

Analisis Perbandingan Algoritma Floyd-Warshall Dengan Algoritma Bellman-Ford Dalam Pencarian Rute Terpendek Menuju Museum di Jakarta

Jason Octavianus Agung¹⁾, Thomas Efendi²⁾, Halim Agung³⁾

Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia
Jalan Lodan Raya No.2, Pademangan, Jakarta Utara, 14430

¹⁾jasonoctavianus05@gmail.com

²⁾joshuathomasefendi66@gmail.com

³⁾hagung@bundamulia.ac.id

Abstract: Currently the tourist attraction of the museum has become the spotlight of Jakarta residents, museum visitors not only come from within the country but many also come from abroad. The number of visitors to the museum lately reduced, because many visitors who do not know the access or shortest route to the museum in Jakarta. The algorithm used in this research is Bellman-Ford algorithm and Floyd-Warshall algorithm. The result of this research is that Bellman-Ford algorithm and Floyd-Warshall algorithm can calculate the shortest route to museum in Jakarta, with success rate for Bellman-Ford algorithm that is 41,17% and Floyd-Warshall algorithm success rate also 41,17%. The error rate on the program is 58,83%. The average accuracy of predictions on the program is only 41,17% which can be concluded that the Floyd-Warshall algorithm and the Bellman-Ford algorithm have calculations with a success percentage below 50%.

Keywords: Bellman-Ford, comparison, Floyd-Warshall, museum, shortest reoute

Abstrak: Saat ini objek wisata museum sudah menjadi sorotan warga Jakarta, pengunjung museum tidaklah hanya berasal dari dalam negeri tetapi banyak juga yang berasal dari luar negeri. Jumlah pengunjung museum belakangan ini berkurang, dikarenakan banyak pengunjung yang tidak mengetahui akses atau rute terpendek menuju museum di Jakarta. Algoritma yang digunakan dalam penelitian ini adalah algoritma Bellman-Ford dan algoritma Floyd-Warshall. Hasil dari penelitian ini adalah bahwa algoritma Bellman-Ford dan algoritma Floyd-Warshall dapat menghitung rute terpendek menuju museum di Jakarta, dengan tingkat keberhasilan untuk algoritma Bellman-Ford yaitu 41,17% dan tingkat keberhasilan algoritma Floyd-Warshall juga 41,17%. Tingkat error pada program sejumlah 58,83%. Rata-rata keakuratan prediksi pada program hanyalah sejumlah 41,17% yang dapat disimpulkan bahwa algoritma Floyd-Warshall dan algoritma Bellman-Ford memiliki penghitungan dengan persentase keberhasilan di bawah 50%.

Kata kunci: Bellman-Ford, Floyd-Warshall, museum, perbandingan, rute terpendek

I. PENDAHULUAN

Museum merupakan salah satu tempat wisata yang menjadi sorotan masyarakat, di dalam museum terdapat benda benda bersejarah yang berasal dari jaman kuno atau jaman terdahulu. Masyarakat yang ingin berpergian ke museum yang berada di Ibukota Jakarta ini harus melakukan pertimbangan museum mana yang paling dekat dengannya atau rute mana yang merupakan rute terpendek yang harus ditempuh oleh masyarakat agar dapat sampai ke museum tujuan yang diinginkan. Pencarian rute terpendek pada sebuah peta atau map di dalamnya tentu terdapat

sebuah algoritma pemrograman untuk menjalankan proses pencariannya, menurut Kamus Besar Bahasa Indonesia (KBBI) algoritma didefinisikan sebagai prosedur sistematis untuk memecahkan masalah matematis dalam langkah-langkah terbatas. Algoritma sendiri merupakan urutan terbatas dari operasi-operasi terdefinisi dengan baik, yang masing masing membutuhkan memori dan waktu yang terbatas untuk menyelesaikan suatu masalah (Donald E. Knuth). Algoritma yang akan digunakan dalam penelitian ini adalah Algoritma Floyd-Warshall dan Algoritma Bellman-Ford dalam pencarian rute terpendek menuju museum di DKI Jakarta.

Berdasarkan penelitian sebelumnya yang berjudul Pencarian Titik Lokasi Dengan Pemanfaatan Algoritma Floyd-Warshall yang ditulis oleh Anggoro pada tahun 2015 mendapatkan hasil yaitu metode Floyd-Warshall dapat memberikan bobot yang kecil menuju suatu titik tujuan menuju ke sekolah [1].

Berdasarkan penelitian yang dilakukan oleh Budiarsyah pada tahun 2012 yang berjudul Algoritma Dijkstra, Bellman-Ford, dan Floyd-Warshall untuk mencari rute terpendek dari suatu graf, mendapatkan hasil yaitu algoritma Bellman-Ford dapat menyelesaikan permasalahan graf berbobot negatif yang tidak dapat diselesaikan oleh algoritma Dijkstra dan algoritma Floyd-Warshall [2].

Pada penelitian yang berjudul Penentuan Jalur Terpendek Menuju Café di Kota Malang Menggunakan Metode Bellman-Ford dengan Location Based Service berbasis android yang ditulis oleh M. Rofiq pada tahun 2014 mendapatkan hasil berupa algoritma Bellman-Ford dapat melakukan pencarian rute terpendek menuju café yang terdapat di kota Malang [3].

Algoritma Floyd-Warshall dalam jurnal penelitian yang ditulis oleh Ni Ketut Dewi Ari Jayanti pada tahun 2015 berhasil digunakan untuk menyelesaikan permasalahan jalur terpendek pada tata letak parkir dengan menghitung seluruh lintasan pada blok parker [4]. Berdasarkan jurnal penelitian yang ditulis oleh Fenny Anggraini pada tahun 2014 algoritma Bellman-Ford dapat menemukan rute terpendek berdasarkan pencarian lokasi perseroan terbatas di PT. Jakarta Industrial Estate Pulogadung, aplikasi dapat membeikan informasi berupa nama perusahaan, alamat, nomor telepon, dan jalur menuju perseroan terbatas tersebut [5].

II. METODE PENELITIAN

Heuristic [1] Kata Heuristic berasal dari bahasa Yunani yang berarti menemukan. Heuristic merupakan salah satu strategi untuk melakukan proses pencarian (search) ruang permasalahan secara selektif, yang memandu proses pencarian yang kita lakukan di sepanjang jalur yang memungkinkan sukses paling besar. Pada dunia pemrograman kata Heuristic merupakan lawan dari kata algoritmik, sehingga Heuristic diartikan sebagai suatu proses yang mungkin menyelesaikan permasalahan tetapi tidak menjamin bahwa setiap solusi akan selalu ditemukan. Ketika di dalam metode pencarian kata Heuristic ini menjadi dapat diartikan sebagai suatu fungsi yang memberikan sebuah nilai berupakan biaya (estimasi) dari suatu solusi.

Teori graf [2] adalah pokok bahasan yang sudah cukup lama digunakan namun masih memiliki banyak terapan yang dapat di pakai sampai saat ini. Graph sendiri di digunakan untuk mendefinisikan dan mempresentasikan objek-objek diskrit dan hubungan antar objek-objek tersebut.

Algoritma Floyd [3] - warshall adalah algoritma yang akan memilih satu jalur terpendek dan teraman dari beberapa alternatif jalur yang telah dihasilkan dari proses kalkulasi. Dalam mencari lintasan terpendek, algoritma floyd - warshall memulai iterasi dari titik awalnya kemudian memperpanjang lintasan dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang semimumum mungkin. Secara sederhana pengertian algoritma floyd - warshall adalah algoritma yang menghitung semua rute dan menghubungkan semua titik hingga ke titik tujuan.

Dasar algoritma floyd - warshall adalah sebagai berikut [3]: (1) Asumsikan semua simpul graf berarah G adalah $V = \{1, 2, 3, 4, \dots, n\}$, perhatikan subset $\{1, 2, 3, \dots, k\}$; (2) Untuk setiap pasangan simpul i, j pada V , perhatikan semua lintasan dari i ke j dimana semua simpul pertengahan diambil dari $\{1, 2, \dots, k\}$, dan p adalah lintasan berbobot minimum diantara semuanya; (3) Algoritma ini mengeksplorasi relasi antara lintasan p dan lintasan terpendek dari i ke j dengan semua simpul pertengahan berada pada himpunan $\{1, 2, \dots, k-1\}$; dan (4) Relasi tersebut bergantung pada apakah k adalah simpul pertengahan pada lintasan p .

Algoritma floyd - warshall untuk mencari lintasan terpendek adalah sebagai berikut [6]:

$$1) W = W_0$$

2) Untuk $k = 1$ hingga n , lakukan:

Untuk $i = 1$ hingga n , lakukan:

Untuk $j = 1$ hingga n lakukan.,

Jika $W[i, j] > W[i, k] + W[k, j]$ maka

Tukar $W[i, j]$ dengan $W[i, k] + W[k, j]$

$$3) W^* = W$$

Keterangan:

$W =$ matriks

$W_0 =$ Matrik hubung graf mula-mula

$k =$ iterasi 1 sampai ke- n

$i =$ titik awal pada vi

$j =$ titik akhir pada vj

$W^* =$ hasil matriks setelah perbandingan

Dalam iterasinya untuk mencari lintasan terpendek, algoritma Floyd-Warshall membentuk n matriks sesuai dengan iterasi- k . Itu menyebabkan waktu prosesnya lambat, terutama untuk n yang besar. Algoritma Bellman-Ford [3] merupakan

algoritma untuk menghitung jalur terpendek yang berasal dari satu sumber node pada sebuah digraph yang berbobot, yang dimaksud berasal dari satu sumber adalah algoritma bellman-Ford menghitung semua jarak terpendek yang berawal dari satu titik node. Algoritma Bellman-Ford digunakan apa bila ada sisi dengan bobot negatif, yang merupakan salah satu keunggulan juga dari algoritma Bellman-Ford ini sendiri karena algoritma ini dapat menghubungkan dua simpul yang merupakan bilangan negatif.

Algoritma Bellman-Ford dalam rumus matematika dapat di tuliskan sebagai berikut :

$M [i,v] = \min(M [i-1,v] , (M [i-1,n]+ Cvn))$
 dengan $i = \text{iterasi}$, $v = \text{vertex} = \text{node}$, $n = \text{node neighbor}$, $C = \text{cost}$

Langkah paling pertama yang harus dilakukan dalam analisis graf dengan algoritma Bellman-Ford adalah menentukan titik asal setelah menetapkan titik asal dari lintasan lalu mulai melakukan penandaan atau marking, dalam hal ini semua titik harus di beri tanda dengan symbol infinity sedangkan titik asal sendiri akan di beri tanda dengan angka 0.

Langkah berikutnya adalah melakukan relaxing pada simpul yang terdapat pada graf. Simpul yang akan di relaxing adalah semua simpul lain selain simpul asal. Relaxing sendiri adalah membandingkan bobot antar titik yang ada, dalam hal ini yang sudah di tandai dengan symbol infinity, dengan titik titik lain yang ada di sekitarnya yang dihubungkan pada titik awal yang di beri tanda 0..

Unified Modelling Language (UML) [7] adalah suatu notasi pendiagraman yang tepat yang mampu mengijinkan suatu desain program untuk direpresentasikan dan didiskusikan. Pada UML sendiri terdapat beberapa diagram yang digunakan untuk merepresentasikan program

Use case diagram [7] adalah diagram yang digunakan untuk memperlihatkan cara kerja sebuah sistem yang bermanfaat untuk mengidentifikasi dan mendokumentasi spesifikasi yang dibutuhkan oleh sistem. Activity Diagram [7] menggambarkan aktivitas yang terjadi pada sebuah sistem secara berurutan dari awal mulai aktivitas sampai akhir dari aktivitas. Sequence Diagram [7] adalah diagram yang menggambarkan bagaimana proses suatu sistem bekerja dalam sebuah periode waktu yang berjalan. Sequence Diagram lebih bersifat dinamis dibandingkan bersifat statis.

Class diagram [7] menunjukkan hubungan relasi antar satu class dengan class lainnya. Class merupakan komponen yang paling dasar dari pemrograman berorientasi objek. Pada class sendiri

terdapat komponen-komponen seperti nama yang unik, atribut, dan metode yang digunakan.

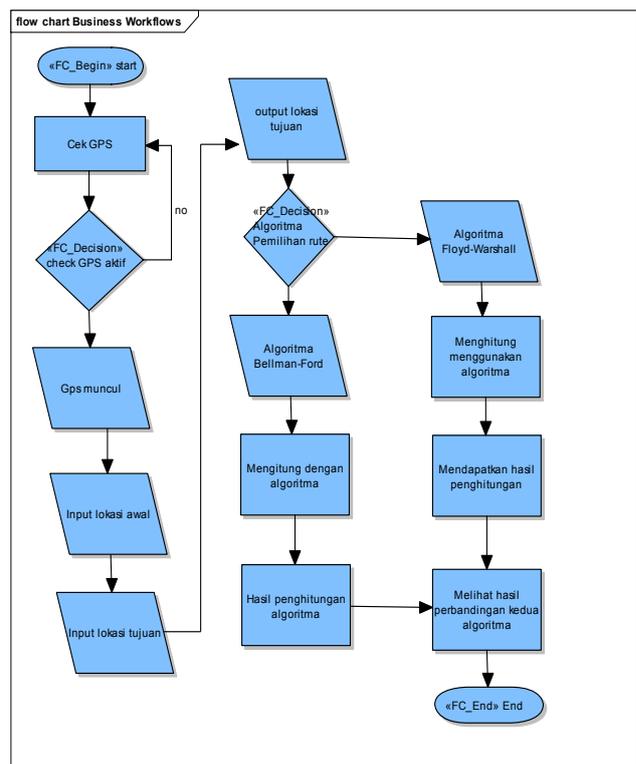
III. HASIL DAN PEMBAHASAN

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang bisa terdapat dalam suatu sistem yang akan digunakan oleh pengguna. Kebutuhan fungsional dari aplikasi yang dirancang ini adalah sebagai berikut: (1) Menerima masukkan dari user berupa update lokasi untuk dimodifikasi; dan (2) Menerima masukkan dari user berupa ketepatan

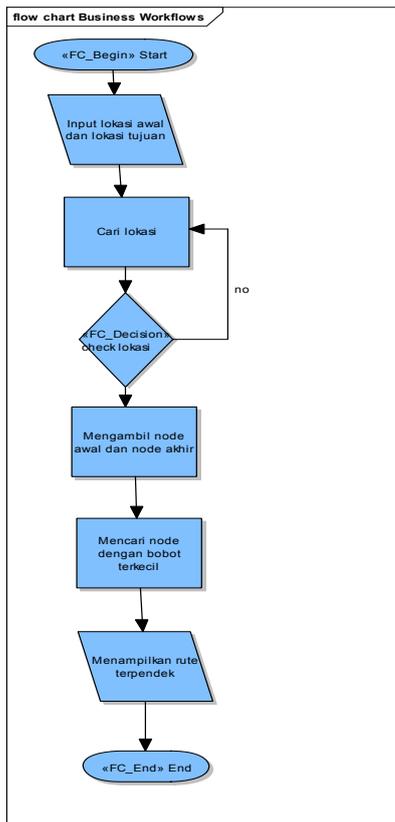
Algoritma yang digunakan dalam pembuatan aplikasi ini adalah algoritma Floyd-Warshall dan algoritma Bellman-ford. Algoritma tersebut dipilih karena kedua algoritma tersebut digunakan untuk mencari rute terpendek dan kedua algoritma tersebut termasuk kedalam algoritma yang cukup baik dan akurat dalam melakukan pencarian rute terpendek.

Perancangan flowchart diagram bertujuan untuk menggambarkan aliran proses dalam sistem. Flowchart diagram dari aplikasi ini ditunjukkan pada Gambar 1, Flowchart diagram dari Algoritma Bellman-Ford ditunjukkan pada Gambar 2, Flowchart diagram dari Algoritma Floyd-Warshall ditunjukkan pada Gambar 3.

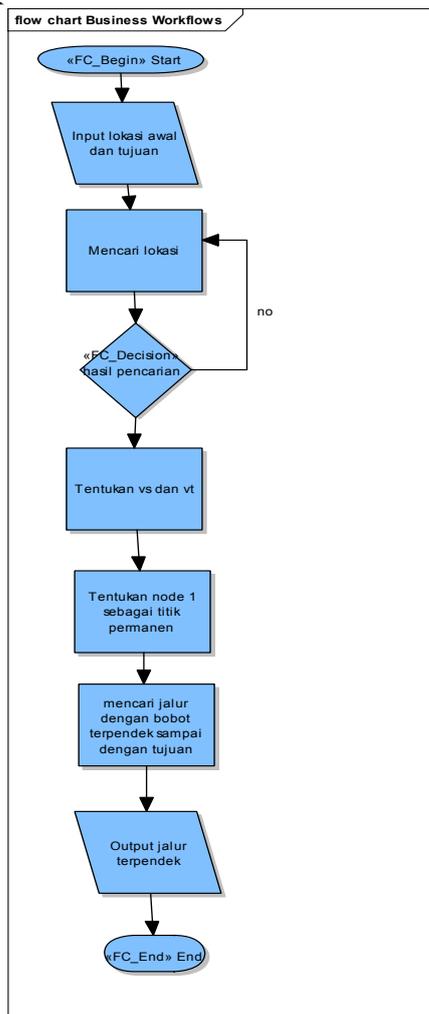
Use case diagram identifikasi menghitung rute terpendek menggambarkan apa saja yang dapat dilakukan oleh pengguna didalam sistem.



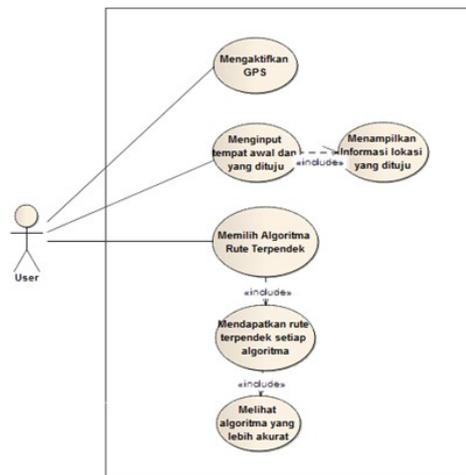
Gambar 1. Flowchart aplikasi



Gambar 2. Flowchart Bellman-Ford aplikasi



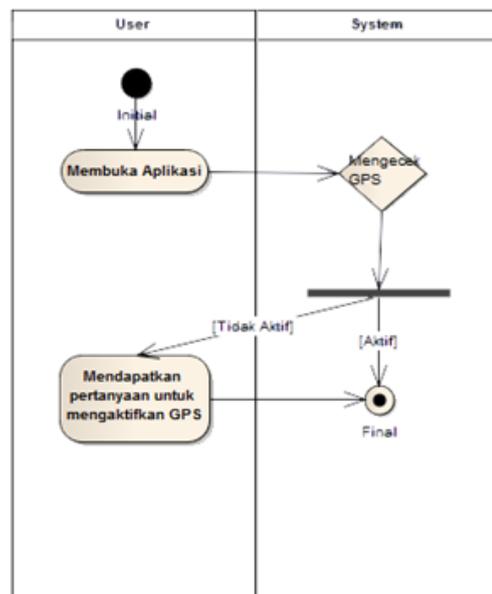
Gambar 3. Flowchart Floyd-Warshall aplikasi



Gambar 4. Usecase diagram

Seperti yang terlihat pada Gambar 4, interaksi antara pengguna (user) dengan aplikasi adalah sebagai berikut: (1) User membuka aplikasi terlebih dahulu; (2) User akan diminta untuk menyalakan GPS terlebih dahulu; (3) User akan diminta untuk melakukan input lokasi awal dan input lokasi tujuan; (4) Aplikasi akan melakukan penghitungan menggunakan algoritma yang terdapat di dalam aplikasi; dan (5) Aplikasi akan menampilkan hasil dari penghitungan rute terpendek berdasarkan algoritma yang dipilih oleh user

Perancangan activity diagram ini bertujuan untuk menggambarkan aktivitas-aktivitas yang terjadi di dalam sistem/aplikasi secara sistematis. Activity diagram cek GPS ditunjukkan pada Gambar 5.

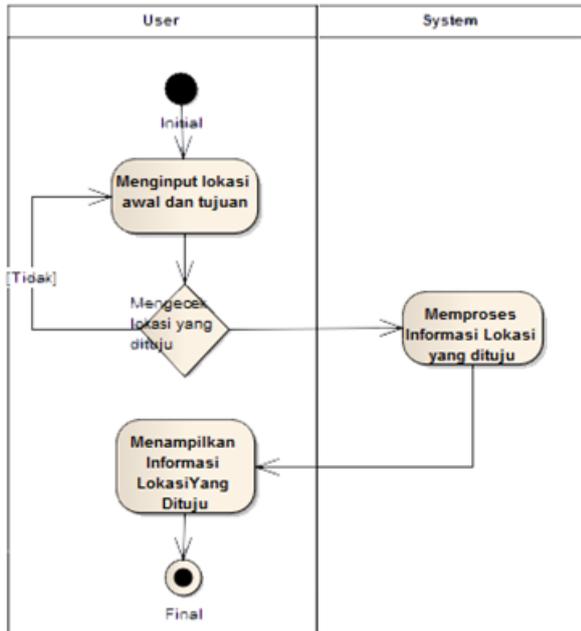


Gambar 5 Activity cek GPS

Aktivitas yang dilakukan oleh aplikasi dalam melakukan interaksi dengan user seperti yang di gambarkan pada Gambar 5 adalah sebagai berikut: (1) Aplikasi melakukan pengecekan GPS pada smartphone; (2) Aplikasi akan meminta user untuk mengaktifkan GPS apabila GPS belum menyala;

dan (3) Setelah GPS menyala maka user akan dapat masuk ke dalam tampilan utama dari aplikasi.

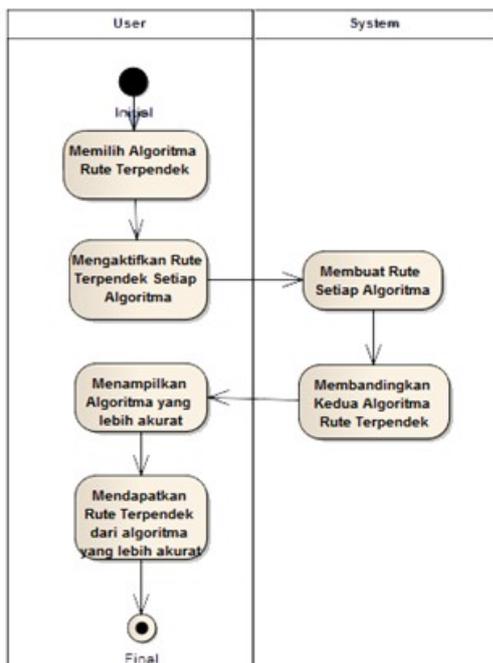
Activity diagram input dan output lokasi ditunjukkan pada gambar 6.



Gambar 6. Activity input dan output

Aktivitas yang dilakukan oleh user dalam penggunaan aplikasi digambarkan dengan activity diagram yang ditunjukkan pada Gambar 6: (1) User akan melakukan input lokasi awal dan lokasi tujuan pada aplikasi; (2) Aplikasi akan mengecek lokasi dan memproses lokasi awal dan lokasi tujuan; dan (3) Aplikasi akan menampilkan lokasi tujuan yang dipilih oleh user.

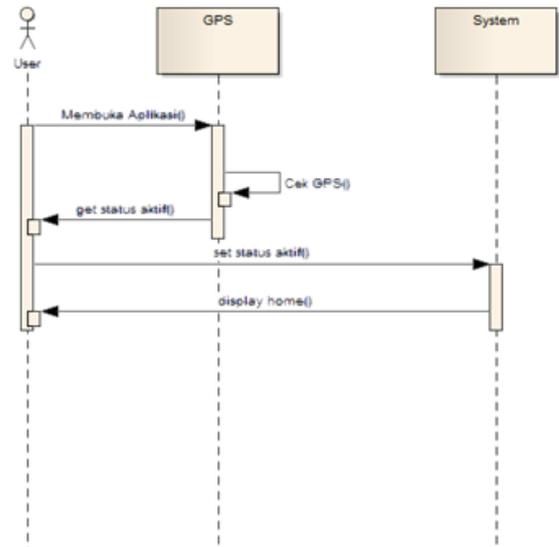
Activity diagram proses kerja algoritma ditunjukkan pada Gambar 7.



Gambar 7. Activity diagram proses kerja algoritma

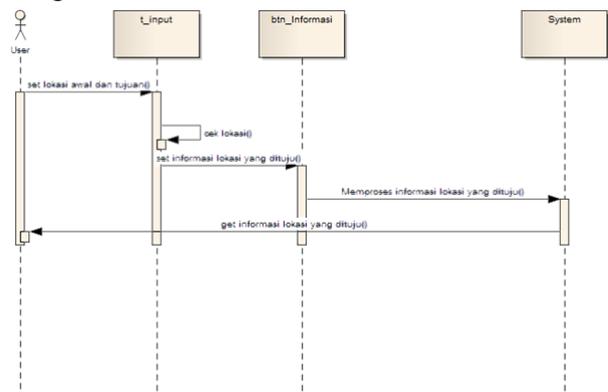
Proses user dalam memilih algoritma dapat dilihat dalam activity diagram algoritma pada gambar 7: (1) User akan memilih algoritma dari pilihan algoritma yang ada, yaitu algoritma Bellman-Ford dan algoritma Floyd-Warshall; (2) Sistem akan mencari algoritma yang sesuai dengan pilihan user; (3) Sistem akan menggambarkan rute terdekat berdasarkan node-node yang telah tersedia; dan (4) Aplikasi akan menunjukkan hasil penghitungan algoritma dalam bentuk angka yang berupa satuan meter.

Sequence diagram check GPS ditunjukkan pada gambar 8.



Gambar 8 Sequence diagram cek GPS

Sequence diagram pada Gambar 8 menggambarkan proses aplikasi dalam melakukan pengecekan GPS sudah menyala pada smartphone user atau belum. Sequence diagram system ditunjukkan pada Gambar 9.

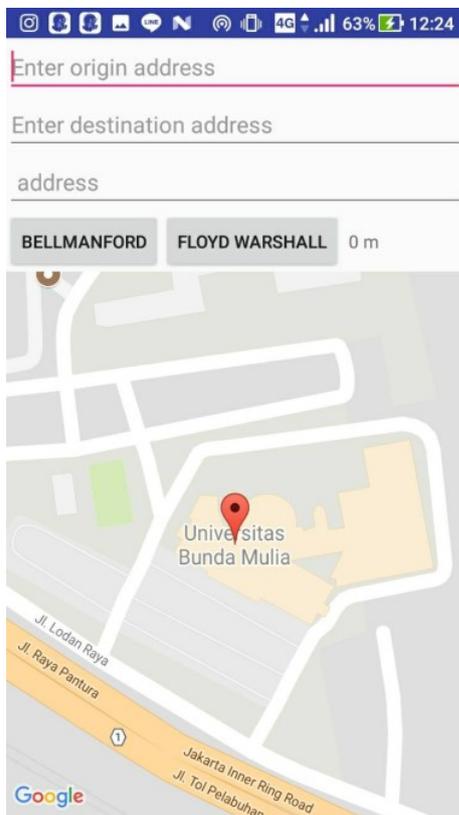


Gambar 9 Sequence diagram system

Sequence diagram pada Gambar 9 menggambarkan proses aplikasi dari awal user melakukan input lokasi awal dan lokasi tujuan pada aplikasi sampai pada aplikasi mencari rute terpendek menggunakan algoritma yang akan dipilih oleh user sehingga mendapatkan hasil berupa rute terpendek.

A. Implementasi Antarmuka Pengguna

Gambar 10 adalah tampilan pada saat user membuka program pertama kali sehingga merupakan tampilan awal program. Ketika user berada pada tampilan awal program ini user dapat melihat adanya 2 buah edittext yang dapat diisi. Edittext pertama adalah Enter Origin Address yaitu edittext yang akan diisi dengan lokasi awal yang user, sedangkan edittext yang kedua adalah Enter Destination Address yaitu lokasi tujuan yang akan dituju oleh user. User juga akan menjumpai adanya 2 buah button yang berfungsi yaitu button BellmanFord dan button FloydWarshall kedua button tersebut memiliki fungsi yaitu ketika salah satu button diklik maka aplikasi akan mulai menghitung dengan algoritma sesuai dengan yang tertulis pada button yang dipilih, kemudian user baru dapat membandingkan hasil dari kedua buah algoritma yang diterapkan di dalam aplikasi.



Gambar 10. Tampilan awal program

B. Pengujian Aplikasi dan Algoritma

Dari Gambar 11 dan 12 dapat disimpulkan bahwa hasil penghitungan dari kedua algoritma adalah sama, dikarenakan logika yang digunakan oleh algoritma Bellman-Ford dan algoritma Floyd-Warshall adalah sama, yang membedakan kedua algoritma tersebut adalah algoritma Bellman-Ford mempunyai kelebihan bahwa Bellman-Ford dapat menghitung node yang memiliki bobot negatif, tetapi

sayangnya dalam penghitungan node untuk pencarian jarak tidak ada node dengan bobot negative.

No	Lokasi Awal	Lokasi Tujuan	Hasil Floyd (m)	Hasil Bellman-Ford(m)	Hasil Real (m)
1	Universitas Bunda Mulia	Museum Kebaharian Jakarta	1600	1600	1538
2	Universitas Bunda Mulia	Museum Fatahillah	2213	2213	2100
3	Apartemen Sunter Icon	Sunter Mall 21	3512	3512	2500
4	Universitas Bunda Mulia	Rumah Sakit Pluit	2853	2853	3000
5	Ancol Selatan 2	Samsat Jakarta Utara	4426	4426	5200
6	Season City	Monumen Nasional	6800	6800	7400
7	Apartemen Sunter Icon	Kalbis Institute	14518	14518	15000
8	Pasar Cipete	Blok M Square	2227	2227	3900
9	Gang Makmur Dalam	Season City	4893	4893	5000
10	Super Indo Sunter	Mall Artha Gading	3743	3743	3700
11	Mall Artha Gading	Mall Of Indonesia	715	715	600
12	McDonald's Sunter	Sumobo	3434	3434	3400
13	Central Petojo	Gereja Bunda Hati Kudus	700	700	750
14	Gereja Bunda Hati Kudus	Sekolah Bunda Mulia	2400	2400	2100
15	Northland Ancol Residence	Restoran Sederhana Gunung Sahari	1598	1598	1500
16	Sekolah Menengah Atas Jubilee	Lapangan Golf Bandar Kemayoran	2337	2337	2300
17	Food Centrum	RSPI Prof Dr Sulianti Saroso	2704	2704	2700

Gambar 11 Pengujian Algoritma

Data uji		
Algoritma	Floyd-warshall	Bellman-Ford
Hasil Rute Terdekat	42	8
Akurasi Floyd-warshall(%)	$= \frac{\text{jumlah lokasi rute terdekat}}{\text{total lokasi pengujian}} \times 100\%$ $= \frac{7}{17} \times 100\%$ $= 41,17\%$	
Akurasi Bellman-Ford(%)	$= \frac{\text{jumlah lokasi rute terdekat}}{\text{total lokasi pengujian}} \times 100\%$ $= \frac{7}{17} \times 100\%$ $= 41,17\%$	

Gambar 12 Tingkat akurasi algoritma

IV. SIMPULAN

Dari penelitian dan pengujian yang dilakukan maka dapat disimpulkan sebagai berikut: (1) Algoritma Floyd-warshall dan Bellman-Ford dapat diimplementasikan untuk mencari rute terdekat; (2) Algoritma Floyd-warshall dan Algoritma Bellman-Ford memberikan bobot yang sama untuk mencapai tujuan; (3) Dari hasil pengujian, dapat diketahui bahwa algoritma pencarian rute terpendek yang akurat dengan tingkat akurasi 41,17%; dan (4) Algoritma Bellman-Ford tidak banyak digunakan dalam pencarian rute terpendek karena algoritma Bellman-Ford hanya memiliki kelebihan menghitung jalur dengan node negatif, tetapi pada pencarian node jalan tidak ada node yang berbobot negatif.

V. DAFTAR RUJUKAN

[1] A. A. Anggoro, Pencarian Titik Lokasi Dengan Pemanfaatan Algoritma Floyd-Warshall Sebagai Perhitungan Jarak Terdekat di Institut Teknologi Bandung. Jurnal LPKIA, Vol. 1 No.1. hlm 36, Bandung. 2015.

- [2] K. D. Budiarsyah, Algoritma Dijkstra, Bellman-Ford, dan Floyd – Warshall Untuk Mencari Rute Terpendek Pada Graf, Sekolah Teknik Elektro dan Informatika. Hlm 62, 2012
- [3] M. Rofiq, Penentuan Jalur Terpendek Menuju Café di Kota Malang Menggunakan Metode Bellman-Ford Dengan Location Based Service Berbasis Android, Jurnal Ilmiah Teknologi dan Informasi Asia, Volume 8, nomor 2. Hlm 29, 2014.
- [4] N. K. D. A. Jayanti, Penggunaan Algoritma Floyd-Warshall Dalam Masalah Jalur Terpendek Pada Penentuan Tata Letak Parkir. Seminar Nasional Informatika. Denpasar. 43, 2014.
- [5] F. Anggraini, Penerapan Algoritma Bellman-Ford Dalam Aplikasi Pencarian Lokasi Perseroan Terbatas di PT. Jakarta Industrial Estate Pulogadung (PT.JIEP). Universitas Muhammadiyah Jakarta. Volume 7. nomor 1. ISSN: 2085-1669. Hlm 57, 2014.
- [6] R. Kriswanto., K. J. Bendi. & A. Aliyanto, Penentuan Jarak Terpendek Rute Transmisi dengan Algoritma Floyd-Warshall. Proceeding Seminar Nasional Teknologi Informasi & Komunikasi Terapan. ISSN : 979-26-0276-3. Hlm 45, 2014.
- [7] S. Kendal, Object Oriented Programming Using Java. ISBN: 978-87-7681-501-1. Hlm, 25, 2009.