PERANCANGAN APLIKASI DAN IMPLEMENTASI PENCARIAN LOKASI TERDEKAT PADA KAWASAN TELKOM UNIVERSITY BERBASIS ANDROID

Kevin Prathama Nugraha[1], Dr. Ir. Bambang Hidayat, IPM[2], Leanna Vidya Yovita, S.T., M.T.

Program Studi Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University Bandung

kevinpnmail@gmail.com[1], avenir.telkom@gmail.com[2], leanna@telkomuniversity.ac.id[3]

Abstrak

Pencarian rute pada peta geografis mempunyai banyak aplikasi di banyak bidang. Rute yang dicari biasanya merupakan rute terpendek dalam artian dengan biaya kecil. Algoritma Dijkstra merupakan salah satu dari banyak algoritma pencarian rute pada peta geografis. Keunggulan dari algoritma ini adalah efisiensi waktu dengan tidak mengorbankan perhitungan biaya. Hal tersebut dimungkinkan karena selain memperhitungkan biaya, algoritma ini juga menggunakan estimasi untuk memprioritaskan arah pencarian yang benar.

Pada tugas akhir ini telah dikembangkan salah satu fitur LBS (Location Base Service) , yaitu *shortest path finder* atau pencari rute terpendek, dengan pembuatan suatu aplikasi berbasis android yang bertujuan memberikan layanan berupa informasi lokasi kepada pengguna perangkat *mobile* di kawasan kampus Telkom University. Pada sistem kerja aplikasi ini pengguna dari perangkat *mobile* akan dideteksi posisinya oleh GPS yang terintegrasi pada aplikasi ini. Selanjutnya Pengguna perangkat *mobile* menentukan tempat yang diinginkan. Setelah itu aplikasi akan memberikan rute terpendek ketempat tujuan dari posisi pengguna perangkat *mobile* yang sebelumnya telah terdeteksi oleh GPS.

Hasil yang dicapai dalam tugas akhir ini terbentuknya aplikasi berbasis android yang dapat menentukan rute terpendek dengan algoritma Dijkstra dari suatu titik ke suatu titik tujuan yang telah ditentukan pada kawasan kampus Telkom University.

Kata kunci : LBS, *Shortest Path*, Android, Algoritma Dijkstra, GPS

Abstract

Route search on geographic map has many applications in many fields. The searched route is usually the shortest route in terms of a small fee. Dijkstra's algorithm is one of many route search algorithms on a geographic map. The advantage of this algorithm is time efficiency by not compromising cost calculations. This is possible because in addition to taking into account costs, the algorithm also uses estimates to prioritize the correct search direction.

In this final project has been developed one of the features of LBS (Location Base Service), which is the shortest path finder or the shortest route finder, with the creation of an android based application that aims to provide location information services to mobile device users in the campus area of Telkom University. In the working system of this application the user of the mobile device will be detected its position by GPS integrated in this application. Next Mobile device users determine where they want to be. After that the application will provide the shortest route to the destination destination of mobile device user positions previously detected by GPS.

The results achieved in this final task the formation of android-based applications that can determine the shortest route with Dijkstra algorithm from a point to a destination point that has been determined on the campus of Telkom University.

Keywords: LBS, Shortest Path, Android, Dijkstra Algorithm, GPS

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pada zaman sekarang ini teknologi berkembang dengan pesat, contohnya aplikasi *android*. Aplikasi tersebut banyak membantu para pengguna perangkat *mobile* yang berbasis *android* dalam kehidupan sekarang ini, contohnya aplikasi pengolahan informasi. Pengolahan informasi dibutuhkan bagi daerah yang memiliki mobilitas tinggi. Seiring berkembangnya teknologi, masyarakat ingin kepraktisan untuk mengakses atau memperoleh pengolahan informasi. Pada tugas akhir ini penulis membuat aplikasi berbasis *android* untuk memberikan informasi rute terpendek dan petunjuk arah yang tepat untuk mencapai suatu tujuan.

Penggunaan aplikasi ini awalnya pengguna aplikasi akan dideteksi posisinya dengan GPS yang tersedia di aplikasi untuk menentukan jarak pengguna dengan tempat yang ingin dituju. Aplikasi ini memberikan informasi rute terpendek dan petunjuk arah yang tepat untuk mencapai tempat yang diinginkan

Saat ini ada beberapa metoda pencarian jalur terpendek, seperti algoritma Dijkstra. Untuk aplikasi ini akan menggunakan algoritma Dijkstra, untuk menghitung jarak antara posisi pengguna menuju tempat yang diinginkan

1.2. Permasalahan

Beberapa permasalahan pada tugas akhir ini dapat dirumuskan sebagai berikut :

1. Bagaimana menentukan rute terpendek menggunakan algoritma Dijsktra?
2. Bagaimana mengimplementasikan pencarian jalur terpendek berbasis android?
3. Bagaimana merancang design yang menarik?
4. Fitur fitur android apa saja yang digunakan untuk merealisasikan aplikasi tersebut?

1.3.Batasan Masalah

Pada tugas akhir ini penulis membatasi permasalahan sebagai berikut :

1. Kawasan yang dijadikan penelitian adalah kampus Fakultas Teknik TELKOM UNIVERSITY
2. Pembuatan aplikasi menggunakan *eclipse* Bahasa pemrograman *java*
3. Metoda penentuan jalur menggunakan algoritma Dijkstra.
4. *Platform android* yang digunakan adalah android versi 4.4
5. Simulasi aplikasi tersebut menggunakan *smartphone* yang mendukung sistem operasi android versi 4.4 atau versi yang lebih baru
6. Pendeteksian lokasi perangkat *mobile* dilakukan di peta yang telah dibuat
7. Pembuatan peta menggunakan layanan dari googlemap
8. Jalan yang digunakan di peta aplikasi ini adalah jalan untuk pejalan kaki

1.4.Tujuan

Tujuan pembuatan tugas akhir ini yaitu :

1. Memahami cara kerja algoritma Dijkstra.
2. Memahami pembuatan aplikasi android pada eclipse
3. Membuat design yang mudah dipahami pengguna
4. Menggunakan fitur LBS, GPS, googlemaps dan fitur java pada pembuatan aplikasi android

1.5. Metode Penelititan

Metode penelitian tugas akhir ini sebagai berikut :

1. Penelitian dilakukan dengan eksperimen bertujuan untuk menganalisis unjuk kerja pada aplikasi ini.
2. Menganalisis ketepatan algoritma yang digunakan untuk pencarian lokasi terdekat

1.6. Sistematika Penulisan

Tugas akhir ini dibagi beberapa topik pembahasan yang disusun secara sistematis sebagai berikut:

BAB I Pendahuluan

Bab ini membahas latar belakang, permasalahan, batasan masalah, tujuan, metode penelitian, dan sistematika penulisan.

BAB II Dasar Teori

Bab ini membahas android, GPS, algoritma Dijkstra, Googlemap

BAB III Rancangan

Bab ini berisi perancangan sistem dan skema pengujian

BAB IV Analisis dan Pengujian

Bab ini berisi tentang pengujian dan analisis aplikasi yang telah dibuat sebelumnya.

BAB V Kesimpulan dan Saran

Bab ini berisi kesimpulan dari implementasi aplikasi dan saran untuk pengembangan di masa mendatang.

BAB II
DASAR TEORI

2.1. Android

2.1.1. Android OS

Android adalah sistem operasi yang berbasis *open source*. Android banyak digunakan untuk sistem operasi *smartphone* maupun tablet untuk mendukung kecerdasan buatan dalam *smartphone* maupun tablet. Sistem operasi android sudah banyak dikembangkan untuk membuat aplikasi yang mendukung untuk kecerdasan sebuah *smartphone* maupun tablet. [3]

2.1.2. Android SDK

Android SDK adalah *tools* API (Application Programming Interface) yang diperlukan untuk mulai pengembangan aplikasi pada *platform* android menggunakan bahasa pemrograman *java*. Sebagai *platform* aplikasi netral, android memberi kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan aplikasi bawaan dari *smartphone* maupun tablet.

2.1.3. Arsitektur Android

Secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut :

2.1.3.1. Application dan Widgets

Application dan *widgets* merupakan sebuah *layer* yang berhubungan dengan aplikasi saja. Setelah mengunduh aplikasi kemudian melakukan instalasi dan

menjalankan aplikasi tersebut. Aplikasi dibuat menggunakan bahasa pemrograman *java*.

2.1.3.2. Application Framework

Application framework adalah *layer* dimana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem android. Komponen-komponen yang termasuk dalam *application framework* adalah sebagai berikut :

a. Viewss
b. *Content Provider*
c. *Resource Manager*
d. *Notification Manager*
e. *Activity Manager*

2.1.3.3. Libraries

Libraries adalah *layer* dimana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses ini untuk menjalankan aplikasinya. Berjalan diatas *kernel*, *layer* ini meliputi berbagai *libraries* antara lain :

a. *Libraries* media untuk pemutaran media video dan audio
b. *Libraries* untuk manajemen tampilan
c. *Libraries Graphics*
d. *Libraries* SQLite
e. *Libraries* SSL dan *webkit* terintegrasi dengan *web browser* dan *security*

2.1.3.4. Android Run Time

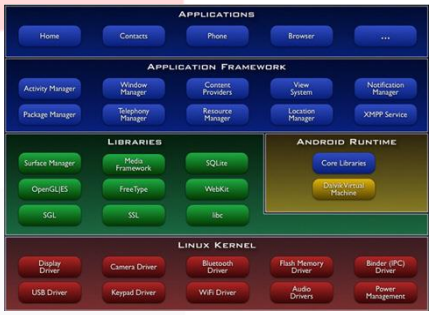
Layer yang membuat aplikasi android dapat dijalankan dimana dalam proses menggunakan implementasi *linux*. *Dalvik Virtual Machine*

(DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. Dalam *android run time* dibagi menjadi dua bagian yakni :

a. *Core libraries* yang berfungsi menerjemahkan bahasa *java* ke bahasa C maupun sebaliknya
b. *Dalvik virtual machine* merupakan virtual mesin berbasis register yang dioptimalkan untuk menjalani fungsi-fungsi secara efisien.

2.1.3.5. Linux Kernel

Linux kernel seperti pada gambar 2.1 adalah *layer* dimana inti dari *operating system* dari android itu berada. Berisi file-file sistem yang mengatur sistem *operating*, *memory*, *resource*, dan sistem-sistem operasi android lainnya.



Gambar 2.1 arsitektur linux kernel 1 [3]

2.1.4. Versi Android

2.1.4.1. Android versi 4.4 (KitKat)

Android versi 4.4 (KitKat) diluncurkan pada 31 Oktober 2013. Perbaikan android ini dari android sebelumnya yaitu perbaikan sistem penyimpanan sementara pada penggunaan memori, yang mana kinerja prosesor telah diminimalisir terhadap

penyimpanan *registry* data sementara RAM dan secara langsung akan ditampung oleh kapasitas memori internal yang tersedia, sehingga prosesor akan terasa lebih ringan. Kelebihan android versi 4.4 adalah sebagai berikut :

a. Warna layar transparan menyelimuti tombol virtual sentuh pada layar, sehingga tampilan warna yang terlihat sama dengan warna dari *wallpaper* yang digunakan.
b. Aplikasi Google *now* muncul sebagai salah satu bagian dari panel *homescreen*, sehingga lebih mudah digunakan
c. Dalam fitur yang disuguhkan Google telah mengubah latar *App Drawer* KitKat dari versi sebelumnya yang berwarna hitam menjadi transparan.
d. Ukuran ikon menu pada android KitKat sengaja dibuat lebih besar, sehingga sangat nyaman digunakan untuk dioperasikan oleh pengguna.
e. Pada bagian *widget* Google telah memberikan

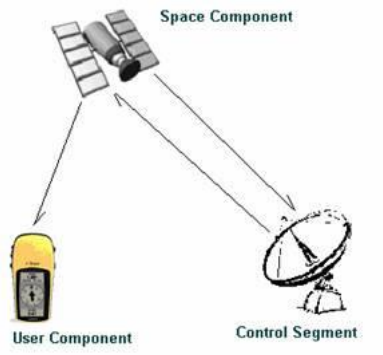
tampilan halaman menu supaya terlihat sederhana dengan menyingkirkan tab *Apps* melalui mode luring

f. Memiliki inovasi penghemat *batery*.
g. Memiliki kemampuan untuk meningkatkan memori di dalam *smartphone* yang bisa meningkatkan *responsive* dari layar *touchscreen*.
h. *Support* prosesor sebagai pendukung perangkat telah menggunakan *tricore*, sehingga memiliki kemampuan kecepatan tinggi pada kapasitas pemakaian.

2.2. GPS

GPS adalah sistem satelit navigasi dan penentuan posisi menggunakan satelit. Nama formal dari GPS yaitu NAVSTAR GPS, kependekan dari *Navigation Satellite Timing and Ranging Global Position System*. Sistem yang dapat digunakan banyak orang sekaligus dalam segala cuaca ini didesain untuk menentukan posisi, kecepatan tiga dimensi yang teliti, dan informasi mengenai waktu secara kontinyu diseluruh dunia. [1]

Pada dasarnya GPS terdiri dari tiga segmen utama, yaitu segmen angkasa terdiri dari satelit-satelit GPS, segmen sistem kontrol terdiri dari stasiun-stasiun pemonitor dan pengontrol satelit, dan segmen pengguna terdiri dari pemakaian GPS termasuk alat-alat penerima dan pengolah sinyal serta data GPS. Ketiga segmen dapat dilihat skemanya seperti gambar 2.2 berikut.



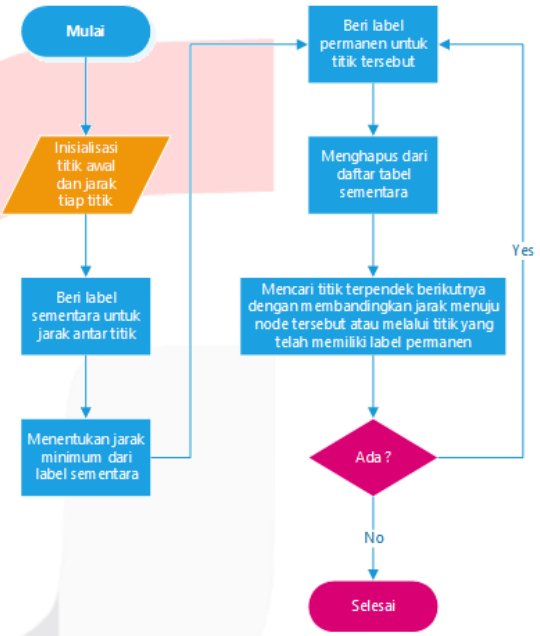
Gambar 2.2 skema GPS [1]

2.3. Algoritma Dijkstra

Algortima ini ditemukan oleh Edsger W. Dijkstra dan di publikasi pada tahun 1959 pada sebuah jurnal Numerische Mathematik yang berjudul "*A Note on Two Problems in Connexion with Graphs*". Algoritma ini sering digambarkan sebagai algoritma greedy (tamak). Sebagai contoh, ada pada buku *Algorithmics*. [5]

Djikstra merupakan salah satu varian bentuk algoritma popular dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks a ke z dalam graph berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyelesaian yang diberikan adalah infiniti (Tak Hingga). Pada algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute lintasan terpendek. [2]

Berikut flow chart dari algoritma Dijkstra.



Gambar 2.3 flowchart algoritma Dijkstra

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titk lainnya. Misalnya titik mengambarkan gedung dan garis menggambarkan jalan, maka algoritma Dijkstra melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik. Pertama-tama tentukan titik mana yang akan menjadikan node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu persatu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ke titik selanjutnya tahap demi tahap inilah

urutan logika dari algoritma Dijkstra seperti gambar 2.3 :

1. Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi)
2. Set semua node "Belum Terjamah" dan set node awal sebagai "Node keberangkatan"
3. Dari no keberangkatan, pertimbangkan node tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi 6+2=8. Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai "Node terjamah". Node terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set "Node belum terjamah" dengan jarak terkecil (dari node keberangkatan) sebagai "Node Keberangkatan" selajutnya dan lanjutkan dengan kembali ke step 3

2.4. Graf

Menurut Deiby T. Salaki (Suatu Graf G=(V,E) didefinisikan sebagai pasangan himpunan sisi dan simpul dengan V(G) = Himpunan simpul {v1, v2, ... , vn} dan E(G) = Himpunan sisi {e1, e2, ... , en}.

Setiap sisi berhubungan dengan satu atau dua simpul. Dua buah simpul dikatakan berhubungan atau bertetangga (adjacent) jika ada sisi yang menghubungkan keduanya.

Berdasarkan orientasi yang ada pada sisinya, graf dapat dikelompokkan menjadi dua yaitu: Graf berarah (direct graf) yaitu graf yang setiap sisinya diberikan arah sehingga untuk dua simpul vi dan vj, maka (vi,vj vj,vi) dan graf tak berarah (undirect graf) yaitu graf yang sisinya tidak mengandung arah sehingga untuk dua simpul vi dan vj, maka (vi,vj) (vj,vi). Selain itu juga dikenal graf berbobot yaitu graf yang sisinya memiliki bobot. [2]

2.4.1 Terminologi dasar

1. Bertetangga (adjacent) adalah 2 buah graf yang terhubung langsung dengan sebuah sisi.
2. Bersisian (insident) adalah sembarang sisi yang bersisian dengan simpul u dan v
3. Simpul terpencil (isolated vertex) adalah simpul yang tidak bertetanggaan dengan simpul-simpul lainnya
4. Graf kosong (null graph or empty graph) adalah graf yang himpunan sisinya adalah himpunan kosong
5. Derajat (degree) adalah suatu simpul pada graf tak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut
6. Lintasan (path) adalah panjang dari simpul awal hingga akhir
7. Siklus (cycle)/ sirkuit (circuit) adalah lintasan yang berawal dan berahir pada simpul yang sama
8. Terhubung (connected) adalah dua simpul yang terhubung
9. Graf berbobot (Weight graph) adalah graf yang setiap sisinya diberi sebuah harga (bobot)

2.4.2 Lintasan

1. Lintasan dan sirkuit euler

Lintasan euler adalah lintasan yang melalui masing-masing sisi di dalam graf tepat satu kali. bila lintasan tersebut kembali ke asal, sehingga membentuk lintasan tertutup maka disebut sirkuit euler.

2 Lintasan dan sirkuit Hamilton

Lintasan hamilton adalah lintasan yang melalui tiap simpul didalam graf tepat satu kali. bila lintasan itu kembali ke asal membentuk lintasan tertutup(sirkuit), maka lintasan tersebut adalah sirkuit hamilton. setiap graf lengkap adalah graf hamilton

3 Lintasan terpendek (shortest path)

Graf yang digunakan mencari lintasan terpendek adalah graf berbobot (weighted graph). ada beberapa macam persoalan lintasan terpendek antara lain :

1. Lintasan terpendek antara 2 buah simpul tertentu
2. Lintasan terpendek antara semua pasangan simpul
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
4. Lintasan terpendek antara dua buah simpul yang

melalui beberapa simpul tertentu.

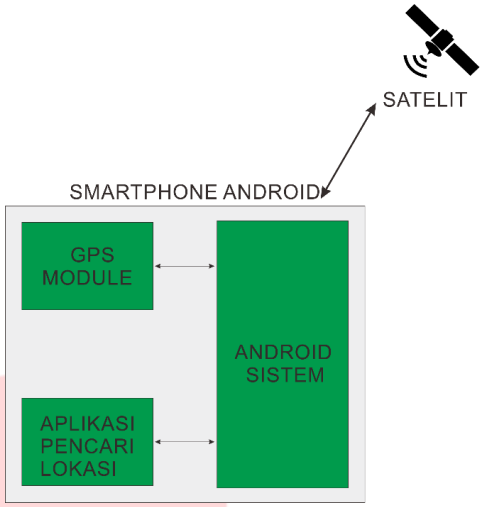
2.5. Googlemaps

Googlemaps adalah sebuah *online tool* yang memberikan *user* berbagai fitur-fitur map seperti tampilan *street maps*, arahan kemudi *point-to-point*, dan jalur-jalur untuk mencari lokasi bisnis di berbagai kota. Dengan tambahan *street map* dan *terrain view*, *satellite* atau *aerial views* dapat memberikan tampilan yang mudah dipahami *user* dan dapat diakses siapa saja melalui *online connection.* Dengan adanya googlemaps ini sangat memudahkan siapa saja untuk membuat aplikasi peta secara bebas untuk ikut berkontribusi, sehingga *maps* yang nanti dibuat bisa memudahkan yang lainnya kedepannya. [6]

BAB III
PERANCANGAN SISTEM

3.1.Perancangan Sistem Umum

Perancangan umum untuk pembuatan aplikasi ini adalah sebagai berikut

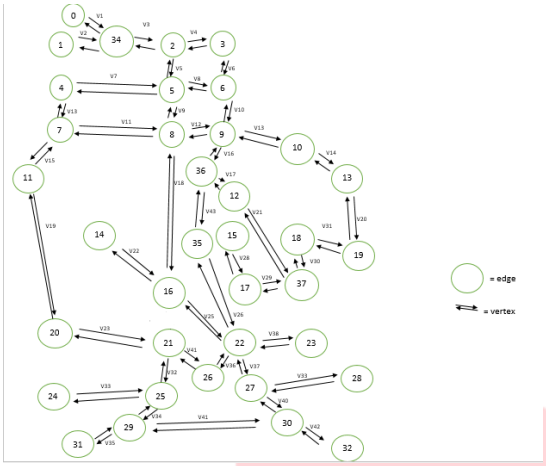


Gambar 3. 1 rancangan sistem secara umum

Dari gambar 3.1 diatas dapat dilihat aplikasi pencari lokasi memanggil gps *module* melalui android sistem untuk mencari lokasi sekarang, kemudian *smartphone* android menhubungi satelit untuk mendapatkan lokasi pengguna. Kemudian aplikasi akan menghitung jarak lokasi dari tempat pengguna berada dengan tempat yang ingin dikunjungi.

3.2.Skema Pengujian

Berikut gambar dari kawasan Telkom University menjadi edge dan vertex untuk pencarian jalur tercepat yang ingin dibuat



Gambar 3.2 kawasan Telkom University dalam Edge dan Vertex

Tabel 3.1 Tabel Edge

Tabel Edge	
No Edge	Nama Gedung / nama jalan
0	Gedung Tokong Nanas
1	Gedung LC
2	Simpang Gedung A
3	Gedung A
4	Gedung K
5	Simpang Gedung B
6	Gedung B
7	Simpang Gedung K
8	Simpang Gedung H
9	Simpang Jam
10	Simpang MSU
11	Gedung N
12	Gedung C
13	MSU
14	Gedung D
15	Gedung G
16	Simpang Gedung D
17	Simpang Gedung G
18	Simpang Parkiran
19	Parkiran
20	Gedung O
21	Simpang Gedung O
22	Simpang Gedung E
23	Gedung E
24	Gedung N
25	Simpang Gedung N
26	Gedung H
27	Simpang Gedung F
28	Gedung F
29	Simpang Student Center
30	Simpang Kantin
31	Student Center
32	Kantin
34	Samping LC
35	Samping Gedung G dan D
36	Samping Gedung C
37	Samping Gedung C-2

Tabel 3.2 Tabel Vertex

Tabel Vertex		
No Vertex	Jarak (M)	Edge
V1	243.69	0-34,34-0
V2	25.82	1-34,34-1
V3	46.87	2-34,34-2
V4	57.81	2-3,3-2
V5	55.49	2-5,5-2
V6	53.02	3-6,6-3
V7	68.17	4-5,5-4
V8	57.17	5-6,6-5
V9	43.13	5-8,8-5
V10	42.82	6-9,9-6
V11	72.11	7-8,8-7
V12	51.42	8-9,9-8
V13	72.95	9-10,10-9
V14	75.52	10-13,13-10
V15	57.39	7-11,11-7
V16	38.86	9-36,36-9
V17	29.91	36-12,12-36
V18	66.65	8-16,16-8
V19	79.85	11-20,20-11
V20	67.81	13-19,19-13
V21	87.62	12-37,37-12

V22	19.61	14-16,16-14
V23	34.20	20-21,21-20
V24	50.49	21-22,22-21
V25	43.55	16-22,22-16
V26	37.76	22-35,35-22
V27	24.25	35-17,17-35
V28	24.81	15-17,17-15
V29	41.30	17-37,37-17
V30	29.62	18-37,37-18
V31	52.82	18-19,19-18
V32	50.07	21-25,25-21
V33	49.88	24-25,25-24
V34	75.91	25-29,29-25
V35	80.85	29-31,31-29
V36	56.15	22-26,26-22
V37	67.27	22-27,27-22
V38	33.74	22-23,23-22
V39	25.99	27-28,28-27
V40	28.07	27-30,30-27
V41	73.48	29-30,30-29
V42	63.13	30-32,32-30
V43	66.87	35-36,36-35

Pada tugas akhir ini skema pengujian akan dibagi menjadi tiga bagian. Yaitu

1. Pencarian jalur tercepat dengan 2 lokasi Edge yang berdekatan

   Pada skema pengujian ini akan diuji pencarian lokasi dari gedung A ke Gedung B (Edge 3 ke Edge 6). Karena lokasi gedung A bisa langsung menuju ke Gedung B tanpa melalui Edge Perantara.

2. Pencarian jalur tercepat dengan 2 lokasi Edge yang memiliki Edge perantara

   Pada skema pengujian ini dilakukan dua pengujian, yaitu dari Gedung Tokong Nanas ke MSU (Edge 0 ke Edge 13), dan Student Center ke Kantin (Edge 31 ke Edge 32).

3. Pencarian jalur tercepat dengan 2 lokasi Edge terjauh

   Pada Skema pengujian ini dilakukan pengujian dari Gedung Tokong nanas ke Student Center (Edge 0 ke Edge 31).

3.3 Perancangan Aplikasi

Untuk perancangan aplikasi digambarkan pada gambar berikut ini :

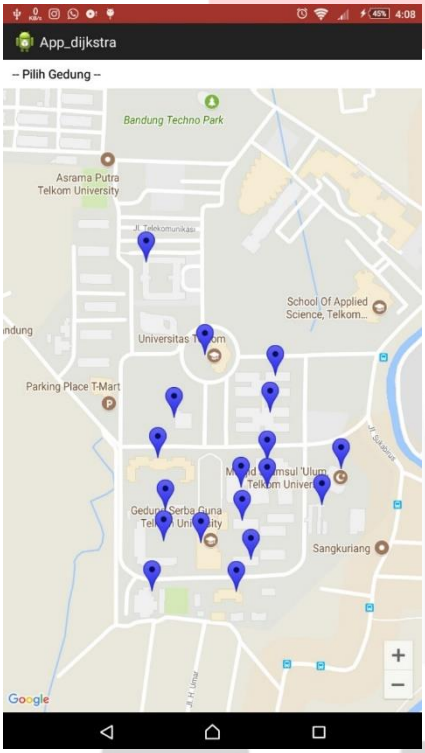
Gambar 3.3 perancangan aplikasi

Pada perancangan aplikasi dapat dilihat pada gambar 3.3 diatas. Pengguna membuka aplikasi lalu memilih lokasi tujuan dilanjutkan dengan memilih lokasi awal berada dan hasil jalur terpendek akan terbentuk.

BAB IV
ANALISIS DAN PENGUJIAN

4.1.Tampilan Aplikasi

Hasil implementasi dari aplikasi pencarian lokasi terdekat pada kawasan Telkom University berbasis Android memiliki tampilan aplikasi seperti berikut :

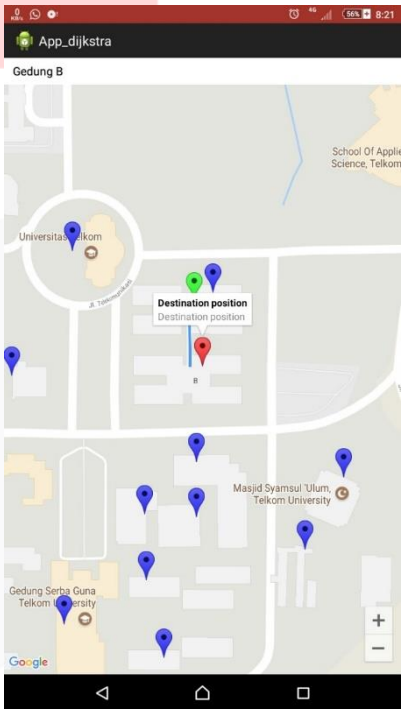


Gambar 4.1 Tampilan Aplikasi

4.2. Hasil Skema Pengujian

4.2.1. Pencarian jalur tercepat dengan 2 lokasi Edge yang berdekatan

Berikut hasil pencarian lokasi terdekat dari gedung A menuju gedung B dari aplikasi android yang telah dibuat

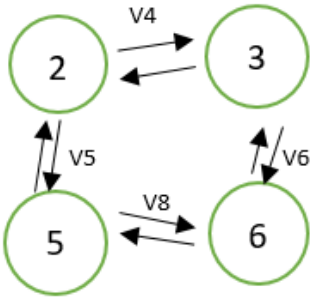


Gambar 4.2 Gedung A ke Gedung B

Pada gambar 4.2 dapat dilihat dari gedung A ke gedung B jalur yang dipilih adalah V6 dengan jarak 53,02 M.

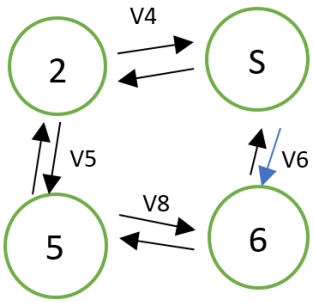
Hasil pencarian jalur tercepat secara hitung manual dijelaskan

melalui gambar-gambar dibawah ini

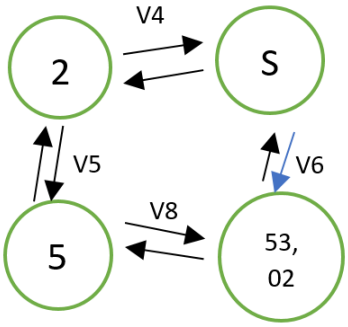


Gambar 4.3 Start Gedung A ke Gedung B

Dari gambar 4.3 diatas dapat dilihat pencarian dari gedung A menuju gedung B(Edge 3 menuju Edge 6)



Gambar 4.4 pemilihan jalur pertama dari gedung A ke gedung B



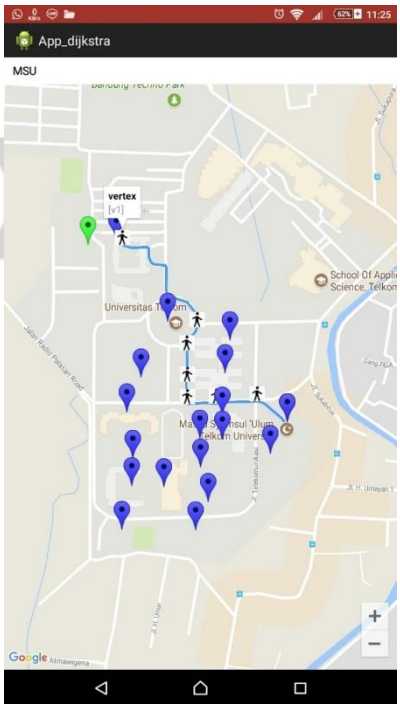
Gambar 4.5 Pencarian jalur tercepat dari Gedung A ke Gedung B selesai

Dari penjelasan gambar diatas, pencarian jalur tercepat dari gedung A menuju gedung B selesai dalam 1 langkah pencarian dikarenakan Edge dari gedung B langsung berhubungan dengan gedung A sehingga pencarian selesai dalam 1 tahap saja dengan jarak 53,02 M

4.2.2. Pencarian jalur tercepat dengan 2 lokasi Edge yang memiliki Edge perantara

- **Tokong Nanas Menuju MSU**

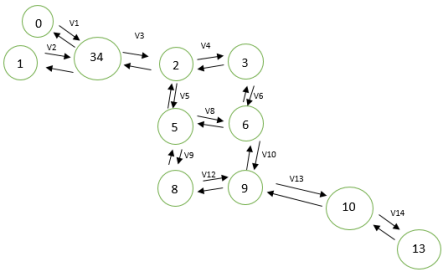
  Hasil pencarian jalur tercepat dari Tokong Nanas menuju MSU dari aplikasi adalah sebagai berikut



Gambar 4.6 Tokong Nanas ke MSU

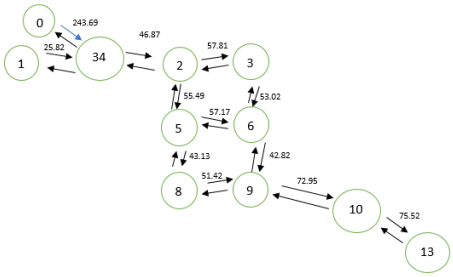
Dari gambar 4.6 diatas dapat dilihat jalur yang dipilih dari Tokong Nanas menuju MSU adalah V1, V3, V5, V9, V12, V13, dan V14 dengan jarak 589,07 M.

Hasil pencarian jalur tercepat secara hitung manual dijelaskan melalui gambar-gambar dibawah ini



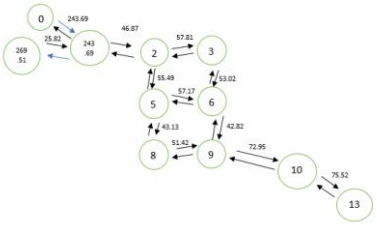
Gambar 4.7 Start jalur Tokong Nanas Menuju MSU

Dari gambar 4.7 diatas pencarian dari Tokong Nanas menuju MSU(Edge 0 menuju Edge 13)



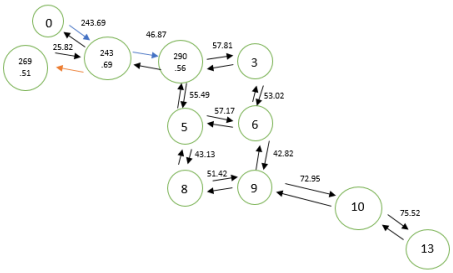
Gambar 4.8 tahap 1 pemilihan jalur Tokong Nanas menuju MSU

Pada gambar 4.8 dari edge 0 akan langsung memilih jalur ke edge 34 karena hanya edge 34 saja yang berhubungan langsung menuju edge 0.



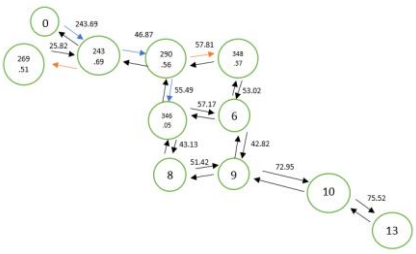
Gambar 4.9 tahap 2 pemilihan jalur Tokong Nanas menuju MSU

Pada gambar 4.9 dapat dilihat jika algoritma memilih menuju edge 1 terlebih dahulu dikarenakan bobot nilai yang kecil terlebih dahulu.



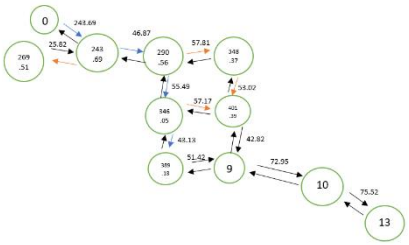
Gambar 4.10 tahap 3 pemilihan jalur Tokong Nanas menuju MSU

Pada gambar 4.10 pencarian berlanjut dengan memilih jalur lainnya yaitu menuju Edge 2.



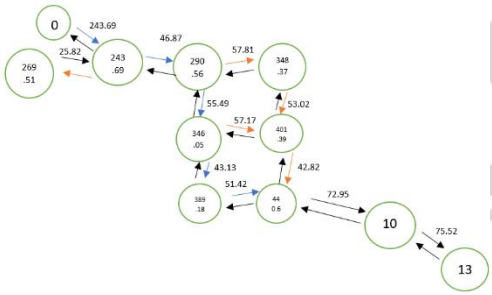
Gambar 4.11 tahap 4 pemilihan jalur Tokong Nanas menuju MSU

Pada gambar 4.11 algoritma memilih menuju edge 5 dikarenakan lebih kecil bobot nilainya dibandingkan dengan edge 3



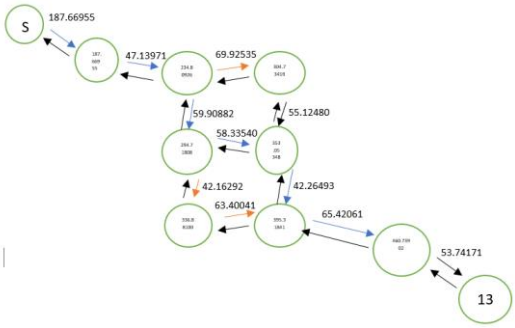
Gambar 4.12 tahap 5 pemilihan jalur Tokong Nanas menuju MSU

Pada gambar 4.12 algoritma memilih menuju edge 8 karena bobot lebih murah dibandingkan menuju edge 6. Dan edge 6 dapat nilai baru jika melewati edge 3 lebih murah dibandingkan melewati edge 5.



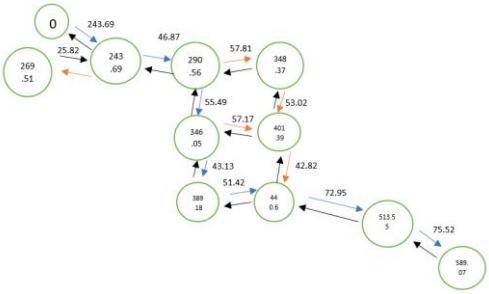
Gambar 4.13 tahap 6 pemilihan jalur Tokong Nanas menuju MSU

Pada gambar 4.13 terlihat dari edge 8 menuju edge 9 paling murah untuk dilewati. Sehingga edge 9 bernilai 440,6 M.



Gambar 4.14 tahap 7 pemilihan jalur Tokong Nanas menuju MSU

Pada gambar 4.14 dari edge 9 akan menuju edge 10.

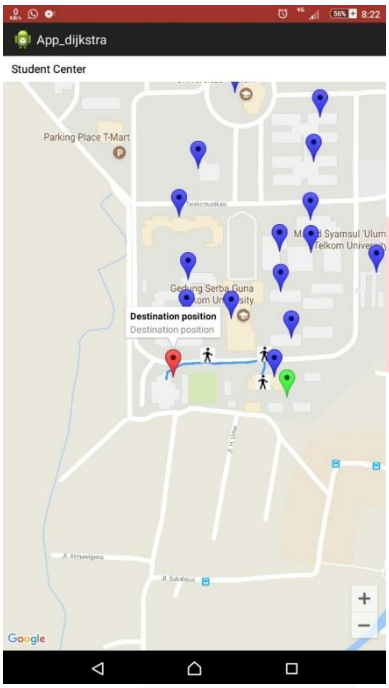


Gambar 4.15 tahap 8 pemilihan jalur Tokong Nanas menuju MSU

Dari penjelasan gambar 4.15, pencarian jalur tercepat dari Tokong Nanas menuju MSU selesai dalam 8 langkah pencarian dikarenakan adanya beberapa edge dalam pencariaan jalur tersebut dengan jarak 589,07 M

- **Kantin menuju Student Center**

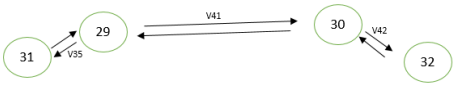
Hasil pencarian jalur tercepat dari Kantin menuju Student Center dari aplikasi adalah sebagai berikut



Gambar 4.16 Kantin ke Student Center

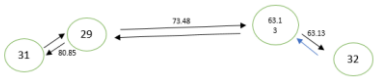
Pada gambar 4.16 dari Kantin menuju Student Center jalur yang dipilih yaitu V42, V40, V41, dan V35 dengan jarak 217,46 M.

Hasil pencarian jalur tercepat secara hitung manual dijelaskan melalui gambar-gambar dibawah ini



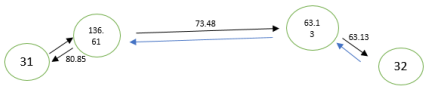
Gambar 4.17 Start jalur Kantin menuju Student Center

Pada gambar 4.17 diatas pencarian dari Kantin menuju Student Center(edge 32 menuju edge 31).



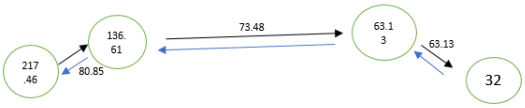
Gambar 4.18 tahap 1 pemilihan jalur Kantin menuju Student Center

Pada gambar 4.18 dilihat dari edge 32 menuju edge 30 dikarenakan hanya ada 1 saja jalur yang bisa dipilih.



Gambar 4.19 tahap 2 pemilihan jalur Kantin menuju Student Center

Pada gambar 4.19 terlihat dari edge 30 langsung menuju edge 29 dikarenakan hanya ada 1 saja jalur yang bisa dipilih.



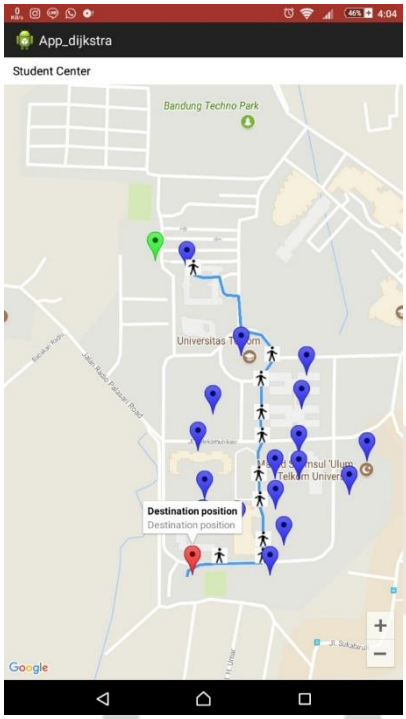
Gambar 4.20 tahap 3 pemilihan jalur Kantin menuju Student Center

Dari penjelasan gambar diatas, pencarian jalur tercepat dari Kantin menuju Student Center selesai

dalam 3 langkah pencarian dengan jarak 217,46 M

4.2.3. Pencarian jalur tercepat dengan 2 lokasi Edge terjauh

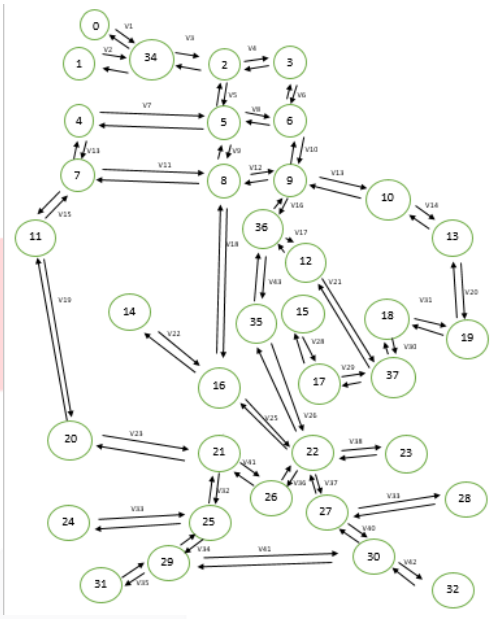
Hasil pencarian jalur tercepat dari Tokong Nanas menuju Student Center dari aplikasi adalah sebagai berikut



Gambar 4.21 Tokong Nanas ke Student Center

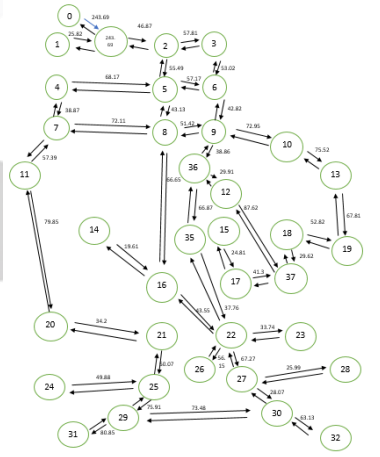
Pada gambar 4.21dari Tokong Nanas menuju Student Center jalur yang dipilih adalah V1, V3, V5, V9, V18, V25, V37, V40, V41,dan V35 dengan jarak 749,06 M.

Hasil pencarian jalur tercepat secara hitung manual dijelaskan melalui gambar-gambar dibawah ini



Gambar 4.22 Start Jalur Tokong Nanas menuju Student Center

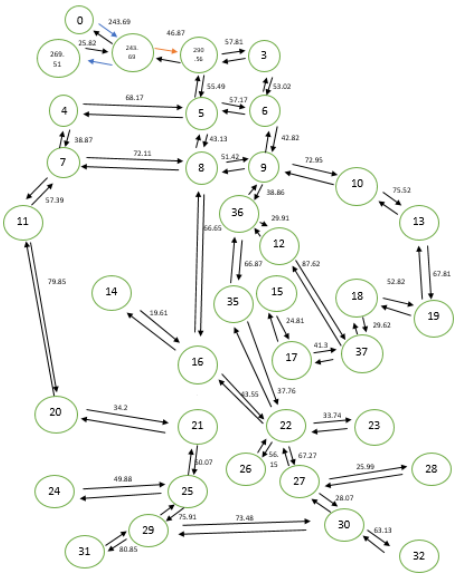
Pada gambar 4.22 pencarian dari Tokong Nanas menuju Student Center (edge 0 ke edge 31).



Gambar 4.23 tahap 1 pemilihan jalur Tokong Nanas Menuju Student Center

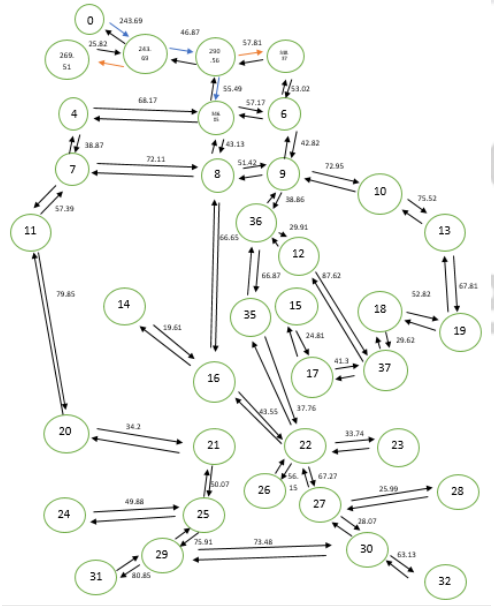
Pada gambar 4.23 diatas dari edge 0 langsung menuju edge 34

dikarenakan hanya ada 1 jalur yang bisa dipilih.



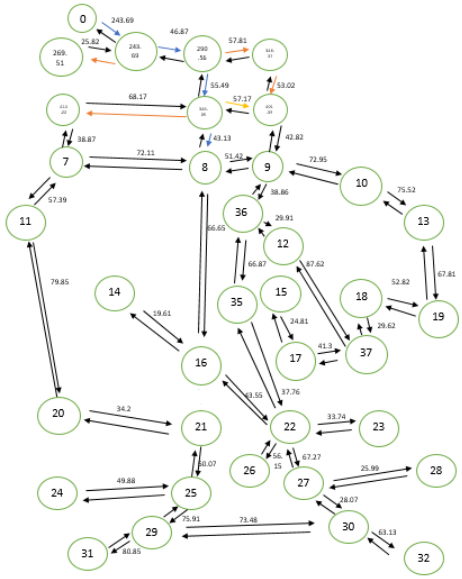
Gambar 4.24 tahap 2 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.24 dari edge 34 akan memilih edge 1 terlebih dahulu dikarenakan bobot yang lebih kecil dibandingkan dengan edge 2.



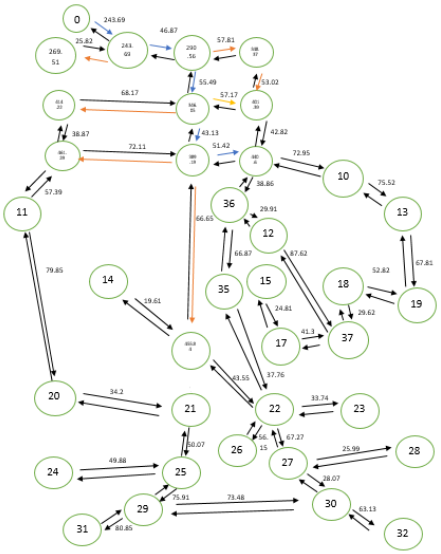
Gambar 4.25 tahap 3 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.25 dari edge 2 menuju edge 5 dikarenakan bobot yang lebih kecil dibandingkan edge 3.



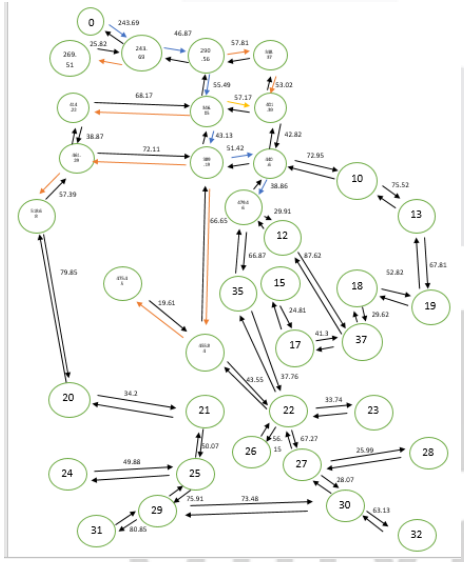
Gambar 4.26 tahap 4 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.26 dari edge 5 menuju edge 8 karena bobot paling kecil.



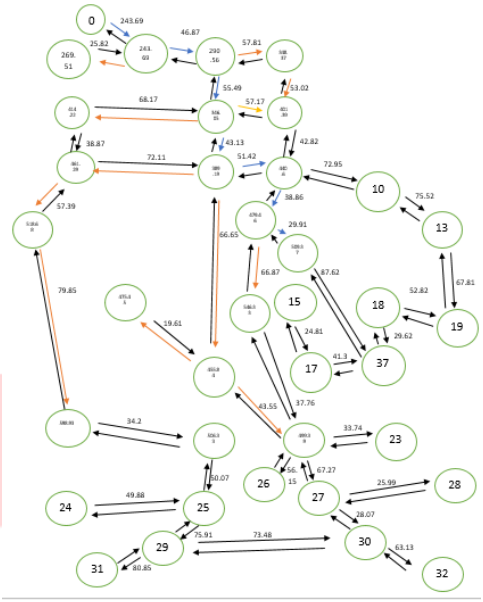
Gambar 4.27 tahap 5 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.27 dari edge 8 menuju edge 9 karena bobot paling kecil.



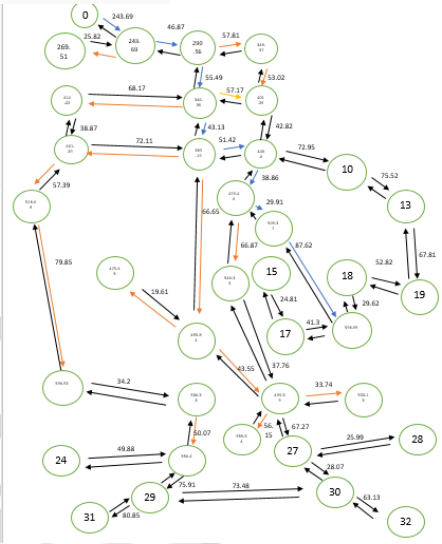
Gambar 4.28 tahap 6 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.28 dari edge 9 menuju edge 36.



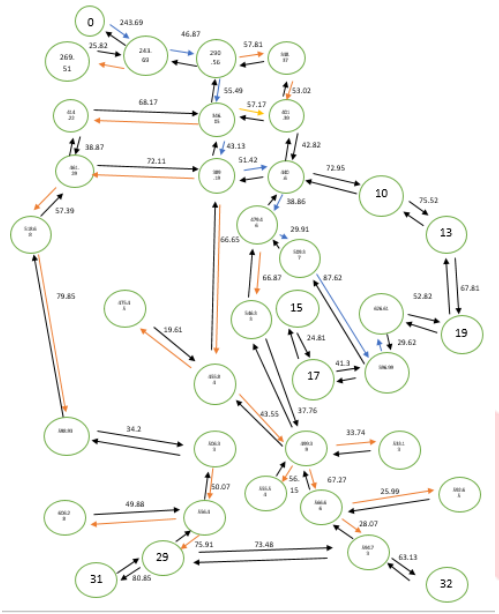
Gambar 4.29 tahap 7 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.29 edge 36 memilih ke edge 12 karena bobot yang paling kecil



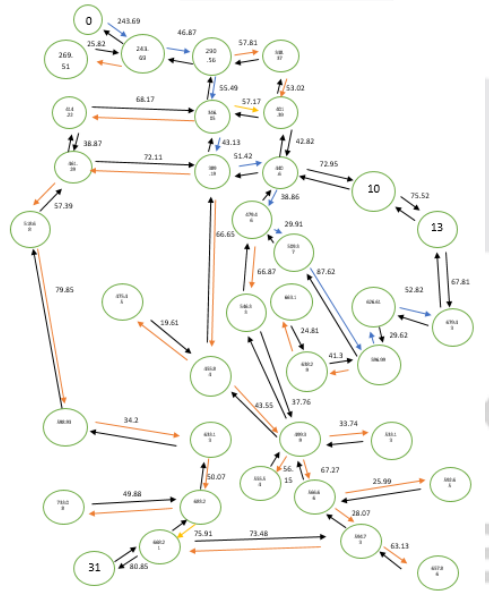
Gambar 4.30 tahap 8 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.30 dari edge 12 langsung memilih edge37 karena hanya 1 jalur saja.



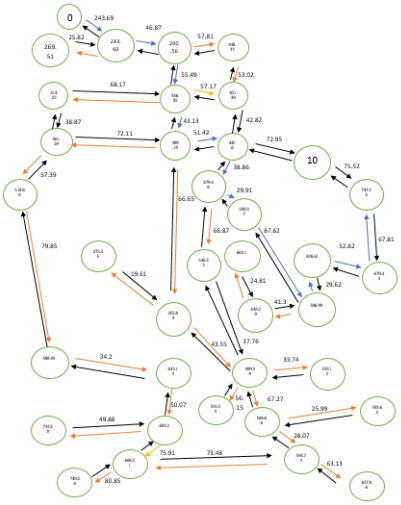
Gambar 4.31 tahap 9 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.31 dari edge 37 lanjut memilih ke edge 18.



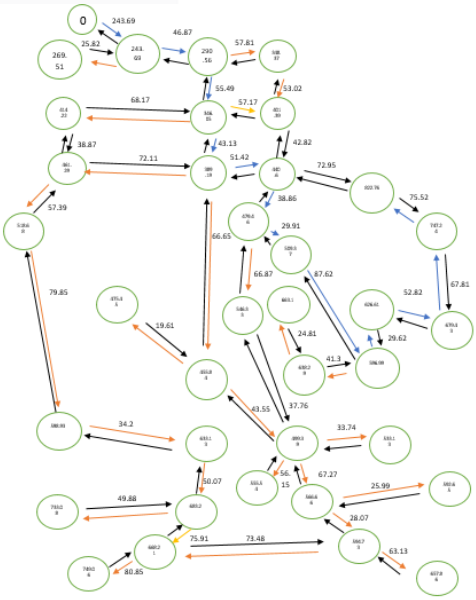
Gambar 4.32 tahap 10 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.32 dari edge18 menuju edge 19.



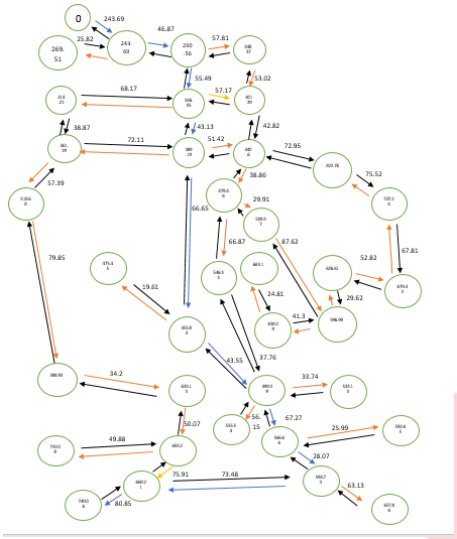
Gambar 4.33 tahap 11 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.33 dari edge 19 menuju edge 13.



Gambar 4.34 tahap 12 pemilihan jalur Tokong Nanas Menuju Student Center

Pada gambar 4.34 dari edge 13 ke edge 10. Dari sini belum selesai dikarenakan belum mencapai tujuan sehingga jalur yang dipilih akan menjadi gambar selanjutnya.



Gambar 4.35 tahap 13 pemilihan jalur Tokong Nanas menuju Student Center

Dari penjelasan gambar diatas, pencarian jalur tercepat dari Tokong Nanas menuju Student Center selesai dalam 13 langkah pencarian dengan panah biru dengan jarak 749,06 M

4.3. Analisa

Dari hasil penghitungan secara manual dan dengan aplikasi dapat disimpulkan bahwa program aplikasi pencarian lokasi terdekat pada kawasan Telkom University berbasis android dapat diimplementasikan. Dengan beberapa contoh skema pengujian dapat dilihat hasil dari aplikasi maupun dari hasil penghitungan menghasilkan jalur yang sama dan tepat.

Kedepannya aplikasi ini dapat dikembangkan untuk lebih lanjut agar informasi seputar kawasan Telkom University bias dipetakan secara lengkap sehingga jika ada mahasiswa baru maupun pendatang yang mendatangi kawasan Telkom University dapat mengetahui lokasi-lokasi seputar kawasan Telkom University dengan lengkap.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. Perancangan dan Implementasi aplikasi pencarian lokasi terdekat pada kawasan Telkom University berbasis android dapat diimplementasikan dengan baik dan tepat dengan menggunakan algoritma Dijkstra.
2. Dari hasil penghitungan secara manual dan dengan aplikasi dapat disimpulkan bahwa program aplikasi pencarian lokasi terdekat pada kawasan Telkom University berbasis android dapat diimplementasikan. Dengan

beberapa contoh skema pengujian dapat dilihat hasil dari aplikasi maupun dari hasil penghitungan menghasilkan jalur yang sama dan tepat.

5.2. Saran

1. Penulis sangat senang apabila ada yang ingin melanjutkan proyek ini. Jika menggunakan algoritma yang sama, lengkapi wilayah Telkom University yang belum ada dalam proyek ini
2. Bandingkan algoritma Dijkstra dengan algoritma lainnya seperti, Algoritma Floyd Warshall dan algoritma Bellman Ford. Cari algoritma yang terbaik
3. Kombinasikan beberapa algoritma untuk mendapatkan hasil yang terbaik

DAFTAR REFERENSI

[1] Abidin, hasanudin.2007.*Penentuan Posisi dengan GPS dan Aplikasinya*.Jakarta: Penerbit PT Pradnya Paramita

[2] Munir,rinaldi.2007.*Matematika Diskrit*.Bandung : Penerbit Informatika

[3] Safaat, nazruddin.2011.*Android Pemrograman Aplikasi Smartphone dan Tablet PC Berbasis Android*.Bandung : Penerbit Informatika

[4] Suyanto.2007.*Artificial Intelegent*.Bandung : Penerbit Informatika

[5]https://wirasetiawan29.wordpress.com/2015/04/02/tentang-algoritma-dijkstra/ diakses tanggal 2 oktober 2015

[6] https://www.google.com/maps/about/ diakses 10 Juli 2017

[7] http://digilib.uin-suka.ac.id/7247/ diakses 1 januari 2016