Implementation of Various Heuristic Information in Planning a Trip by KRL Commuter Line

Audry Nyonata, 13515087
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
audrynyonata@gmail.com, 13515087@std.stei.itb.ac.id

***Abstract*—Commuting has long been a common habit, and is an inevitable part in some people's everyday routines, especially for those who have their daily activities (e.g. work, study) done in a big city, for instance, Jakarta. KRL Commuter Line is a commuter rail system which offers transportation services in the Jakarta metropolitan area, by utilizing electric multiple unit (EMU) to carry its passengers from/to numerous stations in Jakarta and its surrounding territories comprising the cities of Bogor, Depok, Tangerang, and Bekasi. This paper discuss multiple factors, such as the fare of the trip, also duration, distance, and number of changing trains, that should be taken into account while planning a trip by KRL Commuter Line in order to get the most optimal route possible. This paper also contains analytical result of the route-planning experiment which is run in Java environment by implementing the various heuristic information as bounds for the Branch and Bound algorithm [1].**

Keywords—route-planning, optimal, heuristic, branch&bound.

I. INTRODUCTION

Aside from being the capital, as well the administrative and political center of Indonesia, Jakarta is also home to most economic activities performed in the country. According to a commuter survey conducted by BPS-Statistics of DKI Jakarta Province, in 2014, Jakarta's daytime population reached the number of 11,2 million, and stood at only 10 million during the night [2]. The difference between those two demonstrates the number of people who commute from out of the city to life their live in Jakarta, whether it's studying, working, or just enrolling in a particular course.

The city's dense population and its relatively high cost of living caused some of its citizens to refuse becoming permanent residents of Jakarta, they prefer to live in the outskirts of Jakarta, those are Bogor, Depok, Tangerang, and Bekasi. People would travel regulary from time to time to their workplace and return home on the same day.

Despite the fact that private vehicles such as cars and motorcycles are the most popular transportation used for commuting, public transportation like bus and trains are still crucial for the well-being of lots of people. Public transportation is also the key to solve Jakarta's renowned problem, specifically its huge traffic. Thereby, the problem of public transportation has always been an issue worth discussing.

About 950.000 passengers in average benefited from the KRL Commuter Line service every day [3]. KRL Commuter Line is operated by PT KAI Commuter Jabodetabek (KCJ), a subsidiary of PT Kereta Api Indonesia, responsible for handling the KRL Commuter Line services in ticket revenues, rolling stock maintenance, and station management. While all operational matters like scheduling, stations, and infrastructures remained under PT KAI's responsibility [4].

Fig. 1. PT KAI Commuter Jabodetabek Logo

Back in 2008, Divisi Anguktan Perkotaan Jabotabek was created under PT Kereta Api Indonesia. Later, this division form PT KCJ in accordance with Inpres No. 5 tahun 2008 and Surat Menteri Negara BUMN No. S-653/MBU/2008, 12 Agustus 2008. The current form of electric train we know today is the result of modernization of the commuter railway system began in 2011, in which there was done a simplification of the existing lines and reduce them into five main line, removing express KRL services, providing units specified for women, and also some renovations of rail tracks and stations. PT KCJ now aims for 1,2 million daily passengers in 2019 [5]. To reach that goal, it is necessary to set up a study about their passengers' commuting behavior and pattern.

In addition, we could try providing a way to improve the passengers' experience of using KRL Commuter Line service. An optimal commuting trip is shown by the cheap fare, the short travel time, the short distance compared to the displacement, or the least number of changing lines needed to travel from one station to get to the destination. These information needs to be evaluated carefully, or else the different priority may lead to non-optimal results.

This paper demonstrates the implementation of branch and bound algorithm to get the best possible solution, using the information which has been assigned to four categories i.e. fare, duration, distance, and number of line changes, as bounds. The information about KRL Commuter Line service such as schedules and distance between stations used in this paper are real data that can be accesed through their official website.

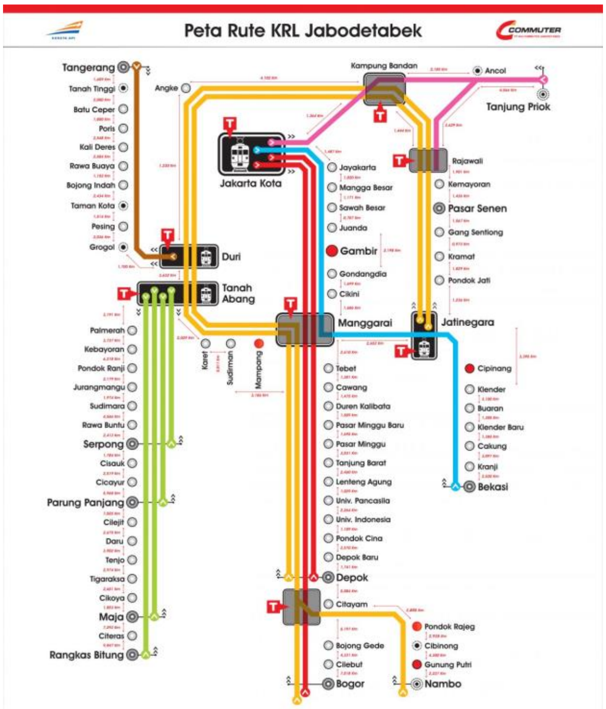


Fig. 2. KRL Commuter Line operation area showing 13 different lines, main lines are described in following: red (central line), yellow (loop line), blue (bekasi line), green (rangkasbitung line), brown (tangerang line), pink (tanjung priok line).

II. Background Concepts & Theories

A. Graph

A graph is a representation of objects and its connections with other objects. Graph is usually written as an ordered pair G = (V,E), consisting of a set of V vertices (vertex) or nodes together with a set of E edges. V is usually taken to be nonempty [6]. An edge shows that a pair of vertices are related in some sense, it could be directed or undirected. A vertex might have an edge connecting to itself, moreover it might exists in a graph and doesn't belong to any edge of the graph.

A weighted graph is a graph in which each edge is given a numerical value. Such number might represent costs, distances, etc., depending on what problem the graph is illustrated for.

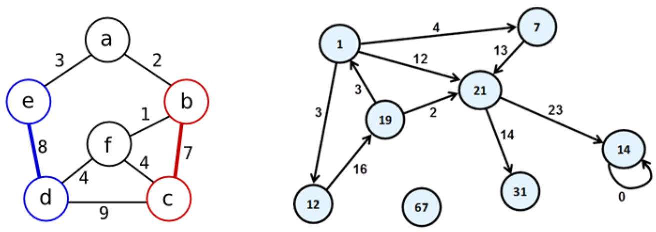


Fig. 3. Undirected (left) and directed (right) weighted graphs.

B. State-space Tree

A tree is a specialized graph, in which all the existing edges are undirected and there are no cyclic connection between any pair or the nodes.

A dynamic tree, unlike a static tree, is generated alongside during the process of searching for solution. Repeatedly, a node created is evaluated, if it doesn't lean towards the goal, then the process continues to creation of other nodes [1].

Some terms that are worth mentioning while discussing the state-space tree are,

1) Problem state: the nodes in the dynamic tree fulfill the constraints.

2) Solution state: problem's solution is stated in one or more state.

3) Goal state: solution state which couldn't be expanded any further

4) Solution space: a set of all created solution state

5) State space: all the nodes in the dynamic tree

6) State-space tree: the dynamic tree

7) Initial state: the root node of the state-space tree

Several operations transforms one problem from one state to another.

C. Branch and Bound

The branch and bound (B&B) algorithm is a systematic method to find a solution. B&B generates its solution space using breadth-first scheme, organized in a state-space tree. To shorten the search process, each node is given a cost, only the nodes with least cost search is expanded and evaluated. The cost of node i, c(i), stated the estimation of lowest cost from i to goal node, in other words c(i) serves the lower bound of search cost through i [1]. A bound function is used to evaluate a node, if a node doesn't seem to reach towards the goal, it will not be expanded further.

Heuristic function is applied in order to calculate the estimation, it is written,

$$c(i) = f(i) + g(i)$$

where c(i) is the cost for node i, f(i) is the cost to reach node i from root, and g(i) is the cost to reach goal node from i. The cost of each node is sorted to obtain the node with minimum cost.

In general, the steps of B&B algorithm are [1],

1) Add the root node to a queue Q. If root is the solution that we want (goal node), then the solution has been found. Stop.

2) If queue Q is empty, then the solution doesn't exists.

3) While queue Q is not empty, choose a node i from Q which has least cost.

4) If node i is the solution that we want, then the solution is found. Stop. If node i is not the solution, then expand its child. Return to step 2.

5) For each child node j generated from parent node i, calculate c(j) and add all of them to queue Q.

6) Return to step 2.

There are various mathematical problems, for example, the N-queens problem, 15-puzzle problem, and travelling salesperson problem (TSP), and other optimalization problem that are able to be solved by the B&B algorithm.

Another famous path-finding algorithm is the A* (A star) algorithm. Similiarly, as in A*, look for a bound which is guaranteed lower than the true cost. Then, search the branching tree, there are several methods to chose from, like depth-first, best-first, etc. If heuristic equals the sum of cost and bound, and the tree search method used is best-first, then B&B is the same as A*. But bounds often are much more sophisticated, for example using mathematical programming optimizations [7].

D. KRL Commuter Line Trip Estimation

The train fares between the origin and destination is calculated progressively based on the distance. But shorter distance doesn't necessarily mean cheaper fare. For example, if we traveled only 1 kilometer away we would need to pay the same amount of money as if we traveled for 25 kilometers. The reason is the fare counting policy that regulate for the first 25 kilometers, passengers only need to pay IDR 3000, and for every next 10 kilometers an amout of IDR 1000 is charged in addition [5]. The distance is counted only once, that is when they get out of the the station. So, even if we needed to get off the train to change lines for many times, it wouldn't directly effect the fare charged to us as long the distance didn't increased much. Same thing goes for the elapsed traveling time.

The duration needed to get to one station from another can be obtained from the train schedule. The timetable tells us what time which train arrives in which station. By subtracting the time for two adjacent station, we get the duration between each stops. Normally, the trip will takes longer time on the office start and end hours, due to the increasing amount of passengers, leading to increasing number of train queuing to pass the stations in and out.

An extra amount of time is also needed when we make a transit, because the train which we would continue our trip on may arrives at different schedule than our originating train, it could even be late.

However, the timetable provided in the KRL Commuter Line website is rather difficult to understand, since it doesn't differentiate the line enough. For instance, the schedule for Jakata Kota – Bogor is displayed on the same table along with Jatinegara – Bogor, even though the two lines serve many different stations from one another, resulting in a number of empty cells in the spreadsheet. Another example is there is schedule for Bekasi – Jakarta Kota via Manggarai, along with Bekasi – Jakarta Kota via Pasar Senen, yet the line it is not listed in the route map. Therefore, reading the KRL Commuter Line schedule must be done patiently and thorough.

III. Methodology

This paper evaluates and does some experiments with assistance of writing a program code. This program is written in Java programming language, utilizing the priority queue library. There are four classes made in order to solve the KRL Commuter Line trip optimalization problem, i.e. Main, Stasiun (stations), Jalur (lines), and Simpul (nodes).

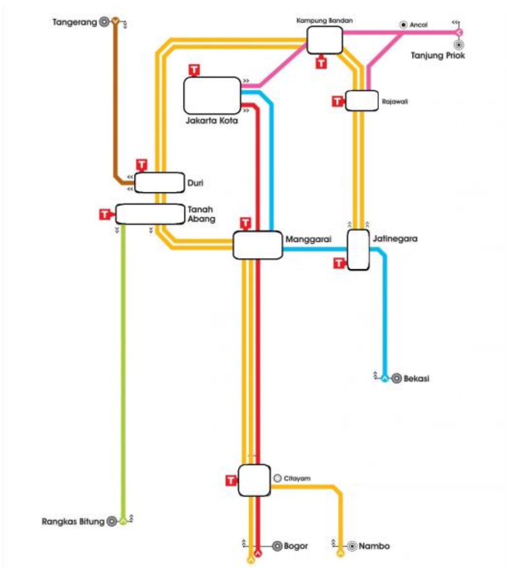


Fig. 4. Simplified KRL Commuter Line route-map used in the program

The 80 stations, class Stasiun, under the KRL Commuter Line operating area (active and inactive) are listed in an array. There is a counter for the number of stations created in the runtime process. As for the train lines, class Jalur, the 13 lines in Fig.2 is simplified into 8 lines that really show differences in the route-planning process. The 8 lines are also listed in a separate array with a specified counter assigned to count their number.

TABLE I. Snapshot of Data Store

```
public static void init() {
  Jalur RedLine = new Jalur("RedLine");
  Jalur YellowLineBogor = new Jalur("YellowLineBogor");
  …

  Stasiun ManggaBesar = new Stasiun("ManggaBesar",2);
  ManggaBesar.add("RedLine");
  ManggaBesar.add("BlueLine");
  …
  Stasiun Serpong = new Stasiun("Serpong",1);
  Serpong.add("GreenLine");
  …

  GreenLine.add(Serpong,350,1784);
  GreenLine.add(RawaBuntu,240,2413);
  GreenLine.add(Sudimara,420,4566);
  …
}
```

The class Stasiun is identified by its name (string), each has their own array of jalurId (string). The class Jalur is identified by its id (string), each has their own array of route (Stasiun), array of duration in seconds, and array of distaces in meters. For index i, the value of arrays[i] corresponds to the Stasiun in route[i]. To mimic the real situation and achieve more accurate result, additional 5 minutes is added for every line changes made. These are the main class to store data and information needed for the program.

In the graph illustration, stations are represented as the nodes, while the accessibility of a station and other particular station is shown by the edges, meaning there is rail track connecting the two stations, and train lines as the paths in te graph. The data structure used to represent this graph is by incidency list, in which every vertex (stations) listed the path (line) that has incidencies with it. Also, every line describe which stations are connected to which, along with the value of duration and distances needed to get there.

TABLE II. INCIDENCY LIST

Mangga Besar	RedLine BlueLine	Serpong	Green Line	…	…
listJalur	Red Line	…	Blue Line	Green Line	…
Red Line	Bogor 0 0	Cilebut 660 7518	…	Mangga Besar 180 1171	…
Green Line	…	Maja 660 7293	…	Serpong 350 1784	Rawa Buntu 240 2413
…	…	…	…	…	…

As for the state-space tree, a priority queue with generic class Simpul is used, in which each Simpul is identified by its Stasiun. Class Simpul keeps track of its duration/elapsed time, elapsed distance, fare, and array of visited stations so far. Fare are calculated from the current distance, and from the array of visited stations it is possible to get the number of changing lines that has been done to get to the current node. By utilizing various queue comparator, it is possible to sort the queue the way we want, whether to choose the cheapest fare as a priority, shorter duration, shorter distance, least number of changing lines, and most stations past so far.

TABLE III. SNAPSHOT OF CUSTOM COMPARATOR

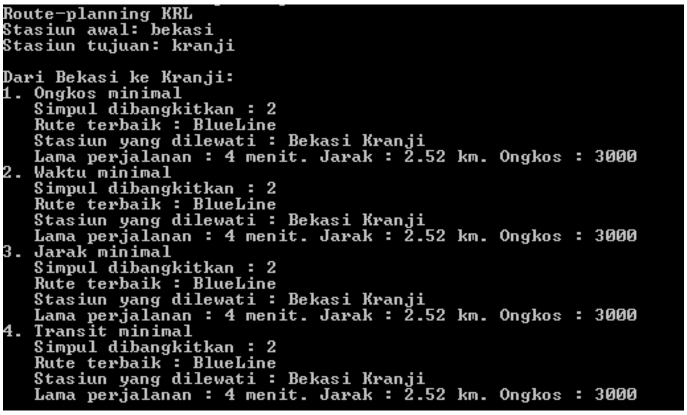
```
Comparator<Simpul> comparator = new Comparator<Simpul>() {
  public int compare(Simpul S1, Simpul S2) {
    if (…) {
      …
    } else { //minimal transit
      if (S1.banyakTransit()==S2.banyakTransit()) {
        if (S1.jarak > S2.jarak) {
          return 1;
        } else if (S1.jarak == S2.jarak) {
          return 0;
        } else {
          return -1;
        }
    } else if (S1.banyakTransit() > S2.banyakTransit()) {
      return 1;
    } else {
      return -1;
    }
};

PriorityQueue<Simpul> queue = new PriorityQueue<Simpul>(3628800,comparator);
```

The bound in this program is to check whether the evaluated node equals the goal node or not, also with assumptions that in one trip there will be no duplicate line since it is very unlikely to be effective, (except for yellow and pink line), the node that has duplicate lines in their route will not be expanded. This leads to much shorter and saves a lot of time in the search process.

IV. RESULTS & ANALYSIS

A. Test Case 1 :Bekasi – Kranji



Since both of Bekasi and Kranji St. have only one train lines passing through, which is BlueLine, only 2 nodes are generated. To reach Kranji from Bekasi, there are no other stations past in between, also no changing lines. From the data, we know that the trip takes 4 minutes on train plus 0 minutes for waiting/transit. The distance is the same as it is listed on the route map, from there, the fare was calculated. So the test results in success optimal result.

Using Google Maps, we can see that the results doesn't really have much difference. Therefore, it is safe to assume that this program code with implemented branch and bound algorithm is reliable.

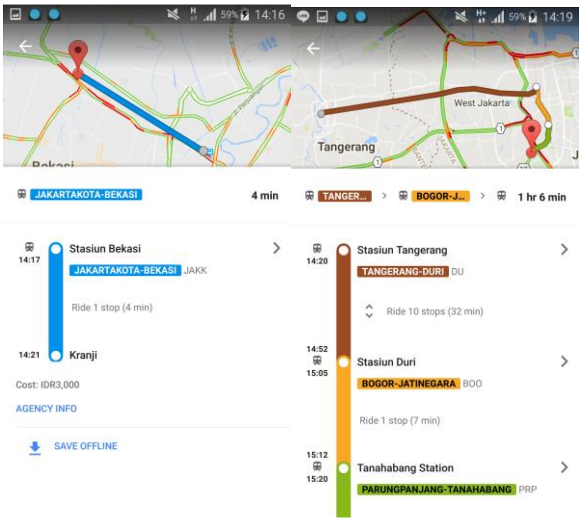


Fig. 5. Google Maps reccomendations for test case 1 (left), which shows the same travel time with test results, and test case 2 (right) with time difference of 10 minutes.

B. *Test Case 2 :Tangerang – Palmerah*

```
Route-planning KRL
Stasiun awal: tangerang
Stasiun tujuan: palmerah

Dari Tangerang ke Palmerah:
1. Ongkos minimal
   Simpul dibangkitkan : 323
   Rute terbaik : BrownLine transit di Duri, sambung YellowLineNambo t
ransit di TanahAbang, sambung GreenLine
   Stasiun yang dilewati : Tangerang TanahTinggi BatuCeper Poris KaliD
eres RawaBuaya BojongIndah TamanKota Pesing Grogol Duri TanahAbang Pal
merah
   Lama perjalanan : 56 menit. Jarak : 26.12 km. Ongkos : 4000
2. Waktu minimal
   Simpul dibangkitkan : 71
   Rute terbaik : BrownLine transit di Duri, sambung YellowLineBogor t
ransit di TanahAbang, sambung GreenLine
   Stasiun yang dilewati : Tangerang TanahTinggi BatuCeper Poris KaliD
eres RawaBuaya BojongIndah TamanKota Pesing Grogol Duri TanahAbang Pal
merah
   Lama perjalanan : 56 menit. Jarak : 26.12 km. Ongkos : 4000
3. Jarak minimal
   Simpul dibangkitkan : 75
   Rute terbaik : BrownLine transit di Duri, sambung YellowLineBogor t
ransit di TanahAbang, sambung GreenLine
   Stasiun yang dilewati : Tangerang TanahTinggi BatuCeper Poris KaliD
eres RawaBuaya BojongIndah TamanKota Pesing Grogol Duri TanahAbang Pal
merah
   Lama perjalanan : 56 menit. Jarak : 26.12 km. Ongkos : 4000
4. Transit minimal
   Simpul dibangkitkan : 190
   Rute terbaik : BrownLine transit di Duri, sambung YellowLineBogor t
ransit di TanahAbang, sambung GreenLine
   Stasiun yang dilewati : Tangerang TanahTinggi BatuCeper Poris KaliD
eres RawaBuaya BojongIndah TamanKota Pesing Grogol Duri TanahAbang Pal
merah
   Lama perjalanan : 56 menit. Jarak : 26.12 km. Ongkos : 4000
```

Tangerang St. has only one train line passing through, which is BrownLine, meanwhile, Palmerah St. can only be reached through GreenLine. In between those two stations, there are Duri St. and Tanah Abang St. which is past through numerous train lines, providing transit and line changing access to other stations in different region of the city.

The algorithm success to achieve optimal result, which requires using three different train lines, Tangerang(brown) -> Duri(yellow) -> Tanah Abang(green) -> Palmerah, with estimated time of 56 minutes, distance of 26 km, from there the fare is calculated IDR 3000 (25 km) + IDR 1000 (1 km) = IDR 4000.

The difference nodes generated by each different priorities are the effects of using the comparator, which has been customized to evaluate all nodes with same minimum values respectively, by fare, duration, distance, and number of changing lines, which will be explained in the next test case.

C. *Test Case 3 :Bogor – Jatinegara*

```
Route-planning KRL
Stasiun awal: bogor
Stasiun tujuan: jatinegara

Dari Bogor ke Jatinegara:
1. Ongkos minimal
   Simpul dibangkitkan : 4300
   Rute terbaik : RedLine transit di Manggarai, sambung BlueLine
   Stasiun yang dilewati : Bogor Cilebut BojongGede Citayam Depok Depo
kBaru PondokCina UniversitasIndonesia UniversitasPancasila LentengAgun
g TanjungBarat PasarMinggu PasarMingguBaru DurenKalibata Cawang Tebet
Manggarai Jatinegara
   Lama perjalanan : 88 menit. Jarak : 47.576 km. Ongkos : 6000
2. Waktu minimal
   Simpul dibangkitkan : 4041
   Rute terbaik : YellowLineBogor transit di Manggarai, sambung BlueLi
ne
   Stasiun yang dilewati : Bogor Cilebut BojongGede Citayam Depok Depo
kBaru PondokCina UniversitasIndonesia UniversitasPancasila LentengAgun
g TanjungBarat PasarMinggu PasarMingguBaru DurenKalibata Cawang Tebet
Manggarai Jatinegara
   Lama perjalanan : 88 menit. Jarak : 47.576 km. Ongkos : 6000
3. Jarak minimal
   Simpul dibangkitkan : 4600
   Rute terbaik : YellowLineBogor transit di Manggarai, sambung BlueLi
ne
   Stasiun yang dilewati : Bogor Cilebut BojongGede Citayam Depok Depo
kBaru PondokCina UniversitasIndonesia UniversitasPancasila LentengAgun
g TanjungBarat PasarMinggu PasarMingguBaru DurenKalibata Cawang Tebet
Manggarai Jatinegara
   Lama perjalanan : 88 menit. Jarak : 47.576 km. Ongkos : 6000
4. Transit minimal
   Simpul dibangkitkan : 146
   Rute terbaik : YellowLineBogor
   Stasiun yang dilewati : Bogor Cilebut BojongGede Citayam Depok Depo
kBaru PondokCina UniversitasIndonesia UniversitasPancasila LentengAgun
g TanjungBarat PasarMinggu PasarMingguBaru DurenKalibata Cawang Tebet
Manggarai Sudirman Karet TanahAbang Duri Angke KampungBandan Rajawali
Kemayoran PasarSenen GangSentiong Kramat PondokJati Jatinegara
   Lama perjalanan : 130 menit. Jarak : 70.3 km. Ongkos : 8000

C:\Users\user\Desktop\krl>
```

Bogor St. has two lines passing through it, which is RedLine and YellowLineBogor.

If we choose to prioritize minimum fare, the algorithm evaluates all possible nodes which have same fare value, therefore many nodes will be generated since the cost is all the same for the first 25 km. If we choose to prioritize the minimum travel time, the algorithm evaluates all possible nodes which have same duration value, in this program it may seem useless, but in the real case, every train has different duration for the same route at different times of the day.

For the minimum distance, the algorithm evaluates a station and its adjacent stations in RedLine, or YellowLine for the minimum distance value. For each node, it reevaluates all the line passing through, in order to find possible transit and changing lines to be made. This is the reason that the algorithm needs to generate a great number of nodes, to assure the accuracy and the quality of the route.

From here we get the result that is optimal, which is Bogor (yellow)-> Manggarai (blue)-> Jatinegara with only 47 km distance, and two changing lines made.

We see different results if we want the result with minimum changing lines number, this leads to the least number of nodes generated since the algorithm evaluates and only expand the nodes that have as close as possible to one train line. The result is Bogor(yellow) -> Jatinegara, with no chaning lines, but sacrificing time, distance, and therefore the fare.

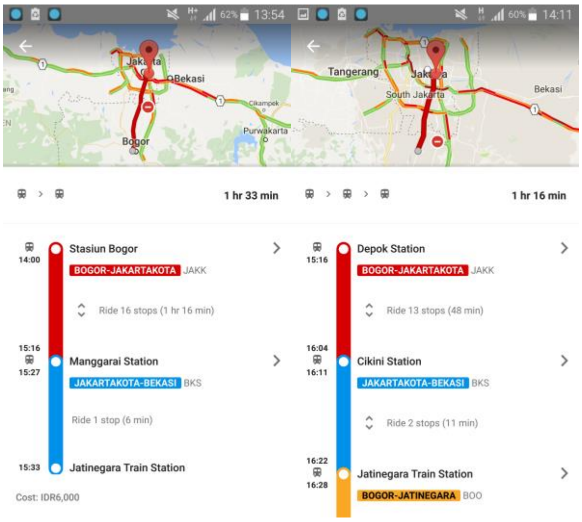


Fig. 6. Google Maps reccomendations for test case 3 (left) and test case 4 (right). Notice the RedLine used is equal with the YellowLine, the different results are caused by the arrival time of the RedLine is closer to the time when the test was made.

D. Test Case 4 :Depok – Pondok Jati

```
Route-planning KRL
Stasiun awal: depok
Stasiun tujuan: pondok jati

Dari Depok ke PondokJati:
1. Ongkos minimal
   Simpul dibangkitkan : 7046
   Rute terbaik : YellowLineNambo transit di Manggarai, sambung BlueLi
ne transit di Jatinegara, sambung YellowLineBogor
   Stasiun yang dilewati : Depok DepokBaru PondokCina UniversitasIndon
esia UniversitasPancasila LentengAgung TanjungBarat PasarMinggu PasarM
ingguBaru DurenKalibata Cawang Tebet Manggarai Jatinegara PondokJati
   Lama perjalanan : 64 menit. Jarak : 26.682 km. Ongkos : 4000
2. Waktu minimal
   Simpul dibangkitkan : 5752
   Rute terbaik : YellowLineNambo transit di Manggarai, sambung BlueLi
ne transit di Jatinegara, sambung YellowLineBogor
   Stasiun yang dilewati : Depok DepokBaru PondokCina UniversitasIndon
esia UniversitasPancasila LentengAgung TanjungBarat PasarMinggu PasarM
ingguBaru DurenKalibata Cawang Tebet Manggarai Jatinegara PondokJati
   Lama perjalanan : 64 menit. Jarak : 26.682 km. Ongkos : 4000
3. Jarak minimal
   Simpul dibangkitkan : 6649
   Rute terbaik : RedLine transit di Manggarai, sambung BlueLine trans
it di Jatinegara, sambung YellowLineBogor
   Stasiun yang dilewati : Depok DepokBaru PondokCina UniversitasIndon
esia UniversitasPancasila LentengAgung TanjungBarat PasarMinggu PasarM
ingguBaru DurenKalibata Cawang Tebet Manggarai Jatinegara PondokJati
   Lama perjalanan : 64 menit. Jarak : 26.682 km. Ongkos : 4000
4. Transit minimal
   Simpul dibangkitkan : 220
   Rute terbaik : YellowLineNambo
   Stasiun yang dilewati : Depok DepokBaru PondokCina UniversitasIndon
esia UniversitasPancasila LentengAgung TanjungBarat PasarMinggu PasarM
ingguBaru DurenKalibata Cawang Tebet Manggarai Sudirman Karet TanahAba
ng Duri Angke KampungBandan Rajawali Kemayoran PasarSenen GangSentiong
 Kramat PondokJati
   Lama perjalanan : 97 menit. Jarak : 46.934 km. Ongkos : 6000
```

The reason this test was done is because this distance optimal trip has a rather unique route, which is getting into YellowLine, get off to continue using BlueLine, and get off again to return to YellowLine. Again from the test we can see that priotizing minimum number of changing lines resulting in minimum number of nodes generated, and higher time, distance, and fare value.

The comparator of prioritizing minimum changing lines yields the same result as the previous test case, Depok(yellow) -> Pondok Jati, which requires zero line changing. The only difference was that the two stations are closer together, leading to shorter time, shorter distance for approximately ~25 km, and therefore cheaper fare than previous test case.

E. Analysis

From the four test conducted, there were some information that we can obtain.

TABLE IV. DIFFERENT NUMBER OF NODES GENERATED DUE TO DIFFERENT USES OF COMPARATOR

Test Case No.	1	2	3	4
Minimum Fare	2	323	4300	7046
Minimum Travel Time	2	71	4041	5752
Minimum Distance	2	75	4600	6649
Minimum Number of Line Changes	2	190	146	220

V. CONCLUSIONS & RECOMMENDATIONS

From the evaluation and experiment elaborated above, we can conclude that,

1) The Branch and Bound algorithm is capable to solve the KRL Commuter Line trip optimalization problem.

2) The closer the two stations are together in the route map, resulting in less number of nodes generated by the algorithm.

3) Prioritizing the fare which is almost same for every node, and priotizing the distance to retain the route quality leads to the maximum amount of nodes generated, in the other hand, prioritizing minimum number of changing lines resulting least number of nodes generated, but the resulting route might have higher fare, longer travel time, and longer distance to travel between two stations.

ACKNOWLEDGMENT

As the author I would like to thank Dr. Masayu Leylia Khodra, S.T, M.T, Dr. Nur Ulfa Maulidevi, S.T, M.Sc. and Dr. Ir. Rinaldi Munir, M.T. as the lecturer of the course IF2211 Strategi Algoritma. I am extremely grateful to them for their expert and valuable guidance along with the knowledge given to their students. I also place on record, my sincere gratitude to all of my friends and family who have supported me through the process of writing this paper.

REFERENCES

[1] R. Munir, Diktat Kuliah IF2211 Strategi Algoritma, Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2009.

[2] https://jakarta.bps.go.id/backend/brs_ind/brsInd-20150220094832.pdf, accessed 18 Mei 2017, 22:08 WIB.

[3] http://megapolitan.kompas.com/read/2017/05/10/08353531/penumpang.krl.jabodetabek.tembus.1.juta.pengguna, accessed 15 Mei 2017, 23:03 WIB.

[4] https://en.wikipedia.org/wiki/KAI_Commuter_Jabodetabek, accessed 18 Mei, 02:19 WIB.

[5] http://www.krl.co.id/, accessed 15 Mei 2017, 23:02 WIB.

[6] R. Munir, Matematika Diskrit, 3rd ed. Bandung: Penerbt INFORMATIKA. Bandung, 2010.

[7] http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/A-Star-Best-FS-dan-USC-(2016).pptx, accessed 18 Mei 2017, 23:07.

[8] https://id.wikipedia.org/wiki/Jabodetabek, accessed 15 Mei 2017, 23:03 WIB.

[9] http://microdata.bps.go.id/mikrodata/index.php/catalog/521, accessed 15 Mei 2017, 23:03 WIB.

[10] http://www.kompasiana.com/prattemm/ada-apa-dengan-layanan-krl-jabodetabek_57aa89c74423bdbf068b456f, accessed 15 Mei 2017, 23:03 WIB.

[11] http://www.thejakartapost.com/news/2015/02/17/138-million-commute-jakarta-daily.html, accessed 18 Mei 2017, 22:08 WIB.

IMAGE SOURCES

1 http://www.krl.co.id/wp-content/uploads/2016/08/logo.png

2 http://www.krl.co.id/wp-content/uploads/2016/06/Peta-Rute-KRL-2017-1-724x1024.jpg

3 http://jorisvr.nl/images/mwmatching.png
http://www.introprogramming.info/wp-content/uploads/2013/07/clip_image030.png

4 Fig.2 with slight modification

5 Google Maps

6 Google Maps

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Mei 2017

Audry Nyonata, 13515087