APLIKASI PETUNJUK LOKASI SARANA UMUM DI KOTA BANDUNG MENGGUNAKAN ALGORITMA DIJKSTRA BERBASIS ANDROID

Selvia Lorena Br. Ginting [1)] , Rizky Rainaldi [2)]
[1), 2)] Jurusan Sistem Komputer Universitas Komputer Indonesia, Bandung
selvialorena@yahoo.com[1)] , rizkyrainaldi@gmail.com [2)]

ABSTRAK

Kota Bandung merupakan kota metropolitan terbesar di Tatar Pasundan, Jawa Barat, sekaligus menjadi ibu kota provinsi tersebut. Sebagai ibu kota Jawa Barat Kota Bandung tentunya sering didatangi oleh pendatang dan wisatawan yang berasal dari daerah maupun luar kota, namun pendatang dan wisatawan tersebut kerap kali kesulitan mencari informasi sarana umum yang berada di Kota Bandung karena tidak mengetahui nama dan dimana lokasi sarana umum tersebut. Adapun pada penggunaan peta manual akan menyulitkan pengguna dalam menentukan rute terpendek untuk mencapai suatu lokasi tujuan. Oleh karena itu penulis memberikan solusi dengan membuat suatu aplikasi berbasis android yang dapat menunjukkan lokasi sarana umum di Kota Bandung dengan menggunakan algoritma Dijkstra yang merupakan salah satu algoritma pemecahan permasalahan pencarian rute terpendek. Cara kerja algoritma Dijkstra ialah memakai strategi greedy, di mana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul lain yang belum terpilih. Hasil yang diharapkan dari diciptakannya aplikasi ini adalah agar pengguna dapat lebih mudah mencari informasi dan lokasi sarana umum di Kota Bandung secara efektif dan efisien dengan tampilan aplikasi yang nyaman dan mudah digunakan.

Kata Kunci : Location Based Services, Google Maps, Algoritma Dijkstra

1 PENDAHULUAN

Kemajuan teknologi termasuk dalam bidang telekomunikasi turut mengambil peranan penting dalam menunjang segala aktifitas manusia. Dalam bidang telekomunikasi terdapat dua hal yang berkembang pesat beberapa tahun terakhir yaitu Internet dan ponsel. Untuk lebih lanjut perkembangan ponsel saat ini lebih condong ke arah *smartphone*. *Smartphone* sebagai alat bukan hanya untuk telepon maupun sms saja tetapi bertambah fitur-fitur yang dimiliki untuk memudahkan pengguna melakukan aktifitasnya. *Smartphone* sendiri merupakan mobile device yang dijalankan dengan sistem operasi tertentu, yang menggabungkan sebagian besar fungsi-fungsi pada *cellular phone, Personal Digital Assistant (PDA), audio player, digital camera* dan *camcorder, Global Positioning System (GPS)* dan *Personal Computer (PC)* sehingga mendukung kemampuan komputasi dan *connectivity* yang lebih modern. Telah banyak sistem operasi yang digunakan pada *smartphone* saat ini, sebut saja salah satunya Android yang dikembangkan oleh Google.

Kota Bandung merupakan kota metropolitan terbesar di Tatar Pasundan, Jawa Barat, sekaligus menjadi ibu kota provinsi tersebut. Sebagai ibu kota Jawa Barat Kota Bandung sering didatangi oleh pendatang dan wisatawan yang berasal dari daerah maupun luar kota, namun pendatang dan wisatawan tersebut kerap kali kesulitan mencari informasi sarana umum yang berada di Kota Bandung karena tidak mengetahui nama dan dimana lokasi sarana umum tersebut. Adapun pada penggunaan peta manual untuk mencapai suatu lokasi akan menyulitkan pengguna dalam menentukan rute terpendek untuk mencapai tempat lokasi tersebut. Oleh karena itu penulis memberikan solusi dengan membuat aplikasi petunjuk lokasi sarana umum di Kota Bandung menggunakan algoritma Dijkstra berbasis Android. Pada penelitian ini, penulis akan mengumpulkan data-data lokasi dan foto dari masing-masing tempat bahasan dan menginputkannya ke dalam aplikasi serta mengimplementasikan algoritma Dijkstra dalam penjalurannya. Adapun data tempat yang dibahas dalam penelitian untuk sarana umum ini yaitu berupa Mall, Kampus, Hotel dan Rumah Sakit. Data tempat, foto dan jalan diperoleh dari dinas terkait, *Google* dan *Google Maps.*

2 TEORI PENUNJANG

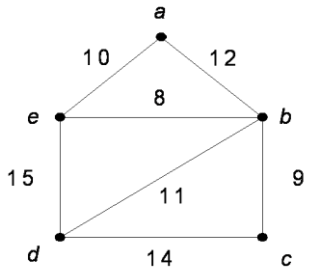
2.1 Sarana Umum

Menurut Kamus Besar Bahasa Indonesia sarana adalah segala sesuatu yang dapat dipakai sebagai alat dalam mencapai maksud atau tujuan[3]. Dikatakan "sarana umum" karena keberadaan wadah atau tempat ini bersifat mempermudah atau memperlancar terpenuhinya kebutuhan-kebutuhan bersama seluruh masyarakat, misalnya di bidang keamanan, komunikasi, rekreasi, kesehatan, administrasi publik, relijius dan sosial-budaya.

2.2 Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/vertex (*edges* atau *arcs*). Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek tersebut. Representasi visual dari graf adalah suatu objek dinyatakan sebagai noktah, bulatan atau titik, dan hubungan antar objek dinyatakan dengan garis. [4].

Secara matematis graf didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi G = (V, E), yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertex* atau *node*) dan E adalah himpunan sisi *(edge)* yang menghubungkan sepasang simpul. Simpul *(vertex)* pada graf dapat dinyatakan dengan huruf, bilangan atau gabungan keduanya. Sedangkan sisi-sisi yang menghubungkan simpul u dengan simpul v dinyatakan dengan pasangan (u, v) atau dinyatakan dengan lambang e1, e2, e3 dan seterusnya.[1]



Gambar 2.1 Contoh Sebuah Graf Sederhana

2.3 Algoritma Dijkstra

Algoritma Dijkstra adalah sebuah algoritma untuk memecahkan permasalahan jalur terpendek *(shortest path problem)* pada sebuah graf berarah *(directed graf)* atau graf tidak berarah *(undirected graf)* dengan bobot-bobot sisi *(edge weights)* yang bernilai tidak negatif. Bila node dari sebuah graf melambangkan kota-kota dan bobot sisi *(edge weights)* melambangkan jalur antara kota-kota tersebut, maka algoritma Dijkstra dapat digunakan untuk menemukan jalur terpendek antara dua kota. [5]

Algoritma Dijkstra merupakan *greedy algorithm* yang berfungsi menentukan rute terpendek berdasarkan bobot terkecil dari satu titik ke titik lain nya. Karena merupakan *greedy algorithm* atau algoritma rakus maka algortima ini hanya memberikan solusi terbaik tanpa memberikan alternatifnya. Adapun nama algoritma Dijkstra sendiri berasal dari penemunya yaitu Edsger Dijkstra pada tahun 1959. [1]

Berikut *Pseudocode* algoritma Dijkstra :

algorithm *DijkstraShortestWeightedPath*(*G*, *s*)
_ ***pre-cond_:*** *G* is a weighted (directed or undirected) graph, and *s* is one of its
nodes.
_ ***post-cond_:*** π specifies a shortest weighted path from *s* to each node of *G*, and
d specifies their lengths.
begin
$d(s) = 0, \pi(s) = _$
for other *v*, $d(v)=\infty$ and $\pi(v) =$ *nil*
handled = ∅
notHandled = priority queue containing all nodes. Priorities given by $d(v)$.
loop
loop-invariant: See above.
exit when *notHandled* = ∅
let *u* be a node from *notHandled* with smallest $d(u)$
for each *v* connected to *u*
foundPathLength = $d(u) + w_{_u,v_}$
if $d(v) >$ *foundPathLength* then
$d(v) =$ *foundPathLength*
$\pi(v) = u$
(update the *notHandled* priority queue)
end if
end for
move *u* from *notHandled* to *handled*
end loop
return $_d, \pi_$
end algorithm.

Adapun langkah-langkah dari algoritma Dijkstra sesuai *pseudocode* yaitu:

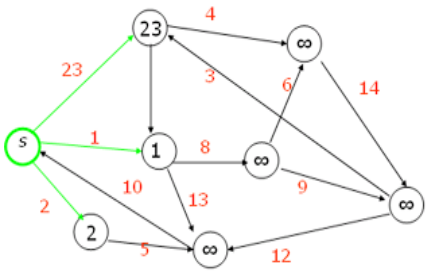
1. Langkah pertama yaitu menetapkan node awal sebagai status ditemukan (*found*) dan kemudian dikunjungi atau ditangani(*handled*)

2. Langkah kedua yaitu dilakukan pencarian terhadap setiap node yang dapat dicapai secara langsung dari node yang sedang dikunjungi

3. Langkah ketiga yaitu:
- Apabila node yang didapatkan pada langkah kedua belum pernah ditemukan, maka rubah statusnya menjadi ditemukan
- Apabila node yang didapatkan sudah pernah ditemukan maka lakukan *update* pada bobotnya, ambil bobot yang lebih kecil

4. Langkah keempat yaitu dilakukan pencarian terhadap node yang memiliki bobot paling kecil dari semua node yang berada pada status ditemukan kemudian mengunjunginya.

5. Lakukan *looping* secara berurutan pada langkah kedua, ketiga dan keempat sampai semua node ditemukan.



Gambar 2.2 Contoh Graph Algoritma Dijkstra

2.4 Android

Android adalah sebuah sistem operasi yang berbasis Linux yang mencakup sistem operasi, *middleware*, dan apikasi. Android menyediakan *platform* terbuka bagi para pengembang sehingga dapat menciptakan aplikasi dengan leluasa untuk digunakan oleh para pengguna smartphone android.

Sebagai sistem operasi untuk perangkat mobile, Android disebut sebagai platform mobile pertama yang Lengkap, Terbuka, dan Bebas.

1. Lengkap (*Complete Platform*) : Para desainer dapat melakukan pendekatan yang komperhensif ketika mereka sedang mengembangkan platform android.
2. Terbuka (*Open Source Platform*) : Platform Android disediakan melalui lisensi open source. Pengembang dapat dengan bebas untuk mengembangkan aplikasi.
3. Bebas (*Free Platform*) : Android adalah platform atau aplikasi yang bebas untuk developer. Tidak ada lisensi atau biaya royalti utuk dikrmbangkan pada platform Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk android dapat didistribusikan dan diperdagangkan dalam bentuk apa pun. [2]

2.5 Google Maps

Google maps merupakan sebuah layanan peta dunia virtual berbasis web yang disediakan oleh Google. Layanan ini gratis dan dapat ditemukan di http://maps.google.com. *Google Maps* menawarkan peta yang dapat digeser *(panned)*, diperbesar (*zoom in*), diperkecil (*zoom out)*, dapat diganti dalam beberapa mode (*map, satellite, hybrid,* dan lain-lain). [3]

2.6 LBS (Location Based Service)

Sistem Layanan Berbasis Lokasi, atau lebih dikenal dengan *Location-Based Service* (LBS), menggabungkan antara proses dari layanan mobile dengan posisi geografis dari penggunanya. Poin pentingnya adalah ketika posisi target, di mana sebuah target bisa jadi adalah pengguna LBS itu sendiri atau entitas lain yang tergabung dalam suatu layanan.[2]

3 ANALISIS & PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Analisis merupakan tahapan yang paling penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya.

3.1.2 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses aktivitas yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik dan sesuai kebutuhan. Berikut adalah kebutuhan sistem yang diperlukan :

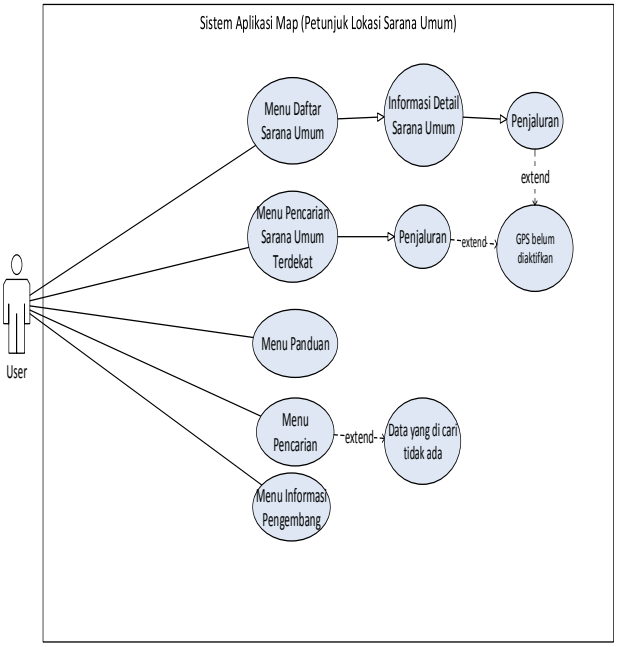
1. Aplikasi petunjuk lokasi dapat berjalan pada sistem operasi Android.
2. Aplikasi dapat mengakses *Google Map* dan mendeteksi lokasi *user* via GPS.
3. Aplikasi dapat menampilkan informasi atau data sesuai database sistem aplikasi.
4. Aplikasi dapat menampilkan penjaluran rute terpendek dengan algoritma Dijkstra dan *Google maps* sebagai tampilan peta.
5. Aplikasi dapat menampilkan lokasi sarana umum terdekat berdasarkan kategori dan koordinat awal.
6. Aplikasi dapat menampilkan fasilitas panduan penggunaan aplikasi.

3.2 Pemodelan Sistem

Pada analisis fungsional ini menggunakan pemodelan UML (*Unifiled Modeling Language*). Pemodelan disini merupakan penggambaran diagram model yang terdiri dari *Use Case* diagram, *Sequence* Diagram dan *Collaboration* Diagram.

3.2.1 Use Case Diagram

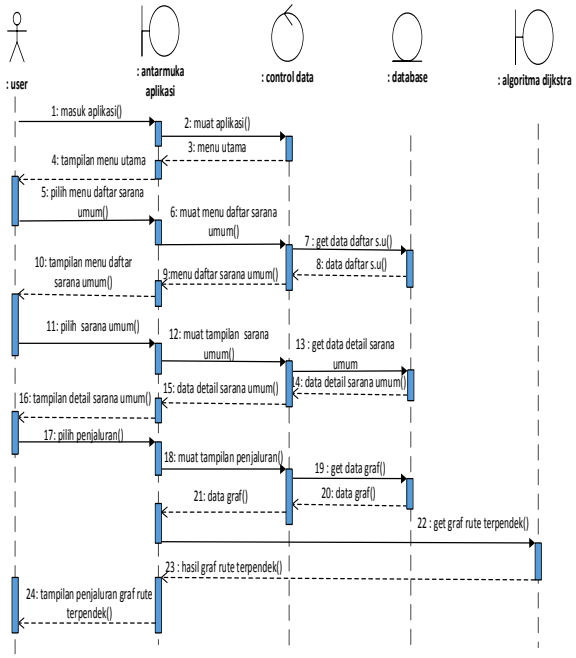
Use Case diagram terdiri dari tiga bagian yaitu definisi *actor*, definisi *use case* dan *Skenario use case*. Berikut adalah *use case diagram* yang dirancang untuk aplikasi petunjuk lokasi sarana umum :



Gambar 3.1 Diagram Use Case Sistem

3.2.2 Sequence Diagram

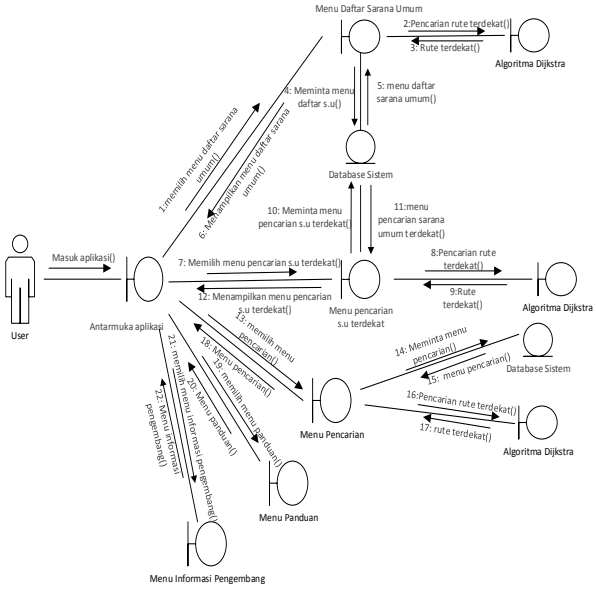
Sequence diagram menggambarkan interaksi antar objek didalam dan disekitar sistem. Pada Sequence diagram ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan pada sistem sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.



Gambar 3.2 Sequence Diagram

3.2.3 Collaboration Diagram

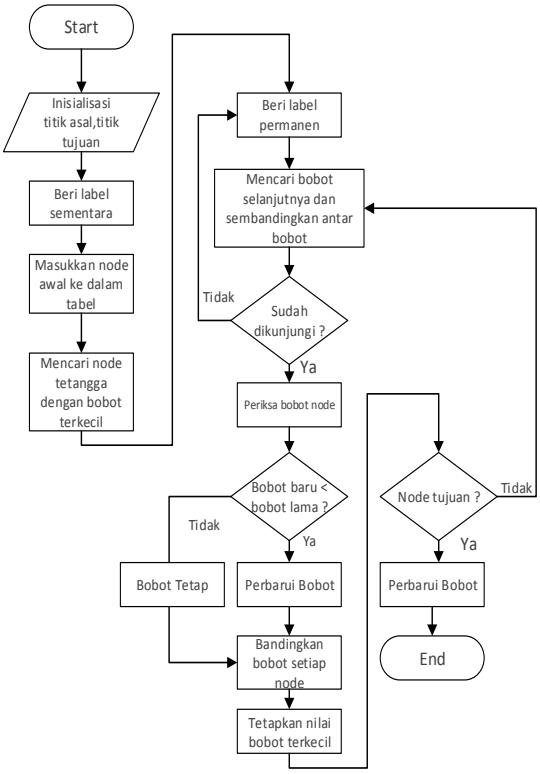
Diagram Collaboration merupakan diagram interaction. Diagram membawa informasi yang sama dengan diagram sequence, tetapi lebih memusatkan atau memfokuskan pada kegiatan objek dari waktu pesan itu dikirimkan.



Gambar 3.3 Collaboration Diagram

3.2.4 Analisis Algoritma

Pencarian rute terpendek termasuk ke dalam materi teori graf. Algoritma yang sangat terkenal untuk menyelesaikan persoalan ini adalah algoritma Dijkstra. Algoritma ini ditemukan oleh seorang ilmuwan komputer berkebangsaan Belanda yang bernama Edsger Dijkstra. "Dijkstra" diartikan sebagai algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Contoh penerapan algoritma dijkstra adalah lintasan terpendek yang menghubungkan antara dua kota berlainan dengan jarak tertentu. Cara kerja algoritma dijktra memakai strategi greedy, di mana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul lain yang belum terpilih. Algoritma dijkstra membutuhkan parameter lokasi asal, jarak dan lokasi tujuan. Secara singkat algoritma Dijkstra dapat dijelaskan dengan *flowchart* seperti dibawah ini :



4 IMPLEMENTASI & PENGUJIAN

4.1 Implementasi

Implementasi aplikasi petunjuk lokasi sarana umum di Kota Bandung menggunakan algoritma Dijkstra berbasis android meliputi lingkungan implementasi perangkat keras, perangkat lunak dan implementasi terhadap antarmuka. Dalam tahap ini dilakukan transformasi algoritma Dijkstra kedalam kode-kode program yang sesuai dengan bahasa pemrograman yang digunakan yang dalam kasus ini adalah bahasa pemrograman android dan java.

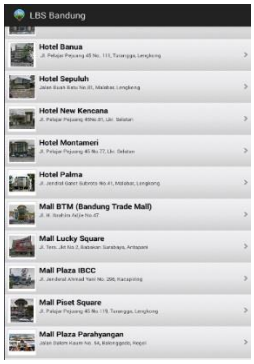
4.1.1 Implementasi Antarmuka Aplikasi

Dalam implementasi antarmuka, dilakukan penerapan hasil perancangan yang sudah dijelaskan pada bab sebelumnya. Berikut adalah hasil perancangan antarmuka yang dibangun.



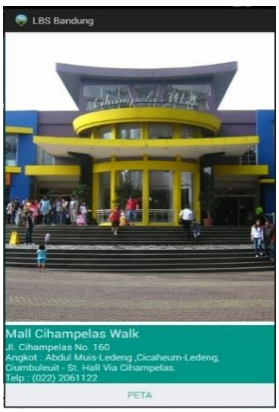
Gambar 4.1 Antarmuka Menu Utama

Gambar 4.1 merupakan tampilan layar utama pada aplikasi yang berfungsi untuk menampilkan nama aplikasi dan menu-menu aplikasi. Pada menu pertama adalah menu daftar seluruh sarana umum, menu kedua adalah menu pencarian sarana umum terdekat berdasarkan kategori, menu ketiga adalah menu panduan, menu keempat adalah menu pencarian dan menu kelima adalah menu informasi pengembang.



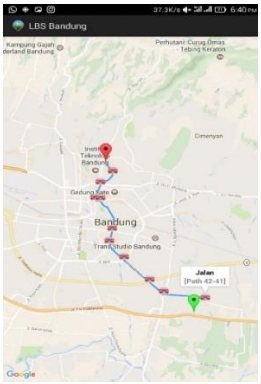
Gambar 4.2 Antarmuka Daftar Sarana Umum

Gambar 4.2 merupakan tampilan daftar sarana umum pada aplikasi yang berfungsi untuk menampilkan sarana umum apa saja yang terdapat pada aplikasi ini. Untuk melihat informasi detail sarana umum pengguna dapat memilih salah satu sarana umum tersebut.



Gambar 4.3 Antarmuka Informasi Detail Sarana Umum

Gambar 4.3 merupakan tampilan informasi detail sarana umum, pada layar tersebut aplikasi menampilkan informasi detail menganai sarna umum yang meliputi nama jalan, nomor telepon dan angkot yang melewati sarana umum tersebut. Pada layar ini juga terdapat tombol peta yang digunakan untuk mencari rute atau jalur yang dapat dilalui pengguna dari posisi asal menuju posisi sarana umum yang dipilih.



Gambar 4.4 Antarmuka Penjaluran

Gambar 4.4 merupakan tampilan penjaluran pada aplikasi yang berfungsi sebagai penunjuk jalan yang dapat dilalui pengguna dari posisi awal ke lokasi sarana umum yang diinginkan. Posisi awal pengguna ditandai dengan node hijau sedangkan posisi sarana umum ditandai dengan node merah. Penjaluran tersebut dipilih dan diseleksi oleh algoritma Dijkstra untuk menemukan rute terpendek yang dapat dilalui pengguna. Aplikasi juga akan memberitahu pengguna apabila pengguna sudah dekat pada lokasi tujuan.



Gambar 4.5 Antarmuka Pencarian Sarana umum terdekat berdasarkan kategori

Gambar 4.5 merupakan tampilan pencarian sarana umum terdekat berdasarkan kategori. Pada layar tersebut pengguna dapat mencari sarana umum terdekat berdasarkan kategori dan koordinat awal yang telah ditentukan.



Gambar 4.6 Antarmuka Menu Panduan

Gambar 4.6 merupakan tampilan menu panduan pada aplikasi yang berfungsi sebagai penunjuk bagaimana cara menjalankan aplikasi.



Gambar 4.7 Antarmuka Menu Informasi

Gambar 4.7 merupakan tampilan menu informasi pada aplikasi yang berfungsi untuk menampilkan informasi seputar pengembang aplikasi.

4.2 Pengujian Sistem

4.2.1 Pengujian Alpha

Pengujian *alpha* dilakukan dengan menggunakan metode *Black box*, yaitu pengujian yang berfokus pada persyaratan fungsional perangkat lunak untuk melihat apakah program aplikasi menghasilkan output yang diinginkan dan sesuai dengan fungsi dari program aplikasi yang dibuat tersebut. Pengujian fungsional dilakukan oleh pengembang. Berikut ini terdapat beberapa pengujian pada aplikasi yang dibangun.

Tabel 4.1 Skenario Pengujian

No	Komponen Pengujian	Skenario Pengujian	Jenis Pengujian
1	Menu Utama	Menampilkan menu utama aplikasi.	*Black Box*
2	Menu Daftar Sarana Umum	Menampilkan menu daftar sarana umum.	*Black Box*
3	Tampilan Informasi Detail Sarana Umum	Menampilkan informasi detail sarana umum	*Black Box*
4	Penjaluran	Menampilkan penjaluran berupa graf dari posisi awal ke posisi tujuan	*Black Box*
5	Menu Pencarian Sarana Umum Terdekat	Menampilkan menu pencarian sarana umum terdekat berdasarkan posisi awal dan kategori.	*Black Box*
6	Menu Panduan	Menampilkan menu panduan penggunaan aplikasi.	*Black Box*
7	Menu Pencarian	Menampilkan sarana umum berdasarkan pencarian nama	*Black Box*
8	Menu Informasi Pengembang	Menampilkan menu informasi penggembang aplikasi.	*Black Box*

4.2.2 Pengujian Beta

Pengujian beta dilakukan pada satu atau lebih pelanggan oleh pemakai akhir perangkat lunak dalam bentuk kuesioner. Dimana kuesioner ini digunakan untuk mengetahui bagaimana tanggapan para responden mengenai aplikasi petunjuk lokasi sarana umum di Kota Bandung yang dibuat oleh penulis dari segi tampilan, performa, penggunaan dan informasi yang didapatkan dari aplikasi tersebut yang mana hasil dari responden akan diolah untuk ditarik kesimpulan. Metode yang digunakan oleh penulis untuk mengetahui bagaimana tanggapan para responden tersebut dengan menggunakan "Skala Likert".Skala *likert* adalah metode perhitungan yang digunakan untuk keperluan riset atas jawaban setuju atau tidaknya seorang responden terhadap suatu pertanyaan.Untuk menghitung skor maksimum setiap jawaban, dengan mengalikan skor dengan jumlah keseluruhan responden, yaitu skor dikali 30 responden. Nilai skor maksimum dapat dilihat pada tabel 4.2

Tabel 4.2 Skor Maksimum

Kategori Jawaban	Keterangan
0% - 20%	Sangat Tidak Setuju
21% - 40%	Tidak Setuju
40% - 60%	Ragu-ragu
61% - 80%	Setuju
81% - 100%	Sangat Setuju

Berdasarkan hasil kuisioner dari keenam pertanyaan yang sudah diolah dapat disimpulkan bahwa aplikasi petunjuk lokasi sarana umm berada pada kategori sangat setuju. Adapun perhitungan secara keseluruhan dapat dilihat pada tabel 4.3

Tabel 4.3 Pengolahan hasil skala kuisioner

Pertanyaan	Nilai Presentase (%)	Keterangan
1	86%	Sangat Setuju
2	89%	Sangat Setuju
3	91%	Sangat Setuju
4	85%	Sangat Setuju
5	85%	Sangat Setuju
6	88%	Sangat Setuju
Total Presentase	524%	Sangat Setuju
Rata-rata	87,33%	

Dari hasil pengujian *beta* dapat diperoleh hasil bahwa pengguna aplikasi dapat mengetahui teknologi *Location Based Service* dari aplikasi yang telah digunakan, lalu pengguna dapat terbantu mencari informasi dan jalur mengenai sarana umum yang berada di Kota Bandung, pengguna sangat setuju bahwa aplikasi dapat menghemat waktu dalam mencari info sarana umum di Kota Bandung, pengguna setuju bahwa aplikasi dapat menggantikan fungsi dari peta manual, pengguna pun setuju bahwa tampilan dan fitur aplikasi mudah dan nyaman digunakan dan responden setuju bahwa informasi sarana umum & peta dalam aplikasi sesuai dengan aslinya.

5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian sistem pada aplikasi petunjuk lokasi sarana umum di Kota Bandung, maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan hasil analisa pengujian Alpha, aplikasi petunjuk lokasi sarana umum ini berfungsi dengan baik dan dapat diterima sesuai dengan skenario rencana pengujian. Serta keluaran dari aplikasi ini sudah seperti yang diharapkan.
2. Aplikasi dapat membantu untuk mencari informasi dan jalur mengenai sarana umum yang berada di Kota Bandung secara efektif, hal ini hal ini ditunjukan dengan persentase 85% dari hasil kuesioner kepada pengguna aplikasi.
3. Dengan adanya aplikasi ini pengguna dapat menghemat waktu dalam mencari info sarana umum di Kota Bandung, hal ini ditunjukan dengan persentase 91% dari hasil kuesioner kepada pengguna aplikasi.

5.2 Saran

Adapun saran untuk meningkatkan kinerja atau kemampuan sistem aplikasi dimasa yang akan datang adalah sebagai berikut:

1. Aplikasi dapat melakukan sistem pencarian lokasi yang tidak terdapat dalam *database.*
2. Menambahkan rute-rute seluruh angkutan umum yang berada dikota Bandung.
3. Perubahan penempatan database sistem aplikasi dari offline menjadi online
4. Aplikasi ini hanya berjalan pada satu *platform*, yaitu pada *platform* android saja. Diharapkan kedepannya bisa berkembang disemua *platform* berbasis *mobile*.
5. Sistem penjaluran kedepannya dapat dikembangkan menggunakan algoritma lain seperti algoritma A*, Belman-Ford dan Floyd Warshall.

6 DAFTAR PUSTAKA

[1] Fauzi,Imron.(2011). *Penggunaan Algoritma Dijkstra Dalam Pencarian Rute Tercepat Dan Rute Terpendek*. Program studi teknik informatika, UIN Syarif Hidayatullah: Jakarta.

[2] Sugiarto, Imam.(2013). *Aplikasi Pencarian Lokasi Terdekat Pelayanan Kesehatan Berbasis Android Di Yogyakarta.* Program studi manajemen informatika, AMIKOM Yogyakarta: Yogyakarta.

[3] Departemen Pendidikan Nasional. 2008. *Kamus Bahasa Indonesia*. Pusat Bahasa : Jakarta.

[4] Diestel, R. (2000). *Graph Theory*. Springer-Verlag: New York.

[5] Gentara,G.M.(2011).*Simulasi Pencarian Jalur Terpendek Menggunakan Algoritma Dijkstra*. Program Studi Teknik Komputer, UNIKOM : Bandung.