application programming interface (API), with low-level control of objects and a true object-oriented model.
- A core language based on the upcoming ECMAScript (ECMA-262) edition 4 draft language specification.
- An XML API based on the ECMAScript for XML (E4X) specification (ECMA-357 edition 2). E4X is a language extension to ECMAScript that adds XML as a native data type of the language.
- An event model based on the Document Object Model (DOM) Level 3 Events Specification. [12]

There are three ways to embed ActionScript 3 to the Flex applications:

- By creating a standalone ActionScript class file (.as), for example:

```
package {
    import mx.core.UIComponent;

    public class InfoWindowTabbedComponent extends UIComponent {
        public function InfoWindowTabbedComponent() { ... }
        .
        .
        .
    }
}
```

- By nesting the ActionScript code within MXML tags, for example [12]:

```

<mx:Button id="alertButton" label="Show Alert">
  <mx:Click>
    <![CDATA[
      Alert.show("Example");
    ]]>
  </mx:Click>
</mx:Button>

```

- By placing the ActionScript code within the MXML script, for example [12]:

```

<mx:Script>
  <![CDATA[
    import mx.controls.Alert;

    private function alertButtonClickHandler():void {
      Alert.show("Example");
    }
  ]]>
</mx:Script>

```

As we already knew, generally, MXML is used to create the user interface and ActionScript is used to build the business logic and data models. However, MXML and ActionScript are similar since MXML gets converted to ActionScript during the compilation and MXML can be interpreted in terms of an ActionScript class [12].

### 2.1.2.3 Google Maps API for Flex

Google provides Google Maps API for Flex so that Flex developers can embed Google Maps into their applications. The ActionScript API allows us to manipulate and add content to the maps interactively through variety of services [14]. The services include map events, map controls, map overlays (markers, polylines, polygons, tile overlay, icons, info window, etc.), geocoding and driving direction services, and 3D maps.

Below is the code for setting the Google Maps key on Flex code.

```
<maps:Map xmlns:maps="com.google.maps.*" id="map"
mapevent_mapready="onMapReady(event)" width="850" height="450"
key="ABQIAAAAVDHMZH5N910COciLu2UyTxT2yXp_ZAY8_ufC3CFXhHIE1NvwkxQTC
zyfhCEHGK8SqPQZgWq7Sf0a6w" x="0" y="0"/>
```

### 2.1.3 PHP

PHP, which stands for Hypertext Preprocessor, is a server-side scripting that was designed for building dynamic websites [15]. PHP was invented by Rasmus Lerdorf in 1994. PHP code can be embedded within an HTML page and will be executed each time the page is visited. The web server interprets PHP code and generates HTML or other presentation that the visitor will see. [16] PHP is an open source product so that we can exploit it freely without any charge.

Advantages of using PHP:

- PHP is free since it is an open source product.
- PHP has high performance moreover it is very efficient since it can run on a single inexpensive server.
- PHP can run on many different operating systems such as Linux, Unix, and Microsoft Windows.
- PHP can be integrated to many database systems, for example is MySQL.
- PHP has many useful built-in functions.
- It is easy to learn PHP since the syntax is similar to C or Java.
- It supports object-oriented programming so that the code can be reusable later on.

Basic syntax of PHP:

- Always start and end with PHP tag: `<?php ... ?>`

For further information about PHP, refers to [15], [16].

#### **2.1.4 Code Igniter**

CodeIgniter is an application development framework or a toolkit for developing web application using PHP. It provides a rich set of libraries to help developers in building project faster without having to write the code from scratch. CodeIgniter is free since it is licensed under an Apache/BSD-style open source license. It is written to be compatible with PHP 4 or later version. Additionally, CodeIgniter has good and fast performance and it is light weight.

CodeIgniter was designed to achieve several goals as follows:

- Dynamic Instantiation → components and routines are loaded and executed only when requested rather than globally so that it will be very light weight.
- Loose Coupling → components are less dependent on each other so that they will be more reusable and flexible.
- Component Singularity → each class and its functions are highly autonomous in order to allow maximum usefulness. [17]

CodeIgniter uses MVC (Model-View-Controller) approach.

- The Model represents data structures, including classes that manipulate the database.
- The View represents pages that will be presented to the user.
- The Controller acts as an intermediary between the Model, the View, and other resources to process the HTTP request and generate a web page.

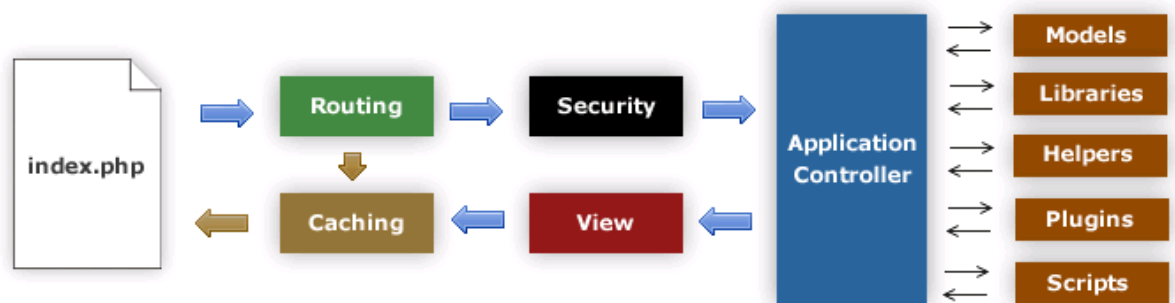


Figure 2.3 CodeIgniter Application Flow Chart

1. The index.php serves as the front controller, initializing the base resources needed to run CodeIgniter.
2. The Router examines the HTTP request to determine what should be done with it.
3. If a cache file exists, it is sent directly to the browser, bypassing the normal system execution.
4. Security. Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.
5. The Controller loads the model, core libraries, plugins, helpers, and any other resources needed to process the specific request.

6. The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served. [17]

### **2.1.5 MySQL**

MySQL, which stands for My Structured Query Language, is a relational database management system (RDBMS) that is very fast and robust. It allows us to store, search, sort, and retrieve data efficiently. MySQL uses the Structured Query Language (SQL) which is the standard database language worldwide so that it will be easy to learn and configure MySQL.

MySQL is a multiuser and multithread server which means that multiple users can access the data concurrently through the MySQL server. Additionally, MySQL is faster and it is an open source. We only need a MySQL license when we need to distribute our application for commercial use. [16]

### **2.1.6 Graph**

Graph is a data structure that consists of a collection of nodes called vertices and a collection of line segments connecting pairs of vertices called edges. In graphs, each node may have multiple predecessors as well as multiple successors. There are two types of graph which are undirected graph and directed graph. [18]

- Undirected graph  $\rightarrow$  there is no direction or arrow head on any edges.
- Directed graph or diagraph  $\rightarrow$  each edge has direction or arrow head called arcs.

Sometimes, the graph can also have weight or cost that is called network [19]. The one that will be used in this thesis is the weighted undirected graph for representing the streets and the street connections. For later development, the weighted directed graph can be used to precisely describe the street connections.

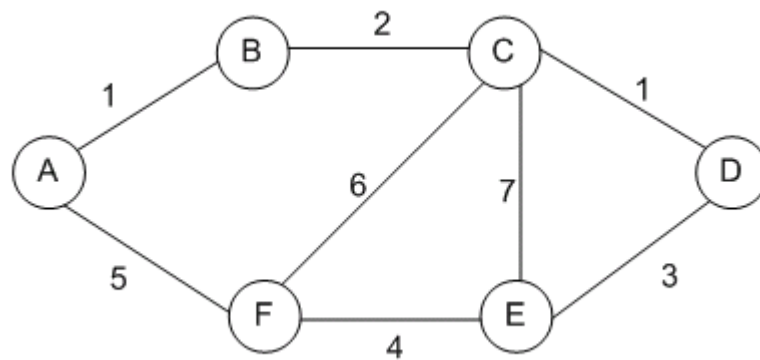


Figure 2.4 Illustration of the weighted undirected graph

The graph consists of the streets that are stored in table vertex in the database and street connections that are stored in the table edge in the database. To attain the street or vertex, we need to define in the database the vertex ID, vertex latitude from the map, vertex longitude from the map, and the street name. In addition, to get the street connection or edge, we need to define the edge ID, start vertex ID, end vertex ID, and the distance between start vertex and end vertex. The start vertex and the end vertex are the foreign keys from the vertex ID in vertex table.

The shortest distance between the start vertex and the end vertex can be obtained since we know each vertex latitude and longitude. We can use the Haversine formula to calculate the distance. For further explanation about the Haversine formula, refers to [20]. Below is the PHP code example of Haversine formula.

```
class Haversine{
    /* start vertex latitude = $lat1; start vertex longitude = $lon1; end vertex latitude = $lat2; end
    vertex longitude = $lon2 */

    function distance_haversine($lat1, $lon1, $lat2, $lon2) {

        $earth_radius = 3960.00; // in miles

        $delta_lat = $lat2 - $lat1;

        $delta_lon = $lon2 - $lon1;

        $alpha  = $delta_lat/2;

        $beta   = $delta_lon/2;

        $a      = sin(deg2rad($alpha)) * sin(deg2rad($alpha)) + cos(deg2rad($lat1)) *
        cos(deg2rad($lat2)) * sin(deg2rad($beta)) * sin(deg2rad($beta)) ;

        $c      = asin(min(1, sqrt($a)));

        $distance = 2*$earth_radius * $c;

        $distance = $distance * 1.609344; //convert to kilometers

        $distance = round($distance, 4);

        return $distance;

    }
}
```

To represent the graph, we need a two-dimensional array called adjacency matrix.

Below is the code example to represent the graph in adjacency matrix.



```

// Size of the matrix is the amount of all vertices
$matrixWidth = count($vertex_id);
for($i=0; $i<count($edge_id); $i++)
{
    /* $points is an array in the following format: (start vertex ID, end vertex ID, distance between
    them or the graph weight)*/
    $points[$i] = array($start_vertex[$i]-1, $end_vertex[$i]-1, $distance[$i]);
}
$ourMap = array();

// Read in the points and push them into the $ourMap adjacency matrix
for ($i=0; $i<count($points); $i++) {
    $x = $points[$i][0];
    $y = $points[$i][1];
    $c = $points[$i][2];
    $ourMap[$x][$y] = $c;
    $ourMap[$y][$x] = $c; // Indicating the undirected graph
}

```

Furthermore, there are several of graphs operations that are adapted to this thesis such as:

- Add vertex → we can insert new vertex or street.
- Delete vertex → we can delete the vertex or the street. This is including deleting all connections or edges related to the vertex.
- Add edge → we can add connections between vertices.
- Delete edge → we can delete connections between vertices.

### 2.1.7 Dijkstra's Algorithm

Dijkstra's algorithm is a best-first graph search algorithm invented by Edsger Dijkstra in 1959 that can be applied for finding shortest path between two vertices in graph which has nonnegative edge weights [18], [19]. Dijkstra's algorithm is one example of greedy algorithm. Greedy algorithm basically solves a problem in stages by doing what appears to be the best thing at each stage [19].

Dijkstra's algorithm will be used in the thesis for finding shortest path between two streets. There is also 'avoid crowd' function that can be selected in finding the shortest path. If it is selected, it means that the 'crowded' vertices except the start and end vertices will be removed or ignored from the graph.

For the problem in this thesis, Dijkstra's algorithm is the most appropriate solution since it is simple to code and has a quite good running time which is  $O(n^2)$  [19]. However, there are several search algorithms compared to Dijkstra's algorithm such as:

- Bellman-Ford algorithm → it also can be used to find shortest path if the edge weights may be negative [19]. Since there will be no negative edge weights in this problem, it is no need to use this algorithm.
- Prim's and Kruskal's Minimum Spanning Tree algorithm → it can be used to return the total minimum weight from the graph. However, the output is a spanning tree which means it returns all the vertices, and this thesis does not require that too.

- Floyd-Warshall algorithm → the algorithm to find shortest paths from all pairs of vertices in the graph [21]. What is needed in this thesis is only a single pair shortest path solution. It is too expensive to use this algorithm.
- A\* algorithm → it is similar to Dijkstra's algorithm and moreover the cost is cheaper than Dijkstra's algorithm since it uses the heuristic function to estimate distance from each current node to the goal node [22]. However, it is more complicated to code considering the limited time in developing this thesis and furthermore there is no heuristic that is used to estimate the distance.

This is the flow of Dijkstra's algorithm:

- First, every node is assigned to a distance value. The start node is set to zero and the other nodes will be set to infinity distance.
- Mark all nodes as unknown or unvisited.
- Set the start node as the current node.
- For the current node, consider all its unvisited neighbors and calculate their distance from the start node. And then, replace the distance if it is smaller than the previous recorded distance.
- All the considered neighbors will be marked as known or visited. A visited node will not be checked again.
- Set the unvisited node with the smallest distance from the start node as the current node and continue looping the step 4 until all nodes are visited.
- Return the shortest path. [23]

Below is the example of Dijkstra's algorithm code in PHP.

```

function Dijkstra(&$ourMap, $infiniteDistance) {
    $this -> infiniteDistance = $infiniteDistance;
    $this -> map = &$ourMap;
    $this -> bestPath = 0;
}

function findShortestPath($start,$to = null) {
    $this -> startnode = $start;
    foreach (array_keys($this->map) as $i) {
        if ($i == $this -> startnode) {
            $this -> visited[$i] = true;
            $this -> distance[$i] = 0;
        } else {
            $this -> visited[$i] = false;
            $this -> distance[$i] = isset($this -> map[$this -> startnode][$i])
                ? $this -> map[$this -> startnode][$i]
                : $this -> infiniteDistance;
        }
        $this -> previousNode[$i] = $this -> startnode;
    }
    $maxTries = count($this->map);
    $tries = 0;
    while (in_array(false,$this -> visited,true) && $tries <= $maxTries) {
        $this -> bestPath = $this->findBestPath($this->distance,array_keys($this ->
visited,false,true));
        if($to !== null && $this -> bestPath === $to) {
            break;
        }
        $this -> updateDistanceAndPrevious($this -> bestPath);
        $this -> visited[$this -> bestPath] = true;
    }
}

```

```
        $tries++;  
    }  
}
```

## 2.2 Theoretical Framework

The research method of finding the solution to the problem is by doing literature study from various sources, such as from the Internet and also text books. Moreover, it is also helped by consulting the problem with friends and thesis supervisor. The development planning of the thesis can be summarized as follows:

- Analyzing the existing problem.
- Gathering information regarding the data and technology for developing the solution.
- After analyzing all the information, the author chose the several appropriate technologies in building the solution considering the time development and the resources available.
- In order to decide the most suitable technology, the author created some prototypes to test the solution. By creating and testing the prototypes, the author finally could decide the best solution regarding the thesis problem.
- Next step is designing and implementing the exact solution.
- And finally, testing and evaluating the solution.