

**ANALISIS PERBANDINGAN ALGORITMA JOHNSON DAN ALGORITMA
BELLMAN-FORD UNTUK PENENTUAN JALUR PENGANTARAN
BARANG OLEH *FREELANCE BICYCLE COURIER***

SKRIPSI

**MUHAMMAD SYARIF RIDHO RAMBE
181401081**



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2023**

**ANALISIS PERBANDINGAN ALGORITMA JOHNSON DAN ALGORITMA
BELLMAN-FORD UNTUK PENENTUAN JALUR PENGANTARAN
BARANG OLEH *FREELANCE BICYCLE COURIER***

SKRIPSI

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer**

MUHAMMAD SYARIF RIDHO RAMBE

181401081



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2023**

PERSETUJUAN

Judul : ANALISIS PERBANDINGAN ALGORITMA
JOHNSON DAN ALGORITMA BELLMAN-
FORD UNTUK PENENTUAN JALUR
PENGANTARAN BARANG OLEH
FREELANCE BICYCLE COURIER

Kategori : SKRIPSI

Nama : MUHAMMAD SYARIF RIDHO RAMBE

Nomor Induk Mahasiswa : 181401081

Program Studi : SARJANA (S1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI
INFORMASI UNIVERSITAS SUMATERA
UTARA

Medan, 6 Desember 2023

Komisi Pembimbing :

Pembimbing II

Pembimbing I



Dr. M. Andri Budiman, S.T., Dian Rachmawati, S.Si, M.Kom.

M.Comp.Sc., M.E.M., S.C.J.P. NIP. 198802192019032016

NIP. 197510082008011011

Diketahui/disetujui oleh

Program Studi SI Ilmu Komputer

Ketua,



Dr. Amalia ST., M.T.

NIP. 197812212014042001

PERNYATAAN

ANALISIS PERBANDINGAN ALGORITMA JOHNSON DAN ALGORITMA BELLMAN-FORD UNTUK PENENTUAN JALUR PENGANTARAN BARANG OLEH *FREELANCE BICYCLE COURIER*

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 6 Desember 2023



Muhammad Syarif Ridho Rambe

181401081

PENGHARGAAN

Dengan rasa syukur kepada Allah SWT tuhan yang maha esa yang sudah melimpahkan karuniaNya, hingga penulis dapat mampu merampungkan tugas akhir ini, yang menjadi salah satu acuan persyaratan dalam memperoleh gelar Sarjana Komputer, pada Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Dengan ini penulis tidak lupa mengirimkan shalawat dan salam kepada Nabi Muhammad Shallallahu 'alaihi Wasallam, yang telah membawa manusia kepada hal yang lebih baik dari sebelumnya serta yang telah menjadi perantara kita dalam memperoleh berkah ilmu pengetahuan yang diridhoi oleh Allah.

Penulis tidak lupa dan sangat menginginkan untuk mengutarakan apresiasi serta hormat yang sebanyak-banyaknya kepada:

1. Bapak Dr. Muryanto Amin, S.Sos, M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
3. Ibu Dr. Amalia ST., M.T. selaku Ketua Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dian Rachmawati, S.Si, M.Kom. selaku dosen pembimbing I yang telah memberikan bimbingan, kritik, semangat, dan saran serta arahan menuju kebaikan kepada penulis dalam menyelesaikan skripsi ini.
5. Bapak Dr. M. Andri Budiman, S.T., M.Comp.Sc., M.E.M., S.C.J.P selaku dosen pembimbing II yang telah memberikan bimbingan, kritik, semangat, dan saran serta arahan menuju kebaikan kepada penulis dalam menyelesaikan skripsi ini.
6. Seluruh Bapak dan Ibu Dosen Program Studi S-1 Ilmu Komputer yang telah memberikan waktu dan tenaga untuk mengajar dan membimbing sehingga penulis dapat sampai kepada tahap penyusunan skripsi ini.
7. Seluruh Staf Pegawai Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah banyak memberikan bantuan kepada penulis selama masa perkuliahan sampai kepada tahap penyusunan skripsi ini.

8. Yang terhormat dan saya sayangi kedua orang tua yang tak pernah lupa dan selalu memberikan kekuatan, doa, dan banyak hal yang penulis butuhkan selama melakukan Pendidikan.
9. Yang saya sayangi kakak-kakak serta abang-abang saya yang terus memberikan arahan dan semangat kepada penulis.
10. Yang saya kasihi teman-teman penulis selama perkuliahan terkhusus kepada Ary Bobby Siregar, Syahrenina Diva Poranc, Fauzan Zaman, Nikita Masaling, Selfhy Agina Ginting yang tidak bosan membantu dan menyemangati penulis dalam merampungkan pendidikan penulis.
11. Teman teman Shredderboiz terutama Joel Purba, Rangga Blek, Angga, Billy, dan Bang Angga Maeswara yang senantiasa menghibur penulis.
12. Teman teman masa kecil penulis, Arif, Rizky, Duan, Reza.
13. Teman teman seperjuangan saya stambuk 2018 terkhusus kom c yang telah memberikan pengalaman serta membantu penulis.
14. Kawan kawan sesama pejuang skripsi Adit, dan Usup.
15. Teman teman Flat Out yang membantu dalam mengerjakan penelitian ini.
16. Teman teman saya Nadhira, Virja, Ibnu, Primus, Ghani, Qila yang memberikan semangat serta arahan dalam mengerjakan tugas akhir saya.

Semoga kasih dan sayang Allah senantiasa melindungi dan memberikan keberkahannya pada semua individu yang telah memberikan dukungan dalam berbagai bentuk, baik dalam bentuk motivasi, panduan, tindakan nyata, atau dukungan moral terhadap saya selama ini. Semoga skripsi yang saya tulis bisa memberikan dampak serta manfaat terhadap diri saya sendiri, keluarga, masyarakat, organisasi, serta negara.

Medan, 6 Desember 2023



Muhammad Syarif Ridho Rambe

181401081

ABSTRAK

Dalam era digital, penjualan dan pembelian barang semakin mudah berkat platform digital. Khususnya, keberhasilan transaksi tergantung pada efisiensi dan kecepatan jasa pengiriman barang. Kurir sepeda, dikenal sebagai Bicycle Messenger, menawarkan solusi ramah lingkungan dan cepat dalam mengantarkan barang. Namun, di Kota Medan, kurir sepeda masih mengandalkan pemahaman personal terhadap rute pengantaran. Freelance Bicycle Courier, yang tidak terikat kontrak, semakin populer dalam mendukung kebutuhan pengantaran barang oleh organisasi seperti Medan Bike Messenger dan JNE. Pencarian rute terpendek adalah tantangan umum dalam ilmu komputer dan teori graf. Algoritma Bellman-Ford dan Algoritma Johnson muncul sebagai solusi untuk mengoptimalkan rute pengantaran barang oleh kurir sepeda. Bellman-Ford dapat memberikan hasil yang efisien dengan memperhitungkan bobot dalam graf, dengan konsep early stopping untuk meningkatkan kecepatan. Sementara itu, Algoritma Johnson, menggabungkan Bellman-Ford dan Dijkstra, memiliki keunggulan dalam mencari semua rute terpendek antara pasangan node. Penelitian ini menyimpulkan bahwa Algoritma Bellman-Ford dan Johnson efektif digunakan untuk menentukan rute pengantaran paket oleh kurir sepeda. Algoritma Bellman-Ford memiliki kompleksitas yang lebih baik dan running time yang lebih rendah ($\Theta(V * E)$) dibandingkan dengan Algoritma Johnson ($\Theta(V^2) + (V * E)$). Jumlah node dalam graf mempengaruhi running time masing-masing algoritma. Algoritma Johnson bersifat "find all shortest path", sementara Algoritma Bellman-Ford berhenti ketika iterasi menghasilkan nilai yang sama dengan iterasi sebelumnya. Hasil penelitian ini dapat memberikan panduan bagi kurir sepeda dan organisasi pengantaran barang untuk meningkatkan efisiensi dan kecepatan dalam layanan pengantaran.

Kata Kunci: Bellman-Ford, Johnson, *Shortest Path*, *Bicycle Courier*.

ABSTRACT

In the current digital era, the buying and selling of various goods have been greatly facilitated by digital platforms. The success of transactions, particularly in the realm of e-commerce, relies heavily on the efficiency and speed of delivery services. Bicycle couriers, commonly known as Bicycle Messengers, offer an environmentally friendly and swift solution for delivering goods. However, in the city of Medan, bicycle couriers still rely on personal knowledge of delivery routes. Freelance Bicycle Couriers, individuals not bound by a fixed contract, have become increasingly popular in supporting the delivery needs of organizations such as Medan Bike Messenger and JNE. The challenge of finding the shortest route is a common issue in computer science and graph theory. The Bellman-Ford algorithm and the Johnson algorithm have emerged as solutions to optimize the delivery routes for bicycle couriers. Bellman-Ford provides efficient results by considering the weights of the graph, incorporating the concept of early stopping to enhance speed. Meanwhile, the Johnson algorithm, combining Bellman-Ford and Dijkstra, excels in finding all the shortest routes between pairs of nodes. This research concludes that the Bellman-Ford and Johnson algorithms are effective in determining delivery routes for bicycle couriers. The Bellman-Ford algorithm exhibits better complexity and lower running time ($\Theta(V * E)$) compared to the Johnson algorithm ($\Theta(V^2) + (V * E)$). The number of nodes in the graph influences the running time of each algorithm. The Johnson algorithm is characterized by "find all shortest paths," while the Bellman-Ford algorithm stops when the current iteration produces the same result as the previous iteration. The findings of this research can guide bicycle couriers and delivery organizations to enhance efficiency and speed in their delivery services.

Keywords: Bellman-Ford, Johnson, Shortest Path, Bicycle Courier.

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN	iv
PENGHARGAAN	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	15
1.1. Latar Belakang	15
1.2. Rumusan Masalah	17
1.3. Batasan Masalah	17
1.4. Tujuan Penelitian	17
1.5. Manfaat Penelitian	18
1.6. Metodologi Penelitian	18
1.7. Sistematika Penulisan	19
BAB II LANDASAN TEORI	21
2.1. Teori Graf	21
2.2. <i>Shortest Path Problem</i>	25
2.3. <i>Minimum Spanning Tree</i>	26
2.4. Algoritma Bellman-Ford	26
2.5. Algoritma Johnson	28
2.6. Kompleksitas Algoritma	29
2.7. Mapbox	30

2.8. Penelitian yang Relevan.....	30
BAB III ANALISIS DAN PERANCANGAN SISTEM	32
3.1. Analisis Sistem.....	32
3.1.1. Analisis Masalah	32
3.1.2. Analisis Kebutuhan	33
3.1.2.1. Kebutuhan Fungsional	33
3.1.2.2. Kebutuhan Non-Fungsional.....	33
3.1.3. Diagram Umum Sistem.....	34
3.2. Pemodelan Sistem.....	34
3.2.1. <i>Use Case Diagram</i>	35
3.2.2. <i>Activity Diagram</i>	36
3.2.2.1. <i>Activity Diagram</i> Pencarian Rute Pengantaran.....	36
3.2.3. <i>Sequence Diagram</i>	36
.....	37
3.3. <i>Flowchart</i>	37
3.3.1. <i>Flowchart</i> Sistem	38
3.3.2. <i>Flowchart</i> Algoritma Bellman-Ford	39
3.3.3. <i>Flowchart</i> Algoritma Johnson.....	40
3.4. Perancangan <i>Interface</i>	41
3.4.1. Halaman Awal.....	41
3.4.2. Halaman Cek Peta	42
3.4.3. Halaman masukkan alamat pengantaran	43
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....	44
4.1. Implementasi.....	44
4.1.1. Halaman <i>List</i> Alamat.....	44
4.1.2. Halaman Pencarian Alamat	45
4.1.3. Halaman Tampilan Peta	46

4.1.4.	Halaman Perhitungan Algoritma.....	47
4.1.5.	Halaman Tampilan Rute Pengantaran.....	48
4.2.	Pengujian Sistem.....	48
4.2.1.	Pengujian Implementasi Algoritma Bellman-Ford	50
4.2.2.	Perhitungan Manual Algoritma Bellman-Ford.....	50
4.2.3.	Pengujian Implementasi Algoritma Johnson.....	54
4.2.4.	Perhitungan Manual Algoritma Johnson.....	54
4.3.	Hasil Pengujian	59
4.4.	Kompleksitas Algoritma	63
4.4.1.	Kompleksitas Algoritma Bellman Ford	64
4.4.2.	Kompleksitas Algoritma Johnson	65
BAB V PENUTUP		67
5.1.	KESIMPULAN.....	67
5.2.	SARAN	67
DAFTAR PUSTAKA		68

DAFTAR GAMBAR

Gambar 2.1 Graf Sederhana	21
Gambar 2.2 Graf Tidak Sederhana	22
Gambar 2.3 Graf Ganda	22
Gambar 2.4 Graf Berarah	23
Gambar 2.5 Graf Tidak Berarah.....	24
Gambar 2.6 Graf Berbobot.....	24
Gambar 2.7 Graf Tidak Berbobot.....	25
Gambar 2.8 Graf Pada Permasalahan Shortest Path.....	25
Gambar 3.1 Ishikawa Diagram.....	32
Gambar 3.2 Diagram Umum Sistem	34
Gambar 3.3 Use Case Diagram	35
Gambar 3.4 Activity Diagram User Pencarian Lokasi	36
Gambar 3.5 Sequence Diagram.....	37
Gambar 3.6 Flowchart Sistem	38
Gambar 3.7 Flowchart Algoritma Bellman-Ford	39
Gambar 3.8 Flowchart Algoritma Johnson	40
Gambar 3.9 Halaman Beranda	41
Gambar 3.10 Halaman Cek Peta	42
Gambar 3.11 Halaman Input Alamat Pengantaran.....	43
Gambar 4.1 Halaman List Alamat.....	44
Gambar 4.2 Pencarian alamat.....	45
Gambar 4.3 Halaman Tampilan Peta.....	46
Gambar 4.4 Halaman Perhitungan Algoritma	47
Gambar 4.5 Halaman Tampilan Rute Pengantaran	48
Gambar 4.6 Titik Awal Pengujian Algoritma	50
Gambar 4.7 Graf Permasalahan Bellman-Ford	51
Gambar 4.8 Langkah Kedua Perhitungan Bellman Ford	51
Gambar 4.9 Bobot Nilai Akhir Setelah Perhitungan.....	53
Gambar 4.10 Graf Permasalahan Johnson.....	54
Gambar 4.11 Langkah Kedua Algoritma Johnson	55

Gambar 4.12 Hasil Bellman Ford Pada Algoritma Johnson	55
Gambar 4.13 Hasil Perhitungan Johnson	59
Gambar 4.14 Chart Perbandingan Running Time Algoritma.....	62
Gambar 4.15 Aplikasi mudah digunakan	63
Gambar 4.16 Aplikasi membantu kurir.....	63

DAFTAR TABEL

Tabel 4.1 Tabel Daftar Lokasi Pengujian.....	49
Tabel 4.2 Pencarian Node 1-6 Iterasi 1	52
Tabel 4.3 Perhitungan Iterasi 1 Node 7-12.....	52
Tabel 4.4 Iterasi Pertama Dan Iterasi Kedua.....	53
Tabel 4.5 Perhitungan Dijkstra Pada Algoritma Johnson	56
Tabel 4.6 Hasil Perhitungan Johnson	57
Tabel 4.7 Lokasi Pengujian.....	59
Tabel 4.8 Lokasi Pengujian.....	60
Tabel 4.9 Lokasi Pengujian.....	61
Tabel 4.10 Lokasi Pengujian.....	62
Tabel 4.11 Kompleksitas Algoritma Bellman Ford	64
Tabel 4.12 Kompleksitas Algoritma Johnson	65
Tabel 4.13 Kompleksitas Algoritma Johnson	66

BAB I

PENDAHULUAN

1.1. Latar Belakang

Penjualan dan pembelian beragam ragam jenis barang saat ini sudah dimudahkan oleh *platform digital*. Banyaknya transaksi mengharuskan jasa pengiriman barang mampu melakukan pekerjaannya dengan seefisien dan secepat mungkin.

Bicycle Messenger atau yang biasa dikenal sebagai kurir sepeda adalah sekelompok orang yang membawa dan mengantarkan barang menggunakan sepeda. Pekerjaan kurir sepeda relatif sederhana, Pekerjaan ini melibatkan penjemputan barang atau paket maupun dokumen dari satu lokasi dan mengantarkannya ke lokasi yang dituju (Fincham, 2008). Kurir sepeda memiliki beberapa kelebihan seperti lebih ramah lingkungan serta dapat bergerak lebih cepat karena bisa menghindari kemacetan ataupun kepadatan lalu lintas yang ada pada kota kota besar (Maes & Vanelslender, 2012). Dalam pengantaran barang yang dilakukan oleh kurir sepeda, para kurir harus dituntut mampu mengantarkan barang dengan cepat dan juga dengan aman, namun dalam prakteknya yang terjadi di Kota Medan kurir sepeda masih menggunakan kemampuan mencari rute menggunakan pemahamannya terhadap daerah pengantarannya.

Freelance Bicycle Courier adalah orang orang yang melakukan pekerjaan pengantaran barang menggunakan sepeda yang tidak terikat kontrak tetap pada sebuah organisasi atau perusahaan, biasanya mereka akan mengambil pesanan pengantaran ketika diminta oleh organisasi atau perusahaan yang bergerak dibidang pengantaran barang menggunakan sepeda. Di Kota Medan *freelance bicycle courier* biasanya bekerja sama dengan Medan Bike Messenger dan JNE yang memerlukan sumber daya manusia dalam mengantarkan barang barang yang akan didistribusikan, namun tidak jarang juga *freelance bicycle courier* mendapatkan pesanan dari pelanggan lainnya.

Pencarian rute terpendek merupakan permasalahan umum di bidang ilmu komputer, Permasalahan dalam pencarian rute terpendek sudah dipakai secara luas

dalam berbagai bidang dengan niat untuk mengoptimalkan suatu sistem (Purwananto, et al., 2005). Jalur terpendek adalah topik penelitian penting dalam teori graf. Dalam dan luar negeri membahasnya secara mendalam dan mengusulkan beragam algoritma untuk memecahkan pertanyaan jalur terpendek, algoritma jalur terpendek dapat dibagi menjadi jalur terpendek sumber tunggal dan jalur terpendek multi-sumber (Wang, 2018). Dalam mencari rute terpendek yang dapat digunakan oleh kurir sepeda bisa menggunakan banyak algoritma *shortest path* yang dapat menghasilkan rute yang efisien, Algoritma Bellman-Ford dan Algoritma Johnson dapat digunakan untuk mendapatkan hasil rute yang efisien.

Dalam mencari rute terpendek yang di dalamnya memiliki bobot dapat digunakan suatu Algoritma yang bernama Bellman-Ford. Untuk menemukan solusi rute terpendek, Bellman-Ford akan menghitung setiap node (Mustofa, et al., 2022). Bellman-Ford akan menghasilkan jawaban yang benar jika di dalam graf yang akan diproses tidak didapati adanya suatu *loop* yang memiliki bobot negatif yang menuju ke titik tujuan (Bawole & Chernovita, 2019). Algoritma Bellman Ford juga dapat berhenti lebih cepat dalam kasus tertentu, jika didapati iterasi yang dilakukan dan iterasi sebelumnya memiliki hasil yang sama, konsep ini dikenal sebagai *early stopping* atau henti cepat.

Dalam pencarian rute terpendek adapula algoritma lainnya yang dapat digunakan salah satunya adalah Algoritma Johnson, Johnson merupakan kombinasi dari 2 algoritma yaitu Bellman-Ford dan juga Dijkstra (Anintha & Ramesh Babu, 2018). Johnson mendapatkan kelebihan dengan *all-pair shortest path* dikarenakan pencampuran dari 2 algoritma, Algoritma Johnson akan mencari rute terpendek menuju semua pasangan node yang ada pada graf (Auparay & Ijtihadie, 2018).

Dikarenakan alasan itu penulis melakukan penelitian mengenai algoritma Johnson dan Bellman-Ford dalam menentukan rute tercepat untuk jalur pengantaran barang oleh *bicycle courier*.

1.2. Rumusan Masalah

Mengacu pada latar permasalahan tersebut, teridentifikasi adanya isu didalam penentuan rute pengiriman barang yang dilakukan oleh kurir masih dilakukan secara manual dengan mengandalkan ingatan dan kemampuan mencari rute sehingga rute yang dilewati tidak efisien, oleh karena itu diperlukan suatu sistem yang membantu kurir dengan menerapkan algoritma Bellman Ford dan algoritma Johnson

1.3. Batasan Masalah

Berikut ini merupakan batasan masalah yang ada pada penelitian ini:

1. Jenis graf yang dihitung ialah graf yang memiliki arah dan bobot nilai. Bobot disini menyatakan jarak antar lokasi.
2. Untuk mengukur kinerja Algoritma Johnson dan Bellman-Ford, parameter kompleksitas yang digunakan adalah *Big-Theta* (Θ), serta *running time* aplikasi menggunakan satuan milisekon (ms).
3. Pengambilan jarak menggunakan layanan *API* yang disediakan oleh Mapbox dengan jarak dalam ukuran kilometer.
4. Permasalahan yang terjadi pada jalur lintasan dianggap tidak ada, seperti kondisi jalan dan kepadatan lalu lintas.

1.4. Tujuan Penelitian

Tujuan yang diharapkan dari penelitian ini adalah:

1. Mendapatkan *software* yang membantu kurir sepeda dalam menentukan rute yang akan dipilih dengan mengimplementasikan algoritma Bellman Ford dan Johnson.
2. Melakukan analisis performa algoritma Bellman-Ford dan Johnson dalam penentuan rute pengantaran kurir sepeda, dengan fokus pada kecepatan eksekusi.

1.5. Manfaat Penelitian

Manfaat dengan adanya penelitian ini dapat membantu organisasi atau perusahaan dalam menentukan rute pengantaran kurir sepeda, dapat memberikan wawasan mengenai bagaimana algoritma Bellman Ford dan Johnson dapat diimplementasikan dalam mengoptimalkan rute pengantaran, dapat menghemat sumber daya seperti waktu dan energi para kurir dengan menunjukkan rute yang dihitung oleh kedua algoritma, serta dapat mempermudah dan mempercepat distribusi barang di Kota Medan yang dilakukan oleh kurir sepeda.

1.6. Metodologi Penelitian

Metode penelitian yang diterapkan dalam studi ini adalah sebagai berikut:

1. Studi Pustaka

Tahapan ini akan dilakukan penelitian mulai dari mencari rujukan terhadap penelitian yang dilakukan. Referensi yang digunakan dalam bentuk jurnal, paper, makalah, artiker ilmiah, skripsi, situs internet yang berhubungan dengan Algoritma Johnson dan Algoritma Bellman-Ford serta data mengenai pengiriman barang yang akan dilakukan oleh kurir sepeda.

2. Analisis dan Perancangan

Dalam melaksanakan penelitian, Algoritma Johnson dan juga Algoritma Bellman-Ford akan di analisa agar didapatkan hal apa saja yang diperlukan dalam penelitian. Dalam tahap ini juga dilakukan pembuatan diagram alir (*flow chart*), diagram Ishikawa, *usecase diagram*, *activity diagram*, *sequence diagram*, dan perancangan *interface*.

3. Implementasi Sistem

Di tahap ini, penulis melakukan pembuatan sistem dengan mengimplementasikan algoritma Bellman-Ford dan juga algoritma Johnson menggunakan *framework flutter*.

4. Pengujian Sistem

Di fase ini nantinya dilakukan percobaan dan pemeriksaan pada sistem yang dibuat sesuai, untuk menentukan sistem yang sudah dikerjakan akan beroperasi sesuai dengan yang diinginkan.

5. Dokumentasi Sistem

Setiap tahapan yang telah dilakukan akan didokumentasikan agar dapat dijadikan kesimpulan yang akan digunakan oleh penulis dalam bentuk skripsi.

1.7. Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima bab, yakni:

BAB I PENDAHULUAN

Bab ini berfungsi sebagai bagian yang memperkenalkan latar belakang penelitian, menguraikan rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian, dan sistematika penulisan. Ini adalah langkah awal yang penting dalam suatu penelitian ilmiah untuk memberikan pemahaman kepada pembaca mengenai konteks dan kerangka kerja penelitian tersebut. Penjabaran masing-masing elemen dalam bab ini dapat membantu memandu pembaca melalui isi penelitian dan merinci aspek-aspek kunci yang berhubungan dengan penelitian tersebut.

BAB II LANDASAN TEORI

Tinjauan teori yang berkaitan dengan Algoritma Johnson dan Algoritma Bellman-Ford akan dibahas pada bagian ini.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini akan menyelidiki permasalahan yang dapat terjadi pada sistem yang nantinya dibangun, lalu akan disambung dengan perancangan sistem dengan menggunakan Algoritma Johnson dan Algoritma Bellman-Ford.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bagian ini memuat bagaimana penerapan juga pengujian pada sistem dengan didasarkan analisis serta perancangan.

BAB V KESIMPULAN DAN SARAN

Bab kali ini menyediakan rangkuman setelah riset selesai dilaksanakan serta berisikan anjuran yang dapat digunakan untuk kepentingan pada riset riset kemudian.

BAB II

LANDASAN TEORI

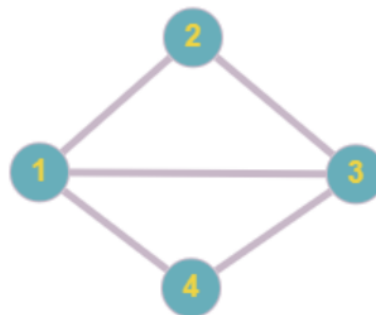
2.1. Teori Graf

Suatu pembelajaran yang mengeksplorasi karakter pada graf dapat dikatakan sebagai teori graf. Sekumpulan objek yang memiliki hubungan dan keterkaitan dapat dikatakan menjadi sebuah graf. Graf ($A = (B, C)$) dapat didefinisikan sebagai struktur yang terdiri dari suatu himpunan objek $B = \{b_1, b_2, \dots\}$, yang dikenal sebagai himpunan titik, dan $C = \{c_1, c_2, \dots\}$, yang merupakan kumpulan sisi. Setiap sisi, C , dalam graf dihubungkan dengan pasangan tak-terurut (b_i, b_j) , di mana b_i dan b_j adalah titik yang terkait dengan sisi tersebut. Oleh karena itu, dalam konteks sisi c_k , pasangan (b_i, b_j) mengacu pada titik ujung sisi c_k yang bersangkutan. (Buhaerah, et al., 2022).

Graf dibagi menjadi dua kelompok yang didasarkan dengan ada atau tidak adanya *loop* atau sisi gandanya, dua kelompok graf yakni graf sederhana dan graf tidak sederhana (Buhaerah, et al., 2022).

1. Graf Sederhana

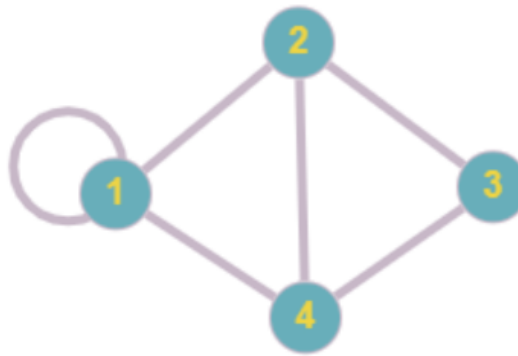
Pada gambar 2.1 di bawah merupakan model daripada graf sederhana yang tidak terdapat loop di dalamnya serta tidak terdapat adanya bagian paralel sehingga graf di bawah termasuk kedalam graf sederhana.



Gambar 2.1 Graf Sederhana

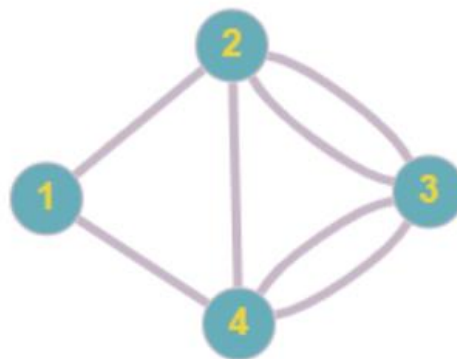
2. Graf Tidak Sederhana

Graf yang tidak bersifat sederhana ialah graf yang memungkinkan keberadaan sisi yang berpasangan. Graf yang tidak bersifat sederhana diklasifikasikan menjadi 2 jenis, yaitu graf semu dan graf ganda.



Gambar 2.2 Graf Tidak Sederhana

Graf semu memiliki (*loop*) yang pada sisi lain yang akan menuju kepada dirinya sendiri seperti pada gambar 2.2 di atas.



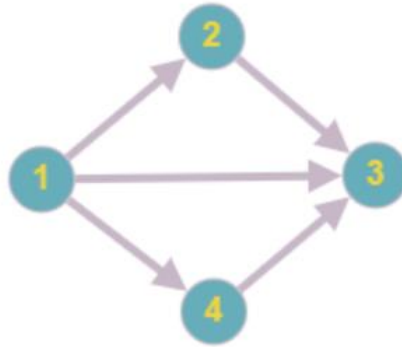
Gambar 2.3 Graf Ganda

Graf ganda mempunyai sisi yang berganda, dengan setiap sisinya dihubungkan sebagai bagian yang tidak terurut sebagaimana ditampilkan pada gambar 2.3 di atas.

Graf juga dibedakan berdasarkan arah pada sisinya yakni graf yang memiliki arah pada simpulnya serta juga graf yang tidak didapati notasi arah pada simpulnya.

1. Graf Berarah (*directed graph*)

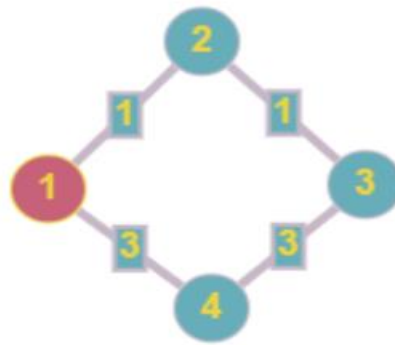
Pada gambar 2.4 di bawah terdapat pemodelan dari graf berarah dimana tanda panah mengorientasikan arah graf.



Gambar 2.4 Graf Berarah

2. Graf Tidak Berarah (*undirected graph*)

Suatu jenis struktur graf yang ditandai oleh sifat bahwa setiap sisi dalam graf tidak memiliki arah atau orientasi tertentu dapat dikatakan sebagai graf yang tidak berarah. Dalam graf ini, tidak ada perbedaan antara simpul awal dan simpul akhir dalam setiap sisi, dan urutan pasangan simpul tidak relevan. Dengan kata lain, graf ini hanya mencatat keterhubungan antara simpul-simpul tanpa memperhatikan arah atau urutan tertentu dalam setiap sisi. Ini membuat graf tidak berarah sering digunakan untuk menggambarkan hubungan atau keterkaitan antara objek atau konsep tanpa mempertimbangkan arah atau urutan tertentu dalam interaksi tersebut.

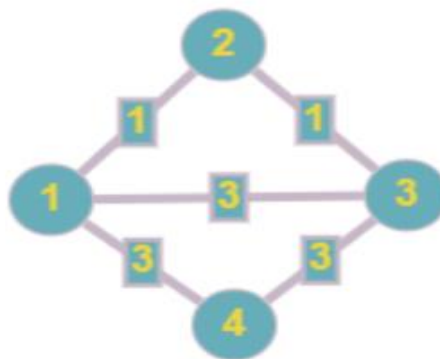


Gambar 2.5 Graf Tidak Berarah

Graf juga dibedakan berdasarkan nilai yang terdapat pada graf tersebut, graf juga dapat dipisahkan ke dalam dua jenis yakni graf yang memiliki bobot nilai didalamnya serta graf yang tidak memiliki bobot nilai.

1. Graf Berbobot (*weighted graph*)

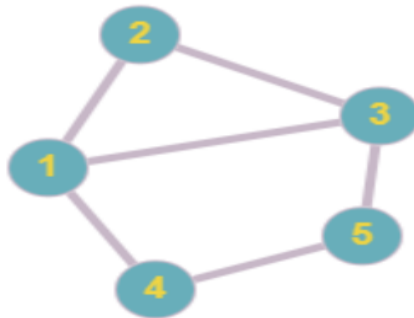
Graf yang dimana terdapat nilai pada sisi simpulnya dapat dikatakan sebagai graf berbobot. Nilai yang ada pada graf dapat memiliki nilai yang bervariasi pada setiap sisinya. Graf yang memiliki nilai bobot dapat dicermati pada gambar 2.6 di bawah.



Gambar 2.6 Graf Berbobot

2. Graf Tidak Berbobot (*unweighted graph*)

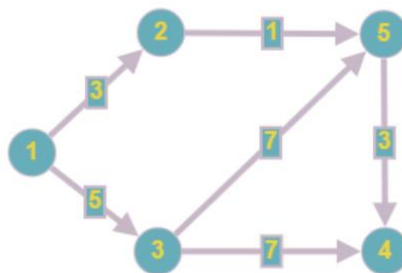
Graf tidak berbobot merupakan jenis graf di mana sisinya tidak didapati nilai bobot, sebagaimana diilustrasikan dalam Gambar 2.7 di bawah.



Gambar 2.7 Graf Tidak Berbobot

2.2. Shortest Path Problem

Shortest Path Problem (STP) adalah permasalahan menemukan rute terpendek di dalam suatu graf dengan bobot paling minimum. Dalam konteks ini, persoalan dapat dimodelkan dalam bentuk graf berbobot. Pada graf berbobot, setiap sisi



Gambar 2.8 Graf Pada Permasalahan Shortest Path

mempunyai nilai yang mewakili aspek yang ingin diukur atau dipecahkan dalam persoalan tersebut. Kata "terpendek" dalam istilah "lintasan terpendek" tidak hanya merujuk pada jarak fisik, tetapi dapat bervariasi tergantung pada jenis persoalan yang sedang dihadapi. Istilah tersebut dapat menggambarkan sejauh mana kemudahan suatu node menuju simpul lainnya (Fitro, et al., 2018).

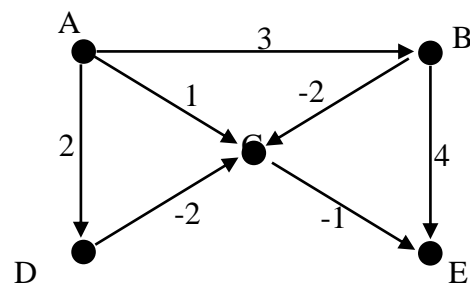
Gambar di atas merupakan contoh graf dengan permasalahan pencarian rute terdekat. Tiap node atau simpul direpresentasikan dalam bentuk lingkaran. Jalur penghubung atau *edge* direpresentasikan dengan garis yang memiliki tanda panah. Tanda panah sendiri memiliki arti bahwa jalur tersebut hanya searah yang berarti graf tersebut merupakan *directed graph*. Tiap *edge* juga memiliki angka di bawah garis yang merepresentasikan bobot atau *weight* yang merupakan *cost* atau biaya untuk perpindahan dari node awal menuju berikutnya.

2.3. *Minimum Spanning Tree*

Minimum Spanning Tree (MST) atau dapat disebut pohon rentang minimum adalah sebuah variasi dari *Shortest Path Problem* yang memiliki perbedaan pada rute yang akan dicari, di pohon rentang minimum, rute yang akan dihitung adalah garis-garis yang akan mengaitkan satu node ke node lainnya yang terdapat di jaringan yang selanjutnya akan mendapatkan panjang busur minimum, sedangkan pada permasalahan jalur terpendek yang dihitung adalah jalur dari node awal ke node terakhir yang menjadi destinasi dimana nantinya akan menghasilkan jarak minimum (Alamsyah, 2010).

2.4. Algoritma Bellman-Ford

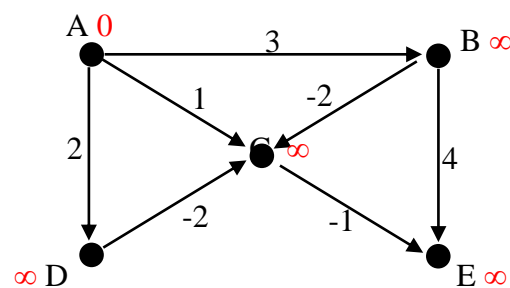
Agar dapat mengatasi masalah jalur terpendek pada suatu graf, dapat dipecahkan dengan menggunakan Bellman-Ford. Dalam pencarian solusi rute terpendek, Bellman-Ford akan menghitung setiap node pada graf. (Mustofa *et al.*, 2022). Pada algoritma Bellman-Ford dinyatakan akurat bilamana di dalam graf yang akan dihitung tidak terdapat adanya *cycle* yang bernilai negatif. (Bawole & Chernovita, 2019). Jika *cycle* berbobot negatif ada di dalam graf maka akan ada setidaknya satu *vertex* yang akan menghasilkan jarak yang akan lebih pendek, sehingga tidak dapat dihasilkannya bobot minimum.



Permasalahan di atas adalah untuk mendapatkan rute terdekat dari *vertex* awal menuju *vertex* lainnya di dalam graf. Adapun langkah penyelesaiannya sebagai berikut:

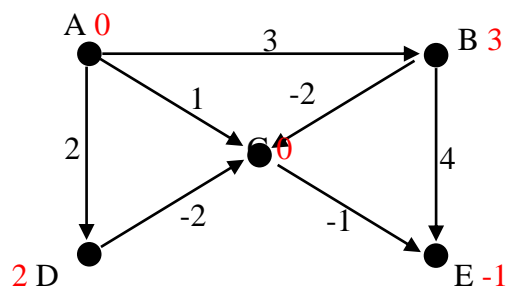
Langkah 1:

Proses pertama diawali dengan memberikan bobot 0 pada node awal atau titik awal dan bobot tak hingga pada node lainnya.



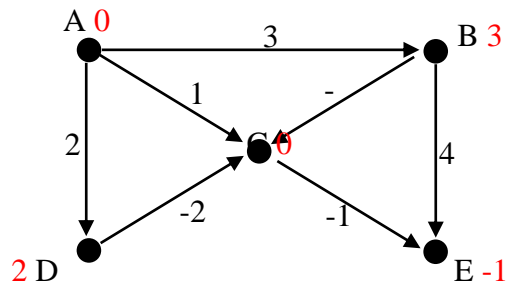
Langkah 2:

Lakukan relaxation pada tiap *edges* yang bersinggungan dengan urutan (AB), (AD), (AC), (BE), (BC), dan (CE). Selanjutnya kita akan mendapatkan jarak dari setiap *edges* setelah proses pertama selesai dilakukan.



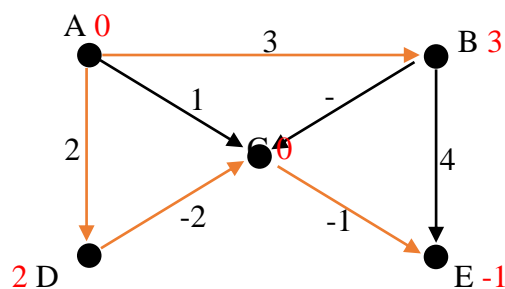
Langkah 3:

Iterasi pertama memberikan seluruh jarak terpendek diantara *edges* ketika 1 kali dilalui. Kita akan mendapatkan hasil selanjutnya ketika iterasi ke-2 sudah dilakukan



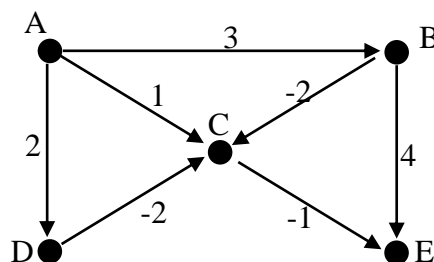
Langkah 4:

Iterasi kedua memberikan hasil ketika setiap *edges* sudah diproses sebanyak 2 kali. Tidak ada perubahan bobot yang terjadi pada iterasi kedua sehingga sudah dapat dihentikan.

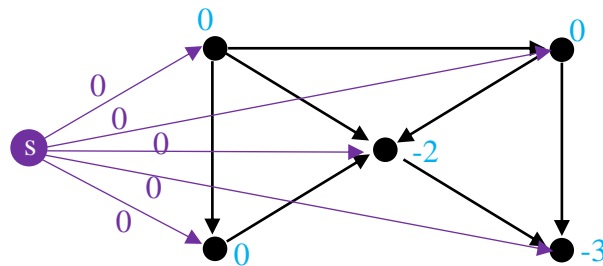


2.5. Algoritma Johnson

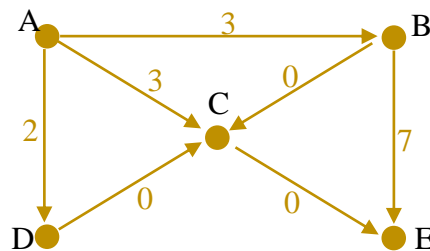
Dalam pencarian rute terpendek banyak algoritma yang mampu dipakai dalam memecahkan permasalahannya, satu di antara yang ada ialah Algoritma Johnson, algoritma ini merupakan penggabungan dari algoritma Bellman-Ford dan juga algoritma Dijkstra (Anintha & Ramesh Babu, 2018). Algoritma Johnson mendemonstrasikan keunggulan dengan memanfaatkan aspek penggabungan dari dua algoritma shortest path yang berbeda, yang menghasilkan efisiensi *runtime* yang lebih tinggi dalam pencarian all-pair shortest path (Auparay & Ijtihadie, 2018). Berikut adalah contoh penggunaan Algoritma Johnson:



1. Didapati sebuah graf dengan bobot $w(u \rightarrow v)$ dengan awal adalah A dan tujuan adalah E.



2. Tambahkan graf bantuan (*augmented graph*) dengan menambahkan node baru yang terhubung ke seluruh node utama dengan bobot nol. Lalu jalankan Algoritma Bellman-Ford untuk mencari bobot terpendek dari node s ke seluruh node utama. Dengan Algoritma Bellman-Ford pula kita bisa mendeteksi apakah ada *weight cycle* pada graf.



3. Setelah dilakukan perhitungan dengan menggunakan algoritma Bellman Ford selanjutnya modifikasi bobot menggunakan $w'(u \rightarrow v) = d_u + w(u \rightarrow v) - d_v$.
4. Karena pada langkah ketiga bobot sudah dimodifikasi sehingga tidak ada lagi bobot bernilai negatif, maka kita bisa menjalankan Algoritma Dijkstra untuk mencari rute terpendek dari seluruh yang ada.

2.6. Kompleksitas Algoritma

Kompleksitas waktu $T(n)$ akan mengukur berapa banyak waktu komputasi yang akan dilakukan oleh algoritma tersebut sebagai fungsi dari masukan n (Jonathan, 2022). Pada penelitian kali ini digunakan notasi Θ (*Big-Theta*) dikarenakan memiliki tingkat akurasi yang lebih baik dibandingkan dengan notasi O (*Big-Oh*) dan juga notasi Ω (*Big-Omega*), Θ (*Big-Theta*) digunakan untuk mengkategorikan sebuah algoritma menjadi sebuah fungsi, yang mana akan dapat diberikan batas atas

serta batas bawah dari fungsi tersebut ketika masukan dari fungsi tersebut bertambah (Candra et al., 2020). Sebuah fungsi $t(n)$ dapat dikatakan sebagai $\Theta(g(n))$, ditulis ulang sebagai $t(n) \in \Theta >(g(n))$, $t(n)$ dibatasi baik diatas maupun dibawah oleh kelipatan konstan dan $g(n)$ untuk keseluruhan n (Levitin, 2011).

2.7. Mapbox

Salah satu penyedia peta yang dapat digunakan menggunakan *API* adalah Mapbox. Untuk menggunakannya pengguna harus mendapatkan *API Key* yang bisa didapatkan dari *website* resmi mereka. Mapbox dapat digunakan pada aplikasi berbasis *desktop*, *web*, dan juga aplikasi *mobile*. Pengguna harus mendaftar agar bisa mendapatkan *API keys* dari Mapbox agar dapat menggunakan berbagai macam layanan yang disediakan oleh Mapbox.

2.8. Penelitian yang Relevan

Berikut adalah penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan, yakni:

1. Pada penelitian Yesti (2022) yang berjudul “*Implementation of Bellman-Ford Algorithm in determining the shortest path of the truck waste disposal*”, tidak optimalnya pengolahan sampah pada Kota Teluk Kuantan menjadi sumber permasalahan yang ada di dalam penelitian ini, oleh karena itu dibuatkan sistem yang dapat mencari rute perjalanan tercepat bagi truk sampah agar proses pengolahan sampah menjadi lebih cepat menggunakan algoritma Bellman Ford. Pemilihan rute yang akan dilalui oleh truk sampah dinilai dapat membantu aktivitas. Bellman-Ford digunakan untuk mendapatkan lintasan terdekat dari satu node ke semua titik node pada graf yang akan diproses.
2. Pada penelitian Farhan (2020) yang berjudul “*Implementasi Bellman-Ford dan Floyd-Warshall dalam menentukan jalur terpendek menuju Universitas Nasional berbasis android*” menunjukkan kebutuhan akan efisiensi terhadap pemilihan rute travel menuju Universitas Nasional. Bellman Ford dan Floyd Warshall dapat dipakai sebagai acuan untuk mendapatkan rute terbaik menuju Universitas Nasional. Hasil pengujian menunjukkan hasil yang sama hanya

saja terdapat perbedaan dalam waktu *running time* untuk mendapatkan perhitungan rute terpendek.

3. Pada penelitian Mubarak (2019) yang berjudul “Sistem informasi optimasi rute pengiriman barang menggunakan algoritma Bellman-Ford (studi kasus JNE cabang Jember)”, dalam penelitian ini adanya masalah efisiensi dalam pengiriman barang yang dilakukan oleh kurir menyebabkan beberapa konsumen mengeluh akan itu, salah satu masalah yang terjadi adalah Ketika kurir tidak efisien dalam hal pemilihan rute yang akan dilewati saat melakukan pengiriman barang. Algoritma Bellman Ford dinilai dapat mempercepat proses menentukan rute yang akan digunakan oleh kurir, yang bermanfaat dapat meningkatkan efisiensi dalam pengantaran barang yang akan dilakukan oleh kurir.
4. Pada penelitian Hendri (2019) yang berjudul “Analisis pencarian lintasan terpendek dengan menggunakan algoritma Bellman-Ford (Studi kasus: Pengantaran Paket Pos di Kecamatan Pontianak Kota)” dalam penelitian ini, didapati permasalahan jalur terpendek pada sebuah perusahaan yang dimiliki oleh negara yang dimana hal pengiriman barang yang dilakukan oleh kantor pos yang mengakibatkan tidak efisiennya dan terjadi keterlambatan waktu pengiriman. Algoritma Bellman Ford digunakan untuk mempercepat proses dalam penentuan rute terpendek agar dapat diperoleh jarak yang efisien.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

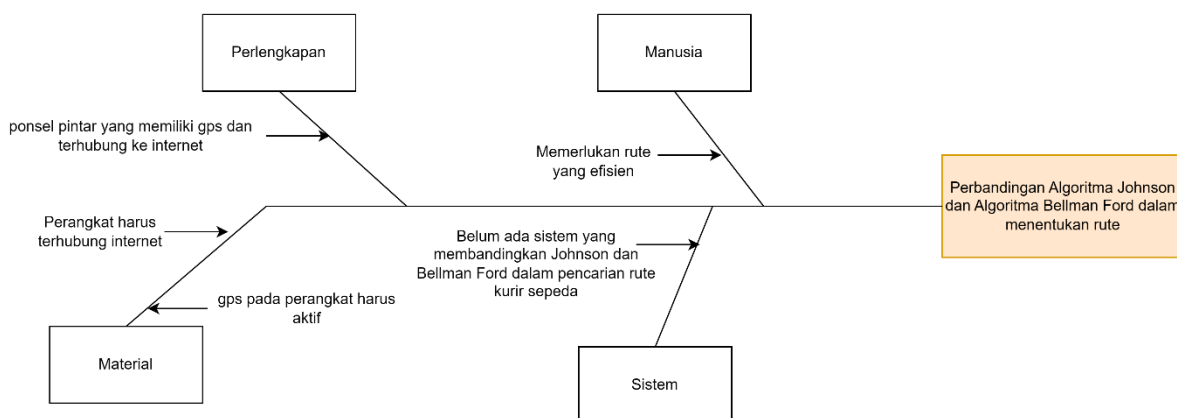
3.1. Analisis Sistem

Analisis sistem ialah fase di dalam penelitian yang akan menjabarkan komponen dan perkara yang dibutuhkan agar suatu sistem dapat bekerja. Di dalamnya akan dibagi menjadi 2 jenis yaitu analisis yang membahas permasalahan serta analisis yang membahas kebutuhan yang masing masingnya akan mengidentifikasi kebutuhan serta masalah yang ada pada sistem yang dibuat.

3.1.1. Analisis Masalah

Dalam sebuah penelitian, analisis masalah diperlukan agar dapat mengungkap pemicu dari permasalahan yang dianalisis lebih lanjut sehingga dapat menghasilkan sebuah sistem yang dapat bekerja dengan optimal. Dalam penelitian ini, permasalahan yang dianalisis adalah penentuan rute pengantaran paket yang akan dilakukan oleh kurir sepeda agar kurir dapat mengantarkan barang lebih cepat dan dekat. Analisis masalah ini akan menggunakan metode *Ishikawa diagram*.

Diagram Ishikawa merupakan diagram yang memiliki bentuk layaknya tulang yang ada pada ikan (*fishbone*), diagram ini akan menunjukkan penyebab dan akibat dari suatu permasalahan. Diagram ini terdiri dari kepala ikan yang berisikan judul dari permasalahan yang akan diurai dan tulang-tulang ikan yang menguraikan



Gambar 3.1 *Ishikawa Diagram*

penyebab dari permasalahan, yakni manusia, sistem, material dan metode. Berikut ini adalah diagram *Ishikawa* yang diuraikan melalui gambar 3.1.

Gambar di atas memperlihatkan diagram Ishikawa yang memuat akar penyebab dari permasalahan yang ingin diselesaikan dalam penelitian kali ini.

3.1.2. Analisis Kebutuhan

Kegunaan analisis ini dalam perancangan sebuah aplikasi ialah agar dapat mengidentifikasi data yang diolah sistem. Analisis kebutuhan melibatkan identifikasi mengenai fungsional dan non-fungsional yang digunakan dalam fase perancangan aplikasi agar dapat berjalan dan dapat memenuhi tujuan dari aplikasi yang dibuat.

3.1.2.1. Kebutuhan Fungsional

Kebutuhan fungsional adalah gabungan berbagai proses yang dijalankan oleh sistem yang dibangun. Berikut adalah kebutuhan fungsional yang ada pada sistem:

1. Sistem dapat membaca lokasi pengguna.
2. Sistem dapat mencari rute pengantaran terpendek menuju alamat pengantaran.
3. Sistem dapat menampilkan arah dari lokasi pengguna menuju alamat pengantaran barang.
4. Sistem dapat menampilkan *runtime* proses Algoritma Johnson dan Algoritma Bellman-Ford.

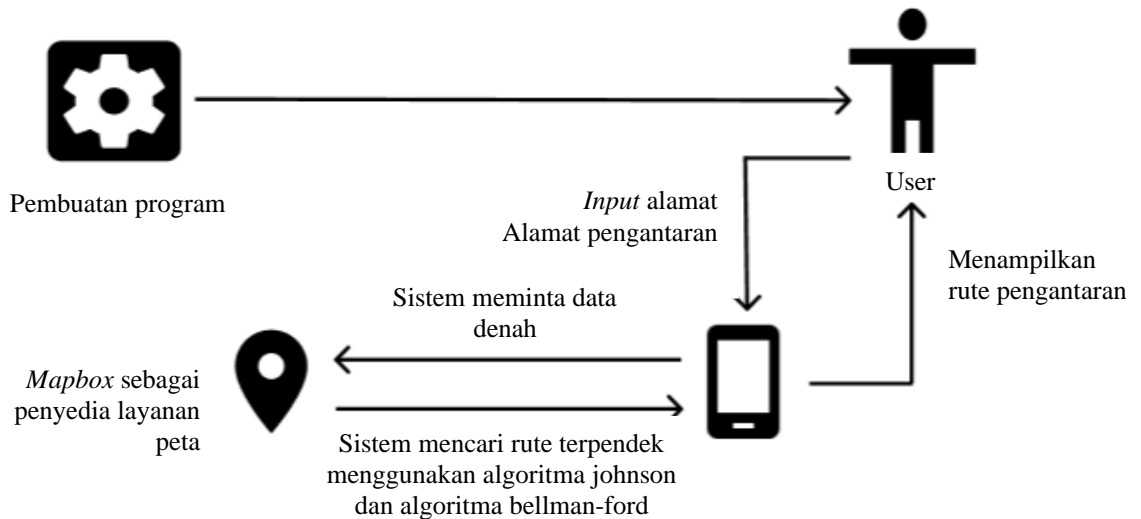
3.1.2.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional ialah hal yang membahas berupa fitur, karakteristik, makna atau kegunaan yang dapat digunakan oleh aplikasi seperti waktu, batasan pengembangan proses, dan standarisasi sistem yang digunakan sebagai pelengkap. Berikut adalah kebutuhan fungsional yang dibutuhkan dalam sistem ini:

1. Tampilan sistem mudah dipahami agar mudah dimengerti oleh pengguna.
2. Dapat memberikan informasi jika ada kesalahan yang dilakukan pengguna.
3. Tidak menggunakan alat tambahan agar mengurangi biaya.

3.1.3. Diagram Umum Sistem

Diagram umum sistem adalah representasi alur yang mengilustrasikan proses serta interaksi yang terjadi di dalam sistem. Perancangan keseluruhan aplikasi ini dijabarkan pada gambar di bawah:



Gambar 3.2 Diagram Umum Sistem

Berikut penjelasan alur proses diagram umum sistem pada gambar di atas:

1. Membuat program dengan mengimplemtasikan kedua algoritma Johnson dan algoritma Bellman-Ford.
2. Lokasi awal user akan dibaca oleh sistem
3. Data alamat pengiriman yang dimasukkan pengguna akan dijadikan parameter untuk mendapatkan rute tercepat dalam pengiriman barang yang dilakukan oleh *bicycle courier*.
4. Dengan algoritma Johnson dan Bellman Ford sistem akan meproses perhitungan guna mendapatkan rute yang efisien.
5. Setelah proses selesai dilakukan, sistem akan menampilkan rute pengantaran terpendek.

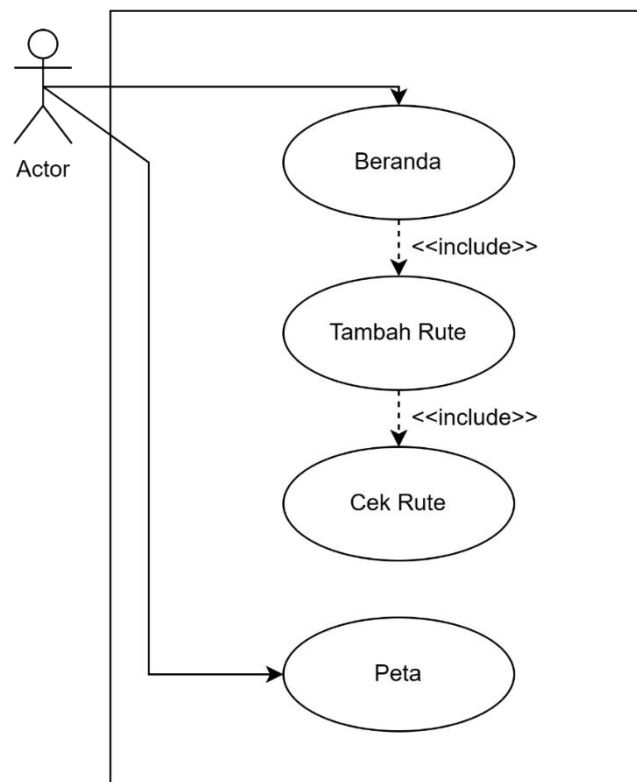
3.2. Pemodelan Sistem

Pemodelan sistem ialah perpanjangan tahapan interaksi antara pengguna dengan sistem yang dibangun agar dapat berfungsi secara maksimal. Pada riset kali ini

akan menggunakan *use case diagram*, *activity*, *sequence*, serta diagram ulir sebagai bentuk pemodelan.

3.2.1. Use Case Diagram

Pemodelan korelasi diantara aktor dengan sistem yang sedang dibangun merupakan pengertian dari *Use case*. Pembuatan diagram ini dapat membantu dalam memahami fungsionalitas sistem dan hubungan antar pelaku dan kasus pengguna. Berikut merupakan contoh diagram kasus pengguna pada sistem yang sedang dibuat.



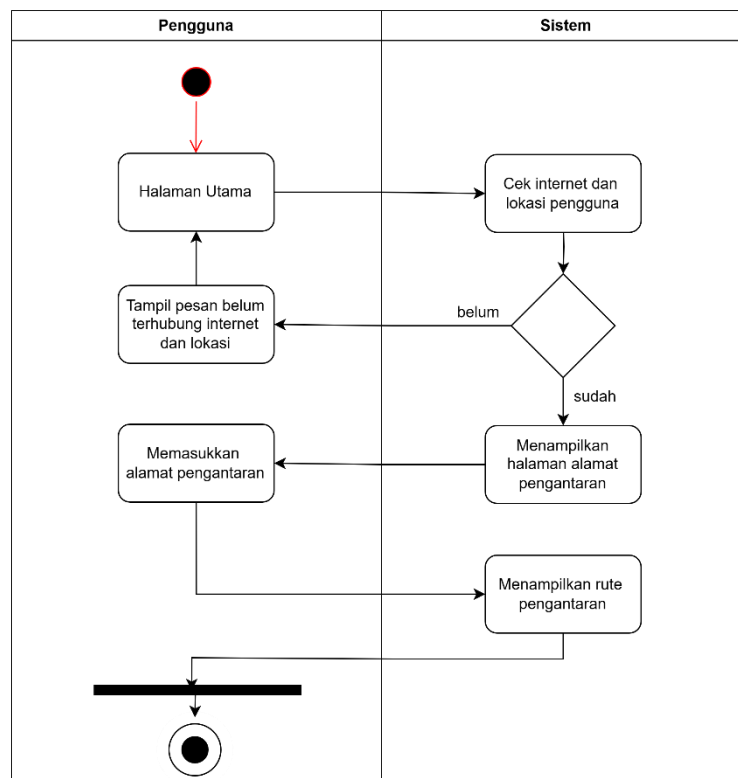
Gambar 3.3 Use Case Diagram

Gambar di atas menunjukkan interaksi dapat dilakukan oleh actor dengan sistem yang dibangun. Pengguna dapat menggunakan fitur untuk melakukan pencarian rute serta dapat menggunakan sistem untuk mengecek lokasi pada peta yang disediakan oleh sistem yang dibuat. Pengguna akan diberikan rute yang terdekat yang dapat digunakan pada pengantaran barang.

3.2.2. Activity Diagram

Activity Diagram adalah penggambaran aliran proses yang terjadi sedari awal sampai akhir. *Activity Diagram* juga dapat digunakan untuk menjelaskan potongan komponen dari *use case diagram*.

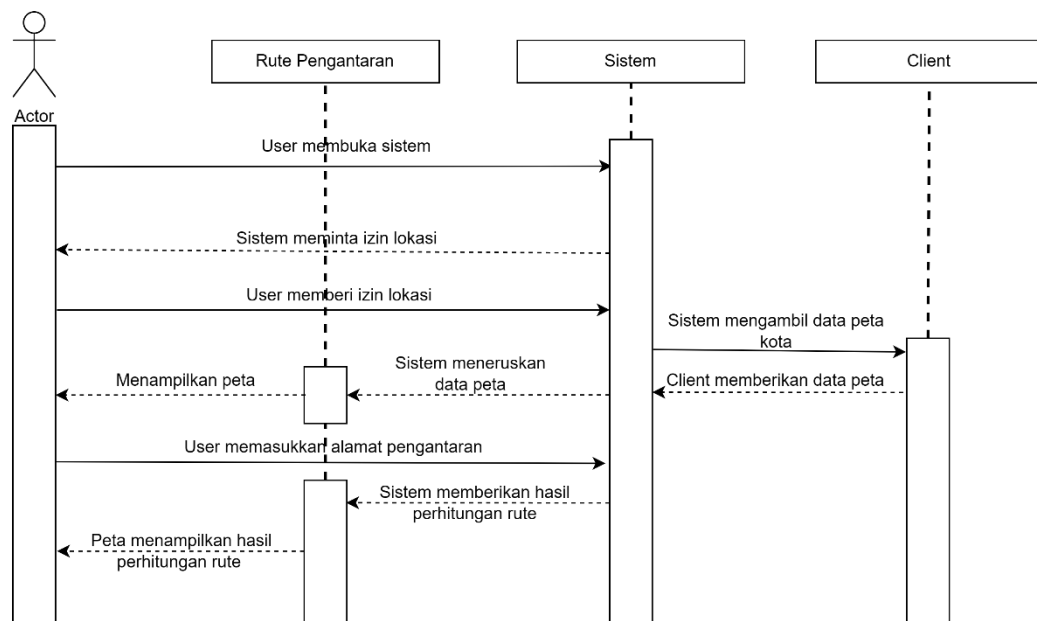
3.2.2.1. Activity Diagram Pencarian Rute Pengantaran



Gambar 3.4 *Activity Diagram* User Pencarian Lokasi

3.2.3. Sequence Diagram

Sequence diagram merupakan tipe pemodelan yang menjabarkan korelasi diantara objek berdasarkan susunan waktu pada sebuah aplikasi sistem. Penggunaan tanda panah menunjukkan interaksi antar objek dan tanda garis putus-putus menunjukkan respon atau jawaban dari *input* yang diberikan pengguna.



Gambar 3.5 *Sequence Diagram*

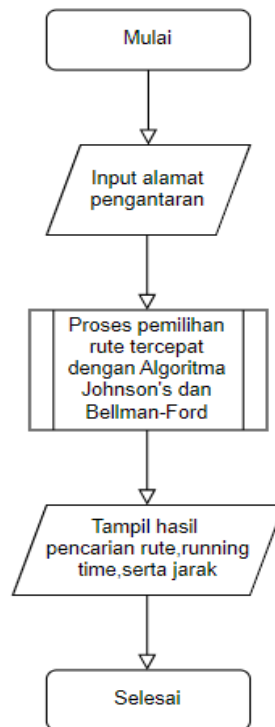
Diagram diatas memiliki 4 komponen yang akan berinteraksi, dimulai dari pengguna yang membuka sistem sampai dengan pengguna mendapatkan rute pengantaran.

3.3. *Flowchart*

Diagram alir representasi grafis maupun simbolik yang menggambarkan rentetan proses dengan bentuk berbagai simbol serta tanda panah yang menggambarkan hubungan antar proses di dalam suatu sistem.

3.3.1. Flowchart Sistem

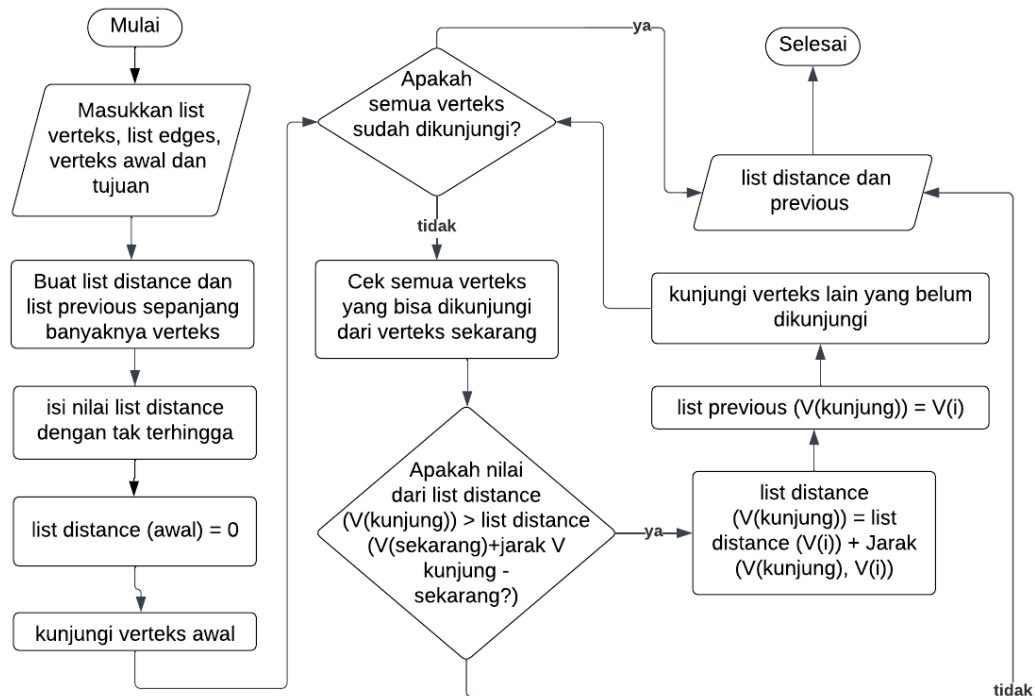
Flowchart sistem akan mengilustrasikan arah kerja pada suatu sistem. Di bawah ini adalah pemodelan daripada diagram alir untuk sistem yang akan dibuat.



Gambar 3.6 *Flowchart* Sistem

Gambar di atas menjelaskan bagaimana tata cara kerja sistem yang akan dibuat. Setelah pengguna membuka aplikasi, pengguna akan memasukkan alamat pengantaran, setelah itu sistem akan mencari rute tercepat menggunakan algoritma bellman-ford dan algoritma johnson. Lalu setelah proses pencarian selesai akan muncul *running time*, rute pengantaran, dan jarak menuju lokasi.

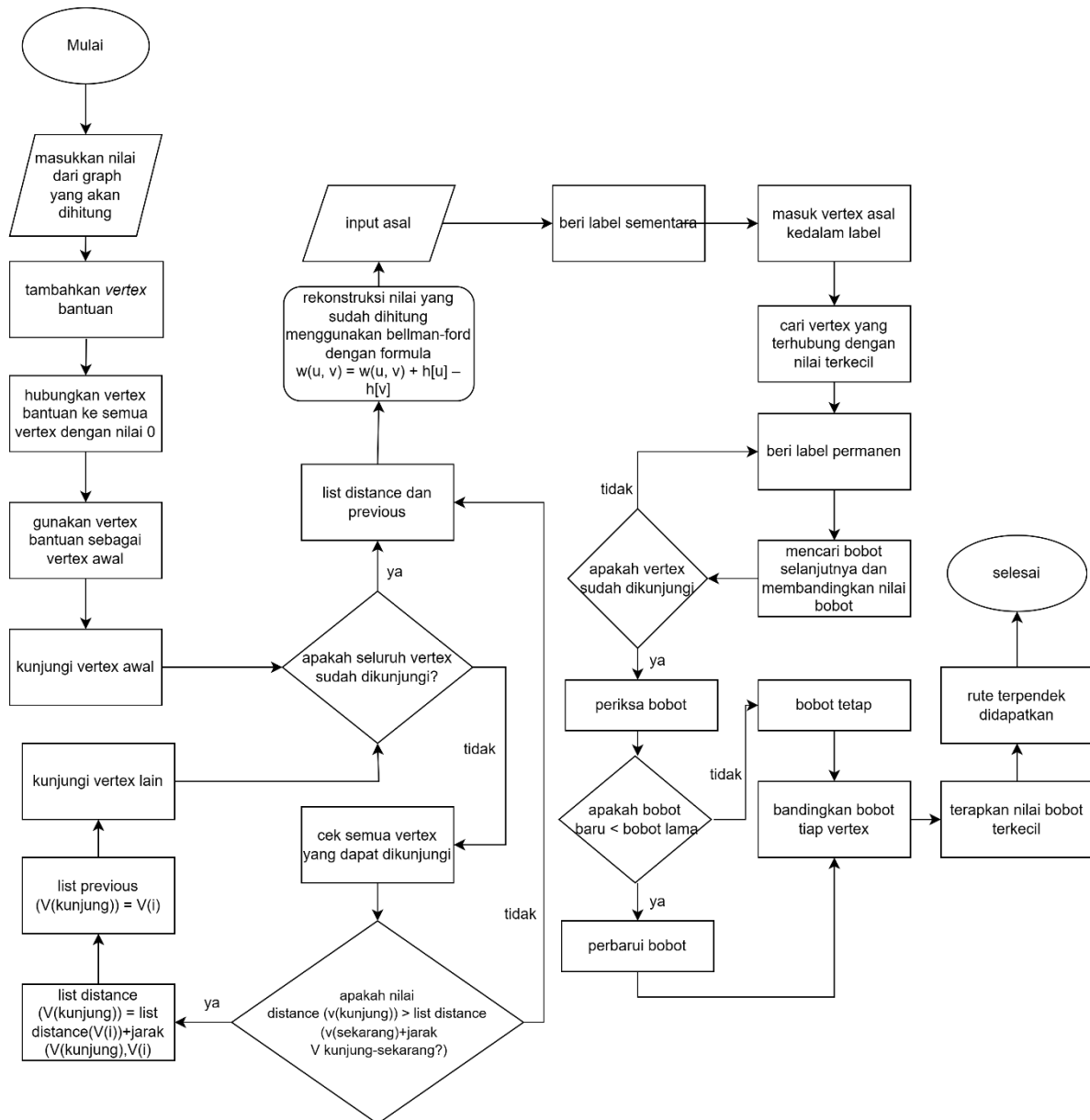
3.3.2. Flowchart Algoritma Bellman-Ford



Gambar 3.7 Flowchart Algoritma Bellman-Ford

Gambar di atas adalah gambaran dari alur kerja Algoritma Bellman-Ford. Alur diatas diawali dengan memasukkan vertex, *edges*, titik awal dan titik akhir. Lalu dilakukan iterasi proses pengecekan maksimal sebanyak jumlah *node* dikurang satu. Iterasi dilakukan untuk menentukan lajur alternatif sehingga didapatkan nilai paling efisien. Proses iterasi akan berhenti bila nilai akhir sama dengan iterasi sebelumnya.

3.3.3. Flowchart Algoritma Johnson



Gambar 3.8 Flowchart Algoritma Johnson

Gambar di atas merupakan representasi dari alur kerja Algoritma Johnson. Langkah pertama diawali dengan pembuatan *vertex* baru sebagai *vertex* pembantu dengan menghubungkannya ke seluruh *vertex* utama. Lalu dijalankan Algoritma Bellman-Ford untuk mendapatkan rute terdekat dari *vertex* pembantu ke seluruh *vertex* yang ada. Selanjutnya modifikasi bobot tiap *edge* dengan rumus $w(u,v) = w(u,v) + h[u]$

– $h[v]$ selanjutnya *node* bantuan tadi akan dihapus. Sekarang graf memiliki *edge* yang bobotnya sudah dimodifikasi sehingga tidak ada lagi *edge* berbobot negatif, tahapan terakhir adalah menggunakan Dijkstra untuk mendapatkan nilai bobot terpendek ke setiap *vertex*.

3.4. Perancangan *Interface*

Tahapan dalam pembuatan desain dari tampilan tatap muka daripada suatu sistem dapat dikatakan sebagai tahap perancangan *interface*, Perancangan *Interface* diperlukan agar pada saat pembuatan sistem memiliki patokan pembuatan serta desain yang akan dibuat.

3.4.1. Halaman Awal

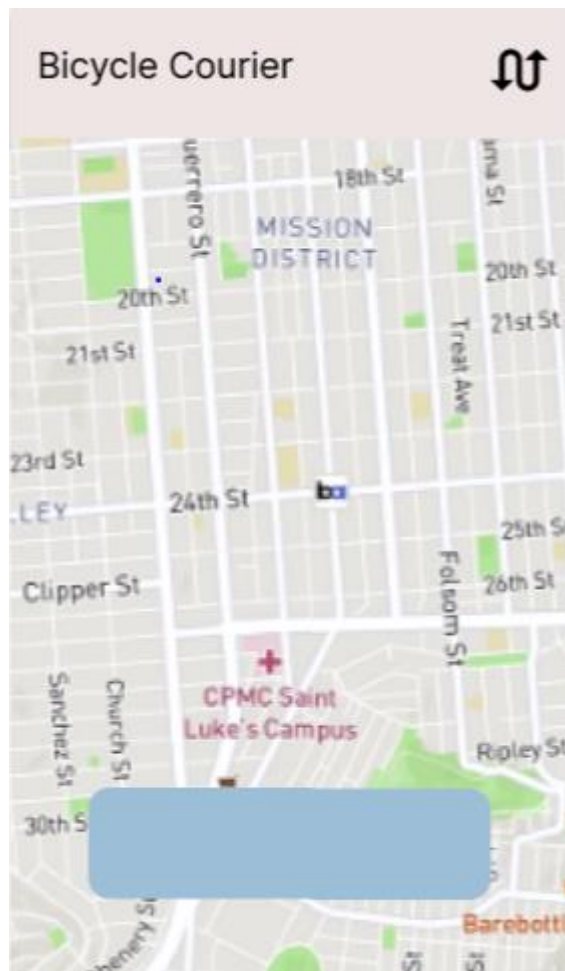


Gambar 3.9 Halaman Beranda

Halaman ini merupakan tampilan awal ketika aplikasi digunakan, pengguna dapat memilih akan menuju ke halaman peta atau akan memasukkan alamat pengantaran. Berikut keterangan komponen yang terdapat di gambar diatas:

1. Tombol cek peta yang akan mengarahkan pengguna ke halaman pengecekan peta dan rute pengantaran
2. Tombol alamat, fungsi tombol ini akan mengarahkan pengguna ke halaman untuk memasukkan alamat pengantaran barang.

3.4.2. Halaman Cek Peta



Gambar 3.10 Halaman Cek Peta

Halaman pada gambar di atas merupakan halaman peta, yang nantinya akan menampilkan rute pengantaran. Berikut keterangan dari gambar di atas:

1. Peta yang akan menampilkan lokasi alamat pengantaran dan rutenya.
2. Slider yang akan menunjukkan titik pengantaran.
3. Tombol yang akan menampilkan waktu running time.

3.4.3. Halaman masukkan alamat pengantaran



Gambar 3.11 Halaman masukkan Alamat Pengantaran

Gambar di atas merupakan desain daripada halaman *input* alamat pengantaran.

1. *Search bar* yang akan mencari alamat pengantaran.
2. *List* yang akan menampilkan alamat yang sudah dimasukkan.
3. Tombol *explore* yang akan mengarahkan ke halaman peta.

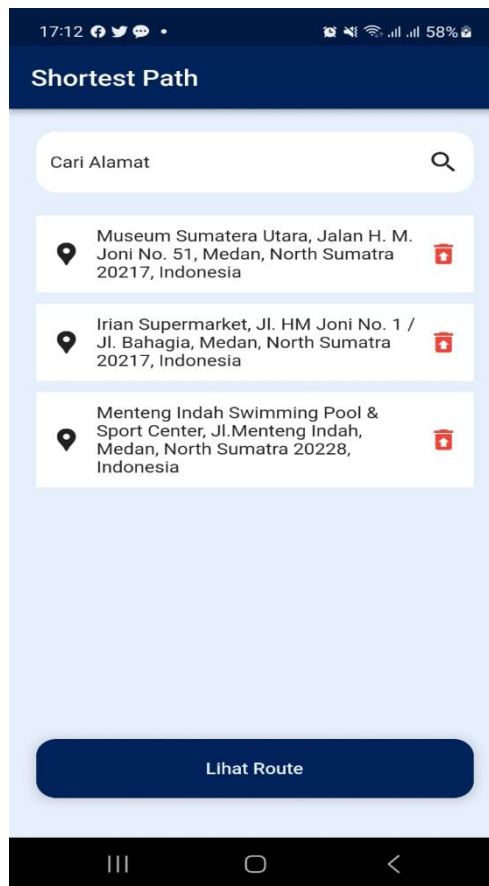
BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi

Penelitian kali ini, sistem yang dibangun dengan mengimplementasikan algoritma Bellman Ford dan juga Johnson didalamnya. Sistem nantinya akan dibangun menggunakan Bahasa Dart serta mengusung *framework* Flutter. Untuk menampilkan peta, sistem akan menggunakan jasa pihak ketiga menggunakan Mapbox.

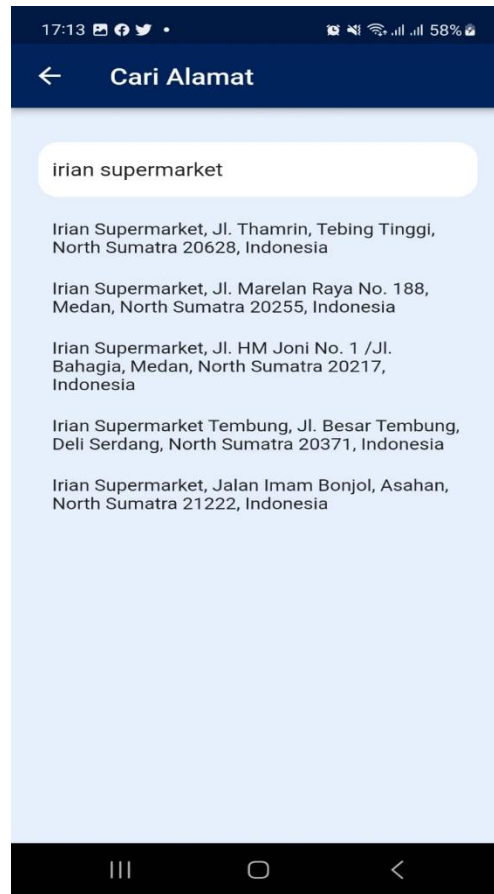
4.1.1. Halaman *List* Alamat



Gambar 4.1 Halaman *List* Alamat

Tampilan *list* alamat diatas akan menjadi tempat dimana alamat ditampilkan dalam bentuk list sesuai dengan yang dimasukkan oleh pengguna pada halaman pencarian alamat.

4.1.2. Halaman Pencarian Alamat



Gambar 4.2 Pencarian alamat

Tampilan diatas merupakan halaman tempat user dapat memasukkan alamat pengantaran, alamat pengantaran yang ada disesuaikan dengan data yang dimiliki oleh Mapbox.

4.1.3. Halaman Tampilan Peta



Gambar 4.3 Halaman Tampilan Peta

Halaman diatas menampilkan titik alamat pengantaran yang sudah dipilih oleh pengguna pada halaman pencarian alamat, pada halaman ini alamat ditampilkan dalam titik *point* pada *map* yang disediakan.

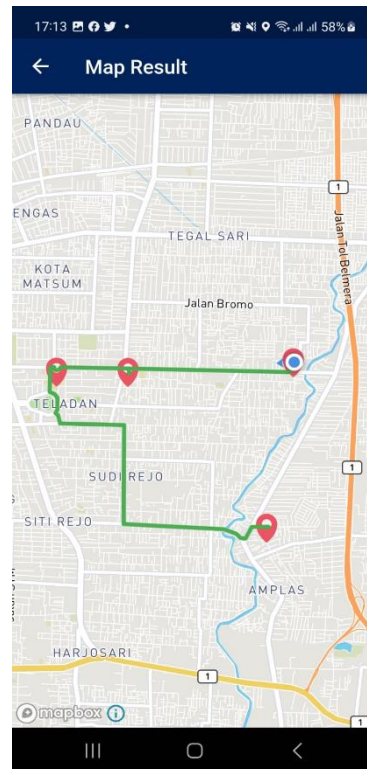
4.1.4. Halaman Perhitungan Algoritma



Gambar 4.4 Halaman Perhitungan Algoritma

Tampilan di atas akan menampilkan node yang dimasukkan serta menampilkan kecepatan perhitungan algoritma dalam bentuk *millisecond* (ms). Ada dua algoritma yang akan dihitung yaitu Algoritma Bellman-Ford dan juga Algoritma Johnson.

4.1.5. Halaman Tampilan Rute Pengantaran



Gambar 4.5 Halaman Tampilan Rute Pengantaran

Pada halaman ini akan ditampilkan rute pengantaran yang dapat dilalui oleh pengguna, rute ditampilkan menggunakan *polyline* yang disediakan Mapbox, pemilihan rute menggunakan hasil perhitungan dari Algoritma sebelumnya.

4.2. Pengujian Sistem

Periode pengetesan dimana ketika sistem telah selesai dikerjakan dan sudah dapat diimplentasikan, selanjutnya akan dilakukannya pengujian terhadap sistem. Sistem yang telah dibangun akan diuji guna mengetahui bahwa aplikasi yang telah dibangun bisa berkerja sesuai harapan. Pada fase pengujian, lokasi pengantaran yang digunakan berdasarkan titik pengantaran yang biasa dilewati oleh kurir sepeda. Aplikasi yang dibangun akan menjalankan dua algoritma yaitu Algoritma Bellman-Ford dan juga Algoritma Johnson yang bertujuan agar mendapatkan

perbandingan proses daripada kedua algoritma yang digunakan. Tabel di bawah adalah daftar lokasi yang digunakan pada tahap pengujian ini.

Tabel 4.1 Tabel Daftar Lokasi Pengujian

No	Lokasi	Longitude	Latitude
1	Menteng Indah	98.724924	3.555071
2	Irian Supermarket	98.70313	3.568458
3	Museum Sumatera Utara	98.69634400000001	3.5683235
4	Lokasi Awal	98.724396	3.561335

Pengujian dilakukan menggunakan perangkat android Samsung S21 Fe dengan spesifikasi sebagai berikut:

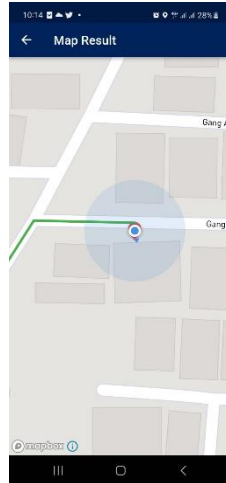
1. Android 13
2. Exynos 2100 8 Core
3. Ram 8 GB

Untuk mendapatkan hasil dibutuhkan beberapa proses agar algoritma dapat bekerja yakni:

1. Pengguna memasukkan alamat antar pada halaman yang sudah disediakan.
2. Aplikasi akan mengambil data alamat serta posisi pengguna yang akan dijadikan acuan dalam pembuatan rute.
3. Data alamat kemudian akan diubah dalam bentuk *longitude* dan *latitude*.
4. Mapbox sebagai penyedia layanan map akan memberikan data yang berisi bobot dalam bentuk jarak, serta titik yang akan dilewati.
5. Aplikasi akan menggunakan *steps* yang dihasilkan oleh Mapbox. *Steps* akan dirubah menjadi grap yang akan dihitung oleh Algoritma Bellman-Ford dan Algoritma Johnson untuk mendapatkan perbandingannya.

4.2.1. Pengujian Implementasi Algoritma Bellman-Ford

Pengujian pada kedua algoritma akan dilakukan dengan titik awal pada longitude 98.724396 dan latitude 3.561335



Gambar 4.6 Titik Awal Pengujian Algoritma

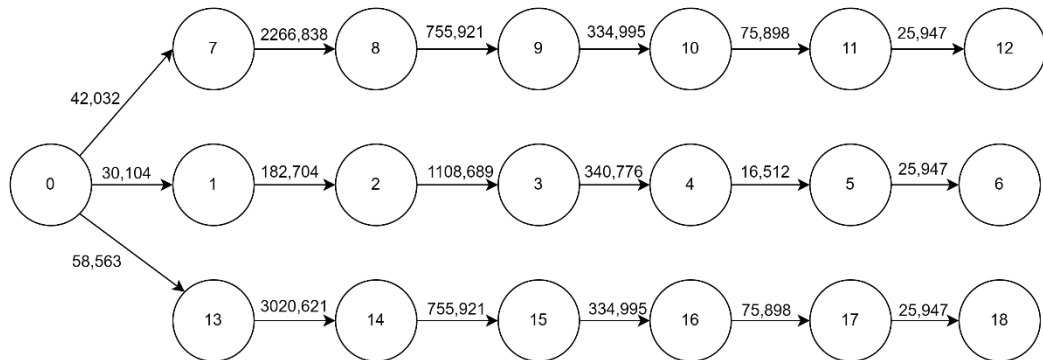
Hasil pengujian mengungkapkan, didapatinya rute pengantaran melalui Menteng Indah, Irian Supermarket, dan Museum Sumatera Utara.

4.2.2. Perhitungan Manual Algoritma Bellman-Ford

Pada perhitungan diatas, titik awal yang digunakan adalah. Terdapat beberapa alamat pengantaran yang akan diantarakan. Berikut adalah langkah-langkah dalam pencarian rute.

Langkah 1:

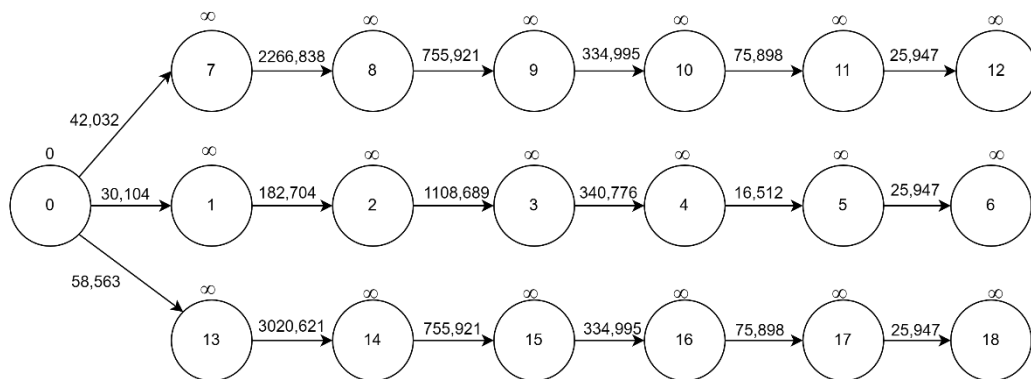
Inisiasi grap awal yang akan dihitung



Gambar 4.7 Graf Permasalahan Bellman-Ford

Langkah 2:

Memulai perhitungan dengan mengimplementasikan algoritma, dimana diletakkannya nilai 0 pada node awal atau titik awal dan ∞ pada node lainnya.

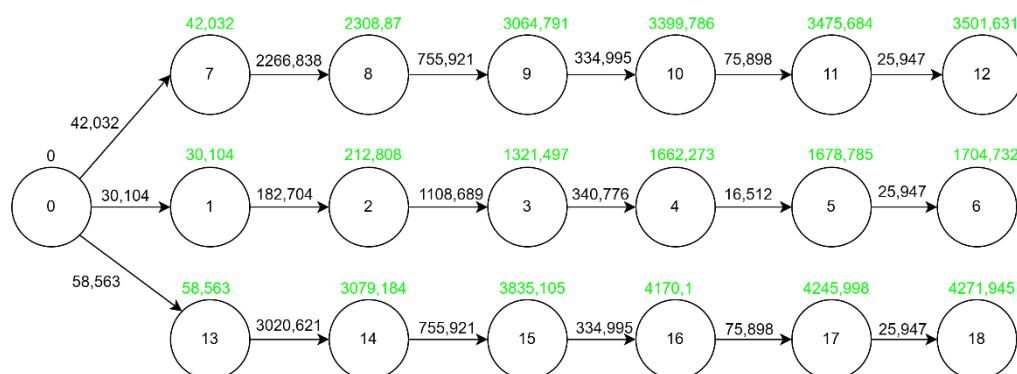


Gambar 4.8 Langkah Kedua Perhitungan Bellman Ford

Tabel 4.4 Iterasi Pertama Dan Iterasi Kedua

Iterasi Pertama						
0	1	2	3	4	5	6
0	30,104	212,808	1321,497	1662,273	1678,785	1704,732
0	7	8	9	10	11	12
0	42,032	2308,87	3064,791	3399,786	3475,684	3501,631
0	13	14	15	16	17	18
0	58,563	3079,184	3835,105	4170,1	4245,998	4271,945
Iterasi Kedua						
0	1	2	3	4	5	6
0	30,104	212,808	1321,497	1662,273	1678,785	1704,732
0	7	8	9	10	11	12
0	42,032	2308,87	3064,791	3399,786	3475,684	3501,631
0	13	14	15	16	17	18
0	58,563	3079,184	3835,105	4170,1	4245,998	4271,945

Dikarenakan iterasi kedua dan iterasi pertama memiliki nilai bobot yang sama maka iterasi dihentikan pada iterasi kedua. Hasil yang didapatkan akan berbentuk sebagai berikut.

**Gambar 4.9** Bobot Nilai Akhir Setelah Perhitungan

4.2.3. Pengujian Implementasi Algoritma Johnson

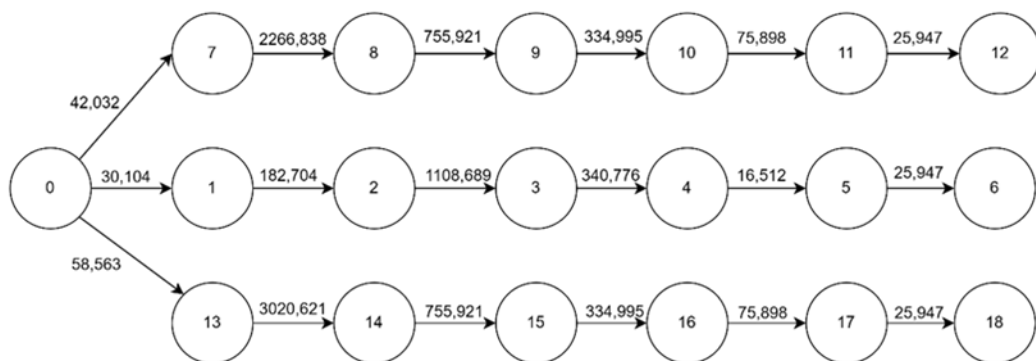
Pengujian algoritma Johnson dilakukan dengan lokasi titik awal yang sama dengan pengujian yang dilakukan dengan Algoritma Bellman-Ford, ini dilakukan agar kedua algoritma dijalankan menggunakan dua grap yang sama.

Dari hasil perhitungan Algoritma Johnson, didapatkan rute pengantaran melalui Menteng Indah, Irian, lalu selanjutnya menuju Museum Sumatera Utara.

4.2.4. Perhitungan Manual Algoritma Johnson

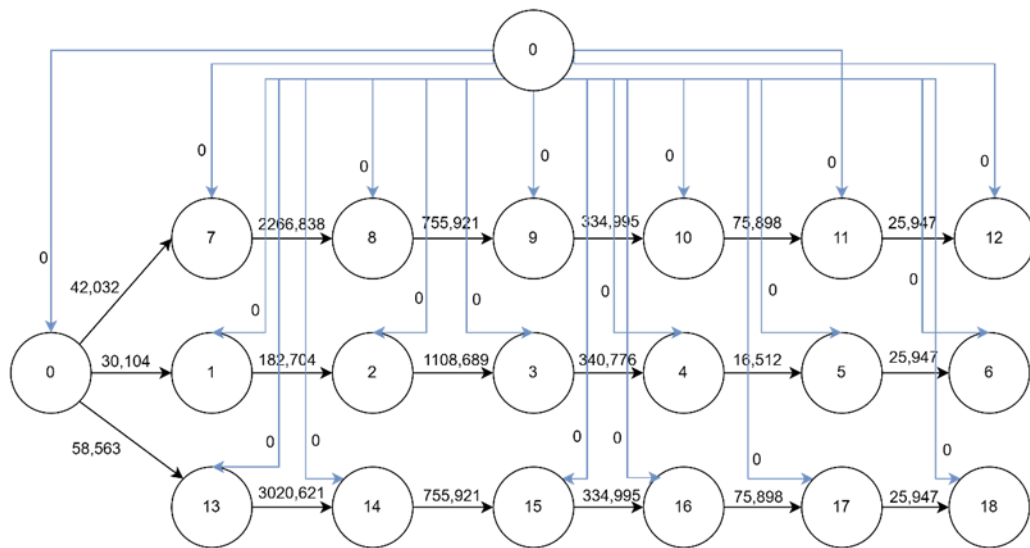
Pada perhitungan diatas posisi titik awal adalah . Lokasi titik awal yang digunakan sama dengan Algoritma Bellman-Ford yang sebelumnya sudah dicoba. Penggunaan titik awal yang sama dilakukan agar kedua algoritma bekerja dengan titik yang sama. Berikut merupakan langkah dalam perhitungan Algoritma Johnson :

Langkah 1



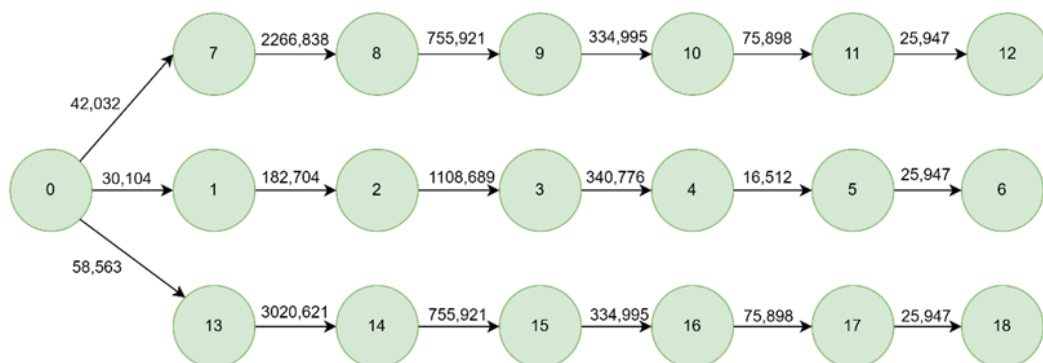
Gambar 4.10 Graf Permasalahan Johnson

Langkah 2



Gambar 4.11 Langkah Kedua Algoritma Johnson

Setelah ditambahkannya graf bantuan, selanjutnya akan dilakukan perhitungan menggunakan Algoritma Bellman-Ford, setelah perhitungan Algoritma Bellman-Ford selesai, nantinya akan didapatkan hasil perhitungan, bobot yang didapatkan selanjutnya akan dimodifikasi menggunakan rumus $w'(u \rightarrow v) = d_u + w(u \rightarrow v) - d_v$ yang bertujuan untuk menghilangkan bobot negatif yang ada pada graf.



Gambar 4.12 Hasil Bellman Ford Pada Algoritma Johnson

Langkah 3

Setelah dilakukannya perhitungan menggunakan Algoritma Bellman Ford, seluruh graf sudah tidak memiliki nilai bobot yang bernilai negative, Ketika graf sudah tidak memiliki bobot bernilai negatif akan dilakukan perhitungan menggunakan Algoritma Dijkstra. Algoritma Dijkstra akan mencari nilai terkecil antar *node* tetangga pada graf.

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
14	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3079,184	3835,105	∞	∞	∞

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
10	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	3399,786	3475,684	∞	∞	∞	∞	∞	∞	∞

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
11	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3475,684	3501,631	∞	∞	∞	∞	∞	∞

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
12	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3501,631	∞	∞	∞	∞	∞	∞

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
15	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3835,105	4170,1	∞	∞

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
16	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	4170,1	4245,998	∞

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
17	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	4245,998	4271,945

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
18	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	4271,945

Dari tabel diatas didapatkan hasil perhitungan akhir sebagai berikut.

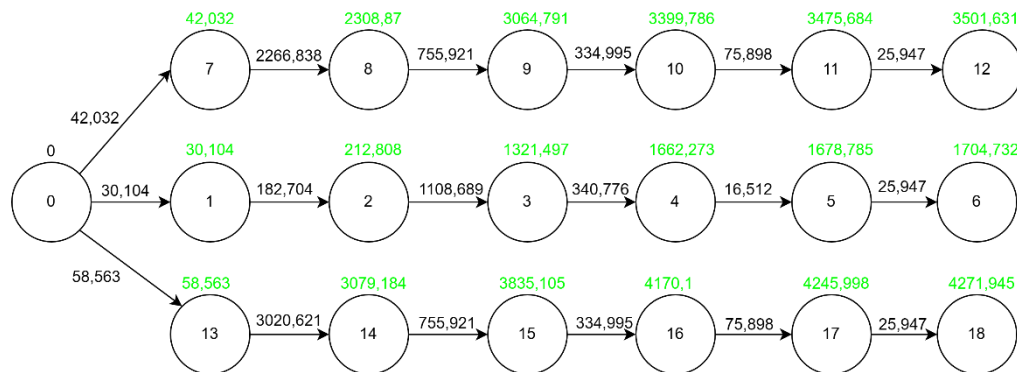
Tabel 4.6 Hasil Perhitungan Johnson

Dari 0 menuju 1 berjarak 30,104
Dari 0 menuju 7 berjarak 42,032
Dari 0 menuju 13 berjarak 58,563
Dari 1 menuju 2 berjarak 212,808
Dari 2 menuju 3 berjarak 1321,497
Dari 3 menuju 4 berjarak 1662,273

Dari 4 menuju 5 berjarak 1678,785
Dari 5 menuju 6 berjarak 1704,732
Dari 7 menuju 8 berjarak 2308,87
Dari 8 menuju 8 berjarak 3064,791
Dari 13 menuju 14 berjarak 3079,184
Dari 9 menuju 10 berjarak 3399,786
Dari 10 menuju 11 berjarak 3475,634
Dari 11 menuju 12 berjarak 3501,631
Dari 14 menuju 15 berjarak 3835,105
Dari 15 menuju 16 berjarak 4170,1
Dari 16 menuju 17 berjarak 4245,998
Dari 17 menuju 18 berjarak 4271,945

Node 0 : 0
Node 1 : 30,104
Node 2 : 212,808
Node 3 : 1321,497
Node 4 : 1662,273
Node 5 : 1678,785
Node 6 : 1704,732
Node 7 : 42,032
Node 8 : 2308,87
Node 9 : 3064,791
Node 10 : 3399,786
Node 11 : 3475,684
Node 12 : 3501,631
Node 13 : 58,563
Node 14 : 3079,184
Node 15 : 3835,105
Node 16 : 4170,1
Node 17: 4245,998

Node 18 : 4271,945



Gambar 4.13 Hasil Perhitungan Johnson

4.3. Hasil Pengujian

Setelah melakukan pengujian dilakukan, hasil pengujian daripada dua algoritma akan dibandingkan. *Running time* masing masing algoritma dari titik awal dan tujuan yang sama akan dibandingkan. Pada tahapan ini akan dilakukan perbandingan menggunakan beberapa lokasi tujuan yang akan dimasukkan secara acak. Hasil daripada pengujian akan ditampilkan pada table dibawah.

Tabel 4.7 Lokasi Pengujian

No	Lokasi pengantaran	<i>Running time</i> (detik)	
		Bellman Ford	Johnson
1	98.717041,3.552365 (Menteng indah), 98.70313,3.568458 (Irian Supermarket), 98.69634400000001,3.5683235 (Museum Sumut)	0,9195	1,4815

Tabel 4.8 Lokasi Pengujian

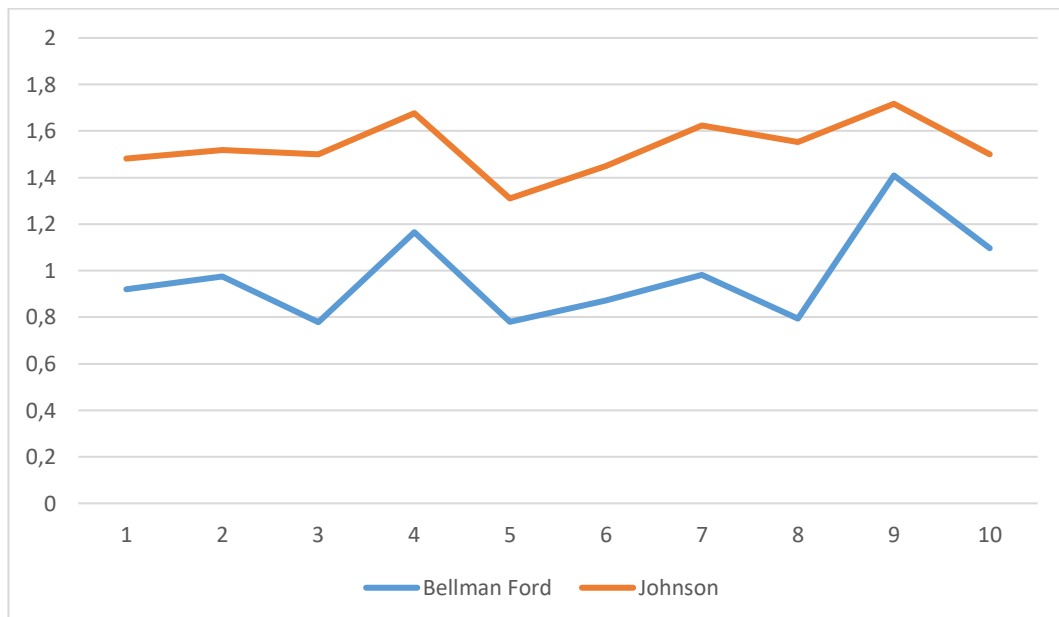
No	Lokasi Pengantaran	<i>Running time</i> (detik)	
		Bellman Ford	Johnson
2	98.69564219379949,3.56488916 41072207 (Stadion Teladan), 98.68471790914016,3.57521344 413353 (Istana Maimun) 98.6785350036021,3.591863048 955103 (Merdeka Walk)	0,9748	1,5187
3	3.5763160169790034, 98.6764288170965 (Taman Ahmad Yani), 3.581376288509491, 98.66930858641527 (Sma Negeri 1 Medan), 3.553329925621439, 98.66241106872093 (Polonia Medan)	0,7784	1,4996
4	3.561932991092327, 98.65759756751268 (Universitas Sumatera Utara), 3.5690019803497237, 98.64709449379954 (Total Futsal), 3.5678199964063677, 98.65738826496437(RS USU)	1,1650	1,6763

Tabel 4.9 Lokasi Pengujian

No	Lokasi Pengantaran	<i>Running time</i> (detik)	
		Bellman Ford	Johnson
5	3.5942727041121776, 98.67488019379967(Deli Park Medan), 3.59174868085892, 98.68105597845906 (Centre Point), 3.5795022647745705, 98.67212976311835 (GKN)	0,7803	1,3102
6	3.5792617280282646, 98.66746913612928 (Rahmat Galery), 3.596436113943953, 98.6758968782779 (JW Marriot), 3.592129088624682, 98.66347142263486 (Carefour Gatsu)	0,8725	1,449
7	3.5751729043549356, 98.68733467845895 (Mesjid Raya Medan), 3.574542164547754, 98.66960769379959 (Rumah Gubernur) 3.5906366942784738, 98.65478846496454 (Berastagi Supermarket)	0,982	1,6233
8	98.698457,3.545403 (Auto 2000), 98.6983865,3.5652530000000000 2 (SMA N 5 Medan), 98.702772,3.586220 (Sukaramai)	0,794	1,552

Tabel 4.10 Lokasi Pengujian

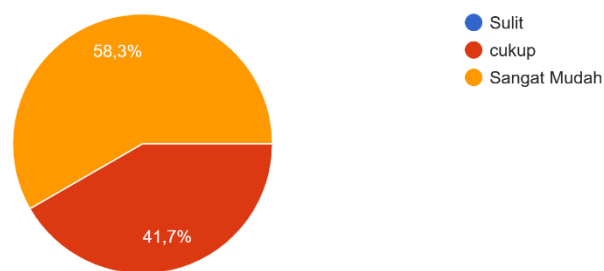
No	Lokasi pengantaran	Running time (detik)	
		Bellman Ford	Johnson
9	98.672403,3.572003 (Pardede hotel), 98.67642599999999,3.590091(Grand Aston City Hall), 98.667406,3.58473(Cambridge hotel)	1,409	1,717
10	98.674718,3.581261 (Tiara Convention) 98.686518,3.59906 (Kantor Indosat Sumut) 98.688276,3.597951 (RSUP Dr. Pirngadi)	1,096	1,500
Rata-Rata		0,9771	1,3846

**Gambar 4.14** Chart Perbandingan Running Time Algoritma

4.3.1 Pengujian User

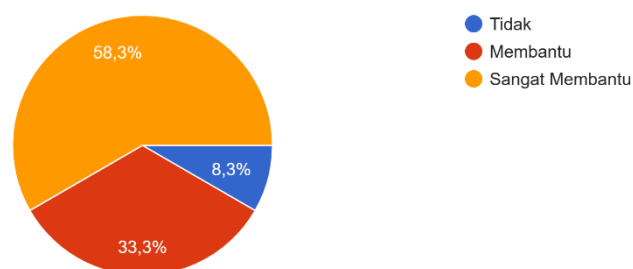
Berikut adalah hasil pengujian sistem dengan melakukan survei dan pengisian kuesioner pada beberapa kurir sepeda.

Apakah aplikasi mudah digunakan
12 jawaban



Gambar 4.15 Aplikasi mudah digunakan

Apakah aplikasi membantu mencari rute
12 jawaban



Gambar 4.16 Aplikasi membantu kurir

4.4. Kompleksitas Algoritma

Kompleksitas algoritma adalah perkiraan waktu yang dibutuhkan algoritma untuk bisa menyelesaikan proses kalkulasi. Pada bagian ini kompleksitas algoritma berfokus pada kompleksitas waktu. Memperhitungkan kompleksitas waktu dari algoritma penting dilakukan agar bisa mengetahui tingkat efisiensi dari sebuah

4.4.1. Kompleksitas Algoritma Bellman Ford

Kompleksitas waktu dari Algoritma Bellman Ford akan ditampilkan pada tabel dibawah.

Tabel 4.11 Kompleksitas Algoritma Bellman Ford

Map<String, double> distances = {};	$\Theta(1)$
Map<String, String> predecessors = {};	$\Theta(1)$
currentGraph.forEach((key, value) { distances[key] = double.infinity;	$\Theta(V)$
distances[start] = 0;	$\Theta(1)$
currentGraph.forEach((u, edges) { edges.forEach((v, weight) {	$\Theta(V * E)$
currentGraph.forEach((u, edges) { edges.forEach((v, weight) {	$\Theta(V * E)$
$\Theta(1 + 1 + V + 1 + V * E + V * E = \Theta(V * E)$	$\Theta(V * E)$

Dari perhitungan diatas didapatkan kompleksitas sebesar $\Theta(V * E)$ dimana V sebagai *Vertex* dan E sebagai *Edge*, yang dinandakan waktu yang diperlukan akan berkembang mengikuti besaran V dan E yang akan diproses.

4.4.2. Kompleksitas Algoritma Johnson

Pada tabel dibawah akan ditampilkan kompleksitas dari Johnson

Tabel 4.12 Kompleksitas Algoritma Johnson

<code>Map<Vertex, Map<Vertex, double>> distance = {};</code>	$\Theta(V)$
<code>Map<Vertex, double> bellDist = bellmanFord(g, g.getVertex('q'));</code>	$\Theta(V)$
<code>g.addVertex('q');</code>	$\Theta(1)$
<code>g.vertices.forEach((_, v) => distance[v] = double.infinity);</code>	$\Theta(V)$
<code>g.vertices.forEach((_, v) { for (Vertex n in v.getNeighbours()) { double w = v.getWeight(n).toDouble(); v.setWeight(n, w + bellDist[v]! - bellDist[n]!); }); });</code>	$\Theta(V * E)$
<code>g.vertices.remove('q');</code>	$\Theta(1)$
<code>Map<Vertex, Map<Vertex, double>> distance = {};</code>	$\Theta(V)$
<code>g.vertices.forEach((_, v) { distance[v] = dijkstra(g, v); });</code>	$\Theta(V)$

Tabel 4.13 Kompleksitas Algoritma Johnson

<pre> g.vertices.forEach((_ , v) { g.vertices.forEach((_ , w) { distance[v]![w] = distance[v]![w]! + (bellDist[w]! - bellDist[v]!); }); }); </pre>	$\Theta(V^2)$
<pre> g.vertices.forEach((_ , v) { for (Vertex n in v.getNeighbours()) { double w = v.getWeight(n); v.setWeight(n, w + bellDist[n]! - bellDist[v]!); }); </pre>	$\Theta(V * E)$
$\Theta(V + V + 1 + V + V * E + V * E + 1 + V + V + V^2 + V * E) = \Theta(V^2 + V * E)$	$\Theta(V^2) + (V * E)$

Berdasarkan perhitungan proses yang dilalui Algoritma Johnson diatas didapatkan hasil $\Theta(V^2 + V * E)$, berdasarkan kompleksitasnya Algoritma Bellman Ford seharusnya dapat memproses dengan lebih efisien dan lebih baik dibandingkan dengan Algoritma Johnson pada kasus ini karena memiliki kompleksitas yang lebih baik.

BAB V

PENUTUP

5.1. KESIMPULAN

Setelah melakukan beberapa tahapan pengujian meliputi analisa, perancangan implementasi, serta pengujian didapatkan kesimpulan mengenai algoritma Bellman Ford serta algoritma Johnson dalam mencari rute pengantaran bagi kurir sepeda sebagai berikut:

1. Algoritma yang digunakan yakni Algoritma Bellman Ford dan juga Algoritma Johnson dapat digunakan dalam menentukan rute pengantaran paket oleh kurir sepeda.
2. Algoritma Bellman Ford memiliki kompleksitas yang lebih baik yaitu $\Theta(V * E)$ dibandingkan Algoritma Johnson $\Theta(V^2) + (V * E)$, sehingga mendapatkan angka *running time* lebih rendah.
3. Jumlah *node* yang menjadi parameter masukan bagi algoritma mempengaruhi *running time* masing-masing algoritma.
4. Algoritma Johnson bersifat “*find all shortest path*” yang tetap meneruskan proses walaupun tujuan sudah didapatkan. Sementara Algoritma Bellman Ford akan berhenti ketika iterasi yang dilakukan dengan iterasi sebelumnya memiliki nilai yang sama.

5.2. SARAN

Berikut adalah saran yang dapat dipertimbangkan untuk penelitian selanjutnya.

1. Membuat fitur menambah lokasi pengantaran langsung pada peta.
2. Menambahkan fitur rute alternatif.
3. Sistem dapat membuat nilai daripada graf bersifat dinamis.

DAFTAR PUSTAKA

- Alamsyah. (2010). *PEMANFAATAN METODE HEURISTIK DALAM PENCARIAN MINIMUM SPANNING TREE DENGAN ALGORITMA SEMUT*.
- Anintha, S., & Ramesh Babu, M. (n.d.). *Proceedings of the 2018 4th International Conference on Electrical Energy Systems (ICEES): 7-9 February 2018, Department of Electrical and Electronics Engineering, SSN College of Engineering, Chennai*.
- Auparay, E., & Ijtihadie, R. M. (2018). REPLIKASI DATA MENGGUNAKAN DETECTION CONTROLLER MODULE UNTUK MENCEGAH CONGESTION DI DATA CENTER. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 16(1), 10. <https://doi.org/10.12962/j24068535.v16i1.a590>
- Bawole, D., & Chernovita, H. (2019). Algoritma Bellman-Ford untuk Menentukan Jalur Terpendek dalam Survey Klaim Asuransi (Studi Kasus : PT. Asuransi Sinar Mas, Jakarta). *INOBIIS: Jurnal Inovasi Bisnis Dan Manajemen Indonesia*, 3, 41–51. <https://doi.org/10.31842/jurnal-inobis.v3i1.119>
- Buhaerah, Busrah, Z., & Sanjaya, H. (2022). *TEORI GRAF DAN APLIKASINYA*.
- Candra, A., Budiman, M. A., & Hartanto, K. (2020). *Dijkstra's and A-Star in Finding the Shortest Path: a Tutorial*.
- Farhan, S., Andryana, S., & Hayati, N. (2020). *IMPLEMENTASI BELLMAN-FORD DAN FLOYD-WARSHALL DALAM MENENTUKAN JALUR TERPENDEK MENUJU UNIVERSITAS NASIONAL BERBASIS ANDROID*.
- Fincham, B. (2008). Balance is everything: Bicycle messengers, work and leisure. In *Sociology* (Vol. 42, Issue 4, pp. 618–634). <https://doi.org/10.1177/0038038508091619>
- Fitro, A., Saeful Bachri, O., Ilham, A., Purnomo, S., & Frendianata, I. (2018). Article ID: IJMET_09_02_078 Cite this Article: Achmad Fitro, Otong Saeful Bachri, Arif Ilham Sulistio Purnomo and Indra Frendianata, Shortest Path

Finding in Geographical Information Systems using Node Combination and Dijkstra Algorithm. *International Journal of Mechanical Engineering and Technology* (IJMET, 9(2), 755–760. <http://www.iaeme.com/IJMET/index.asp755http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=9&IType=2http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=9&IType=2http://www.iaeme.com/IJMET/index.asp756>

Hendri, Kiftiah, M., & Fran, F. (2019). ANALISIS PENCARIAN LINTASAN TERPENDEK DENGAN MENGGUNAKAN ALGORITMA BELLMAN-FORD (Studi Kasus: Pengantaran Paket Pos di Kecamatan Pontianak Kota). In *Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster)* (Vol. 08, Issue 3).

Jonathan, M. K. (2022). *Penarapan Teori Graf dan Kompleksitas Algoritma dalam Penyelesaian Rubik's Cube menggunakan strategi Meet in the Middle*.

Levitin, A. (2011). *Introduction to The Design & Analysis of Algorithms*.

Maes, J., & Vanelslender, T. (2012). The Use of Bicycle Messengers in the Logistics Chain, Concepts Further Revised. *Procedia - Social and Behavioral Sciences*, 39, 409–423. <https://doi.org/10.1016/j.sbspro.2012.03.118>

Mubarak, R. (2019). *Sistem Informasi Optimasi Rute Pengiriman Barang menggunakan Algoritma Bellman Ford*. <http://repository.unej.ac.id/handle/123456789/96449>

Mustofa, D., Arifudin, D., Setyanto, A., Intan, D., Saputra, S., & Wirasto, A. (2022). The Bellman-Ford Algorithm Search for the Best Routes to Tourist Attractions amid the COVID-19 Pandemic. In *Webology* (Vol. 19, Issue 2). <http://www.webology.orghttp://www.webology.org>

Purwananto, Y., Purwitasari, D., & Wahyu Wibowo, A. (2005). *IMPLEMENTASI DAN ANALISIS ALGORITMA PENCARIAN RUTE TERPENDEK DI KOTA SURABAYA* (Vol. 10, Issue 2).

Wang, X. Z. (2018). The Comparison of Three Algorithms in Shortest Path Issue. *Journal of Physics: Conference Series*, 1087(2). <https://doi.org/10.1088/1742-6596/1087/2/022011>

Wisudawan, W. F. (2008). *Kompleksitas Algoritma Sorting yang Populer Dipakai*. <http://www.nist.gov/dads/HTML/algorithm.html>

Yesti, S. A. (2022). IMPLEMENTASI ALGORITMA BELLMAN-FORD DALAM MENENTUKAN LINTASAN TERPENDEK TRUK PEMBUANGAN SAMPAH IMPLEMENTASI ALGORITMA BELLMAN-FORD DALAM MENENTUKAN LINTASAN TERPENDEK TRUK PEMBUANGAN SAMPAH. *Journal*.