

Implementasi *Graph Database* untuk Menentukan Rute Perjalanan Transportasi Umum

M. Syauqi Hanif Ardani ¹⁾, Muhammad Ainul Yaqin ²⁾, Suhartono ³⁾

^{1),2),3)}Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang
Jl. Gajayana 50 Malang
Email: 14650056@student.uin-malang.ac.id

Abstrak. Transportasi umum terutama angkot sebagai sarana transportasi masal sudah lama menjadi moda transportasi yang dipakai masyarakat. Namun, ada beberapa kendala yang ditemui ketika masyarakat hendak memakainya. Salah satunya yaitu tidak hafal kode-kode jalur angkutan kota. Faktor tersebut menjadi penyebab beberapa pengguna angkot tidak tahu kode jalur trayek, sehingga mereka lebih memilih transportasi berbasis digital karena dianggap lebih mudah. Penerapan *Graph Database* dengan menggunakan *Weighted Directed Acyclic Graph (WDAG)* dalam *Neo4j* sebagai *Database Management System (DBMS)* juga dengan penentuan peringkat bobot menggunakan *Analytic Hierarchy Process (AHP)* mampu menyelesaikan persoalan trayek terbaik mana yang bisa digunakan masyarakat jika hendak bepergian dari satu tempat ke tempat lain dengan batasan maksimal 2 kali transfer dan menghasilkan hasil akurasi yang cukup baik yaitu 99%.

Kata kunci: Transportasi Umum, *Graph Database*, *WDAG*, *AHP*, *Neo4j*.

1. Pendahuluan

1.1. Latar Belakang

Angkutan umum yang biasa orang-orang menyebutnya dengan angkot bukan menjadi rahasia umum lagi akan urgensitasnya dalam mengatasi kemacetan yang terjadi di kota Malang ini. Selain dengan mudah aksesnya serta kemurahan juga menjadi prioritas utama kendaraan yang dibutuhkan oleh masyarakat sekitar kota Malang [1].

Pencarian rute terpendek merupakan suatu masalah yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950. Pencarian rute terpendek ini telah diterapkan di berbagai bidang untuk mengoptimasi kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat jalannya suatu proses. Salah satu aplikasi pencarian rute terpendek yang paling menarik untuk dibahas adalah pada masalah transportasi [2].

Banyak masalah alternatif jalur angkot yang dapat diselesaikan dengan menghitung jarak terpendek menggunakan pemodelan yang cocok, graf berbobot yang merepresentasikan jaringan transportasi. Pada transportasi publik, perlu diperhitungkan waktu yang diperlukan untuk melakukan transfer antar rute [3]. Semakin banyak transfer antar rute yang dilakukan, maka semakin banyak waktu dan biaya yang harus dikeluarkan oleh penumpang [4].

Pada data relasional, operasi *join* akan menurunkan kinerja, terlebih untuk data dengan volume yang besar. Namun, dibandingkan dengan data relasional, kinerja *graph* cenderung tetap, dan relatif konstan, walaupun terjadi pertumbuhan data yang besar. Hal tersebut disebabkan karena pada umumnya *query* pada *graph* terbatas pada suatu segmen atau bagian dari *graph* serta kemampuan *graph* untuk merambat dari satu *node* ke *node* yang lain dengan mudah. Waktu eksekusi untuk setiap *query* adalah hanya proporsional ke ukuran dari segmen graf yang terkait dengan *query* dan tidak dengan ukuran totalnya [5].

Analytical Hierarchy Process (AHP) merupakan suatu model pengambilan keputusan yang komprehensif dengan memperhitungkan hal – hal yang bersifat kualitatif dan kuantitatif. *AHP* pada penelitian ini berguna untuk menentukan bobot akhir dari beberapa bobot yang ditentukan, dalam hal ini adalah jarak dan jumlah angkot, sehingga hasil yang ada lebih akurat sesuai dengan bobot kriteria yang ditentukan.

1.2. Rumusan Masalah

1. Bagaimana mengimplementasikan *Graph Database* untuk menemukan rute angkutan umum menggunakan metode *Weighted Directed Acyclic Graphs (WDAG)*?

2. Bagaimana menentukan rute perjalanan terbaik dengan menggunakan metode *Analytical Hierarchy Process (AHP)*?

1.3. Tujuan Penelitian

1. Mengimplementasikan *Graph Database* untuk menemukan rute angkutan umum menggunakan metode WDAG.
2. Mengetahui rute perjalanan terbaik dengan menggunakan metode AHP.

1.4. Tinjauan Pustaka

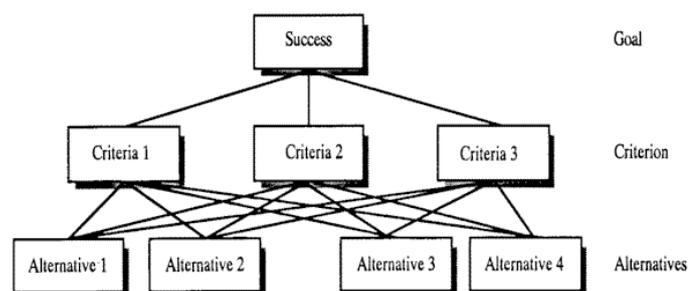
Graph Database Management System (selanjutnya, *Graph Database*) adalah sistem manajemen basis data *online* dengan metode *Create, Read, Update, dan Delete (CRUD)* yang mengekspos model data grafik. Database grafik umumnya dibuat untuk digunakan dengan sistem transaksional (OLTP). Dengan demikian, biasanya dioptimalkan untuk kinerja transaksional, dan direkayasa dengan integritas transaksional dan operasional [6].

Directed Graph $G = (V, E)$ terdiri dari dua set yaitu simpul atau *node* atau *vertice* dan tepi atau *edge*. Simpul-simpul yang ada memiliki asosiasi dengan simpul yang lainnya dengan tidak memiliki urutan [7].

Dalam dunia matematika dan ilmu komputer, *Directed Acyclic Graph (DAG)*, adalah graf berarah terbatas tanpa siklus. Yaitu, terdiri dari banyak *node* dan *edge* dengan masing-masing ujung diarahkan dari satu *node* ke *node* yang lain, sehingga tidak ada cara untuk memulai pada titik manapun dan mengikuti urutan ujung yang diarahkan secara konsisten yang akhirnya kembali ke V lagi. Dengan kata lain, *DAG* adalah graf berarah yang memiliki urutan topologi, urutan *node* sedemikian rupa sehingga setiap *edge* diarahkan dari *node* awal ke urutan selanjutnya.

Weighted Directed Acyclic Graphs (WDAG) adalah pasangan dari *edge* dan *node* yang berada di urutan standar yang setiap *edge* memiliki bobot dari *node* satu ke *node* tetangga sehingga bisa dihitung dari setiap *edge* yang ada.

AHP yang dikembangkan oleh Thomas L. Saaty, dapat memecahkan masalah kompleks. Kompleksitas ini disebabkan oleh struktur masalah yang belum jelas, ketidakpastian persepsi pengambil keputusan serta ketidakpastian tersedianya data statistik yang akurat atau bahkan tidak ada sama sekali [8]. Struktur hirarki AHP ditampilkan pada gambar 1.

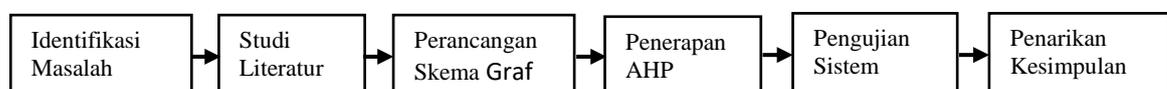


Gambar 1. Struktur Hirarki AHP

Dengan adanya penentuan trayek transportasi umum dengan mengimplementasikan *Graph Database* juga menggunakan metode *Weighted Directed Acyclic Graphs (WDAG)* dan *Analytic Hierarchy Process (AHP)*, maka diharapkan masyarakat pengguna angkot bisa mengetahui trayek yang dibutuhkan jika hendak bepergian ke suatu tempat dengan mudah dan cepat serta akurat.

2. Metodologi Penelitian

Dalam penelitian ini, ada beberapa langkah-langkah yang dilakukan dijelaskan dalam gambar 1.2.



Gambar 2. Metode Penelitian yang Dijalankan

3. Pembahasan

Pada penelitian ini, saya menggunakan data trayek angkutan kota (angkot) yang didapatkan dari dinas perhubungan Kota Malang dan Kota Batu. Data trayek angkot Kota Malang bias dilihat pada tabel 1 dan data trayek angkot Kota Batu pada tabel 2.

Tabel 1. Data Trayek Angkot Kota Malang

No	Jalur / Kode	Rute 1
1	AG / AH	Terminal Arjosari – Jl. Simpang R. Panji Suroso – Jl. Raden Intan – Jl. Jend A. Yani – Jl. Letjen S. Parman – Jl. Letjen Sutoyo – Jl. Jukung Suprpto – Jl. Basuki Rahmat – Merdeka Utara – Jl. Merdeka Timur – Jl. Sukarjowiryo Panoto – Jl. Pasar Besar – Jl. Sersan Harun – Jl. Prof. Moh. yamin – Jl. Sartono SH – Jl. Kol. Sugiono – Terminal Gadang
2	AL	Terminal Arjosari – Jl. R. Panji Suroso – Jl. Laksda Adi Sucipto – Jl. Tenaga – Jl. Karya Timur – Jl. Mahakam – Jl. W. R. Supratman – Jl. Panglima Sudirman – Jl. Patimura – Jl. Trunojoyo – Jl. Kertanegara – Jl. Tugu – Jl. Kahuripan – Jl. Semeru – Jl. Ijen – Jl. Retawu – Jl. Bondowoso – Jl. Jombang – Jl. Surabaya – Jl. Jakarta – Jl. Bogor – Jl. Veteran – Jl. Sumpersari – Terminal Landung Sari
3	ADL	Terminal Arjosari – Jl. Simpang R. Panji Suroso – Jl. Raden Intan – Jl. Jend. A. Yani – Jl. Letjen S. Parman – Jl. Letjen Sutoyo – Jl. W. R. Supratman – Jl. Panglima Sudirman – Jl. Patimura – Jl. Trunojoyo- Jl. Kertanegara – Jl. Kahuripan – Jl Semeru – Jl. Ijen – Jl. Bandung – Jl. Terusan Bogor – Jl. Mayjen Panjaitan – Jl. Mayjen Haryono – Jl. Tlogomas – Terminal Landung Sari

Tabel 2. Data Trayek Angkot Kota Batu

No	Jalur / Kode	Rute
1	BSS	Terminal Batu - Jl. Dewi Sartika - Jl. A.Salim - Jl. Sultan Agung - Jl. A.Yani - Brantas - Selecta - Junggo - Sumberbrantas
2	BJL	Terminal Batu - Jl. Dewi Sartika - Oro-oro Ombo - Junrejo - Dau - Sengkaling - Terminal Landungsari
3	BNK	Terminal Batu - Jl. Dewi Sartika - Jl. Agus Salim - Jl. Sultan Agung - Jl. Suropati - Jl.Hasanudin - Jl. Trunojoyo - Pujon - Ngantang - Kasembon

Penelitian ini dimulai dengan pemodelan *Graph Database* yang merepresentasikan jalur-jalur trayek angkot di Kota Malang dan Kota Batu. Data yang sudah didapatkan berupa data trayek dari setiap angkot, perlu dimasukkan kedalam *Graph Database* melalui beberapa cara, yaitu:

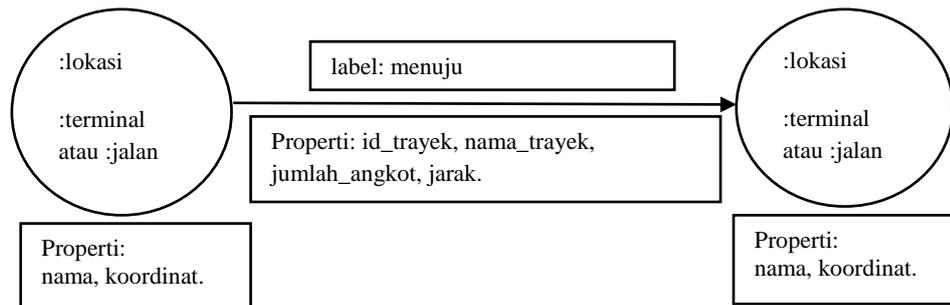
- Titik pemberhentian atau jalan yang dilewati oleh angkot, di create menjadi sebuah *node*
- Jalur atau arah kemana angkot melaju, di create menjadi sebuah relasi atau bisa disebut *edge*.

Skema *Graph Database* yang dibuat adalah sebagai berikut pada tabel 3.

Tabel 3. Skema graf yang dibangun

No	Node/Edge	Label	Properti	Keterangan
1	<i>Node</i>	lokasi, terminal	nama, koordinat	Untuk membuat node berupa terminal
2	<i>Node</i>	lokasi, jalan	tgl_pembuatan, tgl_modifikasi	Untuk membuat node berupa jalan
3	<i>Edge</i>	menuju	id_trayek, nama_trayek, jumlah_angkot, jarak	Untuk membuat relasi antar node dalam satu trayek

Atau jika digambarkan dalam bentuk visual, ditampilkan pada gambar 3.



Gambar 3. Skema Graf yang Dibangun

Susunan skema graf diatas kemudian diaplikasikan kedalam *Graph Database Neo4j* menggunakan *Cypher Query*.

A. Pembuatan Node

Untuk merepresentasikan setiap lokasi yang ada, dibuatlah *node-node* dengan *cypher query* menggunakan klausa *CREATE* dengan *property* seperti dijelaskan di gambar 3. Sebagai contoh dibuatkan satu *node* yang mewakili Terminal Arjosari dengan *cypher* sebagai berikut:

CREATE (:lokasi:terminal {nama:" Terminal Arjosari", koordinat: "-7.934936, 112.658836"})

B. Pembuatan Relasi

Setiap jalur yang dilalui trayek angkutan tertentu, dibuatkan relasi dari satu lokasi ke lokasi lainnya. Langkah pertama adalah dengan menginisialisasi node awal dan node tujuan dengan menggunakan klausa *MATCH*. Kemudian, menggunakan klausa *CREATE* untuk membuat relasi antar *node* awal dan *node* tujuan dengan *property* yang sudah disebutkan pada gambar 3. Misal dalam trayek angkutan dengan kode jalur AL, yang mana melewati Jalan Karya Timur kemudian Jalan Mahakam, maka *Cypher Query* yang digunakan menjadi sebagai berikut:

MATCH (n:lokasi {nama:"Jalan Karya Timur"}),(m:lokasi {nama:"Jalan Mahakan"}) CREATE (n)-[r:menuju {id_trayek:1, nama_trayek:"AL - Arjosari", jumlah_angkot:1, jarak:1600}]-> (m)

4. Pengujian

Dalam AHP, dibutuhkan penentuan bobot kriteria yang berfungsi untuk menginisialisasi bobot mana yang lebih prioritas dari bobot lainnya. Ada sembilan bobot kriteria yang bisa digunakan. Dalam penelitian ini, setelah melalui beberapa survei, maka didapatkan bahwa rute dengan jumlah transfer lebih sedikit merupakan sebagai prioritas dibandingkan dengan jarak tempuhnya agar biaya yang dikeluarkan lebih sedikit sehingga pemberian bobot ditentukan seperti pada tabel 4, kemudian dihitung nilai eigen nya.

Tabel 4. Kriteria Pembobotan Dengan AHP antara Jumlah Angkot dan Jarak

No	Kriteria	Bobot Kriteria
1	Jumlah Angkot	9
2	Jarak	3

Tabel 5. Normalisasi matrik pembobotan untuk menemukan nilai *Eigen*

Kriteria	Jumlah Angkot	Jarak	Rata-rata (<i>Eigen</i>)
Jumlah Angkot	1	3	0.75
Jarak	0.333333333	1	0.25
JKP	1.333333333	4	

Pengujian dilakukan dengan melakukan simulasi perjalanan. Pengambilan data semua alternatif trayek dari *Graph Database* dilakukan dengan menggunakan *Cypher Query*. Sebelum memulai dari satu lokasi ke lokasi lain, perlu dihitung dahulu kemungkinan trayek yang dilalui dari semua lokasi yang sudah dipetakan menggunakan *Cypher Query* sebagai berikut

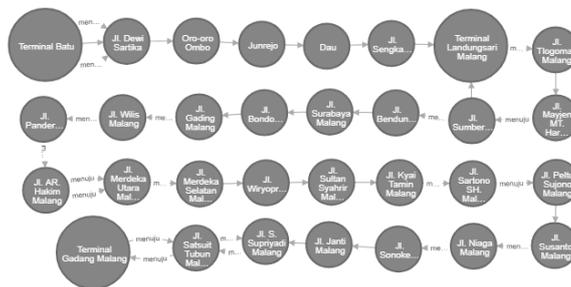
MATCH p = (lokasiAwal)-[:menuju*]->(lokasiTujuan) RETURN COUNT(r)

Cypher query tersebut mengembalikan nilai semua kombinasi relasi antar setiap lokasi yang sudah dibuat dengan model pada gambar 2.1 dan dengan data lokasi pada tabel 2.1 dan tabel 2.2, sehingga didapatkan bahwa ada sebanyak 113288 kombinasi perjalanan dari semua lokasi.

Selanjutnya, dibuat dengan simulasi perjalanan dari Terminal Batu menuju Terminal Gadang dengan Cypher Query sebagai berikut:

MATCH p = (lokasiAwal {nama:"Terminal Batu"}) -[:menuju*]-> (lokasiTujuan {nama: "Terminal Gadang Malang"}) RETURN p AS mencariAngkot, reduce(jumlah_angkot = 0, r in relationships(p) / jumlah_angkot+r.jumlah_angkot) AS totalTrayek, reduce(jarak = 0, r in relationships(p) / jarak+r.jarak) AS jarak;

Maka dihasilkan 6 alternatif trayek yang bisa menyelesaikan perjalanan tersebut dengan jarak yang sama, namun dengan jumlah angkot yang berbeda-beda diterangkan dalam tabel dibawah ini.



Gambar 4. Visualisasi Graf Perjalanan dari Terminal Batu ke Terminal Gadang

Pada gambar 4 didapatkan bahwa ada beberapa rute perjalanan yang bisa dipilih yang memiliki jarak dan jenis angkot yang berbeda-beda seperti ditampilkan pada tabel 6.

Tabel 6. Daftar Alternatif Trayek dari Terminal Batu ke Terminal Gadang

Alternatif	Jarak	Jumlah Angkot	Keterangan	Nama Trayek	Keterangan
Rute 1	33330	4	Tiga kali transfer	BSS, B JL, GA, LG	Transit di Jl. Dewi Sartika, Jl. AR. Hakim Malang, Terminal Landungsari,
Rute 2	33330	3	Dua kali transfer	BSS, B JL, LG	Transit di Jl. Dewi Sartika, Jl. Satsuit Tubun
Rute 3	33330	3	Dua kali transfer	B JL, GA, LG	Transit di Jl. AR. Hakim, Jl. Panderman Malang
Rute 4	33330	2	Satu kali transfer	B JL, LG	Transit di Jl. Satsuit Tubun
Rute 5	33330	4	Tiga kali transfer	BNK, B JL, GA, LG	Transit di Jl. Sengkaling, Jl. AR. Hakim Malang, Jl. Panderman Malang
Rute 6	33330	3	Dua kali transfer	BNK, B JL, LG	Transit di Jl. Sengkaling, Jl. Satsuit Tubun Malang

Pada penelitian ini, diberi batasan berupa jumlah maksimal transfer adalah dua kali atau maksimal menggunakan tiga angkutan yang berbeda, sehingga yang masuk dalam kriteria yaitu rute 2, rute 3, rute 4, dan rute 6. Rute 1 dan rute 5 tidak masuk dalam kriteria karena menggunakan 4 angkutan yang berbeda. Pemberian bobot kriteria juga berdasarkan berapa kali transit rute tersebut, sehingga ditentukan bobot kriteria setiap rute seperti pada tabel 7 dan kemudian dihitung matrik perbandingan pada tabel 8 dan nilai eigen dengan matrik ternormalisasi pada tabel 9.

Tabel 7. Pemberian Bobot Kriteria untuk Trayek Terpilih Berdasarkan Jumlah Kombinasi Angkot

Rute	2	3	4	6
Bobot Kriteria Berdasarkan Jumlah Angkot	7	7	9	7

Tabel 8. Matrik Perbandingan Berpasangan Berdasarkan Jumlah Angkot

Jumlah Angkot	Rute 2	Rute 3	Rute 4	Rute 6
Rute 2	1	1	0.777777778	1
Rute 3	1	1	0.777777778	1
Rute 4	1.28571	1.28571	1	1.2857143
Rute 6	1	1	0.777777778	1
JKP	4.28571	4.28571	3.333333333	4.2857143

Tabel 9. Matrik Ternormalisasi

Jumlah Angkot	Rute 2	Rute 3	Rute 4	Rute 6	Rata-rata (eigen)
Rute 2	0.23333	0.23333	0.233333333	0.2333333	0.233333333
Rute 3	0.23333	0.23333	0.233333333	0.2333333	0.233333333
Rute 4	0.3	0.3	0.3	0.3	0.3
Rute 6	0.23333	0.23333	0.233333333	0.2333333	0.233333333

Untuk perhitungan jarak karena ke empat rute mempunyai jarak yang sama, maka rata-ratanya menjadi seimbang yaitu 0.25. Kemudian matrik berpasangan antara perbandingan jarak dan jumlah angkot dinormalisasi dengan menggunakan nilai *Eigen* dari tabel 5, sehingga dihasilkan bobot akhir untuk setiap rute yang menyelesaikan perjalanan dari Terminal Batu menuju Terminal Gadang adalah Rute 4 yang paling baik yang disusul dengan rute 2, 3, 6 dengan bobot yang sama.

Tabel 10. Bobot Akhir Rute Perjalanan dari Terminal Batu ke Terminal Gadang

No	Alternatif Rute	Bobot Akhir
1	Rute 2	0.174158654
2	Rute 3	0.174158654
3	Rute 4	0.215584936
4	Rute 6	0.174158654

Pada tabel 10 didapatkan bahwa rute 4 memiliki bobot yang terbesar, sehingga merupakan pilihan alternatif rute terbaik untuk melakukan perjalanan dari Terminal Batu menuju Terminal Gadang dengan menggunakan angkot B JL, kemudian di Jalan Satsuit Tubun berpindah menggunakan angkot LG.

Perhitungan Akurasi

Untuk perhitungan akurasi menggunakan *Precision* pada persamaan 1, kemudian *Recall* pada persamaan 2 dan *Accuracy* pada persamaan 3. *Precision* menunjukkan tingkat ketepatan sebuah sistem untuk mengembalikan informasi relevan kepada pengguna. *Recall* merupakan nilai yang menunjukkan tingkat perolehan hasil yang dikembalikan oleh sebuah sistem [9].

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots (1)$$

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots (2)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots (3)$$

TP (*True Positif*) adalah jumlah rute yang bisa menyelesaikan perjalanan dengan maksimal 2x transfer dan muncul di hasil. FP (*False Positif*) adalah semua rute yang bisa menyelesaikan perjalanan walau lebih dari 2x transfer. TN (*True Negative*) semua rute yang dihasilkan walau tidak menyelesaikan perjalanan. FN (*False Negative*) adalah rute yang tidak bisa menyelesaikan perjalanan dan tidak keluar di hasil pencarian.

$$\text{Precision} = \frac{4}{4+2} = \frac{4}{6} = 0.67 \times 100 = 67\%$$

$$\text{Recall} = \frac{4}{4+0} = \frac{4}{4} = 1 \times 100 = 100\%$$

$$\text{Accuracy} = \frac{4 + (113288 - 6)}{113288} = \frac{113286}{113288} = 0.99 \times 100 = 99\%$$

5. Kesimpulan

Untuk perjalanan dari Terminal Landungsari menuju Terminal Gadang ditemukan dua rute dan rute kedua merupakan rute terbaik dengan menggunakan angkutan trayek LG dengan tanpa transfer. Untuk perjalanan dari Terminal Arjosari ke Jalan Wilis ada 4 rute yang bisa dipilih dan rute terbaik adalah rute ke 1 memakai trayek AL, kemudian di Terminal Landungsari pindah memakai trayek LG. Untuk perjalanan dari Terminal Batu menuju Terminal Gadang memiliki 6 rute yang bisa dipilih dengan rute ke 4 merupakan yang terbaik menggunakan trayek B JL kemudian di Jalan Satsuit Tubun berpindah angkot menggunakan LG.

Dari penelitian ini disimpulkan bahwa penggunaan *Graph Database* dan metode dalam menyelesaikan pencarian rute transportasi umum dengan batasan maksimal 2 kali transfer menghasilkan hasil akurasi yang cukup baik yaitu 99%.

Ucapan Terima Kasih

Terimakasih saya sampaikan kepada orang tua saya, juga untuk rekan dan kerabat yang telah menemani berdiskusi dan memberikan masukan dalam penyusunan paper ini.

Daftar Pustaka

- [1]. Rabbani, Imdad. 2017. *Mobile App Pencarian Rute Transportasi Umum Kota Malang dengan Algoritma Best-Path Planning*. Universitas Islan Negeri Maulana Malik Ibrahim Malang: tidak diterbitkan.
- [2]. Purwanantho Yudi, Diana Purwitasari, Agung Wahyu Wibowo. 2005. *Implementasi Dan Analisis Algoritma Pencarian Rute Terpendek Di Kota Surabaya*. Informatika, Surabaya.
- [3]. Geisberger Robert. 2011. *Advanced Route Planning in Transportation Networks*. PhD thesis, Karlsruhe Institute of Technology, February 2011.
- [4]. Liu, Chao-Lin. 2002. *Best-Path Planning For Public Transportation Systems*. Di Proceedings of the IEEE 5th International Conference on Intelligence Transportation Systems, pp. 834-839.
- [5]. Panji Wisnu, Wirawan, Djalal. 2017. *Kajian Implementasi Graph Database pada Rute Bus Rapid Transit*: Universitas Diponegoro, Vol. 3, No. 3.
- [6]. Robinson, Ian, Jim Webber, Emil Eifrem. 2015. *Graph Databases*. Sebastopol: O'Reilly Media, Inc.
- [7]. Jin, Jing. 2006. *Similarity Of Weighted Directed Acyclic Graphs*. The University Of New Brunswick
- [8]. M. Panjaitan, H. dan D. Sitompul, "Implementasi Algoritma K-Means dan Analytic Hierarchy Process (AHP) Untuk Klasterisasi Guru dan Memilih Guru Terbaik (Studi Kasus : SMA Santo Yoseph Medan)," *Jurnal Ilmu Komputer*, pp. 1 - 13.
- [7]. Novianti, Setiawan, Kusumawardani, "Peningkatan Nilai Recall dan Precision pada Penelusuran Informasi Pustaka Berbasis Semantik", Konferensi Nasional Sistem & Informatika 2015.