**PENERAPAN ALGORITMA BEST-PATH PLANNING**

**UNTUK APLIKASI PENCARIAN RUTE TRANSPORTASI PUBLIK BERBASIS ANDROID**

**Selvia Lorena Br Ginting, Yogie Rinaldy Ginting, Tulus Prabudi**

Universitas Komputer Indonesia, Universitas Riau/PhD Student at Curtin University, Universitas Komputer Indonesia

Email: [selvialorena@yahoo.com](mailto:selvialorena@yahoo.com)

**Abstrak**

Seringkali penduduk disuatu kota kesulitan untuk menemukan jalur angkutan (transportasi) umum, terutama di kota yang memiliki jalur transportasi yang rumit seperti di Kota Bandung. Hal ini dikarenakan di Bandung sangat banyak jalur angkutan umum lalu kebanyakan jalur satu arah sehingga mengakibatkan penumpang harus naik angkutan umum berkali-kali. Untuk itu perlu dibangun aplikasi *mobile* berbasis Android yang fungsinya untuk melakukan pencarian rute angkutan umum. Aplikasi ini menerapkan Algoritma *Best-Path­Planning for Public Transportation Systems*, di mana disamping diketahui rute yang akan dijalani juga akan diketahui berapa kali penumpang/pengguna sistem akan melakukan pergantian angkutan. Aplikasi ini membutuhkan membutuhkan masukan lokasi asal dan lokasi tujuan pengguna kemudian lokasi tersebut akan dicocokkan dengan daftar lokasi yang ada di 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 kepada pengguna dan juga berfungsi untuk menghitung jarak antara lokasi tujuan dengan posisi dimana pengguna berada sehingga pengguna dapat mengetahui lokasi yang dituju sudah dekat lewat pesan yang diberikan oleh sistem. Kemudian ada tombol yang fungsinya untuk melakukan pencarian lokasi asal terdekat dengan posisi pengguna, ini dapat membantu pengguna yang belum mengenal lingkungan di mana dia berada. Dengan adanya aplikasi ini diharapkan pengguna dapat terbantu dalam melakukan pencarian rute angkutan umum tedekat di Kota Bandung.

Kata kunci: Rute angkutan umum, Algoritma *Best-PathPlanning*, Android, Kota Bandung.

1. **PENDAHULUAN**

Berawal dari sulitnya menemukan jalur angkutan umum yang dapat digunakan untuk menempuh suatu tujuan di suatu kota khususnya Kota Bandung yang memiliki jalur angkutan umu yang rumit, sehingga perlu dikembangkan aplikasi *mobile* pencarian rute angkutan umum berbasis Android yang fungsinya untuk menangani masalah tersebut. Hal ini didukung dengan perkembangan teknologi dan pertumbuhan pemakaiannya sehingga aplikasi 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 tersebut membutuhkan koneksi internet, GPS (*Global Positioning System*), Google Maps, dan rute-rute angkutan umum yang ada di Kota Bandung beserta data-data pendukungnya, dalam pencarian rute angkutan umum, dengan kata lain dapat berguna sebagai *guide.*

Diharapkan dalam menjalankan aplikasi ini nantinya, pengguna akan mendapati sedikit mungkin tahapan-tahapan yang harus dilakukan dalam proses pencarian rute angkutan umum. Selain itu, dengan aplikasi tersebut, pengguna juga dapat mengetahui posisinya pada peta secara dinamis serta ditambah lagi dengan adanya pemberitahuan kepada pengguna bahwa lokasi yang dituju telah cukup dekat sehingga aplikasi ini diharapkan dapat bertindak sebagai *guide* dalam mencari rute angkutan umum di Kota Bandung.Aplikasi ini mengimplementasikan algoritma *Best-Path Planning* dalam melakukan pencarian rute angkutan umum di Kota Bandung menggunakan *handphone* yang berbasis Android. Target utama yang akan menggunakan aplikasi ini adalah para wisatawan yang berkunjung ke Kota Bandung, penduduk yang masih baru tinggal di Kota Bandung dan belum mengenal angkutan umum Kota Bandung, namun tidak menutup kemungkinan digunakan pula oleh orang yang telah lama bermukim di Kota Bandung.

1. **DASAR TEORI**
   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. Rute angkutan umum yang digunakan dalam penelitian ini ada 2 yaitu rute angkot dan rute damri.

* 1. **Algoritma Best-Path Planning**

Aplikasi dibangun dengan menerapkan *AlgoritmaBest-Path Planning*yang berfungsi untuk mencari rute yang tersedia dari suatu tempat ke tempat lainnya dengan jumlah *transfer* rute sesedikit mungkin. Dikarenakan data yang sedemikian besar dan *resource* perangkat yang dimiliki terbatas, maka pada aplikasi ini jumlah *transfer* rute dibatasi sampai dengan dua kali *transfer*.

* + 1. **Matriks Transisi**

Aplikasi dibangun dengan memanfaatkan matriks transisi yang digunakan untuk mengetahui jumlah transfer rute yang dibutuhkan dari tempat yang satu ke tempat lainnya. Diasumsikan jumlah tempat yang dilalui oleh angkutan umum adalah n. Maka digunakan matriks transisi dengan ukuran n 𝑥 n. Selanjutnya diberi nilai pada *cell* Ti,j dari matriks transisi T yang 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:

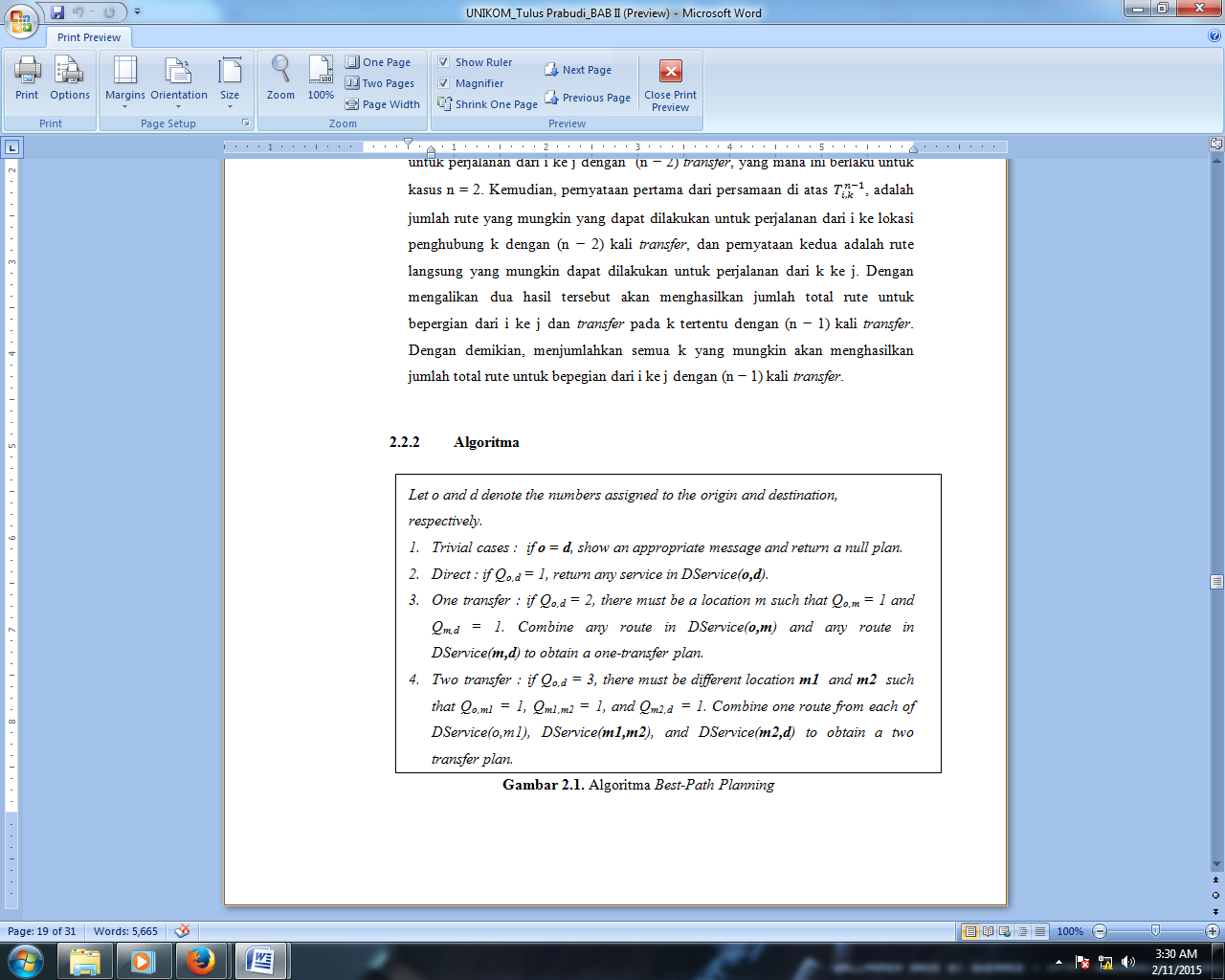
=

Dapat diasumsikan bahwa 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 , 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.*

* + 1. **Algoritma**

Algoritma ini akan memeriksa nilai matriks yang diperoleh 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 pengguna 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*).

Pada Gambar 1 dibawah ini adalah Algoritma *Best-Path Planning* yang di pakai sebagai acuan dalam pembuatan aplikasi yang dibangun. Algoritma ini diambil dari paper *Best-Path Planning for Public Transportation Systems*.



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

Langkah-langkah dari algoritma mengindikasikan bahwa masing-masing angkutan umum memiliki 2 jalur rute. Hal ini bertujuan untuk membedakan tujuan rute angkot/damri tersebut, agar ketika dilakukan pencarian tidak menimbulkan kebingungan bagi si pengguna akan arah angkot/bus mana yang harus pengguna naiki apabila di jalan tersebut terdapat 2 arah yang tiap-tiap arah nya dilalui oleh angkot/damri yang sama.

* + 1. **Data Koordinat**

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. Langkah pertama yang harus 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 Ti,j adalah jumlah rute langsung yang mungkin dari tempat i ke tempat j pada *database* tempat.Dari matriks T diperoleh matriks T2 dan T3 nya dari proses perhitungan yang dilakukan di *MATLAB* yang dimana T adalah rute langsung, T2 adalah rute sekali *transfer* dan T3 adalah rute dua kali *transfer*. Hal ini dikarenakan pada aplikasi ini jumlah *transfer* nya dibatasi sampai dua kali *transfer* saja.

Setelah matriks T, T2, dan T3 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*.

* 1. **Formula Haversine**

Persamaan haversine merupakan sebuah solusi agar lokasi-lokasi terdekat tampak disekitar pengguna dengan mengabaikan bentuk geografis bukit maupun lembah.

Jarak = Acos(Sin(Lat1) x Sin(Lat2) + Cos(Lat1) x Cos(Lat2) x Cos(Long2-Long1) x R

Dimana:

Lat1 : Nilai Latitude Lokasi Pengguna

Lat2 : Nilai Latitude Lokasi yang dituju

Long1 : Nilai Longitude Lokasi Pengguna

Long2 : Nilai Longitude Lokasi yang dituju

R : Radius Bumi (rata-rata radius = 6,371 kilometer)

* 1. **LBS (*Location Based Service)***

*Location Based Service* (LBS) atau Layanan berbasis lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang digunakan. 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 Provider (API Location)**

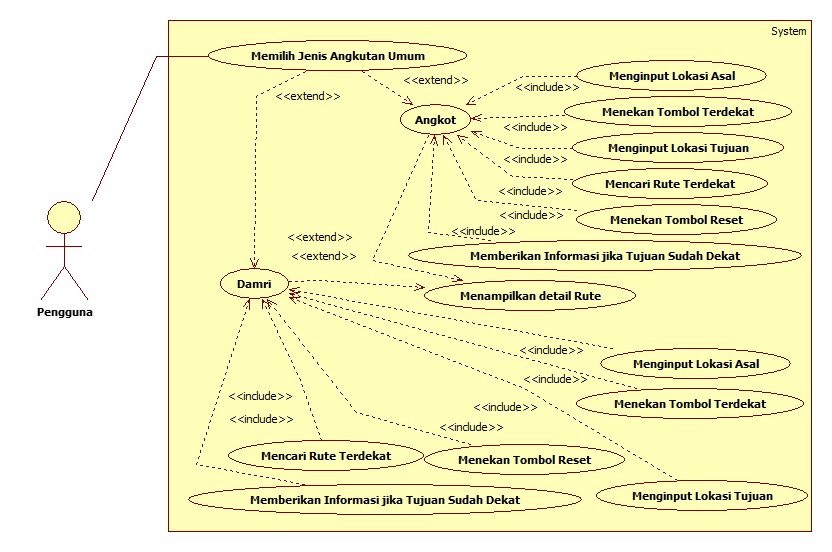
Menyediakan teknologi pencarian lokasi yang digunakan oleh *device*/perangkat. API *Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. API *Location* berapa pada paket android yaitu dalam paket android.location. Dengan *Location Manager*, dapat ditentukan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

1. **ANALISIS DAN PERANCANGAN SISTEM**

Analisis sistem adalah uraian keseluruhan bagaimana sistem yang berjalan saat ini baik dilihat dari analisis fungsional dan analisis nonfungsional.

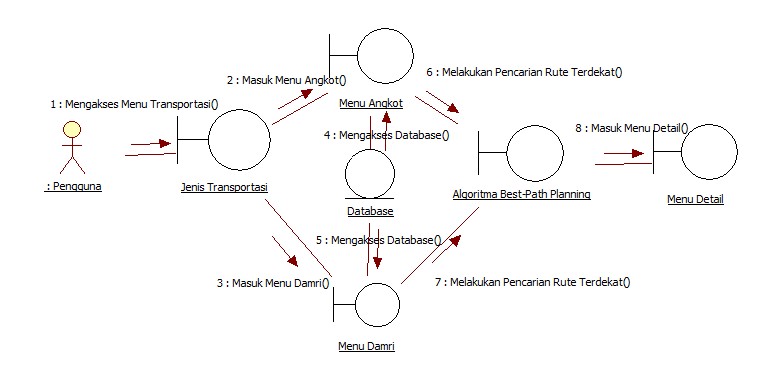
* 1. **Pemodelan Sistem**
     1. **Use Case Diagram**

Terdapat satu aktor yaitu pengguna yang akan mengakses sistem yang mencari rute angkutan umum terdekat.



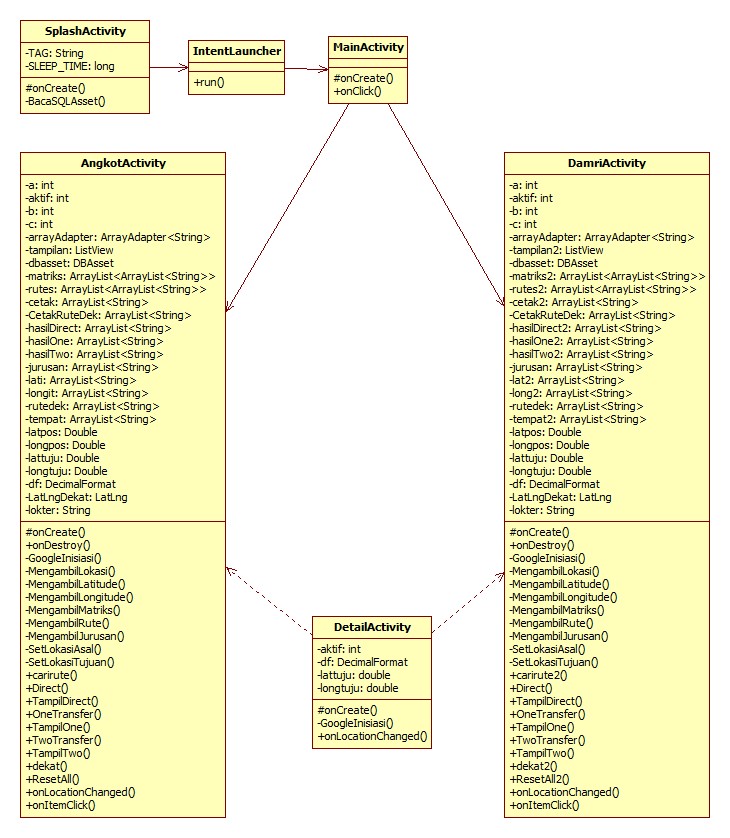
**Gambar 2. Use Case Diagram**

* + 1. **Collaboration Diagram**



**Gambar 3. Collaboration Diagram**

* + 1. **Class Diagram**



**Gambar 4. Class Diagram**

* 1. **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:

1. **Rancangan Data Tabel Database**

**Tabel 1.** Rancangan Tabel 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 |

1. **Rancangan Data Tabel dbMatriks dan dbMatriksB**

**Tabel 2.** Rancangan Data Tabel dbMatriks dan dbMatriksB

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

1. **Rancangan Data Tabel dbRute & dbRuteB**

**Tabel 3.** Rancangan Data Tabel dbRute & dbRuteB

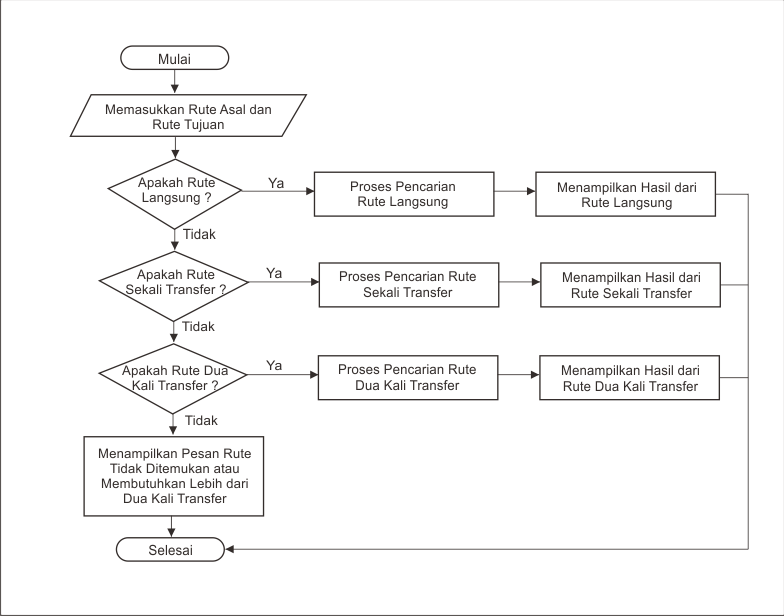
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Colomn ID** | **Name** | **Type** | **Not Null** | **Default Value** | **Primary Key** |
| 0 | id | INTEGER | 1 | null | 1 |
| 1 | namajur | TEXT | 1 | null | 0 |
| 2 | lokasijur | TEXT | 1 | null | 0 |

1. **Rancangan Data Tabel dbTempat & dbTempatB**

**Tabel 4.** Rancangan Data Tabel 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 |

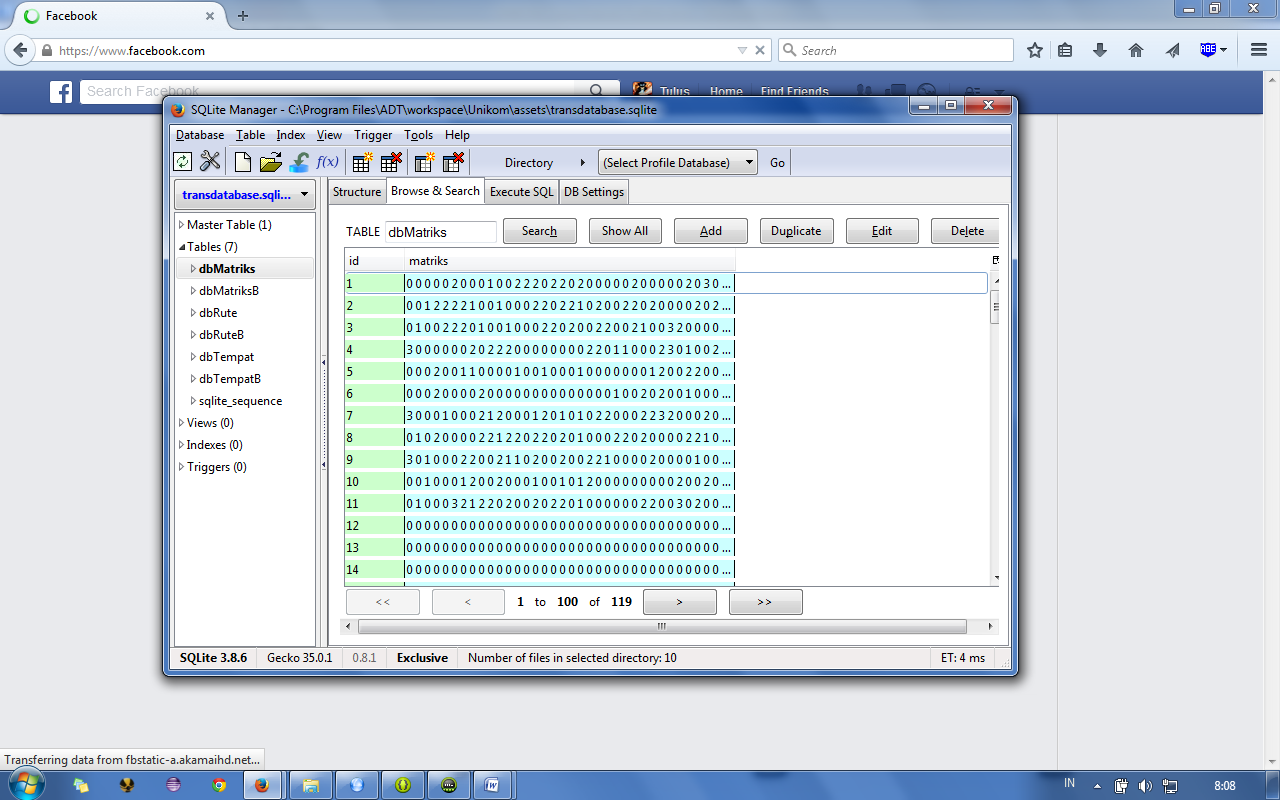
* 1. **Perancangan Prosedur**



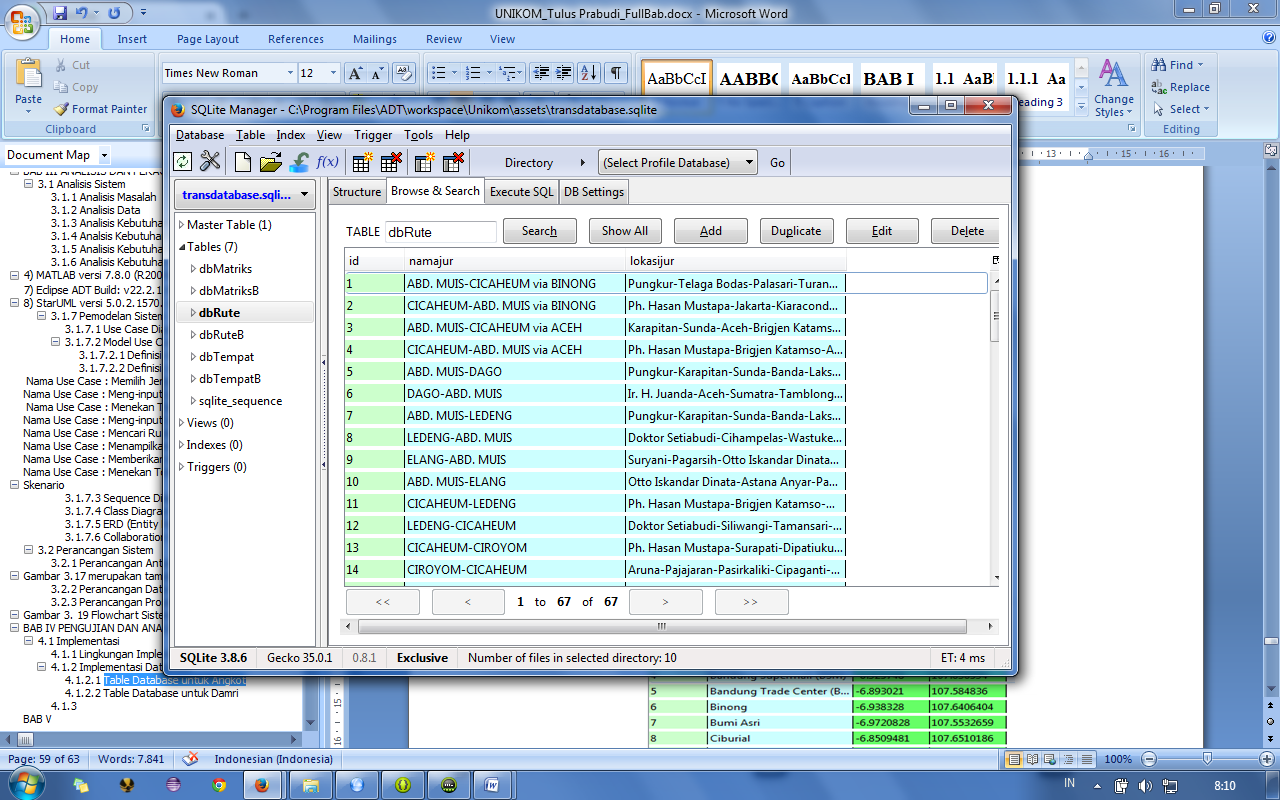
**Gambar 5. Flowchart Sistem**

1. **IMPLEMENTASI DAN PENGUJIAN**
   1. **Implementasi Database**

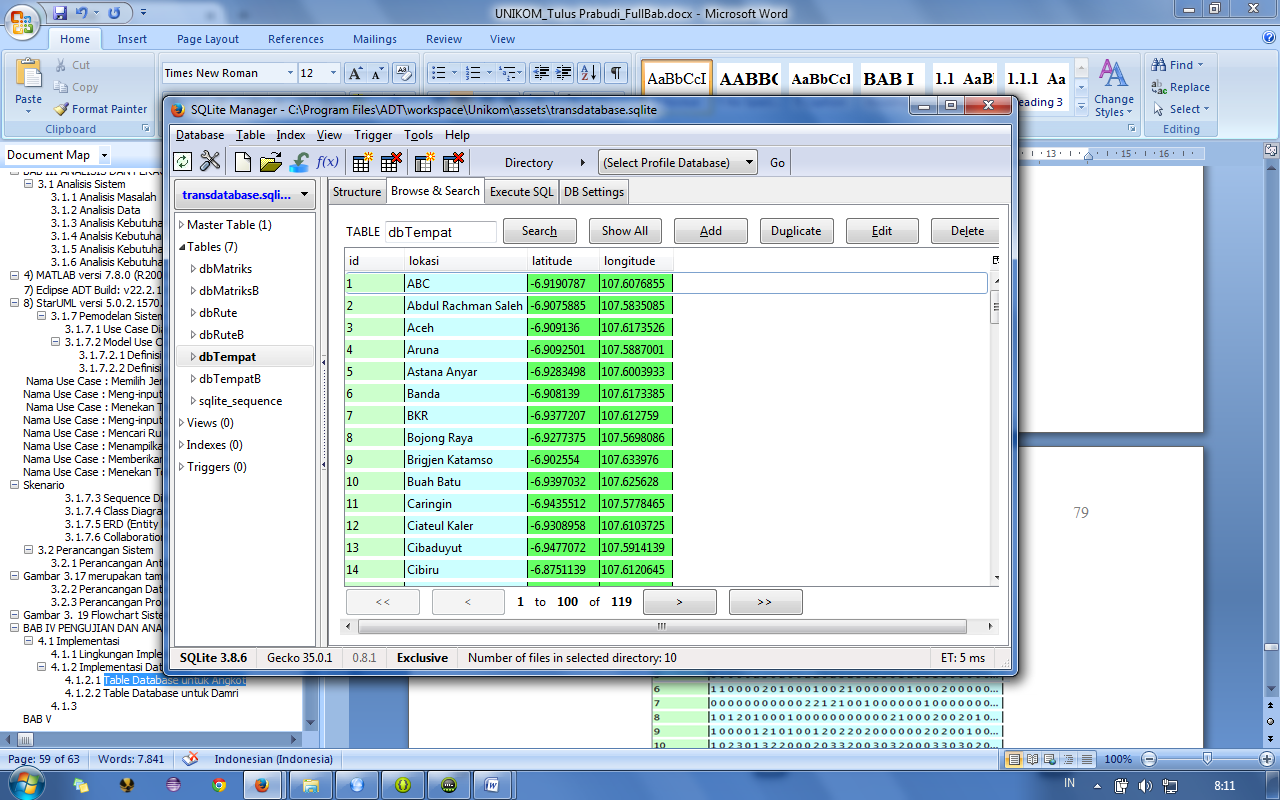
Dalam implementasi *database*, dilakukan penerapan hasil perancangan *database* yang sudah dijelaskan sebelumnya. Berikut adalah cuplikan *database* yang telah dibangun.



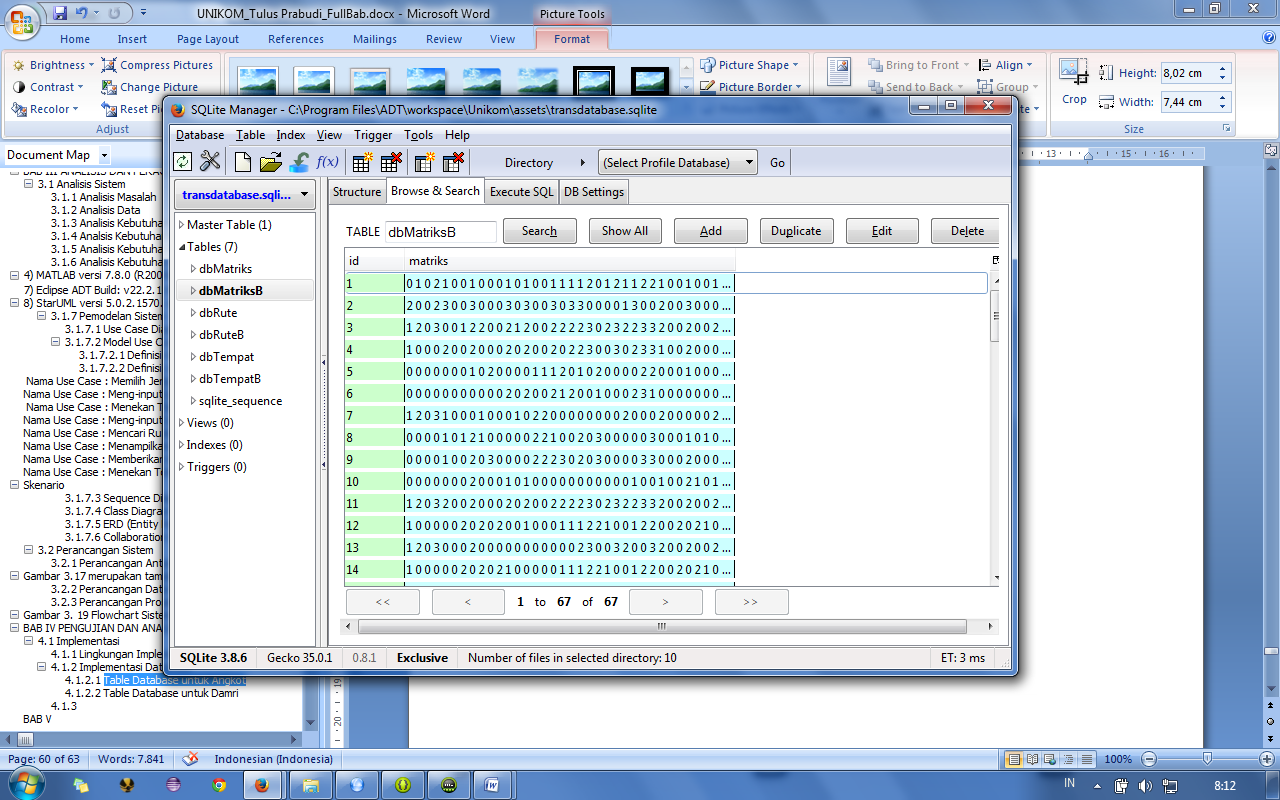
**Gambar 6. Cuplikan Tabel dbMatriks untuk Angkot**



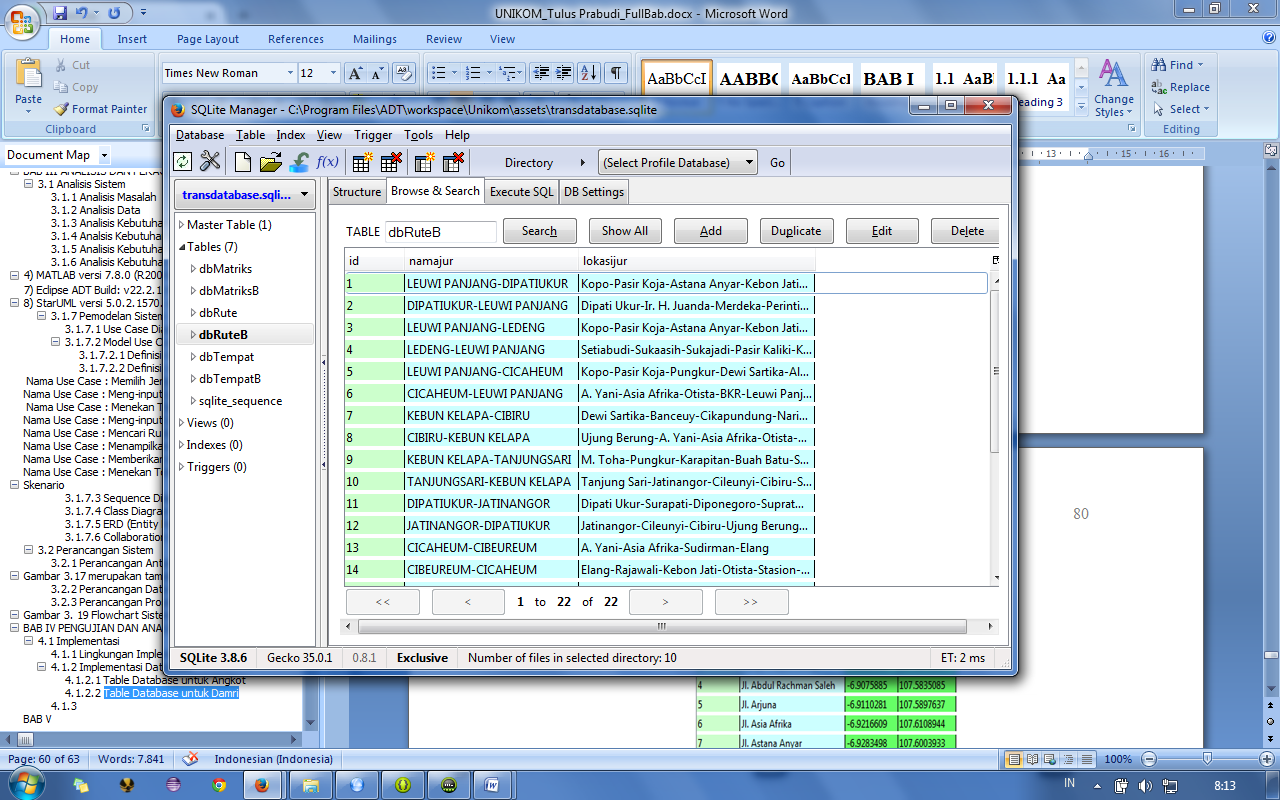
**Gambar 7. Cuplikan Tabel dbRute untuk Angkot**



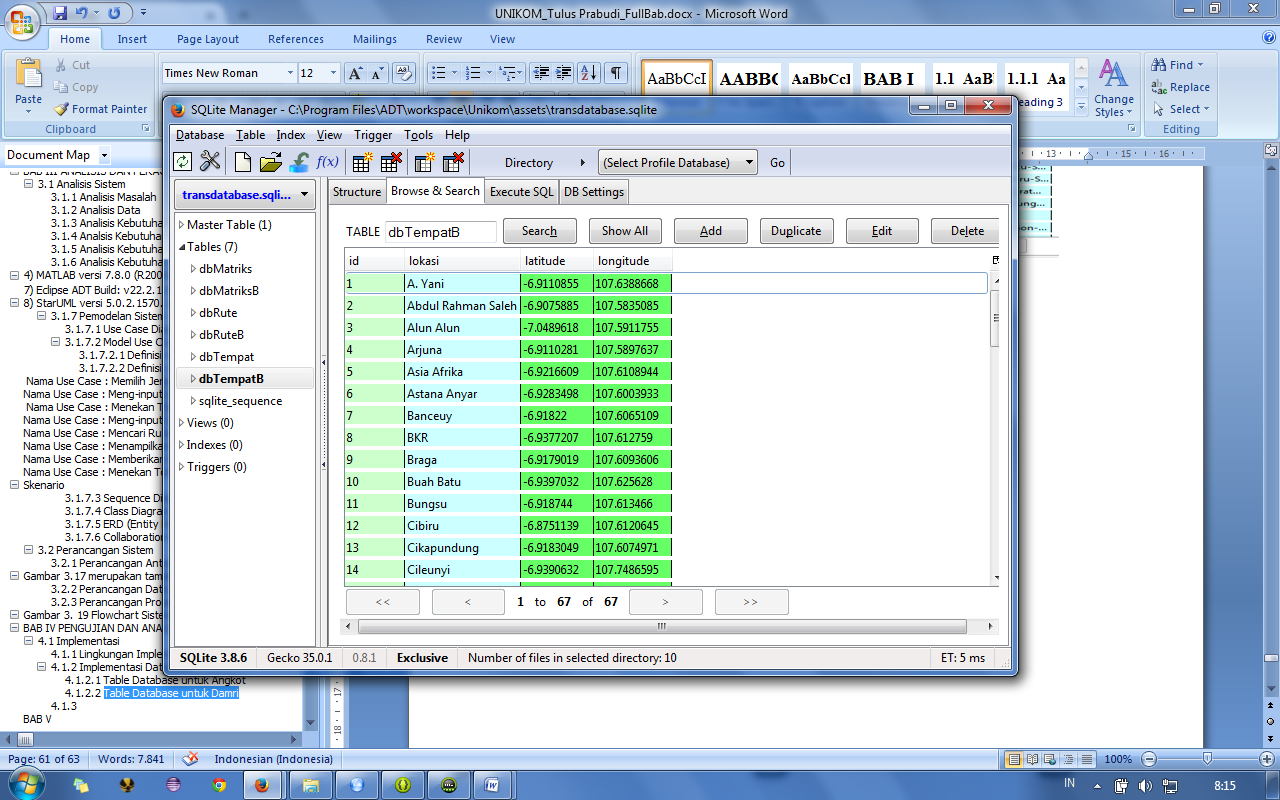
**Gambar 8. Cuplikan Tabel dbTempat untuk Angkot**



**Gambar 9. Cuplikan Tabel dbMatriks untuk Damri**



**Gambar 10. Cuplikan Tabel dbRute untuk Damri**

****

**Gambar 11. Cuplikan Tabel dbTempatB untuk Damri**

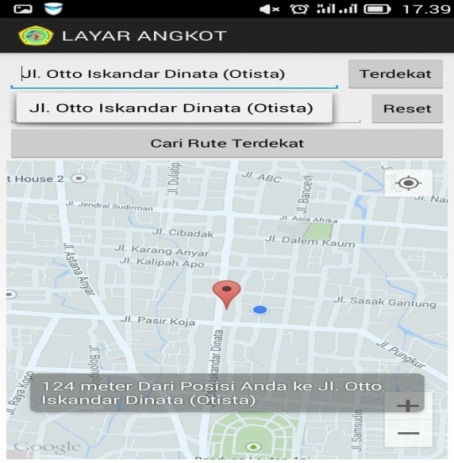
* 1. **Pengujian Sistem**

1. Pengujian proses *input* “lokasi asal” menjelaskan aktivitas pengujian untuk melihat apakah sistem akan menampilkan lokasi yang sesuai dengan masukan yang diberikan pengguna. Berikut pengujian proses *input* lokasi asal.



**Gambar 12. Proses Input Lokasi Asal**

1. 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.



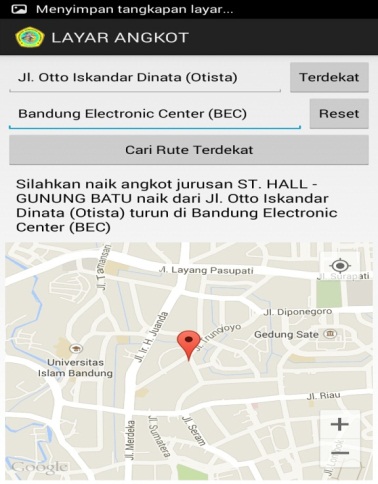
**Gambar 13. Pengguna Menekan Tombol Terdekat**

1. 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.



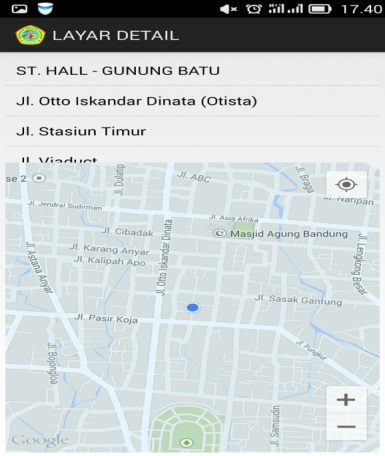
**Gambar 14.** Proses *Input* Lokasi Tujuan

1. 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.



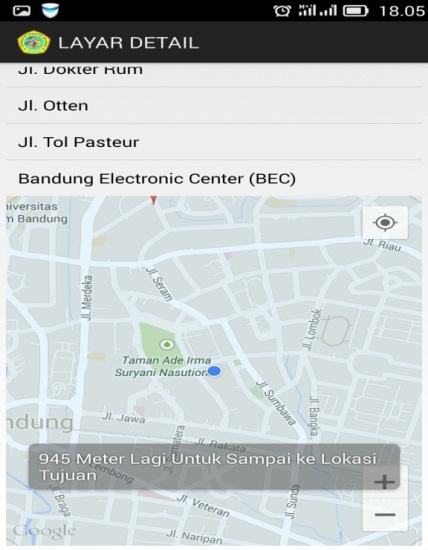
**Gambar 15.** Pengguna Menekan Tombol Terdekat

1. 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.



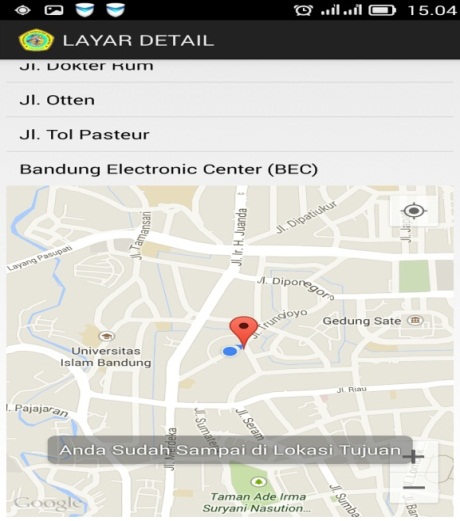
**Gambar 16. Tampilan Sistem Berhasil Menampilkan Detail Rute**

1. 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 17. Sistem Menampilkan Pesan disaat Pengguna Sudah Dekat dengan Lokasi Tujuan**

1. 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.



**Gambar18 Sistem Menampilkan Pesan Disaat Pengguna Sudah Sampai di Lokasi Tujuan**

1. **KESIMPULAN DAN SARAN**

**5.1. Kesimpulan**

Berdasarkan hasil pengujian 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 banyak nilai matriks yang menghasilkan nilai 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 penelitian 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.

**5.2. Saran**

Saran untuk pengembangan selanjutnya agar lebih baik yaitu :

1. Menggunakan algoritma yang lain selain *Algoritma Best-Path Planning for Public Transportation System* seperti metode djikstra dan algoritma A\*.
2. Dapat melakukan sistem pencarian lokasi yang tidak terdapat dalam *database.*
3. **DAFTAR PUSTAKA**

[1] Edy, W., Eng, M., Zaki, Ali. (2011). *Membuat Sendiri Aplikasi Android untuk Pemula*, PT Elex Media Komputindo. Jakarta.

1. Hadihardaja, Joetata. (1997). *Sistem Transportasi*. Gunadarma. Jakarta.
2. Huang, Yong, K., Ahuja, N., A., (1992). *A Potential Field Approach to Path Planning*. Robotics Ard Automation, IEEE Transactions On 8.1: 23-32.
3. Murya, Yosef. (2014). *Pemrograman Android Black Box*. Jasakom. Jakarta.
4. Liu, C. L. (2002). *Best-Path Planning for public Transportation System*. Proceedings of the Fifth International IEEE Conference on Intelligent Transportation System.
5. Pratiwi, Rizki. (2012). *Aplikasi Mobile Pencarian Rute Transportasi Umum dengan Algoritma Best-Path Planning pada Platform Android*.
6. Safaat, N. (2012). *ANDROID Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Informatika Bandung. Bandung.
7. Sinaga, Benyamin, L. (2005). *Pemrograman Berorientasi Objek dengan Java*, Gava Media.Yogyakarta.
8. (Februari 2015), *Trayek Angkot Bandung*,

<URL:http://bandung.go.id/rwd/index.php?fa=infokota.detail&id=17>

1. Februari 2015), *Trayek Damri Bandung*,

<URL:http://bandung.go.id/rwd/index.php?fa=infokota.detail&id=20//>