

# IMPLEMENTASI *BELLMAN-FORD* DAN *FLOYD-WARSHALL* DALAM MENENTUKAN JALUR TERPENDEK MENUJU UNIVERSITAS NASIONAL BERBASIS ANDROID

Syahbani Farhan<sup>1)</sup>, Septi Andryana<sup>2)</sup>, Nur Hayati<sup>3)</sup>

<sup>1, 2, 3)</sup> Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional

e-mail: [syahbanifarhan@gmail.com](mailto:syahbanifarhan@gmail.com)<sup>1)</sup>, [septi.andryana@civitas.unas.ac.id](mailto:septi.andryana@civitas.unas.ac.id)<sup>2)</sup>, [nurhayati@civitas.unas.ac.id](mailto:nurhayati@civitas.unas.ac.id)<sup>3)</sup>

## ABSTRAK

*Disekitar Universitas Nasional memiliki berbagai macam jenis transportasi umum. Sebagian besar masyarakat dan mahasiswa Universitas Nasional masih menggunakan transportasi umum seperti kereta api dan transjakarta, namun masih terkendala jarak antara stasiun dan halte ke Universitas Nasional. Penelitian ini menerapkan algoritma Bellman-Ford dan Floyd-Warshall yang dinilai efektif dan telah banyak digunakan pada penelitian sebelumnya dalam pencarian jalur terpendek diantaranya yaitu pengantaran barang, pencarian kampus dan pencarian lokasi travel. Penelitian ini bertujuan untuk mempermudah masyarakat, khususnya mahasiswa atau peserta didik baru dalam mencari jalur terdekat dari stasiun dan halte menuju Universitas Nasional. Penelitian ini dirancang menggunakan framework flutter dan bahasa pemrograman dart berbasis android dengan pengguna terbanyak pada saat ini. Berdasarkan hasil pengujian, algoritma Bellman-Ford dan Floyd-Warshall untuk kasus pencarian jarak terpendek dari stasiun Pasar Minggu diperoleh jarak terpendek sebesar 1.54 km dengan tingkat keefektifan jarak sebesar 39.40%, sedangkan kasus pencarian jarak terpendek dari halte Jatipadang diperoleh jarak terpendek sebesar 1.97 km dengan tingkat keefektifan jarak sebesar 25.24%.*

**Kata Kunci:** *Android; Bellman-Ford; Flutter; Floyd-Warshall; Jalur Terpendek;*

## ABSTRACT

*Around the University of Nasional has various types of public transportation. Most of the people and University of Nasional students still use public transportation such as trains and transjakarta, but are still constrained by the distance between the station and bus stop to the University of Nasional. This study applies the Bellman-Ford and Floyd-Warshall algorithms which are considered effective and have been widely used in previous studies in the search for the shortest path, including delivery of goods, campus searches and search for travel locations. This study aims to make it easier for the community, especially students or new students to find the closest route from the station and bus stop to the University of Nasional. This study was designed using the flutter framework and the Android-based dart programming language with the most users at this time. Based on the test results, the Bellman-Ford and Floyd-Warshall algorithms for the case of finding the shortest distance from the Pasar Minggu station obtained the shortest distance of 1.54 km with an effectiveness level of 39.40%, while the case of finding the shortest distance from the Jatipadang bus stop obtained the shortest distance of 1.97 km with distance effectiveness level of 25.24%.*

**Keywords:** *Android; Bellman-Ford; Flutter; Floyd-Warshall; Shortest path;*

## I. PENDAHULUAN

DALAM kehidupan sehari-hari implementasi algoritma Bellman-Ford dan Floyd-Warshall telah banyak digunakan dalam pencarian jalur terpendek menuju suatu tempat diantaranya yaitu dalam pengantaran barang[1], pencarian kampus[2], dan pencarian lokasi jasa travel[3]. Jalur terpendek merupakan masalah mendasar dalam teori grafik. Itu Ini juga bisa menjadi salah satu masalah utama dalam jaringan analisis. Masalah jalur terpendek adalah masalah menemukan jalur atau rute terpendek dari titik awal ke titik akhir tujuan.[4] Universitas Nasional merupakan universitas swasta tertua di Jakarta dan tertua kedua di Indonesia. Sebagian besar mahasiswa dan peserta didik baru masih memanfaatkan transportasi umum seperti transjakarta maupun kereta api, tetapi masih terkendala jarak antara stasiun dan halte menuju Universitas Nasional. Permasalahan yang diangkat dalam penelitian ini yaitu semakin maraknya kuliah berbasis online, mahasiswa yang baru mendaftar di Universitas Nasional belum mengetahui dimana lokasi tepatnya Universitas Nasional khususnya bagi mahasiswa pendatang baru dari luar kota Jakarta. Penelitian ini bertujuan untuk mempermudah masyarakat, khususnya mahasiswa dan peserta didik baru dalam mencari jalur terdekat dari stasiun dan halte menuju Universitas Nasional dengan dibuatkan sebuah aplikasi berbasis android.

Berdasarkan kajian jurnal yang dilakukan oleh Paska Marto Hasugian tentang penggunaan algoritma Bellman-

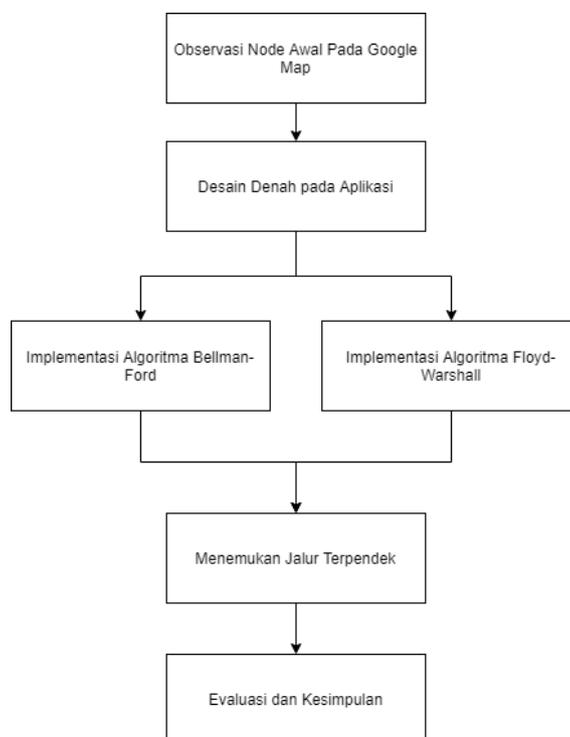
Ford dalam menemukan jalur terpendek pengantaran barang dalam kota, penelitian tersebut memanfaatkan data yang didapat melalui peta dari website Open Street Map dan GPS (Global Positioning System) dan menghasilkan jarak terpendek sejauh 8,8km[1]. Penelitian yang dilakukan oleh Bayu Irawan Ahmad Prasetyo dan Andi Maslan mengkaji tentang perbandingan algoritma Bellman-Ford dan algoritma Dijkstra pada Google Maps. Hasil penelitian tersebut efektif dalam menemukan rute terpendek dari kampus Tembesi ke kampus Tiban, kampus Tembesi ke kampus Nagoya dan kampus Tiban ke kampus Nagoya.[2] Pada penelitian Yulia Darnita, Rozali Toyib, dan Rinaldi melakukan implementasi untuk menentukan letak dan lokasi travel, pada penelitian tersebut menghasilkan nilai yang efektif dari algoritma Floyd-Warshall pada aplikasi yang diujikan.[3]

Berdasarkan kajian penelitian yang lalu implementasinya hanya sebatas perhitungan graf dan aplikasi berbasis web, maka penelitian ini dikembangkan dengan mengimplementasikan ke dalam aplikasi berbasis android flutter dengan membandingkan hasil jarak dan waktu proses dari algoritma Bellman-Ford dan Floyd-Warshall.

## II. METODE PENELITIAN

### A. Flowchart

Berikut merupakan flowchart alur penelitian dalam pembuatan aplikasi dengan mengimplementasikan algoritma Bellman-Ford dan algoritma Floyd-Warshall.



Gambar. 1. Flowchart alur penelitian pencarian jalur terpendek menuju Universitas Nasional

Tahapan flowchart diatas menjelaskan alur dari proses alur penelitian pencarian jalur terpendek menuju Universitas Nasional, diantaranya yaitu.

1. Melakukan observasi pada google map untuk menentukan node-node pada aplikasi.
2. Membuat desain denah graf pada aplikasi berbasis android.
3. Kemudian melakukan implementasi algoritma Bellman-Ford dan Floyd-Warshall pada aplikasi berbasis android.
4. Setelah memproses hasil hitung dengan algoritma Bellman-Ford dan Floyd-Warshall. Lalu menghasilkan jalur terpendek.
5. Melakukan evaluasi ulang aplikasi untuk mencari bug serta rencana pengembangan kedepannya dan kesimpulan dari hasil aplikasi.

### B. Graf

Secara konseptual graf tercipta dari simpul dan sisi yang menghubungkannya.[5] Pada sebuah graf  $G = (V, E)$  yang berisi dari himpunan  $(V_1, V_2, V_3, \dots)$  yang elemennya biasa disebut simpul.[6] Contoh salah satu gambaran graf yaitu peta.[7]

Representasi dari graf yaitu pernyataan objek sebagai bulatan, titik atau disebut verteks, sedangkan penghubung antara objek dinyatakan garis atau edge.[8]

### C. Algoritma Bellman-Ford

Algoritma *Bellman-Ford* merupakan salah satu dalam *single source shortest path algorithm*[9], dan turunan dari kelas algoritma yang sekarang dikenal dengan algoritma *relaxation* atau algoritma *label-correction* untuk menemukan jalur terpendek dari titik awal yang telah ditentukan dalam graf berarah.[10] Berikut merupakan *Pseudocode* dari algoritma *Bellman-Ford*. [11]

```

Bellman-Ford (G,w,s)

1. Initialize-Single-Source (G,s)
2. for i ← 1 to |V| -1
3. do for each edge (a,b) ∈ E
4. do RELAX (a,b,c)
5. for each edge (a,b) ∈ E
6. do if d[b] > d[a] + w(a,b)
7. then return FALSE
8. return TRUE

Relax (a,b,c)

1. if d[b] > d[a] + w(a,b)
2. then d[b] := d[a] + w(a,b)
3. exit

```

Gambar. 2. Pseudocode Algoritma Bellman-Ford

Langkah-langkah penyelesaian algoritma *Bellman-Ford*. [12]

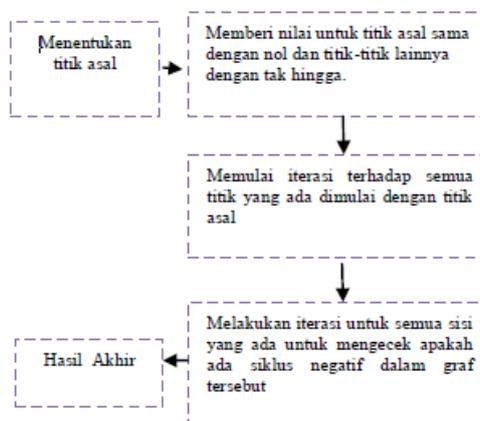
1. Menentukan node asal serta menentukan node selanjutnya maupun sisi-sisinya.
2. Memberikan node awal dengan nilai 0, serta memberikan node lainnya dengan nilai tak terhingga.
3. Memulai perhitungan pada setiap node, dimulai melalui titik awal dengan rumus relaksasi sebagai berikut.

$$d[y] > d[x] + w(x, y) \tag{1}$$

$$d[y] = d[x] + w(x, y)$$

4. Lakukan hingga seluruh node terhitung bobotnya.
5. Lakukan iterasi pada setiap sisi, untuk memverifikasi apakah ada siklus negatif pada graf.

Berikut merupakan langkah pengerjaan algoritma *Bellman-Ford*.



Gambar. 3. Langkah-langkah pengerjaan Algoritma *Bellman-Ford*

#### D. Algoritma Floyd-Warshall

Algoritma *Floyd-Warshall* merupakan suatu pemrograman dinamis yang melaksanakan penyelesaian masalah dengan mengetahui hasil solusi yang diperoleh sebagai keputusan yang terkait satu sama lain.[13] Berikut merupakan *pseudocode* dari algoritma *Floyd-Warshall*. [14]

<p><b>Floyd-Warshall</b></p> <ol style="list-style-type: none"> <li>1. <math>n \leftarrow \text{rows}[W]</math></li> <li>2. <math>D^{(0)} = W</math></li> <li>3. for <math>k \leftarrow 1</math> to <math>n</math></li> <li>4.     do for <math>i \leftarrow 1</math> to <math>n</math></li> <li>5.         do for <math>j \leftarrow 1</math> to <math>n</math></li> <li>6.             <math>D[i,j] &gt; D[i,k] + D[k,j]</math></li> <li>7.             <math>D[i,j] = D[i,k] + D[k,j]</math></li> <li>8. return <math>D^{(n)}</math></li> </ol>
--

Gambar. 4. *Pseudocode* Algoritma *Floyd-Warshall*

Langkah-langkah penyelesaian algoritma *Floyd-Warshall*. [15]

1. Buat sebuah tabel matriks  $n \times n$  jarak pada setiap node.
2. Pada tabel, masukkan nilai sisi yang dilewati masing-masing node.
3. Hitung pada matriks yang telah ditentukan nilainya dengan rumus sebagai berikut.

$$\text{jarak}[a][b] > \text{jarak}[a][c] + \text{jarak}[c][b]$$

$$\text{jarak}[a][b] = \text{jarak}[a][c] + \text{jarak}[c][b]$$

(2)

4. Hitung rumusnya secara berulang sejumlah banyaknya node.

### III. PEMBAHASAN

#### A. Hasil Penelitian

Pada tahapan ini merupakan tahap dimana pengimplementasian hasil rancangan *design layout* kedalam aplikasi menggunakan *platform* Android Studio. Berikut hasil perhitungan dan implementasi ke dalam aplikasi pencarian rute terdekat menuju Universitas Nasional.

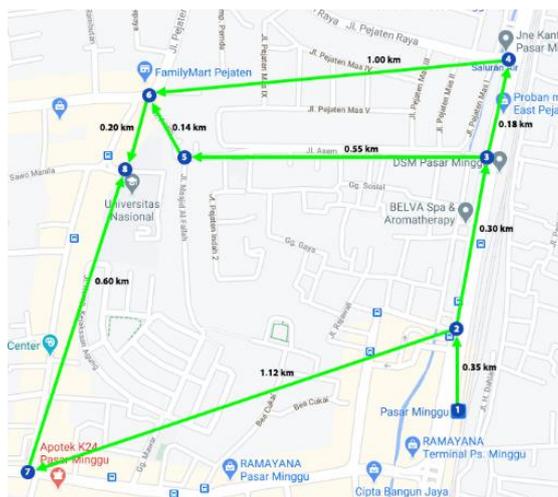
##### 1) Penentuan Node Awal

- a. Stasiun Pasar Minggu

TABEL I  
LOKASI NAMA JALAN DAN JARAK PADA SETIAP NODE  
STASIUN PASAR MINGGU

Nama Jalan	Node	Jarak
Jl. Raya Pasar Minggu 1	1 - 2	0.35 km
Jl. Raya Pasar Minggu 2	2 - 3	0.30 km
Jl. Terminal Baru dan Jl. Raya Ragunan	2 - 7	1.12
Jl. Raya Pasar Minggu 3	3 - 4	0.18 km
Jl. Asem 1	3 - 5	0.55 km
Jl. Pejaten Raya	4 - 6	1.00 km
Jl. Asem 2	5 - 6	0.14 km
Jl. Sawo Manila	6 - 8	0.20 km
Jl. Salihara	7 - 8	0.60 km

Pada Tabel 1 merupakan data nama jalan, node, dan jarak dalam pembuatan desain graf berbobot dari stasiun Pasar Minggu menuju Universitas Nasional. Berikut merupakan tampilan desain graf berbobotnya.



Gambar. 5. Tampilan Graf dari stasiun Pasar Minggu menuju Universitas Nasional

Pada Gambar 5 merupakan graf berbobot yang memiliki 8 node dan 9 sisi. Digunakan sebagai ilustrasi penyelesaian Algoritma Bellman-Ford dan Floyd-Warshall dari stasiun Pasar Minggu menuju Universitas Nasional.

b. Halte Jatipadang

TABEL II  
LOKASI NAMA JALAN DAN JARAK PADA SETIAP NODE  
HALTE JATIPADANG

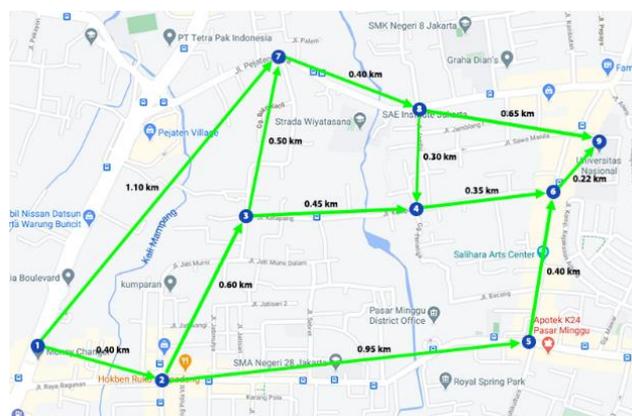
Nama Jalan	Node	Jarak
Jl. Raya Ragunan 1	1 - 2	0.40 km
Jl. Warung Jati Barat dan Jl. Pejaten Raya 1	1 - 7	1.10 km
Jl. Jati Mumi	2 - 3	0.60 km
Jl. Raya Ragunan 2	2 - 5	0.95 km
Jl. Ketapang 1	3 - 4	0.45 km
Jl. Jatipadang Utara	3 - 7	0.50 km
Jl. Ketapang 2	4 - 6	0.35 km
Jl. Salihara 1	5 - 6	0.40 km
Jl. Salihara 2	6 - 9	0.22 km
Jl. Pejaten Raya 2	7 - 8	0.40 km
Jl. Jambu	8 - 4	0.30 km
Jl. Pejaten Raya 3 dan Jl. Sawo Manila	8 - 9	0.65 km

Pada Tabel 2 merupakan data nama jalan, node, dan jarak dalam pembuatan desain graf berbobot dari halte Jatipadang menuju Universitas Nasional. Berikut merupakan tampilan desain graf berbobotnya.

Pada Gambar 6 merupakan graf berbobot yang memiliki 9 node dan 12 sisi. Digunakan sebagai ilustrasi penyelesaian Algoritma Bellman-Ford dan Floyd-Warshall dari halte Jatipadang menuju Universitas Nasional.

2) Proses Pencarian Jalur Terpendek

a. Algoritma Bellman-Ford



Gambar. 6. Tampilan Graf dari halte Jatipadang menuju Universitas Nasional

TABEL III  
HASIL PERHITUNGAN ALGORITMA BELLMAN-FORD DARI STASIUN PASAR MINGGU

Node	Bobot Awal	Iterasi 1	Iterasi 2	Iterasi 3	Iterasi 4	Iterasi 5	Iterasi 6	Iterasi 7
1	0	0	0	0	0	0	0	0
2	$\infty$	0.35	0.35	0.35	0.35	0.35	0.35	0.35
3	$\infty$	$\infty$	0.65	0.65	0.65	0.65	0.65	0.65
4	$\infty$	$\infty$	$\infty$	0.83	0.83	0.83	0.83	0.83
5	$\infty$	$\infty$	$\infty$	1.20	1.20	1.20	1.20	1.20
6	$\infty$	$\infty$	$\infty$	$\infty$	1.83	1.34	1.34	1.34
7	$\infty$	$\infty$	1.47	1.47	1.47	1.47	1.47	1.47
8	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	1.54	1.54

Dapat dilihat pada Tabel 3, merupakan hasil perhitungan dari Algoritma Bellman-Ford. Pada iterasi 1, dikarenakan pada iterasi 2 hasilnya memiliki nilai yang sama pada iterasi 1, maka iterasi dapat dihentikan dan dapat disimpulkan hasil akhir dari perhitungan tersebut yaitu melalui node 1-2-3-5-6-8 dengan melalui Jl. Raya Pasar Minggu 1, Jl. Raya Pasar Minggu 2, Jl. Asem 1, Jl. Asem 2, dan Jl. Sawo Manila jarak total 1.54 km.

TABEL IV  
HASIL PERHITUNGAN ALGORITMA BELLMAN-FORD DARI HALTE JATIPADANG

Node	Bobot Awal	Iterasi 1	Iterasi 2	Iterasi 3	Iterasi 4	Iterasi 5	Iterasi 6	Iterasi 7	Iterasi 8
1	0	0	0	0	0	0	0	0	0
2	$\infty$	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40
3	$\infty$	$\infty$	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	$\infty$	$\infty$	$\infty$	1.45	1.45	1.45	1.45	1.45	1.45
5	$\infty$	$\infty$	1.35	1.35	1.35	1.35	1.35	1.35	1.35
6	$\infty$	$\infty$	$\infty$	$\infty$	1.80	1.75	1.75	1.75	1.75
7	$\infty$	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
8	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	1.50	1.50
9	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	1.97	1.97	1.97

Dapat dilihat pada Tabel 4, merupakan hasil perhitungan dari Algoritma Bellman-Ford. Pada iterasi 1, dikarenakan pada iterasi 2 hasilnya memiliki nilai yang sama pada iterasi 1, maka iterasi dapat dihentikan dan dapat disimpulkan hasil akhir dari perhitungan tersebut yaitu melalui node 1-2-5-6-9 dengan melalui Jl. Raya Ragunan 1, Jl. Raya Ragunan 2, Jl. Salihara 1, dan Jl. Salihara 2 dengan jarak total 1.97 km.

b. Algoritma *Floyd-Warshall*

	1	2	3	4	5	6	7	8
1	0	0.35	0.65	0.83	1.20	1.34	1.47	1.54
2	$\infty$	0	0.30	0.48	0.85	0.99	1.12	1.19
3	$\infty$	$\infty$	0	0.18	0.55	0.69	$\infty$	0.89
4	$\infty$	$\infty$	$\infty$	0	$\infty$	1.00	$\infty$	1.20
5	$\infty$	$\infty$	$\infty$	$\infty$	0	0.14	$\infty$	0.34
6	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	$\infty$	0.20
7	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	0.60
8	$\infty$	0						

Gambar. 7. Matriks hasil algoritma *Floyd-Warshall* dari stasiun Pasar Minggu,

Dapat dilihat pada Gambar 7, merupakan hasil perhitungan dari Algoritma Floyd-Warshall dengan hasil akhirnya yaitu melalui node 1-2-3-5-6-8 dengan melalui Jl. Raya Pasar Minggu 1, Jl. Raya Pasar Minggu 2, Jl. Asem 1, Jl. Asem 2, dan Jl. Sawo Manila jarak total 1.54 km.

	1	2	3	4	5	6	7	8	9
1	0	0.40	1.00	1.45	1.35	1.75	1.10	1.50	1.97
2	∞	0	0.60	1.05	0.95	1.35	1.10	1.50	1.57
3	∞	∞	0	0.45	∞	0.80	0.50	0.90	1.02
4	∞	∞	∞	0	∞	0.35	∞	∞	0.57
5	∞	∞	∞	∞	0	0.40	∞	∞	0.62
6	∞	∞	∞	∞	∞	0	∞	∞	0.22
7	∞	∞	∞	0.70	∞	1.05	0	0.40	1.05
8	∞	∞	∞	0.30	∞	0.65	∞	0	0.65
9	∞	∞	∞	∞	∞	∞	∞	∞	0

Gambar. 8. Matriks hasil algoritma *Floyd-Warshall* dari halte Jatiapdang

Dapat dilihat pada Gambar 8, merupakan matriks hasil perhitungan dari Algoritma Floyd-Warshall dengan hasil akhirnya yaitu melalui node 1-2-5-6-9 dengan melalui Jl. Raya Ragunan 1, Jl. Raya Ragunan 2, Jl. Salihara 1, dan Jl. Salihara 2 dengan jarak total 1.97 km.

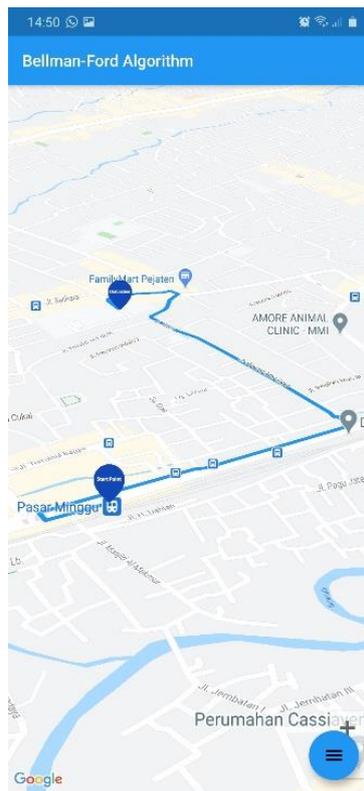
c. Tampilan Aplikasi

Berikut merupakan tampilan dari aplikasi pencarian jalur terdekat menuju Universitas Nasional.



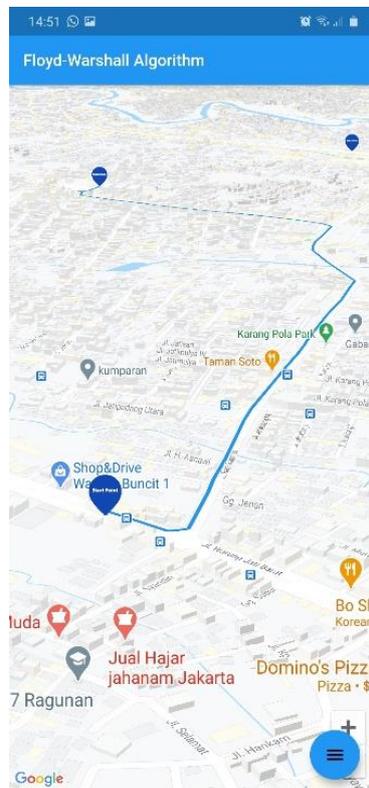
Gambar. 9. Tampilan home aplikasi pencarian jalur terdekat menuju Universitas Nasional

Pada Gambar 9 ,merupakan tampilan awal aplikasi dengan algoritma *Bellman-Ford* dan algoritma *Floyd-Warshall* dalam menentukan jarak terdekat menuju Universitas Nasional.



Gambar. 10. Tampilan aplikasi pencarian jalur terpendek menuju Universitas Nasional menggunakan algoritma Bellman-Ford

Pada Gambar 10 merupakan tampilan aplikasi dengan menggunakan algoritma Bellman-Ford dalam menentukan jarak terpendek dari stasiun Pasar Minggu menuju Universitas Nasional.



Gambar. 11. Tampilan aplikasi pencarian jalur terpendek menuju Universitas Nasional menggunakan algoritma Floyd-Warshall

Gambar 11 merupakan tampilan aplikasi dengan menggunakan algoritma *Floyd-Warshall* dalam menentukan jarak terpendek dari stasiun Pasar Minggu menuju Universitas Nasional.

### 3) Hasil dan Pembahasan

Berikut merupakan tabel perbandingan jarak dan waktu proses dari algoritma *Bellman-Ford* dan *Floyd-Warshall*.

TABEL V  
PERBANDINGAN JARAK DAN RATA-RATA WAKTU PROSES ALGORITMA BELLMAN-FORD

Pengujian	Stasiun Pasar Minggu		Halte Jatipadang	
	Jarak	Waktu	Jarak	Waktu
1	1.54 km	1.36 s	1.97 km	1.33 s
2	1.54 km	1.25 s	1.97 km	1.28 s
3	1.54 km	1.30 s	1.97 km	1.28 s
4	1.54 km	1.31 s	1.97 km	2.77 s
5	1.54 km	1.28 s	1.97 km	2.31 s

Pada Tabel 5, hasil jarak dan rata-rata waktu proses dari algoritma *Bellman-Ford* dari stasiun Pasar Minggu yaitu jarak total 1.54 km, dan rata-rata waktu proses pencarian jarak terpendek yaitu 1.30 s. Sedangkan dari halte Jatipadang yaitu jarak total 1.97 km dan rata-rata waktu proses pencarian jarak terpendek yaitu 1.794 s.

TABEL VI  
PERBANDINGAN JARAK DAN RATA-RATA WAKTU PROSES ALGORITMA FLOYD-WARSHALL

Pengujian	Stasiun Pasar Minggu		Halte Jatipadang	
	Jarak	Waktu	Jarak	Waktu
1	1.54 km	0,75 s	1.97 km	0,74 s
2	1.54 km	0,73 s	1.97 km	0,73 s
3	1.54 km	0,72 s	1.97 km	0,73 s
4	1.54 km	0,73 s	1.97 km	0,72 s
5	1.54 km	0,72 s	1.97 km	0,74 s

Pada Tabel 6, hasil jarak dan rata-rata waktu proses dari algoritma *Bellman-Ford* dari stasiun Pasar Minggu yaitu jarak total 1.54 km, dan rata-rata waktu proses pencarian jarak terpendek yaitu 0.73 s. Sedangkan dari halte Jatipadang yaitu jarak total 1.97 km dan rata-rata waktu proses pencarian jarak terpendek yaitu 0.732 s.

Berdasarkan hasil pengujian, algoritma *Bellman-Ford* dan *Floyd-Warshall* untuk kasus pencarian jarak terpendek dari stasiun Pasar Minggu diperoleh node 1-2-3-5-6-8 dengan melalui Jl. Raya Pasar Minggu 1, Jl. Raya Pasar Minggu 2, Jl. Asem 1, Jl. Asem 2, dan Jl. Sawo Manila dengan jarak total 1.54 km dan tingkat keefektifan jarak sebesar 39.40% dengan rata-rata waktu pencarian jalur dengan algoritma *Bellman-Ford* selama 1.30 s dan rata-rata waktu pencarian jalur dengan algoritma *Floyd-Warshall* yaitu selama 0.73 s. Sedangkan untuk pencarian jarak terpendek dari halte Jatipadang diperoleh node 1-2-5-6-9 dengan melalui Jl. Raya Ragunan 1, Jl. Raya Ragunan 2, Jl. Salihara 1, dan Jl. Salihara 2 dengan jarak total 1.97 km dan tingkat keefektifan jarak sebesar 25.24% dengan rata-rata waktu pencarian jalur dengan algoritma *Bellman-Ford* selama 1.794 s dan rata-rata waktu pencarian jalur dengan algoritma *Floyd-Warshall* yaitu selama 0.732 s.

## IV. KESIMPULAN

Berdasarkan kajian penelitian yang lalu pengimplementasiannya hanya sebatas perhitungan graf dan aplikasi berbasis web, maka penelitian ini dikembangkan dengan mengimplementasikan ke dalam aplikasi berbasis android flutter dengan membandingkan hasil jarak dan waktu proses dari algoritma *Bellman-Ford* dan *Floyd-Warshall*. Dari hasil penelitian yang telah dibahas sebelumnya dapat disimpulkan bahwa algoritma *Bellman-Ford* dan algoritma *Floyd-Warshall* bisa digunakan sebagai acuan dalam pembuatan aplikasi pencarian rute terpendek. Dari hasil pengujian aplikasi dapat dinyatakan bahwa, aplikasi ini dapat membantu masyarakat, mahasiswa dan peserta didik baru untuk mencari jalur terdekat dari stasiun Pasar Minggu dan halte Jatipadang menuju Universitas Nasional. Total jarak yang dihasilkan dari kedua algoritma sama, namun memiliki perbedaan pada waktu pencarian rute. Algoritma *Floyd-Warshall* lebih cepat memproses pencarian rute dibandingkan algoritma *Bellman-Ford* karena proses perhitungannya secara parallel (multiproses) sedangkan untuk perhitungan algoritma *Bellman-Ford* berdasarkan iterasi sebanyak node – 1.

Penelitian ini terbatas pada penentuan total jarak terpendek yang diujikan, aplikasi yang dirancang belum bekerja pada tipe kendaraan yang digunakan, kondisi jalan saat macet maupun lampu merah. Rencana pengembangan penelitian dibuatkan aplikasi dengan mendeteksi jenis kendaraan maupun kondisi jalan agar aplikasi dapat berjalan lebih maksimal.

## DAFTAR PUSTAKA

- [1] P. M. Hasugian, "Analisa Dan Implementasi Algoritma Bellman Ford Dalam Menentukan Jalur Terpendek Pengantaran Baraang Dalam Kota," *J. Ilm.*, vol. 18, no. 2, pp. 1–34, 2015, doi: 10.1017/CBO9781107415324.004.
- [2] B. I. A. Prasetyo and A. Maslan, "ANALISIS PERBANDINGAN PADA ALGORITMA BELLMAN FORD DAN DIJKSTRA PADA GOOGLE MAP," *Khazanah Ilmu Berazam*, vol. 3, no. 2, pp. 337–349, 2020.
- [3] Y. Darnita, R. Toyib, and R. Rinaldi, "Implementasi Algoritma Floyd Warshall Untuk Menentukan Letak Dan Lokasi Perusahaan Travel/Rental Mobil Di Kota Bengkulu," *Pseudocode*, vol. 4, no. 2, pp. 144–155, 2017, doi: 10.33369/pseudocode.4.2.144-155.
- [4] A. Agnes Kikelomo, Y. Nureni Asafe, A. Paul, and L. N. Olawale, "Design and Implementation of Mobile Map Application for Finding Shortest Direction between Two Pair Locations Using Shortest Path Algorithm: A Case Study," *Int. J.*, vol. 3305, no. August, p. 3300, 2017, [Online]. Available: <https://search.proquest.com/openview/f2c8bfc5f764555a045385551b7761b/1?pq-origsite=gscholar&cbl=886380>.
- [5] A. D. Yustita, S. A. Hardiyanti, and I. Yuniwati, "Algoritma Floyd-Warshall Untuk Penentuan Rute Terpendek Model Jaringan Pariwisata Kabupaten Banyuwangi," *J. Mat. dan Pendidik. Mat.*, vol. 3, no. 2, pp. 137–146, 2018.
- [6] E. T. H. Hutasoit, "Pencarian Rute Terpendek Menggunakan Algoritma Bellman-Ford (Studi Kasus: PT. JNE Medan)," *J. Sist. Komput. dan Inform.*, vol. 1, no. 1, p. 20, 2019, doi: 10.30865/json.v1i1.1367.
- [7] S. Hamdi and Prihandoko, "Analisis Algoritma Dijkstra dan Algoritma Bellman-Ford Sebagai Penentuan Jalur Terpendek Menuju Lokasi Kebakaran (Studi Kasus: Kecamatan Praya Kota)," *J. Ilm. Ilmu-Ilmu Tek.*, vol. 8, no. 1, pp. 26–32, 2018.
- [8] F. Anggraini and S. Mingparwoto, "Penerapan Metode Algoritma Bellman – Ford Dalam Aplikasi Pencarian Lokasi Perseroan Terbatas di PT . Jakarta Industrial Estate Pulogadung ( PT . JIEP )," *J. Teknol.*, vol. 7, no. 1, pp. 28–34, 2015.
- [9] R. A. Azdy and F. Darnis, "Implementasi Bellman-Ford untuk Optimasi Rute Pengambilan Sampah di Kota Palembang," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 8, no. 4, p. 327, 2019, doi: 10.22146/jnteti.v8i4.532.
- [10] M. J. Bannister and D. Eppstein, "Randomized speedup of the Bellman-Ford algorithm," *9th Meet. Anal. Algorithmics Comb. 2012, ANALCO 2012*, pp. 41–47, 2012, doi: 10.1137/1.9781611973020.6.
- [11] A. Muzakir and H. Hutrianto, "Bellman-Ford Algorithm for Completion of Route Determination: An Experimental Study," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 6, no. 1, p. 29, 2020, doi: 10.26555/jiteki.v16i1.16943.
- [12] S. Wijaya, "Implementasi Algoritma Bellman Ford Pada Aplikasi Pencarian Pengobatan Patah Tulang Kem Kem Terdekat di Kota Medan Berbasis Android," *J. Ris. Komput.*, vol. 6, no. 1, pp. 30–36, 2019, [Online]. Available: <http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom>.
- [13] J. Nasional, S. Informasi, and V. Apriliani, "Penerapan Algoritma Floyd Warshall dalam Aplikasi Penentuan Rute Terpendek Mencari Lokasi BTS ( Base Tower Station ) pada PT . GCI Palembang," vol. 02, pp. 81–88, 2018.
- [14] F. Widya and T. Andrasto, "Penerapan Algoritma Floyd-Warshall dalam Menentukan Rute Terpendek pada Pemodelan Jaringan Pariwisata di Kota Semarang," *J. Tek. Elektro*, vol. 8, no. 1, pp. 21–24, 2016.
- [15] M. R. Mukti and . M. ., "MENENTUKAN RUTE TERPENDEK DENGAN MENGGUNAKAN ALGORITMA FLOYD-WARSHALL DALAM PENDISTRIBUSIAN BARANG PADA PT. ROPY RAY PUTRATAMA," *KARISMATIKA Kumpul. Artik. Ilmiah, Inform. Stat. Mat. dan Apl.*, vol. 4, no. 1, pp. 39–53, 2018, doi: 10.24114/jmk.v4i1.11857.