APLIKASI PENCARIAN RUTE ANGKUTAN UMUM MENGGUNAKAN METODA BEST-PATH PLANNING BERBASIS ANDROID

Tulus Prabudi[1], Selvia Lorena Br Ginting,S.Si., MT[2]
[1,2] Jurusan Teknik Komputer Unikom, Bandung
[1]tulusprabudi@gmail.com, [2]selvialorena@yahoo.com

ABSTRAK

Berawal dari sulitnya menemukan jalur angkutan umum yang dapat digunakan untuk menempuh suatu tujuan di suatu kota khususnya Kota Bandung, sehingga perlu dikembangkan aplikasi mobile pencarian rute angkutan umum pada Kota Bandung. Aplikasi yang dibuat merupakan aplikasi mobile berbasis android. Pada karya ilmiah ini digunakan Algoritma Best-Path Planning for Public Transportation Systems untuk melakukan pecarian rute angkutan umum. Aplikasi yang dibuat membutuhkan peng-input-an lokasi asal dan lokasi tujuan pengguna yang lokasinya terdapat dalam database untuk melakukan pencarian rute angkutan umum terdekat. Lokasi-lokasi yang digunakan dalam aplikasi ini merupakan lokasi-lokasi yang terdapat dalam Google Maps. Google Maps API digunakan untuk menampilkan peta pada pengguna dan juga berfungsi untuk menghitung jarak antara lokasi tujuan dengan posisi dimana pengguna berada sehingga pengguna dapat mengetahui ketika pengguna sudah dekat dengan lokasi tujuan lewat pesan yang diberikan oleh sistem. Dalam aplikasi yang akan dibuat terdapat tombol terdekat guna mencari lokasi asal yang paling terdekat dengan posisi pengguna apabila pengguna belum mengenal lingkungan dimana pengguna berada. Di harapkan dengan adanya aplikasi ini pengguna dapat terbantu dalam melakukan pencarian rute angkutan umum tedekat di Kota Bandung.

Kata kunci : Rute Angkutan Umum, Algoritma Best-Path Planning, Android.

1. PENDAHULUAN

1.1 Latar Belakang

Berawal dari sulitnya menemukan jalur angkutan umum yang dapat digunakan untuk menempuh suatu tujuan di suatu kota khususnya Kota Bandung, sehingga perlu dikembangkan aplikasi *mobile* pencarian rute angkutan umum. Hal ini didukung dengan perkembangan teknologi dan pertumbuhan pemakaiannya sehingga hal ini nantinya akan sangat memudahkan bagi orang banyak terutama orang-orang yang baru tinggal di Bandung atau yang masih awam dengan angkutan umum di Kota Bandung.

Aplikasi ini merupakan aplikasi *mobile* berbasis android. Dengan menggunakan koneksi internet, GPS (*Global Positioning System*), Google Maps, dan rute-rute angkutan umum yang ada di Kota Bandung beserta data-data pendukungnya, maka akan diciptakanlah sebuah aplikasi yang dapat berguna sebagai *guide* dalam pencarian rute angkutan umum di Kota Bandung.

Diharapkan dalam menjalankan aplikasi ini nantinya, pengguna akan mendapati sesedikit mungkin tahapan-tahapan yang harus dilakukan dalam proses pencarian rute angkutan umum. Selain itu, dengan aplikasi tersebut, diharapkan pengguna juga dapat mengetahui posisinya pada peta secara dinamis serta ditambah lagi dengan adanya pemberitahuan kepada pengguna seandainya tempat tujuan telah cukup dekat sehingga diharapkan akan memudahkan pengguna dalam mencari rute angkutan umum di Kota Bandung.

1.2 Tujuan

Tujuan Tugas Akhir ini adalah mengimplementasikan algoritma *Best-Path Planning* dalam melakukan pencarian rute angkutan umum di Kota Bandung menggunakan *handphone* yang berbasis Android. Agar dapat bermanfaat bagi orang-orang yang baru tinggal di Kota Bandung dan belum mengenal angkutan umum Kota Bandung.

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. Aplikasi ini berjalan di *platform* Android.
2. Aplikasi ini merupakan implementasi dari algoritma *Best-Path Planning for Public*

Transportation Systems untuk pencarian rute angkutan umum di Kota Bandung.
3. Rute angkutan umum yang digunakan berupa rute angkot dan rute damri dalam Kota Bandung.
4. Lokasi-lokasi yang digunakan sebagai input, adalah lokasi-lokasi yang terdapat pada Google Maps™.
5. Koordinat yang diambil adalah titik lokasi dari masing-masing jalan seperti yang telah ditentukan oleh Google Maps.
6. Maksimal *transfer* yang bisa dilakukan adalah dua kali *transfer*.
7. Lokasi yang digunakan merupakan lokasi yang diambil dari situs resmi Pemerintah Kota Bandung.

2. TEORI PENUNJANG

2.1 Rute Angkutan Umum

Angkutan merupakan kegiatan pemindahan penumpang dan barang dari suatu tempat ke tempat yang lain. Definisi angkutan umum menurut Undang-undang adalah angkutan yang dimana penggunanya dipungut biaya sedangkan rute adalah kumpulan ruas jalan yang menghubungkan satu tempat ke tempat lain secara terus menerus [3]. Rute angkutan umum yang digunakan dalam penelitian tugas akhir ini ada 2 rute yaitu rute angkot dan rute damri.

2.2 Algoritma Best Path Planning

Aplikasi yang dibuat dalam Tugas Akhir ini, digunakanlah *Algoritma Best-Path Planning* dimana algoritma ini berfungsi untuk mencari rute yang tersedia dari suatu tempat ke tempat lainnya dengan jumlah *transfer* rute sesedikit mungkin [1]. Karena data yang sedemikian besar dan *resource* perangkat yang dimiliki terbatas, maka pada aplikasi ini jumlah *transfer* rute dibatasi sampai dengan dua kali *transfer* [7].

2.2.1 Matriks Transisi [7]

Aplikasi yang dibuat memanfaatkan matriks transisi yang digunakan untuk mengetahui jumlah transfer rute yang dibutuhkan dari tempat yang satu ke tempat lainnya [1]. Anggap jumlah tempat yang dilalui oleh angkutan umum adalah n. Maka kita menggunakan matriks transisi dengan ukuran n x n. Selanjutnya kita memberi nilai pada cell Ti,j dari matriks transisi T adalah jumlah rute langsung yang mungkin bagi seseorang untuk bepergian dari tempat i ke tempat j. Otomatis nilai Ti,i adalah 0.

Para ahli telah membuktikan bahwa pangkat n dari T adalah jumlah rute yang dapat ditempuh dari lokasi i ke lokasi j dengan (n − 1) kali *transfer*, untuk n ≥ 2.
Definisi standar pangkat n dari T ialah :

$$T_{i,j}^{n} = \sum_k T_{i,k}^{n-1} \; T_{k,j} \ldots\ldots\ldots\ldots\ldots \quad (1)$$

Dapat diasumsikan bahwa $T_{i,j}^{n-1}$ adalah jumlah rute yang dapat dilakukan untuk perjalanan dari i ke j dengan (n − 2) *transfer*, yang mana ini berlaku untuk kasus n = 2. Kemudian, pernyataan pertama dari persamaan di atas $T_{i,k}^{n-1}$, adalah jumlah rute yang mungkin yang dapat dilakukan untuk perjalanan dari i ke lokasi penghubung k dengan (n − 2) kali *transfer*, dan pernyataan kedua adalah rute langsung yang mungkin dapat dilakukan untuk perjalanan dari k ke j. Dengan mengalikan dua hasil tersebut akan menghasilkan jumlah total rute untuk bepergian dari i ke j dan *transfer* pada k tertentu dengan (n − 1) kali *transfer*. Dengan demikian, menjumlahkan semua k yang mungkin akan menghasilkan jumlah total rute untuk bepegian dari i ke j dengan (n − 1) kali *transfer*.

2.2.2 Algoritma

Let o and d denote the numbers assigned to the origin and destination, respectively.
1. *Trivial cases : if* ***o = d****, show an appropriate message and return a null plan.*
2. *Direct : if* $Q_{o,d} = 1$*, return any service in DService(****o,d****).*
3. *One transfer : if* $Q_{o,d} = 2$*, there must be a location m such that* $Q_{o,m} = 1$ *and* $Q_{m,d} = 1$*. Combine any route in DService(****o,m****) and any route in DService(****m,d****) to obtain a one-transfer plan.*
4. *Two transfer : if* $Q_{o,d} = 3$*, there must be different location* ***m1*** *and* ***m2*** *such that* $Q_{o,m1} = 1$*,* $Q_{m1,m2} = 1$*, and* $Q_{m2,d} = 1$*. Combine one route from each of DService(o,m1), DService(****m1,m2****), and DService(****m2,d****) to obtain a two transfer plan.*

Gambar 1. Algoritma *Best-Path Planning* [1]

Gambar 1 adalah algoritma *best-path planning* yang di pakai sebagai acuan dalam pembuatan aplikasi ini [1]. Algoritma ini diambil dari paper *Best-Path Planning for Public Transportation Systems*.

Algoritma ini akan mengecek nilai matriks yang didapatkan dari indeks lokasi awal dan lokasi tujuan secara bertahap. Dimulai dari nilai matriks apabila bernilai 0 atau indeks lokasi awal sama dengan indeks lokasi tujuan, maka diberikan pesan bahwa anda sudah dilokasi yang dituju atau dibutuhkan lebih dari 2 kali transfer. Kemudian dilanjutkan apabila nilai matriks bernilai 1 maka diambil pencarian langsung (*direct*) dari lokasi awal sampai ke lokasi tujuan dan seterusnya hingga nilai matriks bernilai 3 maka diambil pencarian dengan dua kali perpindahan angkutan umum (*two transfer*).

2.2.3 Data Koordinat [7]

Untuk menghitung jarak antara posisi pengguna dengan lokasi tujuan serta untuk menentukan lokasi terdekat maka dibutuhkan sebuah *database* yang di dalamnya berisi koordinat-koordinat dari semua rute yang dilewati oleh angkutan umum yang ada di Kota Bandung.

Untuk mendapatkan hasil matriks yang sesuai dibutuhkan beberapa tahapan. Yang harus dilakukan pertama adalah membuat matriks transisi T berukuran n x n yang berisikan jumlah rute langsung yang mungkin dari satu tempat ke tempat lainnya. n adalah jumlah tempat yang ada pada data tempat sehingga $T_{i,j}$ adalah jumlah rute langsung yang mungkin dari tempat i ke tempat j pada *database* tempat.

Dari matriks T kita bisa mendapatkan matriks T^2 dan T^3 nya dari proses perhitungan yang dilakukan di *MATLAB* yang dimana T adalah rute langsung, T^2 adalah rute sekali *transfer* dan T^3 adalah rute dua kali *transfer*. Hal ini dikarenakan pada aplikasi ini jumlah *transfer* nya dibatasi sampai dua kali *transfer* saja.

Setelah matriks T, T^2, dan T^3 sudah didapatkan, ketiganya kemudian diolah dan nantinya akan digabungkan sehingga membentuk sebuah matriks baru yang digunakan untuk aplikasi ini. Pada Matriks tersebut nilai-nilai elemennya terdiri dari 0, 1, 2, dan 3. T*i,j* = 0 artinya tidak adanya rute yang berangkat dari lokasi i ke lokasi j, atau membutuhkan lebih dari dua kali *transfer*. T*i,j* = 1 artinya rute yang berangkat dari lokasi i ke lokasi j merupakan rute langsung. T*i,j* = 2 artinya rute yang berangkat dari lokasi i ke lokasi j merupakan rute satu kali transfer. Dan untuk T*i,j* = 3 artinya rute yang berangkat dari lokasi i ke lokasi j merupakan rute dua kali *transfer*.

2.3 LBS (*Location Based Service)* [2]

Location Based Service (LBS) atau Layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan. LBS berfungsi untuk mencari dengan teknologi GPS dan *Google's cell-based location*. Dua unsur utama LBS adalah:

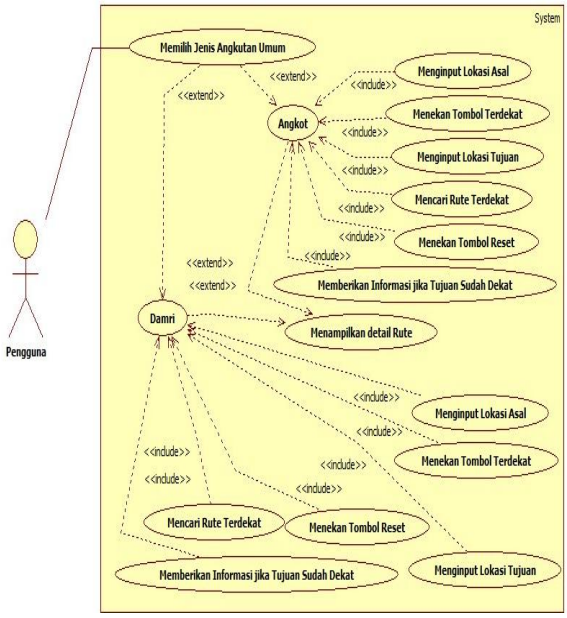
- *Location Manager* (API Maps)
  Menyediakan *tools/source* untuk LBS, *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan, memanipulasi *maps*/peta beserta fitur-fitur lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada com.google.android.maps.

- Location Providers (API Location)
  Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*/perangkat. API *Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. API *Location* berapa pada paket android yaitu dalam paket android.location. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, *Track* gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

3. ANALISA DAN PERANCANGAN SISTEM

3.1 Pemodelan Sistem

3.1.1 Use case Diagram

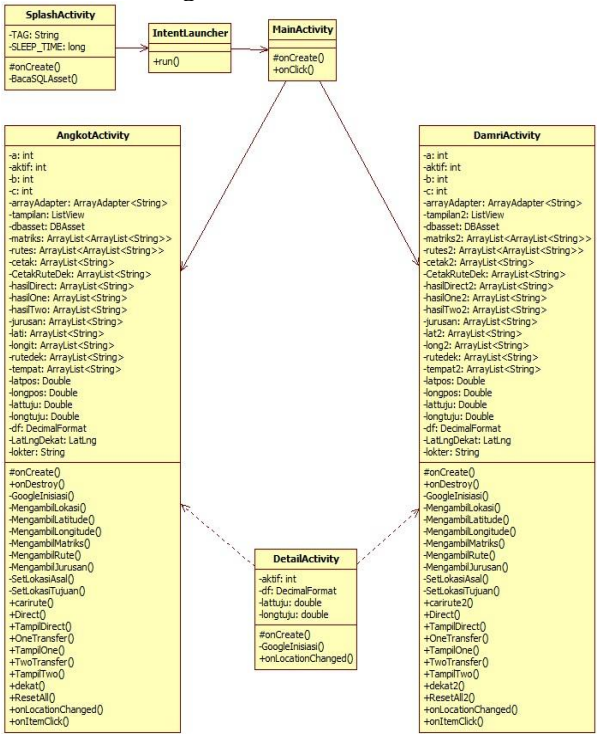


Gambar 2. *Use Case Diagram*

Terdapat Satu Aktor yaitu:

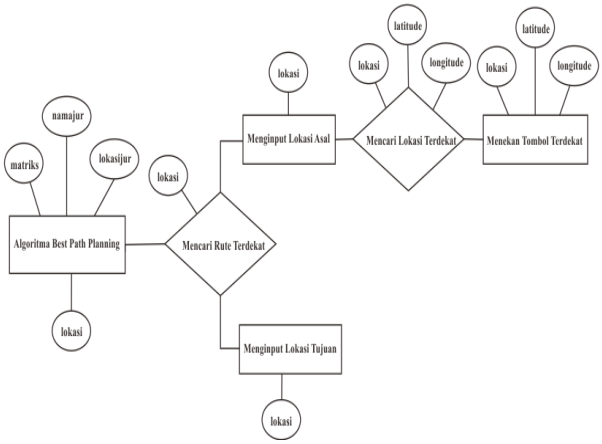
1. Pengguna
   Aktor yang akan melakukan pencarian rute angkutan umum terdekat.

3.1.2 Class Diagram



Gambar 3. *Class Diagram*

3.1.3 ERD (*Entity Relation Diagram*)



Gambar 4. *Entity Relation Diagram*

3.2 Perancangan Data

Aplikasi pencarian rute angkutan umum ini merupakan aplikasi *mobile* yang dapat memberikan informasi rute transportasi umum dari lokasi dimana pengguna berada ke tempat lokasi yang ingin dituju di Kota Bandung sesuai dengan data yang terdapat dalam *dabatase* yang sudah disediakan. Berikut rancangan data yang akan dibuat:

- Rancangan Data Table *Database*

Tabel 1. Rancangan Table *Database*

Row ID	Type	Name	Table Name
1	table	dbMatriks	dbMatriks
2	table	dbMatriksB	dbMatriksB
3	table	dbRute	dbRute
4	table	dbRuteB	dbRuteB
5	table	dbTempat	dbTempat
6	table	dbTempatB	dbTempatB

- Rancangan Data Table dbMatriks & dbMatriksB

Tabel 2. Rancangan Table dbMatriks & dbMatriksB

Colomn ID	Name	Type	Not Null	Default Value	Primary Key
0	id	INTEGER	1	null	1
1	matriks	TEXT	1	null	0

- Rancangan Data Table dbRute & dbRuteB

Tabel 3. Rancangan Table dbRute & dbRuteB

Colomn ID	Name	Type	Not Null	Default Value	Primary Key
0	id	INTEGER	1	null	1
1	namajur	TEXT	1	null	0
2	lokasijur	TEXT	1	null	0

- Rancangan Data Table dbTempat & dbTempatB

Tabel 4. Rancangan Table dbTempat & dbTempatB

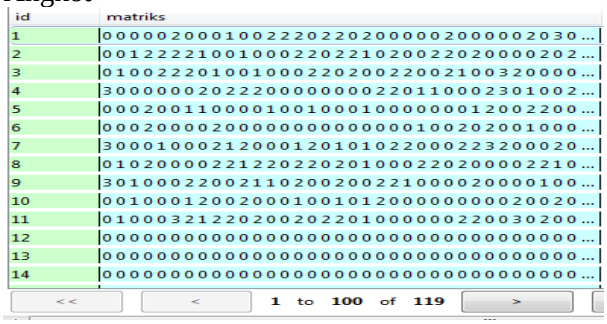
Colomn ID	Name	Type	Not Null	Default Value	Primary Key
0	id	INTEGER	1	null	1
1	lokasi	TEXT	1	null	0
2	latitude	DOUBLE	1	null	0
3	longitude	DOUBLE	1	null	0

4. PENGUJIAN DAN ANALISA

4.1 Implementasi

4.1.1 Implementasi *Database*

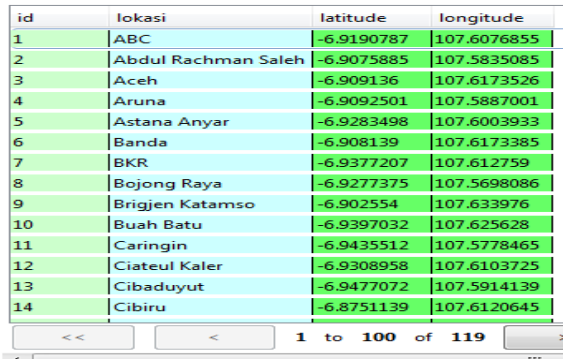
4.1.1.1 Implementasi Table *Database* untuk Angkot



Gambar 5. Cuplikan Table dbMatriks untuk Angkot

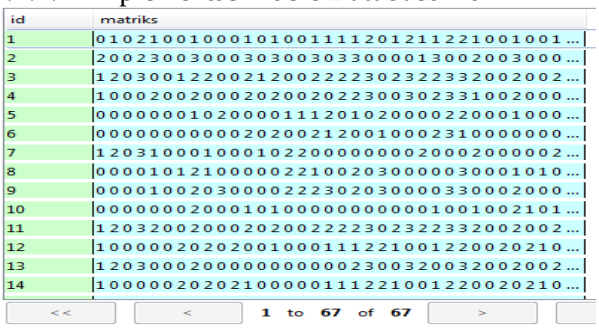


Gambar 6. Cuplikan Table dbRute untuk Angkot



Gambar 7. Cuplikan Table dbTempat untuk Angkot

4.1.1.2 Implementasi Table *Database* Damri



Gambar 8. Cuplikan Table dbMatriks untuk Damri

id	namajur	lokasijur
1	LEUWI PANJANG-DIPATIUKUR	Kopo-Pasir Koja-Astana Anyar-Kebon Jati...
2	DIPATIUKUR-LEUWI PANJANG	Dipati Ukur-Ir. H. Juanda-Merdeka-Perinti...
3	LEUWI PANJANG-LEDENG	Kopo-Pasir Koja-Astana Anyar-Kebon Jati...
4	LEDENG-LEUWI PANJANG	Setiabudi-Sukaasih-Sukajadi-Pasir Kaliki-K...
5	LEUWI PANJANG-CICAHEUM	Kopo-Pasir Koja-Pungkur-Dewi Sartika-Al...
6	CICAHEUM-LEUWI PANJANG	A. Yani-Asia Afrika-Otista-BKR-Leuwi Panj...
7	KEBUN KELAPA-CIBIRU	Dewi Sartika-Banceuy-Cikapundung-Nari...
8	CIBIRU-KEBUN KELAPA	Ujung Berung-A. Yani-Asia Afrika-Otista-...
9	KEBUN KELAPA-TANJUNGSARI	M. Toha-Pungkur-Karapitan-Buah Batu-S...
10	TANJUNGSARI-KEBUN KELAPA	Tanjung Sari-Jatinangor-Cileunyi-Cibiru-S...
11	DIPATIUKUR-JATINANGOR	Dipati Ukur-Surapati-Diponegoro-Suprat...
12	JATINANGOR-DIPATIUKUR	Jatinangor-Cileunyi-Cibiru-Ujung Berung...
13	CICAHEUM-CIBEUREUM	A. Yani-Asia Afrika-Sudirman-Elang
14	CIBEUREUM-CICAHEUM	Elang-Rajawali-Kebon Jati-Otista-Stasion-...

<< < 1 to 22 of 22 > >>

Gambar 9. Cuplikan Table dbRuteB untuk Damri

id	lokasi	latitude	longitude
1	A. Yani	-6.9110855	107.6388668
2	Abdul Rahman Saleh	-6.9075885	107.5835085
3	Alun Alun	-7.0489618	107.5911755
4	Arjuna	-6.9110281	107.5897637
5	Asia Afrika	-6.9216609	107.6108944
6	Astana Anyar	-6.9283498	107.6003933
7	Banceuy	-6.91822	107.6065109
8	BKR	-6.9377207	107.612759
9	Braga	-6.9179019	107.6093606
10	Buah Batu	-6.9397032	107.625628
11	Bungsu	-6.918744	107.613466
12	Cibiru	-6.8751139	107.6120645
13	Cikapundung	-6.9183049	107.6074971
14	Cileunyi	-6.9390632	107.7486595

<< < 1 to 67 of 67 >

Gambar 10. Cuplikan Table dbTempatB untuk Damri

4.2 Pengujian

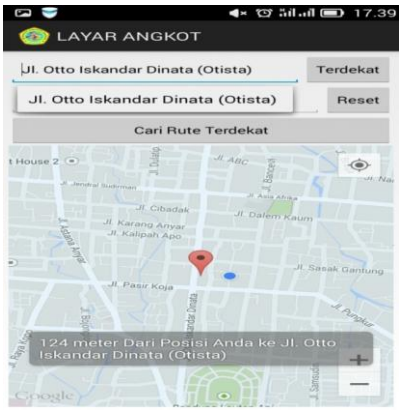
4.2.1 Pengujian Alpha

1. Pengujian proses *input* lokasi asal menjelaskan aktivitas pengujian untuk melihat apakah sistem akan menampilkan lokasi yang sesuai dengan *input*-an pengguna. Berikut pengujian proses *input* lokasi asal.



Gambar 11. Proses *Input* Lokasi Asal

2. Pengujian proses menekan tombol terdekat menjelaskan aktivitas pengujian untuk melihat apakah sistem berhasil menampilkan lokasi terdekat dengan lokasi dimana pengguna berada. Berikut pengujian proses menekan tombol terdekat.



Gambar 12. Pengguna Menekan Tombol Terdekat

3. Pengujian proses *input* lokasi tujuan menjelaskan aktivitas pengujian untuk melihat apakah sistem akan menampilkan lokasi yang sesuai dengan *input*-an pengguna. Berikut pengujian proses *input* lokasi tujuan.



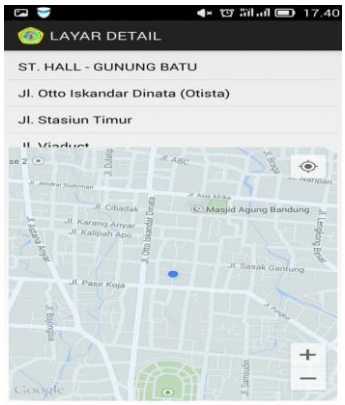
Gambar 13. Proses *Input* Lokasi Tujuan

4. Pengujian proses menekan tombol cari rute terdekat menjelaskan aktivitas pengujian untuk melihat apakah sistem berhasil menampilkan rute terdekat sesuai dengan *input* lokasi asal dan lokasi tujuan pengguna. Berikut pengujian proses menekan tombol cari rute terdekat.



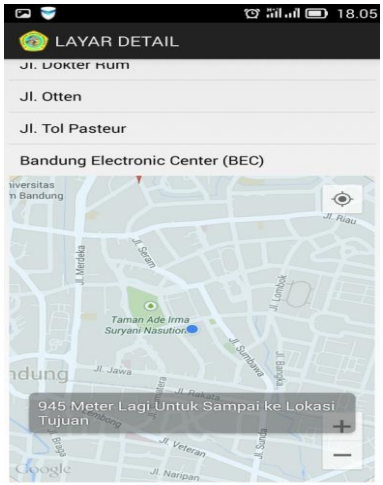
Gambar 14. Pengguna Menekan Tombol Terdekat

5. Pengujian proses menampilkan *detail* rute menjelaskan aktivitas pengujian untuk melihat apakah sistem berhasil menampilkan *detail* rute setelah pengguna menekan *listview* hasil pencarian rute terdekat. Berikut pengujian proses menampilkan *detail* rute.



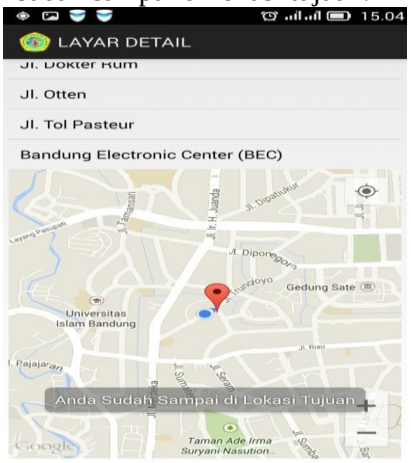
Gambar 15. Tampilan Sistem Berhasil Menampilkan Detail Rute

6. Pengujian proses sistem menampilkan pesan saat pengguna sudah dekat dengan lokasi tujuan menjelaskan aktivitas pengujian untuk melihat apakah sistem berhasil menampilkan pesan ketika pengguna sudah mendekati lokasi tujuan kurang dari 1 km. Berikut pengujian proses sistem menampilkan pesan saat pengguna sudah dekat dengan lokasi tujuan.



Gambar 16. Sistem Menampilkan Pesan Disaat Pengguna Sudah Dekat dengan Lokasi Tujuan

7. Pengujian proses sistem menampilkan pesan saat pengguna sudah sampai di lokasi tujuan menjelaskan aktivitas pengujian untuk melihat apakah sistem berhasil menampilkan pesan ketika pengguna sudah sampai di lokasi tujuan. Berikut pengujian proses sistem menampilkan pesan saat pengguna sudah sampai di lokasi tujuan.



Gambar 17. Sistem Menampilkan Pesan Disaat Pengguna Sudah Sampai di Lokasi Tujuan

5. KESIMPULAN DAN SARAN

Kesimpulan terhadap hasil pengujian yang dilakukan adalah :

Berdasarkan hasil pengujian alpha dan beta dapat disimpulkan bahwa aplikasi yang dibangun sudah sesuai dengan yang diharapkan yaitu dapat melakukan pencarian rute angkutan umum menggunakan *Algoritma Best-Path Planning*. Aplikasi ini hanya dapat mencari rute terdekat yang lokasi-lokasinya terdapat dalam *database* dimana lokasi-lokasinya diambil dari

situs resmi pemerintah Kota Bandung dan lokasinya terdapat dalam *Google Maps*.

Keunggulan aplikasi ini terdapat pada peta, karena dengan bantuan GPS dan *internet* pengguna dapat mengetahui posisi dimana pengguna berada serta dapat melakukan pencarian lokasi terdekat yang dapat dijadikan sebagai lokasi asal.

Kekurangan aplikasi ini terdapat pada hasil matriks transisi yang dihasilkan, karena kebanyakan nilai matriks yang dihasilkan adalah 0, ini menyebabkan banyak rute terdekat yang tidak dapat ditemukan oleh sistem.

Algoritma Best-Path Planning yang digunakan sepertinya kurang cocok digunakan dalam kasus yang diambil karena menghasilkan nilai matriks transisi yang kebanyakan adalah 0, hal ini dikarenakan kasus pada Tugas Akhir ini memiliki rute berangkat dan rute kembali angkutan umum yang tidak sama. Hal tersebut mempengaruhi nilai matriks yang dihasilkan dan menyebabkan sistem sulit menemukan rute angkutan umum terdekat pada Kota Bandung.

Yang dapat disarankan pada pengujian ini adalah:

a) Menggunakan algoritma yang lain selain *Algoritma Best-Path Planning for Public Transportation System* seperti metode djikstra dan algoritma A*.
b) Dapat melakukan sistem pencarian lokasi yang tidak terdapat dalam *database.*
c) Menambahkan rute-rute jenis angkutan umum yang baru selain rute angkot dan damri seperti rute bis sekolah Kota Bandung.

6. DAFTAR PUSTAKA

[1] Chao-Lin liu, 2002, *Best-Path Planning for public Transportation System*, In Proceedings of the Fifth International IEEE Conference on Intelligent Transportation System.

[2] H., Nazruddin Safaat, 2012, *ANDROID Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android,* Informatika Bandung, Bandung.

[3] Hadihardaja, Prof. Ir. Joetata, 1997, *Sistem Transportasi*, Gunadarma, Jakarta.

[4] (Februari 2015), *Trayek Angkot Bandung*, <URL: http://bandung.go.id/rwd/index.php?fa=infokota.detail&id=17>

[5] (Februari 2015), *Trayek Damri Bandung*, <URL: http://bandung.go.id/rwd/index.php?fa=infokota.detail&id=20//>

[6] Murya, Yosef, 2014, *Pemrograman Android Black Box*, Jasakom, Jakarta.

[7] Pratiwi, Rizki, 2012, *Aplikasi Mobile Pencarian Rute Transportasi Umum dengan Algoritma Best-Path Planning pada Platform Android.*

[8] Sholiq, 2006, *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*, Graha Ilmu, Yogyakarta.

[9] Sinaga, Benyamin L., 2005, *Pemrograman Berorientasi Objek dengan Java*, Gava Media, Yogyakarta.

[10] ST, Edy W., Eng, M., Zaki, Ali, Community, SmitDev, 2011, *Membuat Sendiri Aplikasi Android untuk Pemula*, PT Elex Media Komputindo, Jakarta.

[11] ST, Wahyu C., S.T., M.Eng M. A., 2011, *Panduan Belajar Mandiri Matlab*, PT Elex Media Komputindo, Jakarta.