

Pemodelan Jalur Terbaik Transportasi Umum dan Antarmoda di Kota Bandung dengan Metode Graf

Meliana Sitanggang^{#1}, Yoyok Gamaliel^{*2}, Herry Imanta Sitepu^{#3}

[#]Program Studi Sistem Komputer, Institut Teknologi Harapan Bangsa
Jl. Dipatiukur no 80-84, Bandung, Jawa Barat, Indonesia

¹ me.melianas@gmail.com

² yoyok@ithb.ac.id

³ herry@ithb.ac.id

Abstract— Public transportation is one of the main choices for people especially those who do not have private vehicles. However, one of the problems in using public transportation is the lack of information related to public transportation in the city of Bandung. The information such as the best route and travelling time, the shortest path and the shortest time, suitable transportation, and the transportation halts are rarely available. It makes public transportation users need more time in waiting, choosing the transportation, or travelling. This problem needs to be handled properly. Therefore, information system is required to provide the solution. In this research, Graf method is used to find the shortest path information, the shortest travelling time and mileage. Dijkstra and Floyd Warshall algorithms are applied. Throughput and processing time are measured to examine both algorithms. Moreover, in this research Depth First Search (DFS) is used to discover the alternative path in the same route. The implementation can be used for public transportation of Trans Metro Bandung (TMB) bus and city transportation (Angkutan Kota) in Bandung.

Keywords— Public Transportation, Graf Method, Shortes Path, Floyd Warshall, Dijkstra, Depth First Search

Abstrak— Transportasi umum merupakan salah satu pilihan bagi masyarakat yang tidak memiliki kendaraan pribadi. Tetapi masyarakat sering menghadapi permasalahan dalam menggunakan transportasi umum, seperti informasi jalur terbaik, waktu tempuh terpendek, pilihan moda dan lokasi perpindahan moda transportasi. Hal tersebut membuat pengguna transportasi umum membutuhkan lebih banyak waktu dalam menunggu, menentukan transportasi yang digunakan, dan menempuh perjalanan ke tujuan. Untuk menjawab permasalahan tersebut, penelitian ini merancang sistem informasi dengan metode Graf yang dapat memberikan informasi jalur terpendek berdasarkan jarak dan waktu tempuh, moda yang dapat digunakan, dan lokasi perpindahan moda transportasi tersebut. Penelitian ini menerapkan metode jarak terpendek dengan menggunakan algoritme Dijkstra dan Floyd Warshall. *Throughput* dan waktu proses akan diukur untuk meneliti kinerja kedua algoritme tersebut. Penelitian ini juga melakukan pengujian pencarian jalur alternatif yang dapat ditempuh untuk rute yang sama dengan menggunakan algoritme Depth First Search (DFS). Implementasi perancangan ini dapat digunakan untuk transportasi umum berupa bus Trans Metro Bandung (TMB) dan angkutan kota di Bandung.

Kata Kunci— Transportasi Umum, Metode Graf, Jarak Terpendek, Floyd Warshall, Dijkstra, Depth First Search

I. PENDAHULUAN

Informasi mengenai transportasi umum di Kota Bandung masih sangat kurang. Untuk saat ini informasi tersebut masih tersedia secara terpisah, oleh karena itu kondisi ini dirasakan kurang membantu calon pengguna transportasi umum. Informasi tersebut mencakup jarak tempuh, lokasi dimana harus naik/turun transportasi umum, dan lokasi perpindahan moda. Kurangnya informasi ini membuat jarak dan waktu yang ditempuh penggunaannya menjadi lebih panjang dan lebih lama.

Untuk memperoleh informasi jarak dan waktu tempuh transportasi umum masyarakat biasanya menggunakan aplikasi berbasis GPS seperti Google Maps dan Waze. Aplikasi tersebut dapat memberikan informasi jarak dan waktu tempuh untuk setiap moda transportasi berdasarkan lokasi awal dan lokasi tujuan yang dimasukkan oleh pengguna. Tetapi aplikasi tersebut belum dapat menampilkan informasi mengenai titik perpindahan moda transportasi di titik-titik tertentu di Kota Bandung.

Masalah kurangnya informasi mengenai transportasi umum ini perlu ditangani dengan baik. Penanganan masalah tersebut diharapkan dapat membantu masyarakat dalam memperoleh informasi yang lengkap sehingga jarak dan waktu yang ditempuh dapat lebih efisien. Informasi tersebut diharapkan dapat meningkatkan minat masyarakat dalam menggunakan transportasi umum di Kota Bandung. Oleh karena itu diperlukan sebuah aplikasi yang tidak hanya dapat memberikan informasi tentang jarak terpendek dan moda transportasi yang dapat digunakan oleh penggunaannya, tetapi juga dapat memberikan informasi tentang lokasi perpindahan moda beserta pilihan transportasinya. Jarak terpendek di sini merupakan besarnya jarak yang ditempuh dan besarnya waktu tempuh dari satu titik ke titik lain.

Beberapa algoritme yang telah dikembangkan untuk menyelesaikan permasalahan jalur terpendek diantaranya Dijkstra dan Floyd Warshall. Algoritme yang paling efisien untuk permasalahan jalur terpendek dalam jaringan data ialah algoritme Dijkstra. Namun pada jaringan dinamis yang sangat besar, algoritme Dijkstra menjadi tidak efisien dibandingkan dengan Floyd Warshall. Hal tersebut dikarenakan *node-node* pada jaringan akan dikunjungi kembali sehingga banyak perhitungan yang diulang [3]. Penelitian ini menggunakan

kedua algoritme tersebut, Dijkstra dan Floyd Warshall, dalam pencarian jalur terpendek.

Hasil dari penelitian ini adalah sebuah aplikasi yang menggunakan metode Graf dalam merepresentasikan lokasi dan jalur. Algoritme dalam metode Graf yang akan digunakan dalam pencarian jalur terpendek ialah Dijkstra dan Floyd Warshall, dan untuk mencari jalur-jalur alternatif menggunakan algoritme DFS. Aplikasi dapat memberikan informasi jalur terpendek, moda transportasi yang dapat digunakan, lokasi perpindahan moda transportasi, dan jalur-jalur alternatif yang dapat digunakan. Penelitian ini juga akan membandingkan kinerja dari Dijkstra dan Floyd Warshall dalam mencari jalur terpendek pada sistem. Aplikasi diterapkan pada aplikasi berbasis *web service*.

Manfaat dari aplikasi ini adalah masyarakat dapat menggunakan aplikasi untuk memperoleh informasi mengenai transportasi umum di Kota Bandung secara cepat dan akurat. Informasi tersebut berupa jalur dengan jarak tempuh terpendek, jalur dengan waktu tercepat, jalur-jalur alternatif yang dapat ditempuh, dan lokasi perpindahan moda transportasi. Informasi tersebut dapat membantu masyarakat dalam memilih transportasi umum yang akan digunakan.

II. DASAR TEORI

Secara umum, pencarian jalur terpendek dapat dibagi menjadi dua metode, yaitu metode konvensional dan metode heuristik. Metode konvensional cenderung lebih mudah dipahami daripada metode heuristik, tetapi hasil yang diperoleh dari metode heuristik lebih variatif. Metode Graf merupakan metode konvensional dengan algoritme yang menggunakan perhitungan matematis. Metode heuristik merupakan bagian dari kecerdasan buatan untuk pencarian dan penentuan jalur terpendek [3]. Penelitian menggunakan metode Graf dengan algoritme Dijkstra dan Floyd Warshall untuk pencarian jalur terpendek. Jalur terpendek ialah jalur dengan jarak tempuh terpendek maupun jalur dengan waktu tempuh tercepat.

A. Transportasi Antarmoda

Dalam sistem transportasi nasional (Sistranas) terdapat dua istilah, yaitu antarmoda dan multi moda. Transportasi antarmoda merupakan transportasi yang menggunakan lebih dari satu moda transportasi dalam satu perjalanan yang berkesinambungan. Transportasi multi moda merupakan transportasi dengan menggunakan paling sedikit dua moda transportasi yang berbeda [2].

Penelitian ini menggunakan sistem transportasi antarmoda. Transportasi yang digunakan dalam satu perjalanan tidak hanya satu, tetapi bisa lebih dari satu transportasi. Penelitian ini membuat sistem yang dapat menampilkan perpindahan transportasi jika perjalanan yang ditempuh dari lokasi asal ke lokasi tujuan memerlukan lebih dari satu transportasi. Sistem menampilkan lokasi di mana perpindahan transportasi dan transportasi apa yang harus digunakan untuk perjalanan selanjutnya sehingga dapat mencapai lokasi tujuan yang diinginkan.

B. Pemodelan Graf

Suatu Graf terdiri dari lingkaran dan garis. Lingkaran disebut "simpul" (*vertex* atau *node*) yang terhubung oleh garis atau disebut "sisi" (*edge*). Setiap simpul biasanya diberi label dengan cara tertentu, misal dengan huruf-huruf alfabet. Setiap sisi dibatasi oleh dua simpul di ujungnya [7].

Graf yang digunakan dalam penelitian ini adalah Graf berarah dan Graf berbobot. Gambar 1.a menunjukkan model Graf berarah. Graf disebut Graf berarah jika setiap *node* dihubungkan dengan garis yang memiliki arah (anak panah). Pada penelitian ini garis yang memiliki arah pada dua titik *node* merepresentasikan jalur yang dapat dilalui dari satu *node* ke *node* lain serta transportasi yang dilaluinya.

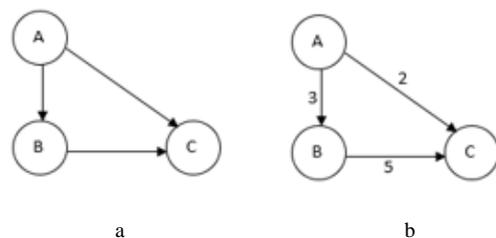
Gambar 1.b menunjukkan model Graf berbobot. Bobot dapat melambangkan beberapa konsep berbeda. Pada penelitian ini bobot merepresentasikan jarak tempuh dan waktu tempuh dari lokasi asal ke lokasi tujuan. Algoritme Dijkstra

Algoritme Dijkstra [1] merupakan varian dari algoritme *greedy*, yaitu algoritme yang melakukan pemecahan persoalan yang terkait dengan masalah optimasi [4]. Algoritme *greedy* membentuk solusi langkah per langkah. Pada setiap langkah solusi terdapat lebih dari satu pilihan yang perlu dieksplorasi sehingga dalam menentukan pilihan di setiap langkah harus dibuat keputusan terbaik.

C. Algoritme Floyd Warshall

Penelitian ini menggunakan algoritme Dijkstra sebagai salah satu alternatif untuk mencari jalur terpendek. Algoritme ini akan mencari panjang lintasan terpendek dari *node* asal ke *node* tujuan dalam sebuah Graf berarah dan berbobot.

Algoritme Floyd Warshall merupakan algoritme sederhana untuk menghitung jarak terpendek dari antara semua *vertex* pada sebuah *edge* dalam Graf berbobot dan berarah [6]. Algoritme Floyd Warshall merupakan varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya. Solusi yang diperoleh tidak hanya satu tetapi ada kemungkinan lebih dari satu [4]. Selain algoritme Dijkstra (algoritme *greedy*), penelitian ini juga menggunakan pemrograman dinamis yaitu algoritme Floyd Warshall sebagai salah satu alternatif untuk mencari jalur terpendek. Algoritme Floyd Warshall memulai iterasi dari titik awalnya kemudian memperpanjang lintasan dengan mengunjungi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang seminimum mungkin.



Gambar 1 (a) Graf berarah, (b) Graf berarah dan berbobot

D. Algoritme Depth First Search

Algoritme Depth-First Search (DFS) merupakan salah satu algoritme yang digunakan untuk mencari jalur dari *node* yang terhubung dalam suatu Graf. DFS menggunakan *stack* untuk mengingat dan menentukan *node* mana yang harus dituju ketika telah mencapai *node* terdalam/terjauh [5].

Penelitian ini akan menggunakan algoritme DFS untuk mencari setiap jalur yang dapat dilalui dari lokasi asal ke lokasi tujuan. Algoritme DFS akan menelusuri struktur Graf berdasarkan kedalaman *node* pada Graf tersebut.

Graf ditelusuri dari *node* paling kiri hingga level *node* terdalamnya. Ketika sudah mencapai *node* level terdalam maka akan kembali lagi ke satu level sebelumnya untuk menelusuri level terdalam dari *node* tersebut, begitu seterusnya hingga seluruh *node* terlewati.

E. Metode Euclidean

Metode Euclidean digunakan pada penelitian ini untuk menghitung waktu tempuh dimana waktu tempuh tersebut terdiri dari variabel waktu dan kemacetan. Waktu tempuh yang diperoleh dari perhitungan metode Euclidean inilah yang akan menjadi nilai yang akan dihitung pada algoritme Dijkstra dan Floyd Warshall untuk mencari jalur dengan waktu tempuh terpendek. Rumus perhitungan metode Euclidean adalah sebagai berikut.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

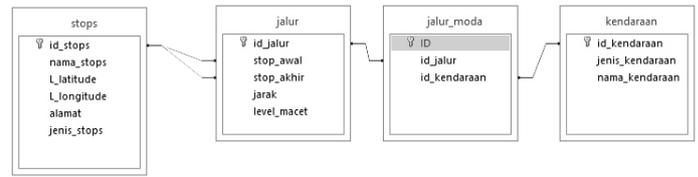
Dimana d_{ij} adalah jarak yang akan dihitung. Pada penelitian ini nilai d_{ij} menunjukkan waktu tempuh yang akan dihitung. Variabel x ialah koordinat x , yang merupakan waktu tempuh. Waktu tempuh diperoleh dari jarak dibagi kecepatan. Variabel y ialah koordinat y , yang merupakan nilai dari variabel kemacetan. Nilai variabel kemacetan terdiri dari level 0 sampai 10, dimana level 0 artinya tidak ada kemacetan dan 10 artinya level kemacetan tertinggi. Variabel i ialah lokasi asal dan j ialah lokasi tujuan.

Nilai waktu tempuh diperoleh dari perhitungan jarak dibagi kecepatan. Nilai kecepatan (v) diperoleh dari perbandingan kecepatan dengan level kemacetan. Nilai maksimum kecepatan ialah 60 km/jam [8]. Nilai kemacetan akan berubah sesuai dengan level kemacetan. Semakin kecil level kemacetan semakin tinggi nilai kecepatan. Perhitungan nilai kecepatan yang dipengaruhi oleh level kemacetan ialah sebagai berikut.

$$v = 60 - (level \times 6) \quad (2)$$

F. Perancangan Basis Data

Gambar 2 merupakan desain basis data yang digunakan pada aplikasi pencarian jalur terpendek. Terdapat 4 tabel yaitu stops, jalur, kendaraan, dan jalur_moda. Tabel stops merupakan tabel yang berisi informasi mengenai titik-titik lokasi tempat naik/turunnya transportasi umum. Tabel kendaraan merupakan tabel yang berisi informasi mengenai transportasi yang tersedia. Tabel jalur merupakan tabel yang berisi informasi mengenai setiap jalur, di mana setiap jalur terdiri dari titik asal, titik akhir, jarak dari kedua titik dan level



Gambar 2 Desain Basis Data

kemacetan dari kedua titik tersebut. Tabel jalur_moda merupakan tabel yang berisi informasi mengenai jalur beserta transportasi yang melewati jalur tersebut.

III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan akan menjelaskan mengenai proses implementasi Graf dan pengujian jalur terbaik Dijkstra dan Floyd Warshall, pengujian jalur alternatif, pengujian antarmoda, dan pengujian kinerja Dijkstra dan Floyd Warshall.

A. Implementasi Metode Graf

Sistem menggunakan Graf yang merepresentasikan lokasi naik/turunnya transportasi umum dan nilai jarak dari dua lokasi. *Node* merepresentasikan setiap lokasi stop asal dan stop tujuan. *Edge* adalah garis antar *node* yang akan merepresentasikan hubungan setiap *node*. Setiap *edge* memiliki arah yang disesuaikan dengan arah suatu transportasi umum dari satu *node* ke *node* lainnya. Setiap *edge* memiliki nilai/bobot yang merepresentasikan jarak tempuh dan waktu tempuh antara dua *node*. Setiap *edge* juga memiliki level kemacetan dan transportasi yang melalui jalur tersebut. Untuk pencarian jalur terpendek, algoritme pada metode Graf yang digunakan adalah Dijkstra dan Floyd Warshall, dan untuk mencari jalur alternatif algoritme yang digunakan adalah DFS.

B. Pengujian Jalur Terbaik Dijkstra

Jalur terbaik yang direkomendasikan sistem berdasarkan algoritme Dijkstra ialah jalur dengan nilai jarak terkecil dan jalur dengan nilai waktu tempuh terkecil. Pengujian dilakukan dengan skenario lokasi awal Terminal Cicaheum dan tujuan Opposite Surapati Core. Nilai level kemacetan di Seberang Yomart Padasuka ke Opposite Surapati Core ialah level 1 dan di Terminal Cicaheum ke Opposite Surapati Core ialah level 2. Tujuan pengujian ialah Dijkstra dapat menampilkan jalur terpendek berdasarkan jarak tempuh terpendek dan jalur tercepat berdasarkan waktu tempuh terpendek. Dari hasil pengujian yang dilakukan pada aplikasi, Dijkstra telah dapat menampilkan jalur terbaik berdasarkan jarak tempuh terpendek dan berdasarkan waktu tempuh terpendek. Jalur yang diperoleh yaitu Terminal Cicaheum – Opposite Surapati Core dengan jarak tempuh 0,41 km dan waktu tempuh 6 menit.

C. Pengujian Jalur Terbaik Floyd Warshall

Jalur terbaik yang direkomendasikan sistem berdasarkan algoritme Floyd Warshall ialah jalur dengan nilai jarak terkecil dan jalur dengan nilai waktu tempuh terkecil. Pengujian dilakukan dengan skenario yang sama dengan pengujian Dijkstra. Lokasi awal Terminal Cicaheum dan tujuan Opposite Surapati Core. Nilai level kemacetan di Seberang Yomart Padasuka ke Opposite Surapati Core ialah

level 1 dan di Terminal Cicaheum ke Opposite Surapati Core ialah level 2. Tujuan pengujian ialah Floyd Warshall dapat menampilkan jalur terpendek berdasarkan jarak tempuh terpendek dan jalur tercepat berdasarkan waktu tempuh terpendek.

Hasil pengujian yang dilakukan pada aplikasi, Floyd Warshall dapat menampilkan jalur terbaik berdasarkan jarak tempuh terpendek dan berdasarkan waktu tempuh terpendek. Jalur yang diperoleh yaitu Terminal Cicaheum – Opposite Surapati Core dengan jarak tempuh 0,41 km dan waktu tempuh 6 menit. Hasil jalur yang diperoleh Dijkstra dan Floyd adalah jalur yang sama.

D. Pengujian Jalur Alternatif

Pengujian jalur alternatif adalah pengujian yang dilakukan di mana terdapat lebih dari satu jalur yang dapat ditempuh dari lokasi asal ke lokasi tujuan. Algoritme yang digunakan dalam pencarian jalur alternatif ialah DFS. pengujian dilakukan dengan skenario lokasi asal Terminal Cicaheum dan tujuan Opposite Surapati Core. Pengujian dilakukan dengan tujuan algoritme DFS dapat menampilkan seluruh alternatif jalur yang dapat dilalui dan mengurutkannya berdasarkan jarak tempuh terpendek dan berdasarkan waktu tempuh terpendek.

Dari hasil pengujian yang dilakukan pada aplikasi, aplikasi telah dapat menampilkan seluruh jalur alternatif dan mengurutkannya berdasarkan jarak tempuh terpendek maupun waktu tempuh terpendek. Hasil jalur yang diperoleh ialah dua jalur alternatif. Jalur pertama yaitu Terminal Cicaheum – Opposite Surapati Core dengan jarak tempuh 0,41 km dan waktu tempuh 12 menit. Jalur kedua yaitu Terminal Cicaheum – Seberang Yomart Padasuka – Opposite Surapati Core dengan jarak tempuh 0,4204 km dan waktu tempuh 6 menit.

E. Pengujian Antarmoda

Pengujian antarmoda ialah pengujian yang dilakukan di mana dalam satu kali perjalanan membutuhkan lebih dari satu transportasi. Pengujian dilakukan dengan lokasi asal Halte Setelah Simpang Punawarman dan lokasi tujuan Seberang The Luxton BNI Dago Bawah. Tujuan pengujian ialah sistem dapat menampilkan semua pilihan jalur yang dapat digunakan. Jika dari hasil jalur yang diperoleh membutuhkan lebih dari satu moda transportasi maka sistem dapat menampilkan lokasi perpindahan transportasi dan pilihan transportasinya.

Dari hasil pengujian yang dilakukan pada aplikasi, aplikasi telah dapat menampilkan transportasi yang dapat digunakan dari lokasi asal ke tujuan. Selain itu lokasi perpindahan moda serta transportasi selanjutnya yang digunakan juga telah berhasil ditampilkan. Hasil jalur adalah Halte Setelah Simpang Purnawarman – Halte XL Riau – The 101 – Seberang The Luxton BNI Dago Bawah. Hasil transportasi ialah transportasi awal Angkot 09 kemudian pindah ke Angkot 02 di The 101.

F. Pengujian Kinerja Dijkstra dan Floyd Warshall

Faktor yang digunakan dalam membandingkan Dijkstra dan Floyd Warshall selain hasil jalur yang diperoleh ialah kinerja dari algoritme. Parameter yang digunakan dalam membandingkan kinerja ialah *throughput* dan *average time*.

Throughput menyatakan banyaknya operasi yang mungkin dilakukan per detik. *Average time* menyatakan rata-rata waktu yang diperlukan dalam suatu operasi.

Pengujian dilakukan dengan melakukan *benchmark* menggunakan *Java Microbenchmark Harness* (JMH). Parameter dalam melakukan pengujian ialah dengan 10 *fork*, 10 *warmup iteration*, dan 10 *measurement iteration*. *Fork* berfungsi untuk pemisahan eksekusi. *Warmup iteration* ialah iterasi yang dilakukan untuk memberikan kesempatan kepada JVM untuk mengoptimalkan kode sebelum pengukuran dimulai. *Measurement iteration* menyatakan berapa banyak iterasi yang harus diukur.

Pengujian dilakukan sebanyak 7 kali dan setiap pengujian terdiri dari 10 iterasi. Pengujian 1 dilakukan dengan 5 *node*, pengujian 2 dengan 15 *node*, pengujian 3 dengan 20 *node*, pengujian 4 dengan 25 *node*, pengujian 5 dengan 30 *node*, pengujian 6 dengan 35 *node*, dan pengujian 7 dengan 43 *node*.

Tabel I merupakan tabel hasil pengujian *throughput* dan *average time* Dijkstra dan Floyd Warshall. Dari hasil pengujian, dapat dilihat bahwa jumlah *node* dalam Graf mempengaruhi jumlah iterasi atau *loop* yang dilakukan suatu algoritme. Jumlah iterasi akan mempengaruhi kinerja dari suatu algoritme. Semakin banyak iterasi yang dilakukan semakin tinggi waktu yang dibutuhkan untuk memproses algoritme.

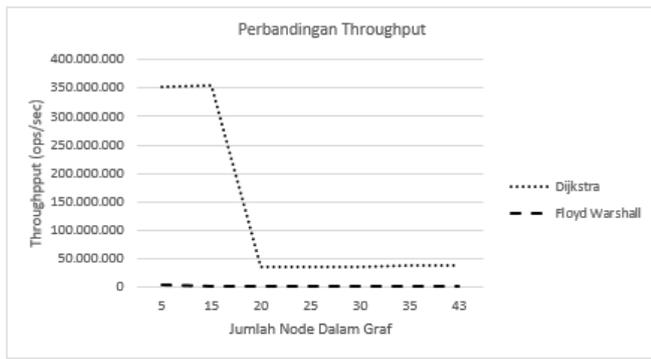
Gambar 3 menunjukkan Grafik perbandingan *throughput* Dijkstra dan Floyd Warshall. Dari Grafik tersebut dapat dilihat bahwa *throughput* Dijkstra lebih besar daripada Floyd Warshall. Hal ini menyatakan bahwa operasi yang dilakukan algoritma Dijkstra dalam satu detik lebih banyak atau lebih besar daripada Floyd Warshall.

Pada pengujian dengan jumlah *node* 5, *throughput* Dijkstra ialah 351.808.019,159 operasi per detik dan Floyd Warshall

TABEL I

HASIL PERBANDINGAN *THROUGHPUT* DAN *AVERAGE TIME*

| No | Algoritme | Jumlah Node | Jumlah Loop | Throughput (ops/sec) | Time (ns) |
|----|----------------|-------------|-------------|----------------------|-------------|
| 1 | Dijkstra | 5 | 4 | 351.808.019,159 | 2,877 |
| | Floyd Warshall | 5 | 125 | 2.976.325,123 | 339,832 |
| 2 | Dijkstra | 15 | 14 | 354.846.415,194 | 2,873 |
| | Floyd Warshall | 15 | 3.375 | 149.797,795 | 6.588,043 |
| 3 | Dijkstra | 20 | 21 | 34.948.778,413 | 28,467 |
| | Floyd Warshall | 20 | 8000 | 65.258,336 | 17.310,583 |
| 4 | Dijkstra | 25 | 26 | 36.341.234,957 | 27,959 |
| | Floyd Warshall | 25 | 15.625 | 37.965,379 | 26.405,104 |
| 5 | Dijkstra | 30 | 31 | 36.430.506,108 | 26,535 |
| | Floyd Warshall | 30 | 27.000 | 21.009,497 | 47.531,71 |
| 6 | Dijkstra | 35 | 36 | 39.261.244,684 | 24,897 |
| | Floyd Warshall | 35 | 42.875 | 9.887,282 | 109.968,734 |
| 7 | Dijkstra | 43 | 44 | 38.484.052,908 | 26,842 |
| | Floyd Warshall | 43 | 79.507 | 5.966,577 | 179.676,391 |



Gambar 3 Grafik Perbandingan *Throughput* Dijkstra dan Floyd Warshall

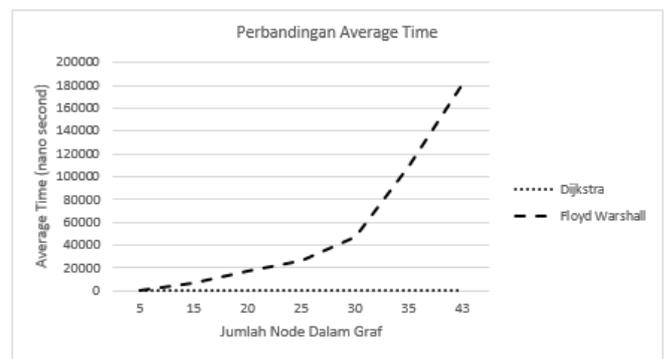
2.976.325,123 operasi per detik. Pada pengujian dengan jumlah *node* 15, *throughput* Dijkstra ialah 354.846.415,194 operasi per detik dan Floyd Warshall 149.797,795 operasi per detik. Pada pengujian dengan jumlah *node* 20, *throughput* Dijkstra ialah 34.948.778,413 operasi per detik dan Floyd Warshall 65.258,336 operasi per detik. Pada pengujian dengan jumlah *node* 25, *throughput* Dijkstra ialah 36.341.234,957 operasi per detik dan Floyd Warshall 37.965,379 operasi per detik. Pada pengujian dengan jumlah *node* 30, *throughput* Dijkstra ialah 36.430.506,108 operasi per detik dan Floyd Warshall 21.009,497 operasi per detik. Pada pengujian dengan jumlah *node* 35, *throughput* Dijkstra ialah 39.261.244,684 operasi per detik dan Floyd Warshall 9.887,282 operasi per detik. Pada pengujian dengan 43 *node*, *throughput* Dijkstra ialah 38.484.052,908 operasi per detik dan Floyd Warshall 5.966,577 operasi per detik.

Gambar 4 menunjukkan Grafik perbandingan *average time* Dijkstra dan Floyd Warshall. Dari Grafik tersebut dapat dilihat bahwa waktu yang diperlukan Dijkstra untuk menjalankan suatu operasi lebih cepat dibandingkan dengan Floyd Warshall.

Pada pengujian dengan jumlah *node* 5, *average time* Dijkstra ialah 2,877 ns dan Floyd Warshall 339,832 ns. Pada pengujian dengan jumlah *node* 15, *average time* Dijkstra ialah 2,873 ns dan Floyd Warshall 6.588,043 ns. Pada pengujian dengan jumlah *node* 20, *average time* Dijkstra ialah 28,467 ns dan Floyd Warshall 17.310,583 ns. Pada pengujian dengan jumlah *node* 25, *average time* Dijkstra ialah 27,959 ns dan Floyd Warshall 26.405,104 ns. Pada pengujian dengan jumlah *node* 30, *average time* Dijkstra ialah 26,535 ns dan Floyd Warshall 47.531,71 ns. Pada pengujian dengan jumlah *node* 35, *average time* Dijkstra ialah 24,897 ns dan Floyd Warshall 109.968,734 ns. Pada pengujian dengan jumlah *node* 43, *average time* Dijkstra ialah 26,842 ns dan Floyd Warshall 179.676,391 ns.

IV. KESIMPULAN

Metode Graf dapat digunakan dalam pemodelan titik-titik lokasi tempat naik atau turun transportasi umum dan jalur yang dilalui transportasi umum di Kota Bandung. Algoritme Dijkstra dan Floyd Warshall yang merupakan algoritme pada metode Graf dapat digunakan untuk mencari jalur terpendek dari dua lokasi yang diketahui. Dari 21 kali pengujian dengan



Gambar 4 Grafik Perbandingan *Average Time* Dijkstra dan Floyd Warshall

tiga Graf berbeda, 21 kali Floyd Warshall dan Dijkstra memperoleh jalur terpendek dengan hasil jalur yang sama.

Dijkstra lebih baik dalam hal *throughput* dan *average time* dibandingkan dengan Floyd Warshall. Dari 7 kali pengujian dengan total 70 iterasi, *throughput* Dijkstra selalu lebih besar daripada Floyd Warshall dan *average running time* Dijkstra selalu lebih kecil daripada Floyd Warshall. Pada *throughput*, untuk *node* terkecil yaitu 5 *node* selisih Dijkstra dan Floyd Warshall ialah 348.831.694 operasi per detik dan pada *node* terbesar yaitu 43 *node* selisihnya ialah 38.478.086,331 operasi per detik. Pada *average time*, untuk *node* terkecil yaitu 5 *node* selisih Dijkstra dan Floyd Warshall ialah 336,955 nano second dan pada *node* terbesar yaitu 43 *node* selisihnya ialah 179.649,549 nano second.

Untuk pengembangan ke arah yang lebih baik parameter lain seperti biaya yang dikeluarkan untuk setiap moda transportasi dan waktu yang dibutuhkan untuk menunggu keberangkatan transportasi umum dapat ditambahkan dalam penelitian berikutnya. Sistem informasi pencarian jalur terpendek dapat dikembangkan dengan beberapa algoritme pada metode Graf lainnya.

DAFTAR REFERENSI

- [1] E. W. Dijkstra, "A Note On Two Problems in Connexion with Graph", *Numerische Mathematik 1*, 1959.
- [2] Menteri Perhubungan Republik Indonesia (2005, Agustus 12). *Sistem Transportasi Nasional (Sistranas)*.
- [3] N. K. D. Ari Jayanti, "Penggunaan Algoritme Floyd Warshall dalam Masalah Jalur Terpendek pada Penentuan Tata Letak Parkir," *Seminar Nasional Informatika STMIK STIKOM Bali*, 2014.
- [4] R. A. D. Novandi, "Perbandingan Algoritme Dijkstra dan Algoritme Floyd-Warshall dalam Penentuan Lintasan Terpendek (*Single Pair Shortest Path*)," *Studi Kasus Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung*, 2007.
- [5] R. Lafore, *Data Structure & Algorithm in Java*, Sams Publishing, United States of America, 2003.
- [6] R. W. Floyd, "Algorithm 97: Shortheest path", *Communications of ACM*, 1962.
- [7] S. Arifianto, "Sistem Aplikasi Penentuan Rute Terpendek Pada Jaringan Multi Moda Transportasi Umum Menggunakan Algoritme Dijkstra," *Program Pascasarjana Universitas Diponegoro Semarang*, 2012.
- [8] UU Nomor 22 Tahun 2009, *Penggunaan dan Perelengkapan Jalan*.

Meliana Sitanggang, kelahiran Porsea, Sumatera Utara tahun 1995 menyelesaikan S1 di Jurusan *Media and Internet Technology* ITHB pada Agustus 2018. Memiliki minat pada *web dan mobile programming* dan *database*.

Yoyok Gamaliel, kelahiran Ciamis tahun 1974 dan memperoleh gelar Sarjana Teknik dari Universitas Kristen Satya Wacana, dan Master of Engineering dari University of South Australia. Minat penelitian pada analisis data serta pemodelan dan simulasi sistem. Saat ini aktif sebagai staf pengajar di Departemen Sistem Komputer dan Teknik Elektro di Institut Teknologi Harapan Bangsa, Bandung.

Herry Imanta Sitepu, menempuh pendidikan S1 di Teknik Elektro ITB dan lulus tahun 1999, dan memperoleh gelar magister dan doktor di jurusan yang sama di ITB. Sejak tahun 2006 aktif sebagai pengajar di Prodi Sistem Komputer Institut Teknologi Harapan Bangsa. Minat penelitian: *computer networking*, *programming* dan *distributed system*.