

**IMPLEMENTASI *GRAPH DATABASE* UNTUK
MENENTUKAN RUTE PERJALANAN
TRANSPORTASI UMUM**

SKRIPSI

**OLEH:
M. SYAUQI HANIF ARDANI
NIM. 14650056**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

**IMPLEMENTASI *GRAPH DATABASE* UNTUK MENENTUKAN
RUTE PERJALANAN TRANSPORTASI UMUM**

SKRIPSI

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
M. SYAUQI HANIF ARDANI
NIM. 14650056**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019**

HALAMAN PERSETUJUAN

**IMPLEMENTASI *GRAPH DATABASE* UNTUK MENENTUKAN
RUTE PERJALANAN TRANSPORTASI UMUM**

SKRIPSI

Oleh:
M. SYAUQI HANIF ARDANI
NIM. 14650056

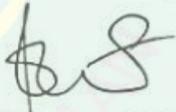
Telah Diperiksa dan Disetujui

Tanggal: Juni 2019

Dosen Pembimbing I

Dosen Pembimbing II


M. Ainul Yaqin, M.Kom
NIP. 19761013 200604 1 004


Dr. Suhartono, M.Kom
NIP. 19680519 200312 1 001

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

IMPLEMENTASI *GRAPH DATABASE* UNTUK MENENTUKAN RUTE PERJALANAN TRANSPORTASI UMUM

SKRIPSI

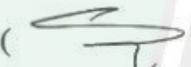
Oleh:
M. SYAUQI HANIF ARDANI
NIM. 14650056

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal : Juni 2019

Susunan Dewan Penguji

Penguji Utama : Syahiduz Zaman, M.Kom
NIP. 19700502 200501 1 005
Ketua Penguji : Khadijah Fahmi H. H., M.Kom
NIDT. 19900626 20160801 2 077
Sekretaris Penguji : M. Ainul Yaqin, M.Kom
NIP. 19761013 200604 1 004
Anggota Penguji : Dr. Suhartono, M.Kom
NIP. 19680519 200312 1 001

Tanda Tangan

()
()
()
()

Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Crysdyo Crysdyo

NIP. 19740424 200901 1 008

HALAMAN PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini,

Nama : M. Syauqi Hanif Ardani

NIM : 14650056

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul Skripsi : IMPLEMENTASI *GRAPH DATABASE* UNTUK
MENENTUKAN RUTE PERJALANAN
TRANSPORTASI UMUM

Menyatakan dengan sebenarnya benarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 21 Juni 2019

Yang membuat pernyataan,



M. Syauqi Hanif Ardani
NIM. 14650056

MOTTO



*“Jadilah Manusia Yang Bisa Berguna Bagi
Siapa Saja Yang Bertemu Dengannya”*

~ @ardan7779

HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim.

Alhamdulillah Rabbil'alamiin, nahmaduhu wa nasta'inuhu wa nastaghfiruh. Wa na'udzubillahi min suruuri anfusina wa min sayyiati a'malina. Tidak lain dan tidak bukan, karena nikmat, rahmat serta karunia-Nya penelitian ini bisa selesai dan sampai kepada anda sang pembaca. Hadirnya berbagai macam bantuan, bimbingan, masukan, serta doa yang dipanjatkan dari berbagai pihak juga sangat memberi kekuatan sendiri bagi saya untuk menyelesaikan skripsi ini. Saya persembahkan karya tulis yang memang masih jauh dari kata sempurna ini kepada:

Mama dan Bapak yang telah memberikan dukungan baik dari segi moril, materil serta doa yang terus mengalir. Adik-adik saya serta saudara yang juga berperan memberikan semangat agar saya tidak putus semangat.

Bapak Ainul Yaqin, M.Kom dan Bapak Dr. Suhartono, M.Kom selaku dosen pembimbing yang selalu memberikan arahan kemana serta bagaimana saya harus melangkah kedepannya. Bapak Cahyo Crysdiyan serta segenap dosen serta karyawan Jurusan Teknik Informatika. Juga segenap civitas akademika UIN Maulana Malik Ibrahim yang juga berperan dalam penyelesaian karya tulis ini.

Seluruh rekan saya di Teknik Informatika, juga kontrakan keat sentosa serta Majelis Sholawat Al Fataa yang sangat berperan dalam menenangkan serta menyeimbangkan hati. Untuk rekan seperjuangan khususnya Faldi, Caca, Hapsa, Wildan, Jun, yang paling sering ngopi bersama. Rekan Pondok Pesantren Anwarul Huda khususnya kamar A-10, rekan dari Keluarga Mahasiswa PLAT-R Malang, juga seluruh rekan yang tidak bisa saya sebut satu persatu, terimakasih semuanya. *Jazakumullahu Khairan Katsira.*

Saya juga persembahkan karya tulis ini kepada siapapun yang hendak memanfaatkan atau melanjutkan penelitian ini, tentu dengan mencantumkan sumber referensi, walaupun memang masih begitu banyak kekurangan, namun semoga bisa tetap dikembangkan hingga kemudian bisa bermanfaat bagi masyarakat luas. Untuk penelitian serta kajian ilmu, "*Jangan Berhenti di Kamu*".

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah segala puji bagi Allah tuhan semesta alam, dengan segala rahmat serta kasih sayang-Nya sehingga penulis mampu menyelesaikan naskah skripsi ini yang semoga bisa bermanfaat bagi generasi selanjutnya di kemudian hari. Tidak lupa juga shalawat serta salam penulis ucapkan untuk junjungan kita, suri tauladan kita, Rasulullah Muhammad SAW yang telah menuntun umatnya dari zaman kegelapan menuju zaman terang dengan Agama Islam yang mulia.

Aku, kamu, kita, semuanya adalah manusia yang berasal dari cairan hina. Manusia akan tetap hina jika tidak memanfaatkan hati serta akal yang diberikan Tuhan kepadanya. Ada banyak jalan bagi setiap individu manusia dalam memanfaatkan akal yang ia punya. Maka dari itu, penelitian ini semoga bisa menjadi salah satu *washilah* saya untuk menggunakan akal dalam rangka berusaha menjadi manusia yang berguna.

Selanjutnya penulis haturkan ucapan terima kasih karena dalam penyelesaian skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari beberapa pihak. Ucapan terima kasih ini penulis sampaikan kepada:

1. Mama dan Bapak saya yang telah memberikan dukungan moral, finansial, serta doa yang tidak terhitung jumlahnya dan tidak akan bisa saya membalasnya walau hingga kaki berdarah untuk menggendong mereka berdua menuju Kota Makkah.
2. Prof. DR. H. Abd. Haris, M.Ag, selaku rektor UIN Maulana Malik Ibrahim Malang beserta seluruh staf. Bakti Bapak dan Ibu sekalian terhadap UIN

Maliki Malang yang menaungi segala kegiatan di kampus UIN Maliki Malang.

3. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh staf. Bapak dan ibu sekalian sangat berjasa memupuk dan menumbuhkan semangat untuk maju kepada penulis.
4. Bapak Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang telah banyak memberi masukan dan motivasi untuk tidak mudah menyerah.
5. Bapak Ainul Yaqin, M.Kom selaku dosen wali sekaligus dosen pembimbing I yang telah bersedia meluangkan waktu untuk membimbing, memberi arahan dan masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
6. Bapak Dr. Suhartono, M.Kom selaku dosen pembimbing II juga senantiasa
7. Seluruh Dosen Teknik Informatika beserta staf yang telah memberikan keilmuan serta pengalaman yang berarti kepada penulis selama ini.
8. Seluruh teman teman Teknik Informatika UIN Maulana Malik Ibrahim Malang lintas angkatan yang telah banyak berbagi ilmu, pengalaman dan menjadi inspirasi untuk terus semangat belajar.
9. Teman teman seperjuangan angkatan 2014 khususnya teman teman TI C dan Biner Teknik Informatika yang telah berjuang bersama dan saling mendukung selama ini.

10. Para peneliti yang telah melakukan penelitian tentang *Graph Database*, AHP, serta transportasi publik yang menjadi acuan penulis dalam pembuatan skripsi ini.

11. Serta semua pihak yang telah membantu yang tidak bisa disebutkan satu persatu. Terimakasih banyak.

Penulis menyadari bahwa skripsi ini masih begitu banyak kekurangan serta jauh dari kesempurnaan dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis sangat mengharapkan saran, masukan serta kritik yang membangun dari berbagai pihak. Dan harapan penulis, semoga skripsi ini dapat memberikan manfaat bagi pembaca dan mendorong peneliti selanjutnya agar lebih baik dalam lagi kedepannya. Amiin.

Wassalamualaikum Wr. Wb.

Malang, 21 Juni 2019

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGAJUAN.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
ABSTRAK	xvii
ABSTRACT.....	xviii
ملخص	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	9
1.3 Tujuan Penelitian.....	9
1.4 Manfaat Penelitian.....	10
1.5 Batasan Masalah.....	10
BAB II LANDASAN TEORI	12
2.1 Pengertian <i>Graph</i>	12
2.2 Property Graph	13
2.3 Resource Description Framework (RDF).....	13
2.4 <i>HyperGraph</i>	14
2.5 <i>Graph Compute Engine</i>	14
2.6 <i>Graph Database</i>	15
2.6.1 Kelebihan <i>Graph Database</i>	16
2.7 <i>Weighted Directed Acyclic Graphs (WDAG)</i>	17
2.8 <i>Analytical Hirarchy Process (AHP)</i>	18
2.9 Penelitian Terkait	22

2.10	Android	23
BAB III METODOLOGI PENELITIAN.....		25
3.1	Desain Penelitian	25
3.2	Jenis Penelitian	25
3.3	Prosedur Penelitian.....	25
3.3.1	Identifikasi Masalah.....	26
3.3.2	Studi Literatur	26
3.3.3	Menentukan Input dan Output Dalam Aplikasi.....	26
3.3.4	Perancangan Sistem	27
3.3.5	Pengumpulan Data.....	29
3.3.6	Pemodelan Struktur Data <i>Graph Database</i>	36
3.3.7	<i>Weighted Directed Acyclic Graphs</i> Dalam <i>Graph Database</i>	38
3.3.8	Analisis Hasil Pengujian.....	45
BAB IV HASIL DAN PEMBAHASAN		51
4.1	Implementasi Sistem	51
4.2	Implementasi <i>Interface</i>	52
4.2.1	Tampilan <i>Splash Screen</i>	52
4.2.2	Tampilan Home / Menu Utama	53
4.2.3	Halaman Cari Alternatif Rute	55
4.2.4	Halaman Rute Terbaik	68
4.2.5	Halaman Lihat Data Lokasi	69
4.2.6	Halaman Semua Rute.....	70
4.2.7	Dialog Tentang Aplikasi	72
4.2.8	Dialog Bantuan.....	73
4.3	Pengujian Fungsionalitas.....	74
4.4	Perhitungan Akurasi	76
4.5	Integrasi Islam	80
BAB V PENUTUP.....		82
5.1	Kesimpulan.....	82
5.2	Saran	82
DAFTAR PUSTAKA		84
LAMPIRAN-LAMPIRAN.....		Error! Bookmark not defined.

DAFTAR GAMBAR

Gambar 1.1 Hasil Uji Tranversing Node	8
Gambar 2.1 Perpaduan Antara <i>Edge</i> dan <i>Node</i> Dalam <i>Graph A</i>	12
Gambar 2.2 Struktur Hirarki AHP	18
Gambar 2.3 Data Pengguna Internet dan Perangkat Mobile di Indonesia Menurut Survei Hootsuite.....	24
Gambar 3.1 <i>Flowchart</i> Sistem Yang Dibangun.....	28
Gambar 3.2 Usecase Diagram.....	29
Gambar 3.3 Skema Graf Yang Dibangun	37
Gambar 3.4 <i>Cypher Query</i> Membuat Node	37
Gambar 3.5 <i>Cypher Query</i> Membuat Relasi.....	38
Gambar 3.6 Alur Relasi Yang Merepresentasikan Jalur Angkutan Umum.....	39
Gambar 3.7 <i>Cypher Query</i> Menghitung Seluruh Kemungkinan Alternatif.....	46
Gambar 3.8 <i>Cypher Query</i> Perjalanan Dari Terminal Batu ke Terminal Gadang	46
Gambar 3.9. Visualisasi Graf dari Terminal Batu ke Terminal Gadang.....	47
Gambar 4.1 Tampilan <i>Splash Screen</i>	53
Gambar 4.2 Tampilan Halaman Utama	54
Gambar 4.3 Tampilan Halaman Cari Alternatif Rute	55
Gambar 4.4 <i>Cypher Query</i> Mengambil Nama Lokasi	56
Gambar 4.5 Tampilan <i>Spinner</i> Yang Menampilkan Lokasi	57
Gambar 4.6 Hasil Fitur Pencarian Dengan Kata Kunci "Batu"	58
Gambar 4.7 <i>Pseudocode</i> Menemukan Alternatif Rute	59
Gambar 4.8 <i>Pseudocode</i> Membatasi Jumlah Maksimal Transfer.....	60
Gambar 4.9 Alternatif Rute Yang Dihasilkan dari Terminal Batu Menuju Terminal Gadang Malang	60
Gambar 4.10 <i>Pseudocode</i> Pembobotan Alternatif Berdasarkan Jumlah Transfer	62
Gambar 4.11 <i>Pseudocode</i> Penentuan Nilai Array Jarak	62
Gambar 4.12 <i>Pseudocode</i> Pembobotan Alternatif Berdasarkan Jarak.....	63
Gambar 4.13 Penentuan Nilai Pembobotan Kriteria.....	65
Gambar 4.14 <i>Pseudocode</i> Pembobotan Matriks Eigen Kriteria	65
Gambar 4.15 <i>Pseudocode</i> Menghitung Jumlah Kolom Perbaris	66
Gambar 4.16 <i>Pseudocode</i> Pembagian Matrik Dengan JKP.....	66

Gambar 4.17 <i>Pseudocode</i> Menentukan Nilai Eigen Ternormalisasi	67
Gambar 4.18 <i>Pseudocode</i> Menentukan Bobot Akhir	67
Gambar 4.19 Halaman Rute Terbaik Menampilkan Alternatif Terpilih.....	68
Gambar 4.20 <i>Cypher Query</i> Mengambil Data Lokasi	69
Gambar 4.21 Halaman Lihat Data Lokasi.....	70
Gambar 4.22 <i>Cypher Query</i> Mengambil Data Semua Trayek Yang Tersedia	71
Gambar 4.23 Halaman Lihat Data Lokasi.....	71
Gambar 4.24 Dialog Tentang Aplikasi	72
Gambar 4.25 Dialog Bantuan.....	73



DAFTAR TABEL

Tabel 1.1 Tingkat Kendala Dalam Persen.....	4
Tabel 3.1 Data Trayek Angkot Kota Malang.....	30
Tabel 3.2 Data Trayek Angkot Kota Batu	30
Tabel 3.3 Data Trayek Menurut Urutan Jalan dan Terminal di Kota Malang	31
Tabel 3.4 Data Trayek Menurut Urutan Jalan dan Terminal di Kota Batu.....	32
Tabel 3.5 Data Lokasi dan Koordinat Kota Malang	32
Tabel 3.6 Data Lokasi dan Koordinat Kota Malang Lanjutan	33
Tabel 3.7 Data Lokasi dan Koordinat Kota Malang Lanjutan	34
Tabel 3.8 Data Lokasi dan Koordinat Kota Malang Lanjutan	35
Tabel 3.9 Data Lokasi dan Koordinat Kota Batu	35
Tabel 3.10 Data Lokasi dan Koordinat Kota Batu Lanjutan.....	36
Tabel 3.11 Skema Graf	37
Tabel 3.12 Jarak Antar Lokasi Trayek AL - Arjosari	40
Tabel 3.13 Jarak Antar Lokasi Trayek AL - Arjosari Lanjutan	41
Tabel 3.14 Jarak Antar Lokasi Trayek GA - Gadang	41
Tabel 3.15 Jarak Antar Lokasi Trayek GA - Gadang Lanjutan	42
Tabel 3.16 Jarak Antar Lokasi Trayek LG - Landungsari	42
Tabel 3.17 Jarak Antar Lokasi Trayek LG - Landungsari Lanjutan	43
Tabel 3.18 Jarak Antar Lokasi Trayek BSS - Terminal Batu	43
Tabel 3.19 Jarak Antar Lokasi Trayek BJJ - Terminal Batu.....	44
Tabel 3.20 Jarak Antar Lokasi Trayek BNK - Terminal Batu.....	44
Tabel 3.21 Kriteria Pembobotan Dengan AHP antara Jumlah Angkot dan Jarak	45
Tabel 3.22 Normalisasi matrik pembobotan untuk menemukan nilai Eigen	45
Tabel 3.23 Daftar Alternatif Trayek dari Terminal Batu ke Terminal Gadang ...	48
Tabel 3.24 Pemberian Bobot Kriteria untuk Trayek Terpilih Berdasarkan Jumlah Kombinasi Angkot	49
Tabel 3.25 Matrik Perbandingan Berpasangan Berdasarkan Jumlah Angkot	49
Tabel 3.26 Matrik Ternormalisasi	49
Tabel 3.27 Bobot Akhir Rute Perjalanan dari Terminal Batu ke Terminal Gadang	50
Tabel 4.1 Autentifikasi Koneksi ke <i>Graph Database</i>	56

Tabel 4.2 Tabel Pembobotan Alternatif Trayek Berdasarkan Jumlah Transfer...	61
Tabel 4.3 Nilai Pembobotan Menurut Saaty	64
Tabel 4.4 Pengujian Fungsionalitas Aplikasi.....	74
Tabel 4.5 Pengujian Fungsionalitas Aplikasi Lanjutan.....	75
Tabel 4.6 Keterangan Nilai-nilai Perhitungan Akurasi.....	77
Tabel 4.7 Daftar Simulasi Yang Dilakukan Serta Alternatif Yang Didapatkan ..	77
Tabel 4.8 Nilai <i>Precision</i> , <i>Recall</i> dan <i>Accuracy</i> Dari Setiap Simulasi	78
Tabel 4.9 Nilai <i>Precision</i> , <i>Recall</i> dan <i>Accuracy</i> Akhir.....	79



ABSTRAK

Ardani, M. Syauqi Hanif. 2019. **Implementasi *Graph Database* Untuk Menentukan Rute Perjalanan Transportasi Umum**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) M. Ainul Yaqin, M.Kom. (II) Dr. Suhartono, M.Kom.

Kata kunci: transportasi umum, *Graph Database*, AHP.

Angkot sebagai sarana transportasi masal sudah lama menjadi moda transportasi yang dipakai masyarakat. Angkot juga menjadi mata pencaharian bagi para supir, serta menjadi alternatif transportasi umum guna mengurangi volume kendaraan serta polusi dijalanan. Namun, ada beberapa kendala yang ditemui ketika masyarakat hendak memakainya sehingga animo masyarakat menjadi berkurang dalam memakai angkot.

Tidak tau rute disebabkan karena banyaknya rute yang ada, serta dengan kode yang mirip sehingga membingungkan calon pengguna. Permasalahan ini yang menjadi topik penelitian karena diharapkan bisa diselesaikan menggunakan sebuah aplikasi yang memudahkan calon pengguna angkot menemukan nama-nama trayek yang harus digunakan. Penelitian ini juga diharapkan bisa memudahkan untuk menemukan jalur yang termudah dan tidak terlalu banyak transfer.

Dalam *Graph Database* Neo4j, lokasi dan trayek dijadikan menjadi *node* dan *edge*. *Node* yang menjadi representasi dari lokasi mempunyai label 'lokasi' dan 'jalan' atau 'terminal', serta properti 'nama' dan 'koordinat'. *Edge* yang menjadi representasi dari trayek mempunyai label 'menuju' serta properti 'id_trayek', 'nama_trayek', 'jarak'.

Penggunaan *Graph Database* untuk menentukan rute angkot dapat dilakukan dengan cara kombinasi *Cypher Query* dengan kemudian dilakukan perhitungan menggunakan *Analytical Hierarchy Process (AHP)* untuk menentukan rute terbaik. Implementasi *Graph Database* ini dapat menyelesaikan persoalan pencarian rute angkot yang ada. Hasil dari penelitian ini menemukan bahwa Implementasi *Graph Database* serta dilakukan perhitungan dengan AHP mendapatkan nilai *Precision* yang berarti bahwa data yang dihasilkan antara alternatif satu dengan yang lain memiliki tingkat kemiripan sebesar 54.86 % , nilai *Recall* yang menandakan jumlah data seharusnya yang dihasilkan sebesar 100%, serta nilai *Accuracy* yang menandakan tingkat akurasi sebesar 82.13 %.

ABSTRACT

Ardani, M. Syauqi Hanif. 2019. **Implementation of Graph Database to Determine Travel Routes for Public Transportation**. Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology Islamic State Maulana Malik Ibrahim University, Malang. Supervisor: (I) M. Ainul Yaqin, M.Kom. (II) Dr. Suhartono, M.Kom.

Keyword: public transportation, Graph Database, AHP.

Public transportation as a means of mass transportation has long been a mode of transportation used by the community. Public transportation is also a livelihood for drivers, as well as being a public transportation alternative to reduce vehicle volume and pollution in the streets. However, there are some obstacles encountered when people want to use them so that the public interest is reduced in using public transportation.

Don't know the route is caused by the many routes that exist, as well as with similar codes that confuse prospective users. This problem is the topic of research because it is expected to be solved using an application that makes it easier for prospective public transportation users to find the names of routes that must be used. This research is also expected to make it easier to find the easiest path and not too many transfers.

In the Neo4j Graph Database, locations and routes are made into nodes and edges. Nodes that are representations of locations have two labels named '*lokasi*' and '*jalan*' or '*terminal*', also properties named '*nama*' and '*koordinat*'. Edge which is the representation of the route has a label named '*menuju*' also properties named '*id_trayek*', '*nama_trayek*', '*jarak*'.

The use of a Graph Database to determine public transportation routes can be done by means of a Cypher Query combination with calculations using the Analytical Hierarchy Process (AHP) to determine the best route. This Graph Database Implementation can solve the search problem for existing public transportation routes. The results of this study found that the implementation of Graph Database and calculations using AHP obtained a Precision value which means that the data generated between alternatives with one another has a similarity rate of 54.86%, the value of Recall which indicates the amount of data that should be generated is 100%, and Accuracy value which indicates an accuracy level of 82.13%.

ملخص

أردان، م. شوقي حنيف. 2019. **تطبيق Graph Database لتحديد طريق السفر لآلات الإتصال العام، البحث العلمي.** قسم المعلوماتية كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف : (1) م. عين اليقين الماجستير، (2) دكتور سهارتونو الماجستير.

الكلمات الرئيسية: آلات الإتصال العام، *Graph Database, AHP*

"أنكوت" من آلات الإتصال الجماعي منذ فترة طويلة ويستخدم المجتمع بها. "أنكوت" هي معيشة للسائقين أيضاً، وأفضل آلات الإتصال لتقليل حجم المركبات والتلوث في الشوارع. لكنه ، هناك بعض العوائق التي يواجهها الأشخاص عندما يرغبون في استخدامها حتى نقص المجتمع في استخدامها.

لا يعرف المجتمع بسبب في هذا المسار هو وجود العديد من المسارات الموجودة، وكذلك مع الرموز المشابهة التي تربك المستخدمين المحتملين. هذه المشكلة هي موضوع البحث لأنه من المتوقع حلها باستخدام تطبيق يسهل على المستخدمين المحتملين على أسمائها التي يجب استخدامها. يُتوقع بهذا البحث العثور على أسهل الطريق قَلَّ المنقول.

في *Graph Database Neo4j* ، يتم إنشاء المواقع والطرق في *node* و *edge*. تحتوي *node* التي تمثل المواقع على تصنيفين باسم "*lokasi*" و "*jalan*" أو "*terminal*" ، وكذلك الخصائص المسماة "*nama*" و "*koordinat*". *edge* التي تمثل المسار لها تسمية تحمل الأسم "*menuju*" وكذلك الخصائص المسماة "*id_trayek*" و "*nama_trayek*" و "*jarak*".

استخدام *graph database* لتحديد طريق "الأنكوت" يستطيع أن يستخدم بطريق مزيج *Cypher Query* مع العمليات الحسابية باستخدام *Analytical Hierarchy Process (AHP)* لتحديد أفضل مسار. يمكن تطبيق *graph database* هذا حل مشكلة البحث عن مشكلة المسار. وجدت نتائج هذه الدراسة أن تطبيق *Graph Database* والحسابات باستخدام *AHP* حصلت على قيمة دقيقة مما يعني أن البيانات المتولدة بين البدائل مع بعضها البعض لديها معدل تشابه بنسبة 54.86% ، تشير قيمة التذكير إلى مقدار البيانات التي يجب أن يكون 100% ، وقيمة الدقة التي تشير إلى مستوى دقة 82.13%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Angkot sebagai sarana transportasi masal sudah lama menjadi moda transportasi yang dipakai masyarakat. Angkot juga menjadi mata pencaharian bagi para supir, serta menjadi alternatif transportasi umum guna mengurangi volume kendaraan serta polusi dijalanan. Namun, ada beberapa kendala yang ditemui ketika masyarakat hendak memakainya sehingga animo masyarakat menjadi berkurang dalam memakai angkot.

Saya melakukan wawancara untuk mengetahui kendala yang dialami ketika menggunakan angkot kepada beberapa pengguna. Ada beberapa point yang narasumber sampaikan terkait dengan pengalaman mereka ketika melakukan perjalanan menggunakan angkot. Berikut beberapa hasil wawancara yang dilakukan.

Isma Farikha Latifatun Nuzulia (23), seorang pelajar serta pendatang asal Kota Wonosobo. Jawa Tengah yang juga pengguna angkot menyampaikan pengalamannya dalam menggunakan angkot. Dia menyebutkan beberapa kendala diantaranya angkot yang memiliki jalur memutar sehingga yang harusnya bisa ditempuh dengan 30 menit malah bisa menjadi 45 menit. Juga terkait dengan penumpang yang sering *overload* atau angkot yang kelebihan muatan sehingga terasa terlalu sesak. Juga perihal banyaknya trayek angkot yang terlalu banyak sehingga membingungkan bagi kebanyakan pengguna. Dia juga menyampaikan bahwa jika kualitasnya diperbaiki, serta rute dan kenyamanannya dibenahi maka

dimungkinkan masyarakat lebih memilih angkot untuk moda transportasi, serta bisa juga untuk mencontoh moda transportasi umum di luar negeri yang lebih tertata serta didukung oleh pemerintah.

Indah Rahayuning Tyas (23) mahasiswi yang berasal dari Kabupaten Purbalingga Jawa Tengah menyampaikan beberapa pengalaman serta kendala yang dialami ketika menggunakan angkot. Dia merasa angkot yang terlalu lama menunggu penumpang menjadi kendala, juga terkait dengan tidak hafal singkatan-singkatan rute angkot yang memang begitu banyak sehingga ketika hendak menuju lokasi bingung hendak menggunakan angkot yang mana. Angkot juga terkadang tidak tertib dengan trayek yang seharusnya dijalani tidak sampai tujuan, namun di tengah jalan harus oper dahulu dengan angkot lain yang mana ini merugikan penumpang. Angkot juga sering menunggu lama sampai penumpang penuh dahulu baru berangkat. Terkait dengan rute angkutan juga pernah mengalami pengalaman kurang mengenakan karena dia tidak tahu rute angkutan dan kemudian tanya kepada supir angkot yang ada, tetapi malah tidak dijawab dengan semestinya karena sebenarnya trayek angkot terkait tidak melewati tujuannya tetapi mengaku bahwa angkot itu melewatinya, sehingga saudara narasumber seakan dibodohi dan merasa dirugikan.

Dimas Wahyu Pratama (22) seorang mahasiswa yang berasal dari Kota Bekasi yang juga merupakan pengguna aktif angkot menyampaikan beberapa kendala yang pernah dirasakan ketika moda transportasi angkot. Dia menyampaikan bahwa salah satu kendala yaitu angkot yang menunggu penumpang terlalu lama, sehingga dibutuhkan jalan dahulu untuk mencari angkot yang sudah siap jalan. Angkot yang tidak konsisten terhadap jalur juga menjadi salah satu

kendala yang mana menurunkan penumpang tidak pada tempatnya padahal tujuan penumpang sebenarnya dilalui angkot tersebut. Mengenai trayek angkot, ia mencari informasi terkait nama-nama rute yang tersedia.

Seorang pelajar dan pendatang asal Kabupaten Cilacap, Jawa Tengah bernama Nisa Amelia (20) yang juga merupakan pengguna aktif angkutan umum (angkot) menyampaikan beberapa kendala dan masukan ketika menggunakan angkot. Kendala utama yang dia rasakan adalah kebingungan ketika hendak bepergian ke suatu tempat. Begitu banyaknya kode trayek serta kemiripan yang ada menyebabkan dia kesusahan untuk menentukan trayek mana yang hendak dipakai dan harus tanya-tanya dulu kepada orang lain yang juga belum tentu mengetahui. Juga terkait dengan rute trayek angkot yang memang tidak bisa menjemput hingga titik pengguna menjadi kendala tersendiri. Selain itu, angkot yang terlalu lama menunggu penumpang (berhenti untuk menunggu penumpang lain) juga menjadi kendala karena lebih membuang waktu penumpang yang sudah berada di dalam angkot tersebut. Nisa menyebutkan, jika kendala-kendala tersebut bisa diatasi maka akan menambah animonya juga masyarakat umum untuk lebih memilih angkot sebagai moda transportasi.

Dari keempat narasumber yang sudah memaparkan pengalamannya dalam menggunakan angkot, disimpulkan bahwa ada beberapa kendala yang masih dimiliki yaitu:

1. Kenyamanan
2. Tidak tau rute yang digunakan
3. Angkot sering menunggu penumpang terlalu lama

4. *Overload* penumpang
5. Angkot yang tidak konsisten jalurnya
6. Penipuan

Dari keenam kendala itu, dilakukan sebuah survei kepada pengguna angkot secara lebih luas. Dilakukan dengan menggunakan formulir kuesioner agar mendapatkan presentasi kendala yang dialami dari masyarakat yang lebih luas. Hasil dari kuesioner ditampilkan dalam Tabel 1.1 berikut.

Tabel 1.1 Tingkat Kendala Dalam Persen

No	Kendala	Presentase
1	Kenyamanan	18,5 %
2	Tidak tau rute yang digunakan	12,9 %
3	Angkot sering menunggu penumpang lama	44,3 %
4	<i>Overload</i> penumpang	18,5 %
5	Angkot yang tidak konsisten jalurnya	2,8 %
6	Penipuan	2,8 %

Angkot menunggu penumpang terlalu lama menjadi faktor kendala dengan presentase yang paling tinggi dari yang lainnya yaitu 44,3 %. Ini menjadi kendala yang titik permasalahan serta solusi berada disisi supir serta regulasi angkot yang berlaku. Hal ini menyebabkan terkait kendala angkot yang menunggu penumpang terlalu lama tidak menjadi pembahasan di penelitian ini.

Overload penumpang menjadi faktor terkuat kedua sebagai kendala yang dialami pengguna angkot dengan presentase 18,5%. *Overload* penumpang menjadikan angkot penuh sesak melebihi kapasitas muatan yang sebenarnya. Titik permasalahan dan solusi berada pada supir angkot dan regulasi yang berlaku sehingga penelitian ini tidak menuju kesitu.

Kenyamanan menjadi faktor terkuat ke tiga dengan presentase 18,5% sama dengan presentase untuk kendala *Overload* penumpang. Kenyamanan juga sangat diperlukan bagi pengguna angkot. Namun, titik permasalahan dan solusi berada pada kendaraan yang harus diperbaiki serta memenuhi regulasi yang berlaku. Pemerintah sangat berperan untuk ikut serta menyelesaikan permasalahan ini.

Tidak mengetahui rute yang digunakan menjadi faktor terkuat yang keempat dengan presentase 12,8%. Tidak tau rute disebabkan karena banyaknya rute yang ada, serta dengan kode yang mirip sehingga membingungkan calon pengguna. Permasalahan ini yang menjadi topik penelitian karena diharapkan bisa diselesaikan menggunakan sebuah aplikasi yang memudahkan calon pengguna angkot menemukan nama-nama trayek yang harus digunakan. Penelitian ini juga diharapkan bisa memudahkan untuk menemukan jalur yang termudah dan tidak terlalu banyak transfer.

Penipuan juga menjadi kendala pengguna angkot dengan presentase 2,8%. Ini karena memang angkot merupakan transportasi publik yang siapapun bisa mengaksesnya. Ada banyak masyarakat yang memakai dan mungkin ada yang berniat tidak baik kepada pengguna lain, namun persentasenya tidak terlalu banyak. Untuk penyelesaian kendala ini perlu adanya peran pemerintah serta masyarakat pada umumnya.

Angkot yang tidak konsisten jalurnya juga menjadi faktor kendala dengan presentasi 2,5%. Angkot terkadang tidak memenuhi rute yang seharusnya dan memilih jalan yang lain. Ini menyebabkan pengguna merasa dirugikan walau memang persentasenya tidak terlalu besar. Kendala ini bisa diselesaikan jika supir lebih tertib dan menaati rute trayek yang sudah ditentukan.

Angkutan umum yang biasa orang-orang menyebutnya dengan angkot bukan menjadi rahasia umum lagi akan urgenitasnya 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 (Rabbani Imdad, 2017).

Dalam Al-Qur'an juga membahas tentang transportasi untuk manusia gunakan dalam kehidupan. Allah SWT menjelaskan tentang transportasi melalui firman-Nya dalam surat An-Nahl ayat 7-8. Allah SWT berfirman:

وَتَحْمِلُ أَثْقَالَكُمْ إِلَىٰ بَلَدٍ لَّمْ تَكُونُوا بَالِغِيهِ إِلَّا بِشِقِّ الْأَنْفُسِ إِنَّ رَبَّكُمْ لَرءُوفٌ رَّحِيمٌ (٧) وَالْخَيْلَ وَالْبِغَالَ وَالْحَمِيرَ لِتَرْكَبُوهَا وَزِينَةً وَيَخْلُقُ مَا لَا تَعْلَمُونَ (٨)

Artinya: “Dan ia (hewan ternak) mengangkut beban-bebanmu ke suatu negeri yang kamu tidak sanggup mencapainya, kecuali dengan susah payah. Sungguh, Rabbmu Maha Pengasih, Maha Penyayang (7). Dan (Dia telah menciptakan) kuda, bagal, dan keledai untuk kamu tunggahi dan (menjadi) perhiasan. Allah menciptakan apa yang tidak kamu ketahui (8)”.

Pencarian rute transportasi merupakan langkah yang cerdas dalam membantu para pengguna angkutan umum. Pencarian rute terpendek merupakan suatu masalah yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950 (Purwananto, 2005). 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 (Geisberger, 2011). Semakin banyak transfer antar rute yang dilakukan, maka semakin banyak waktu dan biaya yang harus dikeluarkan oleh penumpang (Chao, 2002).

Perkembangan teknologi tidak bisa dibendung. Semakin hari semakin maju dan terbukti dapat menyelesaikan berbagai macam masalah sehari-hari. Terutama penggunaan internet yang hampir tidak dapat terlepas dari sistem modern sekarang ini. Kemajuan teknologi juga merupakan sebuah kebesaran Allah SWT yang mana sudah termaktub didalam kitab suci Al-Qur'an surat Al Baqarah ayat 164. Allah SWT berfirman:

إِنَّ فِي خَلْقِ السَّمَاوَاتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ وَالْفُلْكِ الَّتِي تَجْرِي فِي الْبَحْرِ بِمَا يَنْفَع النَّاسَ وَمَا أَنْزَلَ اللَّهُ مِنَ السَّمَاءِ مِنْ مَاءٍ فَأَحْيَا بِهِ الْأَرْضَ بَعْدَ مَوْتِهَا وَبَثَّ فِيهَا مِنْ كُلِّ دَابَّةٍ وَتَصْرِيفِ الرِّيَّاحِ وَالسَّحَابِ الْمُسَخَّرِ بَيْنَ السَّمَاءِ وَالْأَرْضِ لَآيَاتٍ لِقَوْمٍ يَعْقِلُونَ

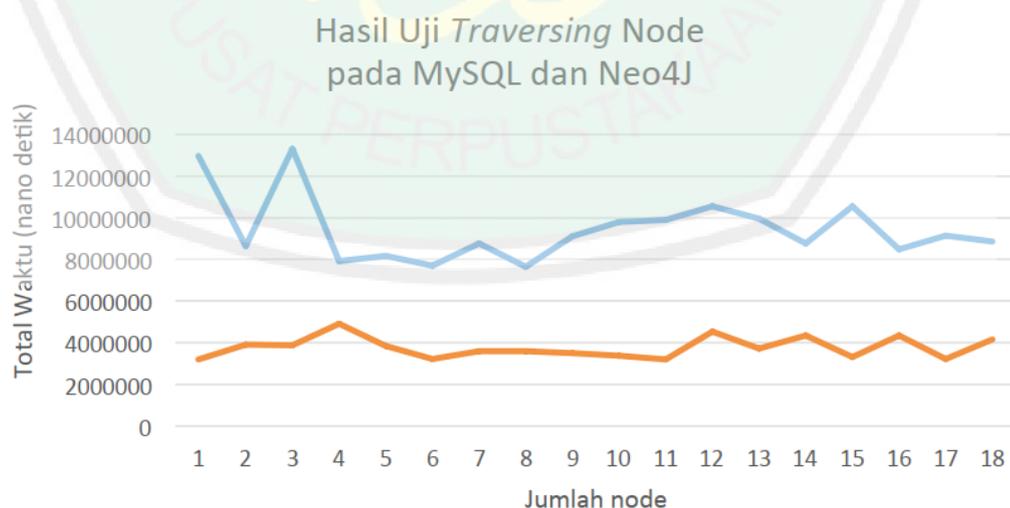
Artinya: “*Sesungguhnya dalam penciptaan langit dan bumi, silih bergantinya malam dan siang, bahtera yang berlayar di laut membawa apa yang berguna bagi manusia, dan apa yang Allah turunkan dari langit berupa air, lalu dengan air itu Dia hidupakan bumi sesudah mati (kering)-nya dan Dia sebarkan di bumi itu segala jenis hewan, dan pengisaran angin dan awan yang dikendalikan antara langit dan bumi; sungguh (terdapat) tanda-tanda (keesaan dan kebesaran Allah) bagi kaum yang memikirkan*”.

Salah satu kemajuan yang terlihat adalah pengolahan data yang semakin besar dengan lebih baik seperti di dalam media sosial yang menampung ribuan hingga jutaan pengguna yang saling terhubung dengan menggunakan sebuah *database* berbasis *Graph*.

Graph Database merupakan sebuah perkembangan teknologi yang mulai ramai dikaji dan diterapkan di berbagai bidang. *Graph Database* memungkinkan database dengan tanpa baris dan kolom seperti *Relational Database*, tetapi merupakan kumpulan *node-node* yang saling terhubung dengan adanya *edge*. Model database seperti ini memungkinkan pemrosesan data dengan lebih cepat, terutama dalam jumlah yang sangat besar.

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 (Wirawan Panji Wisnu, Djalal, 2017).

Pada tahun 2016, Panji dkk menemukan bahwasanya penggunaan *Graph Database* dalam mengatasi masalah perambatan (*traversing*) menunjukkan hasil yang lebih baik daripada menggunakan basis data relasional. Penelitian yang dilakukan merupakan kajian BRT (*Bus Rapid Transit*) yang berada di Kota Semarang. Hasilnya ditampilkan pada Gambar 1.1.



Gambar 1.1 Hasil Uji Tranversing Node

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.

Dengan adanya penentuan trayek transportasi umum dengan mengimplementasikan *Graph Database* juga menggunakan metode *Weighted Directed Acyclic Graphs (WDAG)* serta dihitung mana trayek yang terbaik untuk pengguna pilih dengan menggunakan *Analytical 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.

1.2 Rumusan Masalah

Berdasarkan pokok permasalahan yang telah diuraikan di atas, maka ditentukan rumusan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana mengimplementasikan *Graph Database* untuk menemukan rute angkutan umum?
2. Seberapa akurat *Graph Database* dapat menyelesaikan pencarian rute angkutan umum?

1.3 Tujuan Penelitian

Dalam penelitian yang dilakukan, tidak lepas dari sebuah tujuan yang ingin dicapai. Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan *Graph Database* untuk menemukan rute angkutan umum.
2. Mengetahui akurasi *Graph Database* dalam menyelesaikan pencarian rute angkutan umum.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

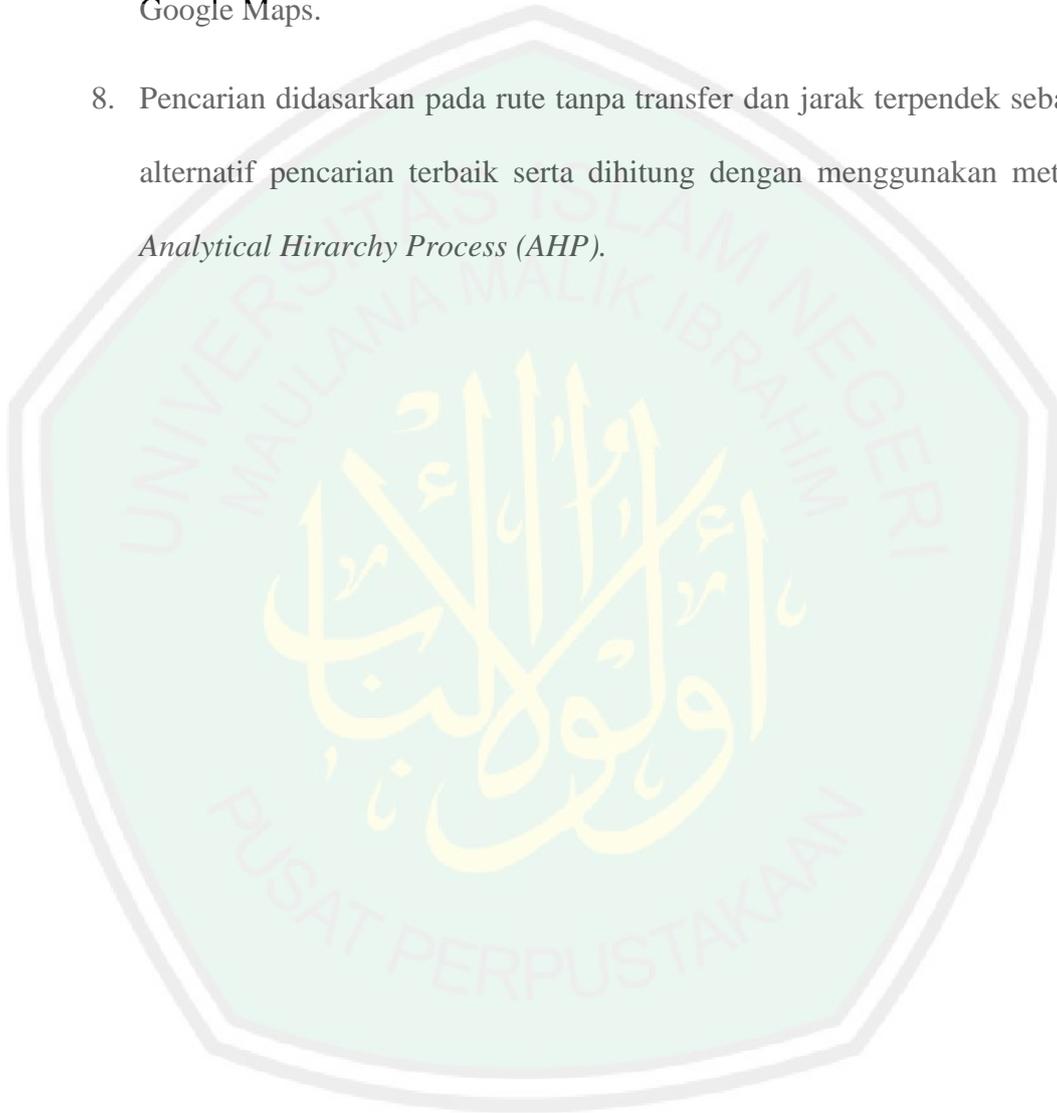
1. Memudahkan pengguna angkot dalam mengetahui trayek yang harus dipakai ketika bepergian ke suatu tempat.
2. Membuka kajian lebih lanjut tentang penerapan *Graph Database* untuk menyelesaikan masalah di masyarakat.

1.5 Batasan Masalah

Untuk menghindari pembahasan yang keluar dari materi, maka ditentukan batasan masalah. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan di kota Malang dan kota Batu
2. Menggunakan *Graph Database Neo4j*
3. Menggunakan *Cypher* dalam pengoperasian logika di *database server*
4. Penelitian ini menyangkup hingga 2 kali transfer atau maksimal 3 transportasi umum dalam satu kali pencarian rute perjalanan.
5. Transportasi Umum yang digunakan Angkutan Kota atau Angkot (kota Malang dan kota Batu)

6. Menggunakan *platform* android sebagai *client side* yang dibangun menggunakan Android Studio.
7. Penentuan setiap lokasi menggunakan titik koordinat yang didapatkan dari Google Maps.
8. Pencarian didasarkan pada rute tanpa transfer dan jarak terpendek sebagai alternatif pencarian terbaik serta dihitung dengan menggunakan metode *Analytical Hierarchy Process (AHP)*.



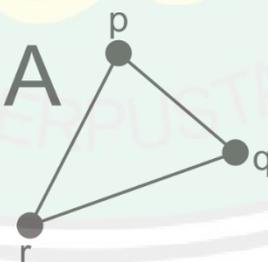
BAB II

LANDASAN TEORI

2.1 Pengertian *Graph*

Secara formal, *Graph* merupakan kumpulan simpul (*node*) dan tepi (*edge*) yang menghubungkannya. *Graph* mewakili entitas sebagai simpul dan cara di mana entitas tersebut berhubungan dengan dunia sebagai hubungan. Struktur ekspresif tujuan umum ini memungkinkan kita untuk memodelkan semua jenis skenario, mulai dari konstruksi roket ruang angkasa, hingga sistem jalan, dan dari rantai pasokan atau asal makanan, riwayat medis untuk populasi, dan seterusnya. (Ian Robinson, 2015).

Menurut Jonathan L. Gross (2006), *Graph* $G = (V, E)$ merupakan struktur matematika yang konsisten antara *Vertices* (atau *Nodes*) dan *Edge* yang terbatas dan setiap *edge* memiliki satu atau dua kumpulan *node* yang saling terhubung dan disebut dengan endpoints.



Gambar 2.1 Perpaduan Antara *Edge* dan *Node* Dalam *Graph* A

Pada Gambar 2.1, ada sebuah *Graph* yang berisi 3 buah *node* dan 3 buah *edge*. Dengan $G = (V, E)$, dapat dijabarkan bahwa $V_A = \{p, q, r\}$ dan $E_A = \{pq, qr, rp\}$. *Graph* A memiliki 3 buah *node* (atau *vertices*) dan juga 3 buah *edge* yang saling menghubungkan antara satu *vertices* ke *vertices* yang lain. *Edge* bertugas untuk

menghubungkan setiap *vertices* dan menentukan apakah maju atau mundur dari setiap *vertices* dalam penggambarannya menggunakan tanda panah diujung *edge* tersebut.

2.2 Property Graph

Property Graph adalah jenis *Graph* yang memiliki *node* dan relasi, dan setiap *node* memiliki propertinya masing-masing. *Node* juga dapat memiliki satu label atau lebih. Relasi-relasi didalam *Property Graph* memiliki nama masing-masing dan memiliki titik *start* dan *end*. Setiap relasi juga memiliki *Property* masing-masing. (Ian Robinson, 2015).

Pada pembahasan *Graph Database* disini akan dilakukan dengan mengaplikasikan jenis *Property Graph*. Setiap *node* dan *edge* memiliki properti masing-masing yang mengimplementasikan lokasi maupun trayek. Beberapa susunan *graph* akan dibahas di Bab III.

2.3 Resource Description Framework (RDF)

Resource Description Framework atau disingkat *RDF* adalah suatu kerangka kerja umum untuk menggambarkan berbagai sumber daya di dalam sebuah sistem web dan isinya dengan mendeskripsikan hubungan antara sumber daya yang ada didalamnya. (Ian Robinson, 2015).

Menurut Rezo Angles (2005) *RDF* dalam *Graph Database* bisa dilihat dari 2 perspektif: (1) Dari *logical perspective*, yang sebagai fragmen logika minimal yang mencakup semua fitur relevan yang diperlukan sebagai bahasa representasi untuk metadata; dan (2) Dari perspektif basis data, sebagai perpanjangan dari model

data yang digunakan dalam komunitas basis data, khususnya model basis data grafik. Sudut pandang sebelumnya telah menjadi bidang penelitian yang aktif.

2.4 HyperGraph

HyperGraph adalah model grafik umum yang dapat menghubungkan sejumlah *node*. Berbeda dengan model grafik properti yang memungkinkan suatu hubungan untuk hanya memiliki satu simpul awal dan satu simpul akhir, model *HyperGraph* memungkinkan sejumlah simpul di salah satu ujung hubungan (*edge*). (Ian Robinson, 2015).

Penggunaan *Graph Database* sudah mulai meluas. Banyaknya data pengguna yang saling terhubung setidaknya dapat menjadi contoh. Bagaimana setiap pengguna dapat terhubung dengan rekan, teman, kerabat, atau pernah bersekolah di lembaga yang sama, juga memiliki beberapa kesamaan yang didalam *Graph* menjadi properti yang saling berhubungan antara satu dan lainnya. Begitu juga beberapa media sosial besar yang menampung banyak data pengguna dan cenderung tidak bisa diatasi hanya dengan *RDBMS (Relational Database Management System)* seperti MySQL, Postgree SQL, dan lain sebagainya.

2.5 Graph Compute Engine

Graph Compute Engine merupakan sebuah mesin yang berfungsi untuk mengatasi berbagai macam operasi yang dibutuhkan dalam pengelolaan *Graph Database* seperti mengidentifikasi sekian banyak kelompok data, memproses perhitungan relasi yang ada atau untuk menghitung berapa banyak simpul dan *Property* yang terkandung dalam sebuah *Graph Database*. (Ian Robinson, 2015).

2.6 Graph Database

Graph Database Management System 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. (Ian Robinson, 2015)

Graph Database adalah teknologi database yang cara penyimpanan datanya seperti pada *Graph* yang terdiri atas kumpulan *edge* dan *vertex* yang dapat diakses secara langsung melalui aplikasi dan mendukung OLTP. (Robinson, Webber, & Eifrem, 2013)

Graph Database adalah sebuah database yang menggunakan struktur *Graph* seperti *node*, *edge*, dan *property* untuk menampilkan dan menyimpan data. (Taft, 2014)

Kebanyakan *Graph Database* menggunakan penyimpanan *database native* yang dirancang khusus untuk skalabilitas dan performa, namun masih terkendala dengan jarangnyanya yang sudah memahami konsep dan penerapannya. Berbeda dengan database *non-native* dan juga *non-Graph Database* seperti *Relational Databases Management System (RDBMS)* yang memiliki karakteristik atau bahasa *query* yang mudah difahami oleh kebanyakan pengembang. Dalam bahasa lainnya, bahwa *Graph Database* memiliki performa yang jauh lebih baik, tetapi memiliki tingkat pemahaman *query* yang lebih sulit.

2.6.1 Kelebihan *Graph Database*

Menurut Ian Robinson (2015), *Graph Database* memiliki beberapa kelebihan, diantaranya:

A. Performa

Graph Database mempunyai peningkatan kinerja dibandingkan dengan *Relational Database* apalagi dalam memproses data yang begitu banyak dan saling berhubungan. *Graph Database* memproses hanya untuk *node* dan *edge* yang terhubung dan tidak memproses keseluruhan data *Graph* yang ada.

B. Fleksibilitas

Graph Database sangat berperan dalam sebuah industri IT yang terus berkembang. Ini dikarenakan bahwa *Graph Database* bisa dikelola dan dikembangkan dengan leluasa, menambah *node*, *edge*, relasi dan lain sebagainya dengan tanpa mengganggu setiap data, fungsi, serta kueri yang sudah ada.

C. Ketangkasan Mengolah Data

Graph Database memungkinkan integrasi dari berbagai aplikasi yang lebih terkendali dan tertata. Sifatnya yang bebas skema, tidak seperti relational database, membuatnya mudah dalam penggunaan data dalam sistem yang berbeda.

2.7 *Weighted Directed Acyclic Graphs (WDAG)*

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 sudah memiliki urutan.

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.

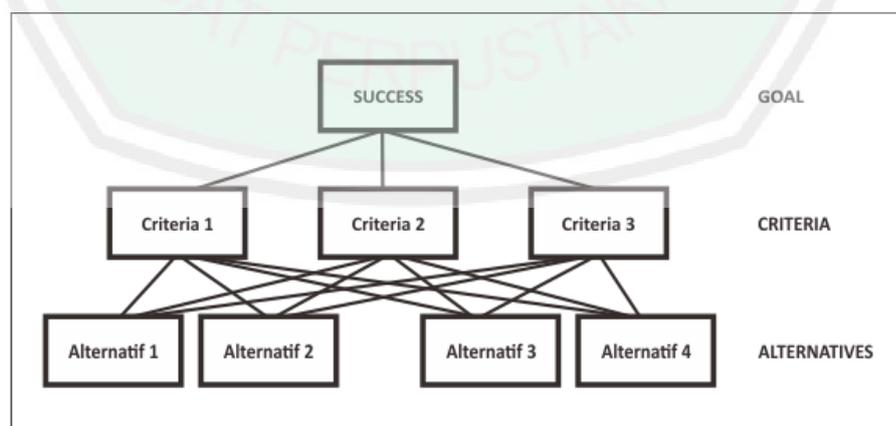
DAG dapat memodelkan berbagai jenis informasi. *Spreadsheet* dapat dimodelkan sebagai *DAG*, dengan *node* untuk masing-masing sel dan *edge* setiap kali rumus dalam satu sel menggunakan nilai dari yang lain. Pengurutan topologi *DAG* ini dapat digunakan untuk memperbarui semua nilai sel saat spreadsheet diubah. Demikian pula, urutan topologi *DAG* dapat digunakan untuk memesan operasi kompilasi dalam sebuah file. Teknik evaluasi dan peninjauan program menggunakan *DAG* untuk memodelkan tonggak dan kegiatan proyek-proyek besar, dan menjadwalkan proyek-proyek ini untuk digunakan semaksimal mungkin. Blok logika kombinasional dalam desain sirkuit elektronik, dan operasi dalam bahasa pemrograman aliran data, melibatkan asiklik pemrosesan jaringan elemen. *DAG* juga dapat mewakili koleksi peristiwa dan pengaruhnya satu sama lain, baik dalam struktur probabilistik seperti jaringan Baye atau sebagai catatan data historis seperti

pohon keluarga atau sejarah versi sistem kontrol yang terdistribusi. *DAG* juga dapat digunakan sebagai representasi data sekuens yang ringkas, seperti representasi grafik asiklik langsung dari kumpulan *string*, atau representasi diagram keputusan *biner* dari urutan pilihan *biner*. Lebih abstrak, hubungan *reachability* dalam *DAG* membentuk urutan parsial, dan urutan parsial yang terbatas dapat diwakili oleh *DAG* menggunakan *reachability*.

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.

2.8 Analytical Hierarchy Process (AHP)

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. Struktur hirarki AHP ditampilkan pada Gambar 2.2.



Gambar 2.2 Struktur Hirarki AHP

Pada hakekatnya AHP merupakan suatu model pengambil keputusan yang komprehensif dengan memperhitungkan hal-hal yang bersifat kualitatif dan kuantitatif. Dalam model pengambilan keputusan dengan AHP pada dasarnya berusaha menutupi semua kekurangan dari model-model sebelumnya. AHP juga memungkinkan ke struktur suatu sistem dan lingkungan kedalam komponen saling berinteraksi dan kemudian menyatukan mereka dengan mengukur dan mengatur dampak dari komponen kesalahan sistem (Saaty, 2001)

Keuntungan yang diperoleh bila seseorang memecahkan masalah dan mengambil keputusan menggunakan AHP antara lain (Saaty, 1993):

1. AHP memberi satu model tunggal yang mudah dimengerti, luwes untuk keanekaragaman persoalan tak terstruktur.
2. AHP memadukan ancangan deduktif dan ancangan berdasarkan sistem dalam memecahkan persoalan kompleks.
3. AHP dapat menangani saling ketergantungan elemen-elemen dalam suatu sistem dan tak memaksakan pemikirang linear.
4. AHP mencerminkan kecenderungan alami pikiran untuk memilah-milah elemen-elemen suatu sistem dalam berbagai tingkat berlainan dan mengelompokkan unsur yang serupa dalam setiap tingkat.
5. AHP memberi suatu skala untuk mengukur halhal dan mewujudkan metode penetapan prioritas.
6. AHP melacak konsistensi logis dan pertimbangan-pertimbangan yang digunakan dalam menggunakan berbagai prioritas.

7. AHP menuntun ke suatu taksiran menyeluruh tentang kebaikan sistem alternatif.
8. AHP mempertimbangkan prioritas relatif dari berbagai faktor sistem dan memungkinkan organisasi memilih alternatif terbaik berdasarkan tujuan-tujuan mereka.
9. AHP tidak memaksakan konsensus tetapi mensintesiskan suatu hasil yang representatif dari berbagai penilaian.
10. AHP memungkinkan organisasi memperhalus definisi mereka pada suatu persoalan dan memperbaiki pertimbangan serta pengertian mereka melalui pengulangan.

Prinsip kerja AHP adalah penyederhanaan suatu persoalan kompleks yang tidak terstruktur, strategik, dan dinamik menjadi bagian-bagiannya, serta menata dalam suatu hierarki. Kemudian tingkat kepentingan setiap variabel diberi nilai numerik secara subjektif tentang arti penting variabel tersebut secara relatif dibandingkan dengan variabel lain. Dari berbagai pertimbangan tersebut kemudian dilakukan sintesa untuk menetapkan variabel yang memiliki prioritas tinggi dan berperan untuk mempengaruhi hasil pada sistem tersebut. Menurut Saaty (1993), terdapat tiga prinsip dalam memecahkan persoalan dengan AHP, yaitu prinsip menyusun hirarki (*Decomposition*), prinsip menentukan prioritas (*Comparative Judgement*), dan prinsip konsistensi logis (*Logical Consistency*).

Terdapat 4 aksioma-aksioma yang terkandung dalam model AHP (Saaty, 2001):

1. Reciprocal Comparison artinya pengambilan keputusan harus dapat memuat perbandingan dan menyatakan preferensinya. Preferensi tersebut harus memenuhi syarat resiprokal yaitu apabila A lebih disukai daripada B dengan skala x , maka B lebih disukai daripada A dengan skala $1/x$.
2. Homogeneity artinya preferensi seseorang harus dapat dinyatakan dalam skala terbatas atau dengan kata lain elemen-elemennya dapat dibandingkan satu sama lainnya. Kalau aksioma ini tidak dipenuhi maka elemen-elemen yang dibandingkan tersebut tidak homogen dan harus dibentuk cluster (kelompok elemen) yang baru.
3. Independence artinya preferensi dinyatakan dengan mengasumsikan bahwa kriteria tidak dipengaruhi oleh alternatif-alternatif yang ada melainkan oleh objektif keseluruhan. Ini menunjukkan bahwa pola ketergantungan dalam AHP adalah searah, maksudnya perbandingan antara elemen-elemen dalam satu tingkat dipengaruhi atau tergantung oleh elemen-elemen pada tingkat di atasnya.
4. Expectation artinya untuk tujuan pengambil keputusan. Struktur hirarki diasumsikan lengkap. Apabila asumsi ini tidak dipenuhi maka pengambil keputusan tidak memakai seluruh kriteria atau objektif yang tersedia atau diperlukan sehingga keputusan yang diambil dianggap tidak lengkap.

2.9 Penelitian Terkait

Chao-lin Liu (2002) menerapkan algoritma *best-path planning* untuk menentukan rute perjalanan transportasi umum. Menggunakan matriks transisi untuk menemukan rute yang dibutuhkan dalam suatu perjalanan.

Panji dkk (2016) meneliti tingkat kecepatan *Graph Database* yang dibandingkan dengan *Relational Database* dalam mengolah data rute perjalanan *Bus Rapid Transit (BRT)* yang dilaksanakan di Kota Semarang. Penelitian ini membuktikan bahwa model *Graph Database* sangat cocok untuk mengolah data rute BRT karena mampu mengolah dengan waktu permintaan *query* lebih cepat dibandingkan dengan *Relational Database MySQL* yang mengandalkan banyak tabel dan dikonekiskan menggunakan JOIN.

Panji, Djalal (2017) melanjutkan penelitiannya terkait dengan *Graph Database*, masih tentang BRT, yang memungkinkan pencarian rute BRT dengan dua kali transit. Namun, disini masih terbatas hanya dengan satu jenis transportasi, yaitu BRT dan belum dibahas terkait jenis transportasi lain seperti angkutan umum, bis, dan kereta yang juga memiliki peran menjadi fasilitas masyarakat secara luas dan juga tidak ada parameter bobot dalam setiap relasi yang dibangun.

Imdad Rabbani (2017) menggunakan algoritma *best-path planning* untuk menemukan rute angkutan kota (angkot) di Kota Malang. Dalam penelitian ini masih menggunakan *database relational* berupa SQLite dan menggunakan *client-side* berupa *platform* android. Penelitian ini masih terbatas dalam tiga rute angkot dan hanya bisa memproses maksimal satu kali transfer.

Aji Sasongko (2017) menggunakan Algoritma AHP untuk menentukan pemilihan karyawan baru. Dalam penelitian tersebut melakukan pembobotan terhadap kriteria dan pelamar. Hasil penelitian berupa aplikasi sistem pemilihan karyawan baru berbasis web yang memberikan rekomendasi sebagai bahan pertimbangan untuk mengambil keputusan secara tepat dan diharapkan dapat mempermudah proses seleksi karyawan baru.

Dalam penelitian yang saya laksanakan, ditawarkan sebuah model penyelesaian pencarian rute transportasi umum dengan menggunakan *Graph Database* Neo4j dan mengimplementasikan metode AHP. Dengan ini, diharapkan akan dapat menjadikan sebuah sistem yang bisa menampilkan alternatif rute transportasi umum dengan lebih akurat dan cepat, bahkan jika data yang dimasukkan terlalu banyak. Di penelitian ini juga ditawarkan sampai dengan maksimal 2 kali transfer yang berarti memiliki kemungkinan jalur yang lebih kompleks.

2.10 Android

Android merupakan sebuah platform mobile yang berbasis linux dan merupakan system operasi sumber terbuka (*Open Source*). Pengguna android semakin bertambah setiap tahunnya.

Gambar 2.3 adalah sebuah survey yang dilaksanakan oleh Hootsuite yang merupakan sebuah situs layanan manajemen konten yang menyediakan layanan media daring yang juga terhubung dengan berbagai situs jejaring sosial lainnya, seperti Facebook, Twitter, LinkedIn, Foursquare, MySpace, dan WordPress, menyebutkan bahwa populasi pengguna perangkat mobile angkanya lebih tinggi lagi yang mencapai 177,9 juta pengguna, dengan tingkat penetrasi mencapai 67%.



Gambar 2.3 Data Pengguna Internet dan Perangkat Mobile di Indonesia Menurut Survei Hootsuite

Android sebagai salah satu sistem operasi yang sangat populer di dunia *mobile device* sudah memberdayakan jutaan perangkat di lebih dari 190 negara di seluruh dunia. Bahkan, setiap hari lebih dari 1 juta perangkat berbasis android baru aktif di seluruh penjuru dunia dan lebih dari 1.5 miliar total unduhan perbulan dari *Google Play* dan terus bertambah.

Indonesia sekarang, menurut data dari *Google Developer*, merupakan urutan ke 8 yang mempunyai pengguna internet aktif dan 100 juta merupakan pengguna *smartphone*. Ini merupakan bukti nyata bagaimana urgensi perangkat mobile didalam setiap kehidupan masyarakat zaman milenial ini.

BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Dalam penelitian ini desain penelitian menjelaskan tentang mekanisme yang digunakan. Proses dilakukan dengan dua tahap, tahap pertama yaitu penentuan jenis penelitian dan tahap kedua yaitu prosedur penelitian. Adapun uraian dari setiap tahap penelitian dijelaskan pada sub pembahasan dibawah ini.

3.2 Jenis Penelitian

Penelitian ini merupakan penelitian kuantitatif yang dilakukan untuk mengoptimalkan pencarian rute transportasi umum dengan menggunakan *Graph Database* dan implementasi metode *Analytic Hierarchy Process (AHP)*.

3.3 Prosedur Penelitian

Dalam prosedur penelitian dijelaskan bagaimana system akan dikerjakan, mulai dari penentuan masalah hingga menemukan solusi dari permasalahan yang ditentukan yaitu pencarian rute transportasi umum sebagai berikut:

- Mulai
- Identifikasi Masalah
- Studi Literatur
- Menentukan *Input* dan *Output* Dalam Aplikasi
- Perancangan Sistem
- Pengumpulan Data

- Pemodelan Struktur Data *Graph Database*
- Penerapan AHP dan Pengujian sistem
- Penarikan kesimpulan
- Selesai.

3.3.1 Identifikasi Masalah

Penjelasan terkait dengan penentuan masalah sebelumnya sudah dibahas di BAB I, yaitu bagaimana menghasilkan pencarian rute transportasi umum di Kota Malang dan Batu dengan *Graph Database* yang dioptimasi menggunakan *Weighted Directed Acyclic Graphs (WDAG)* serta *Analytic Hierarchy Process (AHP)*

3.3.2 Studi Literatur

Studi literature mengenai Aplikasi Pencarian rute transportasi umum di Kota Malang dan Batu Berbasis Android Menggunakan *Graph Database* dan Metode *Analytic Hierarchy Process (AHP)* sudah dijelaskan di BAB II.

3.3.3 Menentukan Input dan Output Dalam Aplikasi

Input dan Output dalam sebuah sistem adalah faktor yang sangat penting yang perlu dijelaskan secara rinci. Penggunaan input yang berbeda dengan proses yang diolah sedemikian rupa menyebabkan output yang dihasilkanpun bisa berbeda. Dalam aplikasi pencarian rute angkot ini, juga memerlukan input dari pengguna yang nantinya output yang dihasilkan digunakan kembali oleh pengguna untuk mengetahui rute trayek perjalanan yang akan dilalui.

A. Input

Input yang diperlukan adalah data posisi pengguna (asal) yang didapat melalui tempat-tempat yang sudah terdata didalam *Graph Database*. Pengguna tinggal mengakses data lokasi yang sudah dimasukkan kedalam Graph Database dan menentukan darimana serta kemana dia akan bepergian. Pengguna juga bisa dengan mudah menggunakan fitur pencarian untuk menemukan lokasi yang dikehendaki. Input berupa data teks yang merepresentasikan properti dari setiap lokasi dalam sebuah node, untuk kemudian menjadi parameter pencarian dalam penentuan rute perjalanan transportasi umum dalam hal ini angkot.

B. Output

Output merupakan keluaran yang dihasilkan dari input yang sudah diproses. Dalam pencarian rute transportasi ini, keluarannya adalah alternatif trayek angkot yang dapat digunakan untuk pengguna mencapai suatu tempat yang dimasukkan.

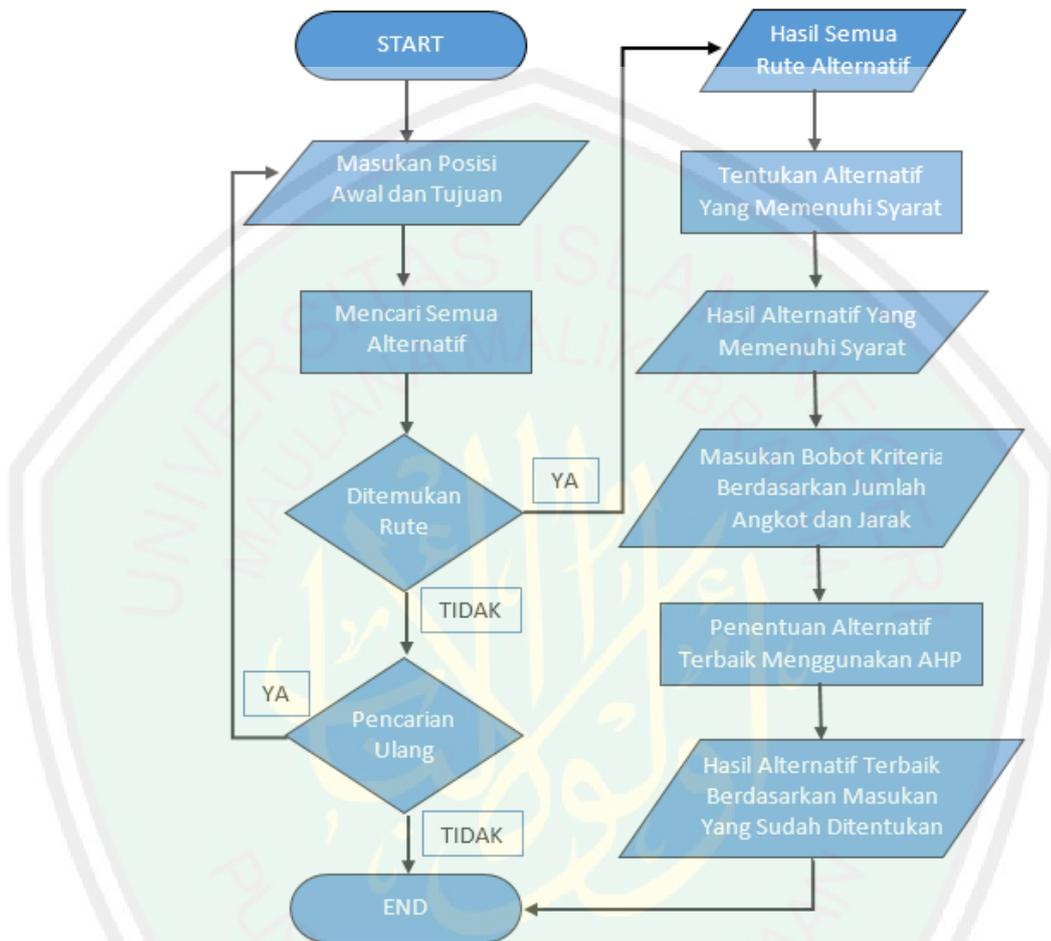
3.3.4 Perancangan Sistem

Ketika akan membangun sebuah sistem, dibutuhkan sebuah rancangan untuk memudahkan langkah demi langkah yang dilakukan. Berikut adalah perancangan sistem yang akan dibangun.

A. Flowchart

Dalam sistem ini, pengguna memasukan lokasi awal dan lokasi tujuan dan kemudian didapatkan semua alternatif rute yang dihasilkan untuk selanjutnya dicari alternatif mana yang terbaik

berdasarkan nilai bobot kriteria menggunakan AHP. Flowchart dijelaskan pada Gambar 3.1.



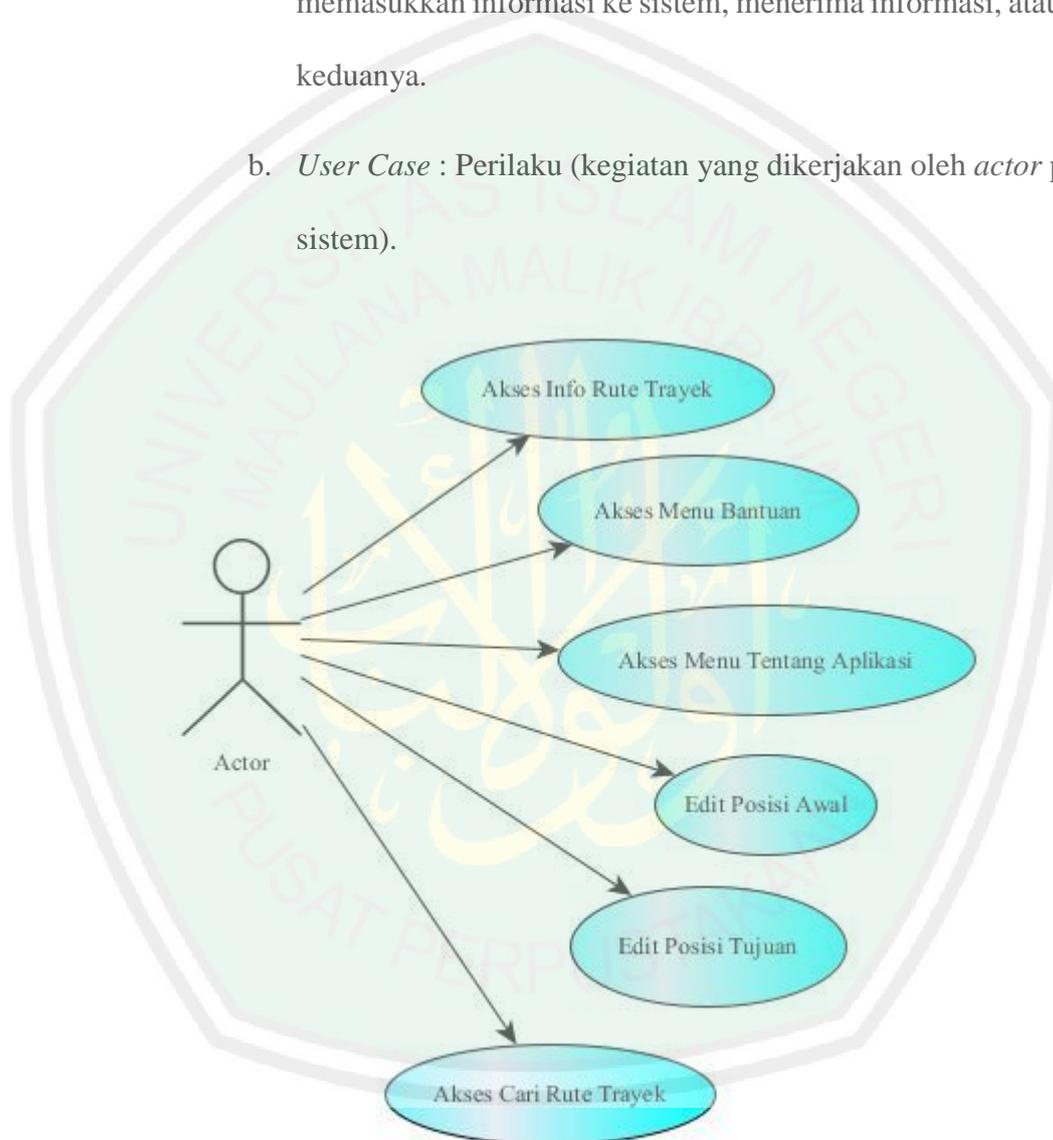
Gambar 3.1 Flowchart Sistem Yang Dibangun

B. Usecase Diagram

Usecase Diagram berguna untuk mengetahui perilaku apa saja yang berjalan dalam sebuah sistem dan actor siapa saja yang berperan didalamnya. Berikut adalah gambaran umum *use case diagram* dari sistem pencarian rute transportasi umum yang akan

dibuat dijelaskan dalam gambar 3.2 serta komponen-komponen dibawah ini. Komponen *use case diagram* :

- a. *Actor* : Pengguna *software* aplikasi. Aktor atau pengguna dapat memasukkan informasi ke sistem, menerima informasi, ataupun keduanya.
- b. *User Case* : Perilaku (kegiatan yang dikerjakan oleh *actor* pada sistem).



Gambar 3.2 Usecase Diagram

3.3.5 Pengumpulan Data

Langkah selanjutnya adalah pengumpulan data. Data yang digunakan adalah data jalur atau trayek angkot di daerah Kota Malang dan Kota Batu. Data

didapatkan dari Dinas Perhubungan kota terkait dan dirangkum dalam sebuah tabel, yang kemudian diolah menjadi suatu data didalam *Graph Database*. Tabel 3.1, Tabel 3.2, Tabel 3.3, Tabel 3.4, merupakan data trayek yang diperoleh dari Dinas Perhubungan setempat.

Tabel 3.1 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 3.2 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

Tabel 3.3 Data Trayek Menurut Urutan Jalan dan Terminal di Kota Malang

AL - Arjosari	GA - Gadang	LG - Landungsari
Terminal Arjosari	Terminal Gadang	Terminal Landungsari
Jl. R. Panji Suroso	Jl. Satsuit Tubun	Jl. Tlogo Mas
Jl. Laksda Adi Sucipto	Jl. S. Supriyadi	Jl. Mayjen MT. Haryono
Jl. Tenaga	Jl. Arif Margono	Jl. Sumbersari
Jl. Karya Timur	Jl. Ade Irma Suryani	Jl. Bendungan Sutami
Jl. Mahakam	Jl. KH. Wahid Hasyim	Jl. Surabaya
Jl. W. R. Supratman	Jl. Kauman	Jl. Bondowoso
Jl. Panglima Sudirman	Jl. KH. Hasyim Asy'ari	Jl. Gading
Jl. Patimura	Jl. AR. Hakim	Jl. Wilis
Jl. Trunojoyo	Jl. Merdeka Utara	Jl. Panderman
Jl. Kertanegara	Jl. S. Wiryopranoto	Jl. AR. Hakim
Jl. Tugu	Jl. Mojopahit	Jl. Merdeka Utara
Jl. Kahuripan	Jl. Tugu	Jl. Merdeka Selatan
Jl. Semeru	Jl. Untung Suropati	Jl. Wiryopranoto
Jl. Ijen	Jl. Pajajaran	Jl. Sultan Syahrir
Jl. Retawu	Jl. Trunojoyo	Jl. Kyai Tamin
Jl. Bondowoso	Jl. Cokroaminoto	Jl. Sartono SH.
Jl. Jombang	Jl. Dr. Cipto	Jl. Peltu Sujono
Jl. Surabaya	Jl. Panglima Sudirman	Jl. Susanto
Jl. Jakarta	Jl. WR. Supratman	Jl. Niaga
Jl. Bogor	Jl. Letjend Sutoyo	Jl. Sonokeling
Jl. Veteran	Jl. Letjend S. Parman	Jl. Janti
Jl. Sumbersari	Jl. Jend. A. Yani	Jl. S. Supriyadi
Terminal Landungsari	Jl. R. Intan	Jl. Satsuit Tubun
	Terminal Arjosari	Terminal Gadang

Tabel 3.4 Data Trayek Menurut Urutan Jalan dan Terminal di Kota Batu

BSS - Terminal Batu	BJL - Terminal Batu	BNK - Terminal Batu
Terminal Batu	Terminal Batu	Terminal Batu
Jl. Dewi Sartika	Jl. Dewi Sartika	Jl. Dewi Sartika
Jl. Agus Salim	Oro-oro Ombo	Jl. Agus Salim
Jl. Sultan Agung	Junrejo	Jl. Sultan Agung
Jl. Ahmad Yani	Dau	Jl. Suropati
Jl. Brantas	Jl. Sengkaling	Jl. Hasanudin
Selecta	Terminal Landungsari	Jl. Trunojoyo Batu
Jl. Junggo		Pujon
Sumber Brantas		Ngantang
		Kasembon

Perlu ada penyederhanaan sebelum dimasukkan kedalam *Graph Database*, karena satu koordinat hanya untuk satu *node*. Di setiap *node* juga diberikan sebuah informasi berupa koordinat yang didapatkan dari *Google Maps*. Penyederhanaan seluruh tempat yang dilalui trayek tersebut ada dalam Tabel 3.5, Tabel 3.6, Tabel 3.7 Tabel 3.8, Tabel 3.9, Tabel 3.10.

Tabel 3.5 Data Lokasi dan Koordinat Kota Malang

No	Nama Jalan / Tempat	Koordinat
1	Jl. Ade Irma Suryani	-7.984336, 112.628573
2	Jl. AR. Hakim	-7.981227, 112.628918
3	Jl. Arif Margono	-7.985423, 112.624119
4	Jl. Bandung	-7.961025, 112.623447
5	Jl. Bendungan Sengguruh	-7.958841, 112.609133
6	Jl. Bendungan Sigura-gura	-7.956887, 112.605740
7	Jl. Bendungan Sutami	-7.960327, 112.613636

Tabel 3.6 Data Lokasi dan Koordinat Kota Malang Lanjutan

No	Nama Jalan / Tempat	Koordinat
8	Jl. Bogor	-7.960183, 112.621454
9	Jl. Bondowoso	-7.967468, 112.615395
10	Jl. Cokroaminoto	-7.970506, 112.636458
11	Jl. Dr. Cipto	-7.968165, 112.635045
12	Jl. Gading	-7.971734, 112.614207
13	Jl. Gajayana	-7.947007, 112.608751
14	Jl. Halmahera	-7.952966, 112.593316
15	Jl. Ijen	-7.974538, 112.621070
16	Jl. Indragiri	-7.958666, 112.640034
17	Jl. Jakarta	-7.964008, 112.620939
18	Jl. Janti	-8.001687, 112.622487
19	Jl. Jend. A. Yani	-7.938672, 112.643702
20	Jl. Jombang	-7.966368, 112.615631
21	Jl. Kahuripan	-7.976589, 112.631553
22	Jl. Kapten Piere Tendean	-7.987010, 112.629465
23	Jl. Karya Timur	-7.950565, 112.641174
24	Jl. Kauman	-7.982822, 112.628361
25	Jl. Kawi	-7.978266, 112.624018
26	Jl. Kertanegara	-7.977428, 112.635222
27	Jl. KH. Hasyim Asy'ari	-8.138307, 112.680309
28	Jl. KH. Wahid Hasyim	-7.983888, 112.629006
29	Jl. Kyai Tamin	-7.988202, 112.632370
30	Jl. Laksamana Adi Sucipto	-7.943530, 112.647718
31	Jl. Letjend S. Parman	-7.950148, 112.639593
32	Jl. Letjend Sutoyo	-7.960288, 112.637030
33	Jl. Mahakam	-7.964102, 112.636753
34	Jl. Mayjen MT. Haryono	-7.937168, 112.606123
35	Jl. Merdeka Selatan	-7.983429, 112.630762

Tabel 3.7 Data Lokasi dan Koordinat Kota Malang Lanjutan

No	Nama Jalan / Tempat	Koordinat
36	Jl. Merdeka Utara	-7.981737, 112.630870
37	Jl. Mojopahit	-7.979980, 112.630977
38	Jl. Niaga	-7.998085, 112.627427
39	Jl. Nusa Kambangan	-7.988073, 112.626610
40	Jl. Pajajaran	-7.975377, 112.635598
41	Jl. Panderman	-7.974817, 112.619739
42	Jl. Panglima Sudirman	-7.972485, 112.638529
43	Jl. Patimura	-7.973420, 112.637016
44	Jl. Peltu Sujono	-7.996088, 112.629959
45	Jl. R. Intan	-7.930977, 112.652767
46	Jl. R. Panji Suroso	-7.930851, 112.650440
47	Jl. Retawu	-7.970533, 112.619590
48	Jl. S. Supriyadi	-8.005926, 112.618485
49	Jl. S. Wiryopranoto	-7.984159, 112.631173
50	Jl. Sartono SH.	-7.993087, 112.631987
51	Jl. Satsuit Tubun	-8.021713, 112.623227
52	Jl. Semeru	-7.974443, 112.625319
53	Jl. Simpang Panji Suroso	-7.934308, 112.654540
54	Jl. Sonokeling	-8.000359, 112.626767
55	Jl. Sultan Syahrir	-7.986210, 112.630717
56	Jl. Sumbersari	-7.954281, 112.610630
57	Jl. Surabaya	-7.965930, 112.617936
58	Jl. Susanto	-7.996140, 112.628294
59	Jl. Tanimbar	-7.990242, 112.627243
60	Jl. Tenaga	-7.944168, 112.643513
61	Jl. Terusan Halmahera	-7.991762, 112.628570
62	Jl. Tlogomas	-7.926371, 112.601617
63	Jl. Trunojoyo	-7.976217, 112.636745

Tabel 3.8 Data Lokasi dan Koordinat Kota Malang Lanjutan

No	Nama Jalan / Tempat	Koordinat
64	Jl. Tugu	-7.977615, 112.634477
65	Jl. Untung Suropati	-7.978808, 112.635409
66	Jl. Veteran	-7.957378, 112.617592
67	Jl. Wilis	-7.972426, 112.620458
68	Jl. Wiryopranoto	-7.984014, 112.631242
69	Jl. Wonogiri	-7.960625, 112.610380
70	Jl. WR. Supratman	-7.965738, 112.637059
71	Terminal Arjosari	-7.934936, 112.658836
72	Terminal Gadang	-8.022361, 112.627851
73	Terminal Landungsari	-7.925288, 112.597926

Tabel 3.9 Data Lokasi dan Koordinat Kota Batu

No	Nama Lokasi	Koordinat
1	Dau	-7.923176, 112.581772
2	Jl. Agus Salim	-7.876931, 112.527954
3	Jl. Ahmad Yani	-7.871537, 112.523501
4	Jl. Brantas	-7.866579, 112.525592
5	Jl. Dewi Sartika	-7.882797, 112.531722
6	Jl. Hasanudin	-7.907530, 112.548239
7	Jl. Junggo	-7.961445, 112.634157
8	Jl. Sengkaling	-7.915877, 112.587233
9	Jl. Sultan Agung	-7.879101, 112.523320
10	Jl. Suropati	-7.873458, 112.516606
11	Jl. Trunojoyo Batu	-7.901091, 112.557170
12	Junrejo	-7.911258, 112.554083
13	Kasembon	-7.782794, 112.327762
14	Ngantang	-7.876588, 112.388967
15	Oro-oro Ombo	-7.901278, 112.535847

Tabel 3.10 Data Lokasi dan Koordinat Kota Batu Lanjutan

No	Nama Lokasi	Koordinat
16	Pujon	-7.843780, 112.447524
17	Selecta	-7.821719, 112.526469
18	Sumber Brantas	-7.754838, 112.531369
19	Terminal Batu	-7.882385, 112.533185
20	Terminal Landungsari	-7.925288, 112.597926
21	Jl. Abdul Gani Batu	-7.876404, 112.520812
22	Jl. Bromo Batu	-7.867650, 112.526431
23	Jl. Diponegoro Batu	-7.875744, 112.531277
24	Jl. Ikhwan Hadi Batu	-7.871661, 112.520370
25	Jl. Imam Bonjol Batu	-7.879913, 112.532078
26	Jl. P. Sudirman Batu	-7.868201, 112.517371
27	Jl. Semeru Batu	-7.869074, 112.527907

3.3.6 Pemodelan Struktur Data *Graph Database*

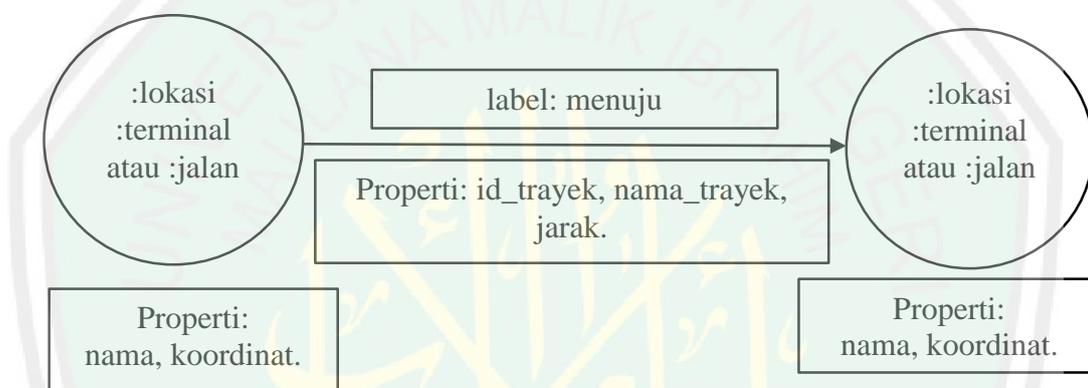
Data yang sudah didapatkan berupa data trayek dari setiap angkot, perlu dimasukkan kedalam *Graph Database* melalui beberapa cara, yaitu:

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

Skema *Graph Database* yang dibuat adalah sebagai berikut pada Tabel 3.11 serta visualisasi dalam Gambar 3.3.

Tabel 3.11 Skema Graf

No	Node/Edge	Label	Properti	Keterangan
1	Node	:lokasi :terminal	nama koordinat	Untuk membuat titik berupa terminal
2	Node	:lokasi :jalan	nama koordinat	Untuk membuat titik berupa jalan
3	Edge	:menuju	id_trayek nama_trayek jarak	Untuk membuat relasi antar node dalam satu trayek



Gambar 3.3 Skema Graf Yang Dibangun

Untuk merepresentasikan setiap lokasi yang ada, dibuatlah *node-node* dengan *Cypher Query* menggunakan klausa *CREATE* dengan *property* seperti dijelaskan di Tabel 3.11. Sebagai contoh dibuatkan satu *node* yang mewakili Terminal Arjosari dengan *cypher* pada Gambar 3.4 sebagai berikut:

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

Gambar 3.4 *Cypher Query* Membuat Node

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 Tabel 3.11. Misal dalam trayek angkutan dengan kode jalur AL, yang mana melewati Jalan Karya Timur kemudian Jalan Mahakam, maka *Cypher Query* yang digunakan ditampilkan pada Gambar 3.5. Setelah itu terbentuklah sebuah graf yang merepresentasikan semua trayek dengan setiap jalur memiliki bobotnya masing-masing sesuai jarak yang ada.

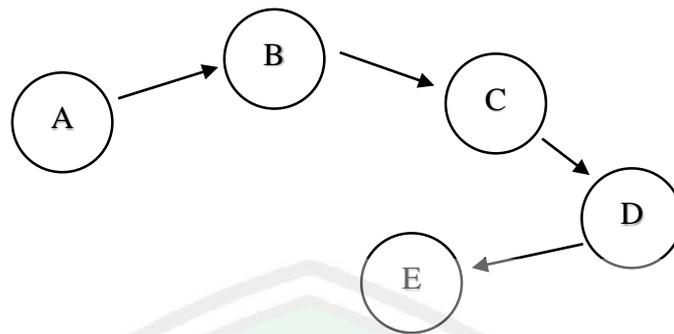
```
MATCH (n:lokasi {nama:"Jalan Karya Timur"}), (m:lokasi
{name:"Jalan Mahakan"}) CREATE (n) -[r:menuju {id_trayek:1,
nama_trayek:"AL - Arjosari", jarak:1600}]-> (m)
```

Gambar 3.5 *Cypher Query* Membuat Relasi

3.3.7 *Weighted Directed Acyclic Graphs* Dalam *Graph Database*

Weighted Directed Acyclic Graphs (WDAG) diterapkan dalam pembangunan aplikasi dalam penelitian ini. Penerapannya yaitu ditambahkan *property* untuk setiap relasi yang dibuat dengan nama “jarak” yang merupakan jarak antara lokasi satu dengan lokasi terdekat lainnya dalam satu trayek. Jarak ini juga nantinya berguna untuk mengetahui total jarak dari setiap perjalanan yang ditempuh, serta sebagai bahan perhitungan dengan AHP agar menemukan alternatif terbaik.

Untuk model graf yang berguna untuk merepresentasikan jalur angkutan umum, setiap lokasi disambungkan dengan relasi masing-masing jalur atau trayek dan dibuat berurutan sesuai jalur yang dilewati. Untuk lebih jelasnya dijelaskan melalui Gambar 3.6.



Gambar 3.6 Alur Relasi Yang Merepresentasikan Jalur Angkutan Umum

Pada Gambar 3.6 dijelaskan bahwa ketika node A menuju node E, maka perpindahan node dilakukan dengan cara berurutan A ke B ke C ke D ke E dan bobot setiap relasi yang dilalui memiliki jaraknya masing-masing.

Setiap *node* yang terhubung diberikan bobot sesuai dengan jarak antar lokasi tersebut. Pemberian jarak didapatkan dengan data lokasi yang didapat dari Dinas Perhubungan kemudian ditentukan berdasarkan koordinat dan jarak antar koordinat tersebut dengan menggunakan *Google Street*.

Dalam basis data graf yang saya modelkan, saya memberikan properti dengan id: jarak untuk menampung jarak dari setiap relasi yang ada antar *node* dengan *node* yang lainnya.

Data disimpan sebagai properti dengan menggunakan variabel properti jarak. Satu node yang terhubung dalam satu jalur trayek akan diberikan nilai properti di relasi yang ada dengan node terdekat dengannya. Node selanjutnya juga dihubungkan dengan relasi node selanjutnya sehingga bisa terhubung relasi satu sama lain dalam satu trayek dari titik awal pemberangkatan angkutan umum menuju titik akhir pemberhentian. Pemberian relasi juga diberikan arah tujuan atau bisa dibidang dengan directed graph sehingga tidak bisa untuk bolak balik atau hanya

bisa melaju sesuai dengan arah yang ditentukan seperti arah laju angkutan umum pada kenyataannya.

Berikut ini merupakan data jarak antar node dari setiap relasi yang menghubungkannya, sesuai dengan penjelasan sebelumnya, dirangkum dalam Tabel 3.12, Tabel 3.13, Tabel 3.14, Tabel 3.15, Tabel 3.16, Tabel 3.17, Tabel 3.18, Tabel 3.19, Tabel 3.20 dengan data jarak lokasi diambil dari *google maps* dan melihat data dari Dinas Perhubungan Kota Malang dan Kota Batu.

Tabel 3.12 Jarak Antar Lokasi Trayek AL - Arjosari

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
1	Terminal Arjosari Malang	Jl. R. Panji Suroso Malang	1200
2	Jl. R. Panji Suroso Malang	Jl. Laksamana Adi Sucipto Malang	1500
3	Jl. Laksamana Adi Sucipto Malang	Jl. Tenaga Malang	650
4	Jl. Tenaga Malang	Jl. Karya Timur Malang	750
5	Jl. Karya Timur Malang	Jl. Mahakam Malang	1600
6	Jl. Mahakam Malang	Jl. WR. Supratman Malang	200
7	Jl. WR. Supratman Malang	Jl. Panglima Sudirman Malang	800
8	Jl. Panglima Sudirman Malang	Jl. Patimura Malang	280
9	Jl. Patimura Malang	Jl. Trunojoyo Malang	350
10	Jl. Trunojoyo Malang	Jl. Kertanegara Malang	350
11	Jl. Kertanegara Malang	Jl. Tugu Malang	100
12	Jl. Tugu Malang	Jl. Kahuripan Malang	400
13	Jl. Kahuripan Malang	Jl. Semeru Malang	800
14	Jl. Semeru Malang	Jl. Ijen Malang	700

Tabel 3.13 Jarak Antar Lokasi Trayek AL - Arjosari Lanjutan

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
15	Jl. Ijen Malang	Jl. Retawu Malang	600
16	Jl. Retawu Malang	Jl. Bondowoso Malang	600
17	Jl. Bondowoso Malang	Jl. Jombang Malang	130
18	Jl. Jombang Malang	Jl. Surabaya Malang	350
19	Jl. Surabaya Malang	Jl. Jakarta Malang	500
20	Jl. Jakarta Malang	Jl. Bogor Malang	500
21	Jl. Bogor Malang	Jl. Veteran Malang	550
22	Jl. Veteran Malang	Jl. Sumbersari Malang	950
23	Jl. Sumbersari Malang	Terminal Landungsari Malang	4100

Tabel 3.14 Jarak Antar Lokasi Trayek GA - Gadang

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
1	Terminal Gadang Malang	Jl. Satsuit Tubun Malang	550
2	Jl. Satsuit Tubun Malang	Jl. S. Supriyadi Malang	2100
3	Jl. S. Supriyadi Malang	Jl. Arif Margono Malang	2400
4	Jl. Arif Margono Malang	Jl. Ade Irma Suryani Malang	550
5	Jl. Ade Irma Suryani Malang	Jl. KH. Wahid Hasyim Malang	58
6	Jl. KH. Wahid Hasyim Malang	Jl. Kauman Malang	200
7	Jl. Kauman Malang	Jl. KH. Hasyim Asy'ari Malang	230
8	Jl. KH. Hasyim Asy'ari Malang	Jl. AR. Hakim Malang	270
9	Jl. AR. Hakim Malang	Jl. Merdeka Utara Malang	220
10	Jl. Merdeka Utara Malang	Jl. S. Wiryopranoto Malang	350
11	Jl. S. Wiryopranoto Malang	Jl. Mojopahit Malang	1500
12	Jl. Mojopahit Malang	Jl. Tugu Malang	750
13	Jl. Tugu Malang	Jl. Untung Suropati Malang	220

Tabel 3.15 Jarak Antar Lokasi Trayek GA - Gadang Lanjutan

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
14	Jl. Untung Suropati Malang	Jl. Pajajaran Malang	900
15	Jl. Pajajaran Malang	Jl. Trunojoyo Malang	180
16	Jl. Trunojoyo Malang	Jl. Cokroaminoto Malang	650
17	Jl. Cokroaminoto Malang	Jl. Dr. Cipto Malang	260
18	Jl. Dr. Cipto Malang	Jl. Panglima Sudirman Malang	250
19	Jl. Panglima Sudirman Malang	Jl. WR. Supratman Malang	300
20	Jl. WR. Supratman Malang	Jl. Letjend Sutoyo Malang	800
21	Jl. Letjend Sutoyo Malang	Jl. Letjend S. Parman Malang	1200
22	Jl. Letjend S. Parman Malang	Jl. Jend. A. Yani Malang	1400
23	Jl. Jend. A. Yani Malang	Jl. R. Intan Malang	1700
24	Jl. R. Intan Malang	Terminal Arjosari Malang	950

Tabel 3.16 Jarak Antar Lokasi Trayek LG - Landungsari

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
1	Terminal Landungsari Malang	Jl. Tlogomas Malang	650
2	Jl. Tlogomas Malang	Jl. Mayjen MT. Haryono Malang	1400
3	Jl. Mayjen MT. Haryono Malang	Jl. Sumbersari Malang	2100
4	Jl. Sumbersari Malang	Jl. Bendungan Sutami Malang	1200
5	Jl. Bendungan Sutami Malang	Jl. Surabaya Malang	700
6	Jl. Surabaya Malang	Jl. Bondowoso Malang	450
7	Jl. Bondowoso Malang	Jl. Gading Malang	650
8	Jl. Gading Malang	Jl. Wilis Malang	800
9	Jl. Wilis Malang	Jl. Panderman Malang	500
10	Jl. Panderman Malang	Jl. AR. Hakim Malang	1300

Tabel 3.17 Jarak Antar Lokasi Trayek LG - Landungsari Lanjutan

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
11	Jl. AR. Hakim Malang	Jl. Merdeka Utara Malang	220
12	Jl. Merdeka Utara Malang	Jl. Merdeka Selatan Malang	350
13	Jl. Merdeka Selatan Malang	Jl. Wiryopranoto Malang	1300
14	Jl. Wiryopranoto Malang	Jl. Sultan Syahrir Malang	260
15	Jl. Sultan Syahrir Malang	Jl. Kyai Tamin Malang	350
16	Jl. Kyai Tamin Malang	Jl. Sartono SH. Malang	600
17	Jl. Sartono SH. Malang	Jl. Peltu Sujono Malang	450
18	Jl. Peltu Sujono Malang	Jl. Susanto Malang	240
19	Jl. Susanto Malang	Jl. Niaga Malang	240
20	Jl. Niaga Malang	Jl. Sonokeling Malang	290
21	Jl. Sonokeling Malang	Jl. Janti Malang	850
22	Jl. Janti Malang	Jl. S. Supriyadi Malang	1100
23	Jl. S. Supriyadi Malang	Jl. Satsuit Tubun Malang	2100
24	Jl. Satsuit Tubun Malang	Terminal Gadang Malang	550

Tabel 3.18 Jarak Antar Lokasi Trayek BSS - Terminal Batu

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
1	Terminal Batu	Jl. Dewi Sartika	180
2	Jl. Dewi Sartika	Jl. Agus Salim	1200
3	Jl. Agus Salim	Jl. Sultan Agung	800
4	Jl. Sultan Agung	Jl. Ahmad Yani	1100
5	Jl. Ahmad Yani	Jl. Brantas	600
6	Jl. Brantas	Selecta	5300
7	Selecta	Jl. Junggo	2200
8	Jl. Junggo	Sumber Brantas	9500

Tabel 3.19 Jarak Antar Lokasi Trayek BJJ - Terminal Batu

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
1	Terminal Batu	Jl. Dewi Sartika	180
2	Jl. Dewi Sartika	Oro-oro Ombo	2700
3	Oro-oro Ombo	Junrejo	4000
4	Junrejo	Dau	4200
5	Dau	Jl. Sengkaling	1600
6	Jl. Sengkaling	Terminal Landungsari	2000

Tabel 3.20 Jarak Antar Lokasi Trayek BNK - Terminal Batu

No	Lokasi Awal	Lokasi Tujuan	Jarak (meter)
1	Terminal Batu	Jl. Dewi Sartika	180
2	Jl. Dewi Sartika	Jl. Agus Salim	1200
3	Jl. Agus Salim	Jl. Sultan Agung	800
4	Jl. Sultan Agung	Jl. Suropati	1200
5	Jl. Suropati	Jl. Hasanudin	6800
6	Jl. Hasanudin	Jl. Trunojoyo Batu	1600
7	Jl. Trunojoyo Batu	Pujon	18600
8	Pujon	Ngantang	10000
9	Ngantang	Kasembon	18000

Langkah dari pencarian rute adalah dengan memetakan semua *node* yang ada dalam *Graph Database* dengan menggunakan klausa *MATCH* yang kemudian disaring untuk menentukan titik lokasi awal dan lokasi akhir. Pencarian dan pengoperasian *Cypher Query* dilakukan dengan arsitektur ASCII untuk sintaks yang terstruktur.

Neo4j menggunakan *Index-free Adjacency* untuk melakukan ekspansi yang berarti setiap *node* mengetahui hubungan yang melekat padanya. Tidak diperlukan

operasi *join* dan pencarian indeks yang diperlukan seperti halnya di *Relational Database*.

Ketika *node* awal dan akhir sudah ditemukan, serta semua relasi yang terhubung sudah didapatkan, selanjutnya adalah dengan melakukan penjumlahan jarak dari relasi yang dihasilkan. Dari *node* ke *node* selanjutnya yang berhasil memecahkan masalah perjalanan dari lokasi awal ke lokasi tujuan dijumlahkan dengan menggunakan klausa untuk dijadikan sebagai jarak total.

3.3.8 Analisis Hasil 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 ditentukan 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 3.21, kemudian dihitung nilai *Eigen*.

Tabel 3.21 Kriteria Pembobotan Dengan AHP antara Jumlah Angkot dan Jarak

No	Kriteria	Bobot Kriteria
1	Jumlah Angkot	9
2	Jarak	3

Tabel 3.22 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* pada Gambar 3.7 sebagai berikut

```
MATCH p = (n)-[r]->() RETURN COUNT(r)
```

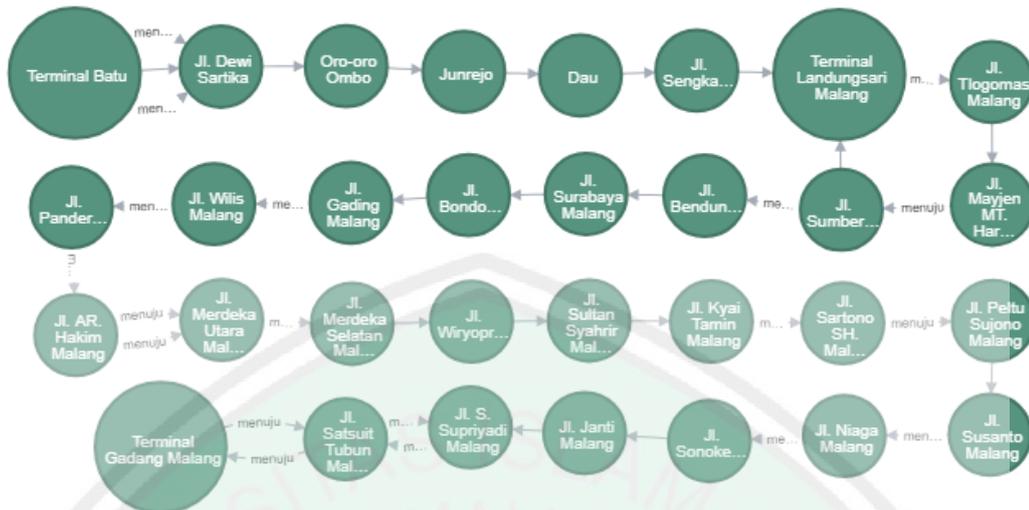
Gambar 3.7 *Cypher Query* Menghitung Seluruh Kemungkinan Alternatif

Cypher Query tersebut mengembalikan nilai semua kombinasi relasi antar setiap lokasi yang sudah dibuat dengan model pada Tabel 3.11 atau Gambar 3.3 dan dengan data setiap lokasi, sehingga didapatkan bahwa ada sebanyak 94 relasi kombinasi perjalanan dari semua lokasi. Selanjutnya, dibuat dengan simulasi perjalanan dari Terminal Batu menuju Terminal Gadang dengan *Cypher Query* pada Gambar 3.8 sebagai berikut:

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

Gambar 3.8 *Cypher Query* Perjalanan Dari Terminal Batu ke Terminal Gadang

Maka dihasilkan 6 alternatif trayek yang bisa menyelesaikan perjalanan tersebut dengan jarak yang sama, namun dengan jumlah angkot yang berbeda-beda diterangkan dalam Tabel 3.23.



Gambar 3.9. Visualisasi Graf dari Terminal Batu ke Terminal Gadang

Pada Gambar 3.9 didapatkan bahwa ada beberapa rute perjalanan yang bisa dipilih yang memiliki jarak dan jenis angkot yang berbeda-beda seperti ditampilkan pada Tabel 3.23. Alternatif trayek yang bisa menyelesaikan perjalanan berjumlah total 6 rute dengan kombinasi jumlah angkutan yang digunakan serta jarak yang berbeda-beda atau terkadang sama tergantung dari relasi yang dihasilkan. Dalam hal ini, enam alternatif rute yang dihasilkan memiliki jumlah jarak yang sama namun jumlah transfer yang berbeda-beda karena memang dalam perjalanan graf yang sudah disusun memiliki percabangan yang berarti perlu dilakukan pergantian trayek angkot atau bisa disebut transfer sehingga penumpang perlu turun dan berganti angkutan sesuai dengan jalur yang dibutuhkan. Transfer ini juga berpengaruh dalam biaya dan efektifitas waktu, karena penumpang cenderung butuh menunggu untuk bisa mendapatkan angkutan selanjutnya. Berikut hasil alternatif rute ditampilkan dalam Tabel 3.23.

Tabel 3.23 Daftar Alternatif Trayek dari Terminal Batu ke Terminal Gadang

Alternatif	Jarak	Jumlah Angkot	Keterangan	Nama Trayek	Keterangan
Rute 1	33330	5	Empat 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	4	Tiga 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	5	Empat 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

Seperti yang sudah disebutkan sebelumnya, 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 4, dan rute 6. Sedangkan rute 1, rute 3 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 3.24 dan kemudian dihitung matrik perbandingan pada Tabel 3.25 dan nilai *Eigen* dengan matrik ternormalisasi pada Tabel 3.26.

Tabel 3.24 Pemberian Bobot Kriteria untuk Trayek Terpilih Berdasarkan Jumlah Kombinasi Angkot

Rute	2	4	6
Bobot Kriteria Berdasarkan Jumlah Angkot	3	5	3

Tabel 3.25 Matrik Perbandingan Berpasangan Berdasarkan Jumlah Angkot

Jumlah Angkot	Rute 2	Rute 4	Rute 6
Rute 2	1	0.6	1
Rute 4	1.666666667	1	1.666666667
Rute 6	1	0.6	1
JKP	3.666666667	2.2	3.666666667

Tabel 3.26 Matrik Ternormalisasi

Jumlah Angkot	Rute 2	Rute 4	Rute 6	Rata-rata (eigen)
Rute 2	0.272727273	0.272727273	0.272727273	0.272727273
Rute 4	0.454545455	0.454545455	0.454545455	0.454545455
Rute 6	0.272727273	0.272727273	0.272727273	0.272727273

Untuk perhitungan jarak karena ke empat rute mempunyai jarak yang sama, maka rata-ratanya menjadi seimbang yaitu 0.333333333. Kemudian matrik berpasangan antara perbandingan jarak dan jumlah angkot dinormalisasi dengan menggunakan nilai *Eigen* dari Tabel 3.21, 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, rute 6 dengan bobot yang sama. Untuk lebih detailnya hasil yang diperoleh dari perhitungan matrik AHP diatas, dijelaskan dan dirangkum pada Tabel 3.27

Tabel 3.27 Bobot Akhir Rute Perjalanan dari Terminal Batu ke Terminal Gadang

No	Alternatif Rute	Bobot Akhir
1	Rute 2	0.287878788
2	Rute 4	0.424242424
3	Rute 6	0.287878788

Pada Tabel 3.27 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 BJJ, kemudian di Jalan Satsuit Tubun berpindah menggunakan angkot LG.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi Sistem adalah penerapan pembahasan yang sudah dijabarkan sebelumnya pada Bab 3 di dalam sebuah aplikasi. Dalam hal ini, aplikasi dibangun dengan teknologi Android. Berikut spesifikasi Hardware dan Software yang digunakan dalam implementasi program.

A. Kebutuhan Hardware

- a. Processor: Intel Core i3-4030U 1,9 GHz
- b. Memory: 6 GB RAM
- c. Disk: SSD Kingstone 240 GB

B. Kebutuhan Software

- a. Operating System: Windows 10 Pro
- b. DBMS: Neo4j 3.4.1
- c. IDE: Android Studio, Netbeans 8.2
- d. Browser: Opera Browser 58.0.3135.118 dan Chrome Browser 73.0.3683.86
- e. Pemodelan: StartUML, Power Designer
- f. Version Control: Git
- g. Nox Android Emulator.
- h. Corel Draw

4.2 Implementasi *Interface*

Pada tahap ini, model yang sudah disusun dalam Bab 3 di implementasikan kedalam software yang dibangun menggunakan IDE Android Studio. Untuk Desain Grafis menggunakan Corel Draw dan Photoshop jika diperlukan.

Implementasi interface ini dirancang dengan seminimalis mungkin, agar bisa memudahkan pengguna dalam menggunakan dan mengoperasikannya. Setelah tampilan *Splash Screen*, pengguna akan langsung dihadapkan dengan 5 tombol utama yaitu:

1. Tombol Cari Alternatif Rute
2. Tombol Lihat Data Lokasi
3. Tombol Lihat Semua Rute
4. Tombol Tentang Aplikasi
5. Tombol Bantuan

4.2.1 Tampilan *Splash Screen*

Splash Screen adalah sebuah tampilan yang ditampilkan pertama kali sebelum sebuah program, khususnya di Android, menuju proses-proses yang lainnya. *Splash Screen* sangat berguna untuk menampilkan beberapa informasi yang dibutuhkan, juga untuk memberikan kesan pertama yang lebih menarik.

Berikut tampilan *Splash Screen* dari aplikasi yang dibangun ditampilkan pada Gambar 4.1. Dalam hal ini, saya memberi nama aplikasi ini dengan nama “Angkotku”.



Gambar 4.1 Tampilan *Splash Screen*

Splash Screen ditampilkan selama 3 detik sebelum memasuki menu-menu utama. Ketika berada di menu utama dan tombol back di tekan, maka langsung menutup aplikasi tanpa kembali ke *Splash Screen* tersebut. *Splash Screen* ditampilkan ketika awal aplikasi dijalankan.

4.2.2 Tampilan Home / Menu Utama

Seperti yang sudah dijelaskan sebelumnya, setelah *Splash Screen* dijalankan, maka selanjutnya akan menuju ke menu-menu atau tombol-tombol utama. Seperti yang sudah disampaikan, bahwa dalam halaman utama ini memiliki tombol-tombol yang berguna untuk menuju halaman yang lainnya yaitu:

1. Halaman Cari Alternatif Rute
2. Halaman Lihat Data Lokasi
3. Halaman Lihat Semua Rute
4. Halaman Tentang Aplikasi
5. Halaman Bantuan

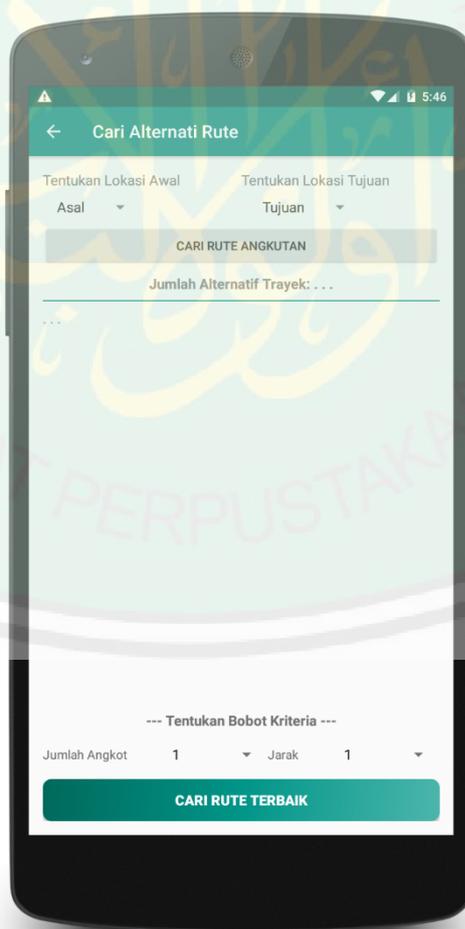
Untuk tampilan halaman utama ditampilkan pada Gambar 4.2 dibawah ini



Gambar 4.2 Tampilan Halaman Utama

4.2.3 Halaman Cari Alternatif Rute

Halaman cari alternatif rute merupakan halaman yang berguna untuk mencari alternatif rute-rute mana saja yang bisa menyelesaikan perjalanan dari satu titik awal menuju titik tujuan. Dalam halaman ini, pengguna dibutuhkan untuk memasukkan titik lokasi dari mana dan hendak kemana dia pergi, untuk selanjutnya pengguna mencari alternatif rute mana saja yang bisa mengantar perjalanan tersebut. Jika sistem menemukan alternatif rute, maka akan ditampilkan dalam halaman ini namun tanpa ada filter rute mana yang terbaik. Berikut tampilan halaman Cari Alternatif Rute pada Gambar 4.3.



Gambar 4.3 Tampilan Halaman Cari Alternatif Rute

Untuk menentukan titik lokasi awal dan tujuan, digunakan sebuah *Spinner* yang berguna untuk menampilkan data semua lokasi yang sudah dimasukkan kedalam *Neo4j Graph Database*.

Dalam aplikasi yang dibangun, setiap fungsi yang berhubungan dengan *Graph Database Neo4j* memerlukan sebuah koneksi yang berguna untuk menghubungkan aplikasi dengan *Graph Database*. Diperlukan juga sebuah Driver untuk melaksanakan koneksi tersebut yaitu dengan mengimplementasikan driver dari `org.neo4j.driver:neo4j-java-driver:1.7.2`. Selanjutnya, untuk melakukan koneksi diperlukan inisialisasi Url, Username, dan Password yang merupakan model Basic Authentication. Detail Authentifikasi yang dilakukan ditampilkan dalam Tabel 4.1 dibawah:

Tabel 4.1 Authentifikasi Koneksi ke *Graph Database*

No	Variabel	Nilai
1	Uri	bolt://172.17.100.2:7687
2	Username	Neo4j
3	Password	9

Untuk memanggil nama-nama lokasi, dalam hal ini di representasikan menjadi *node-node*, dalam *Graph Database* dilakukan dengan menggunakan *Cypher Query* sebagai berikut:

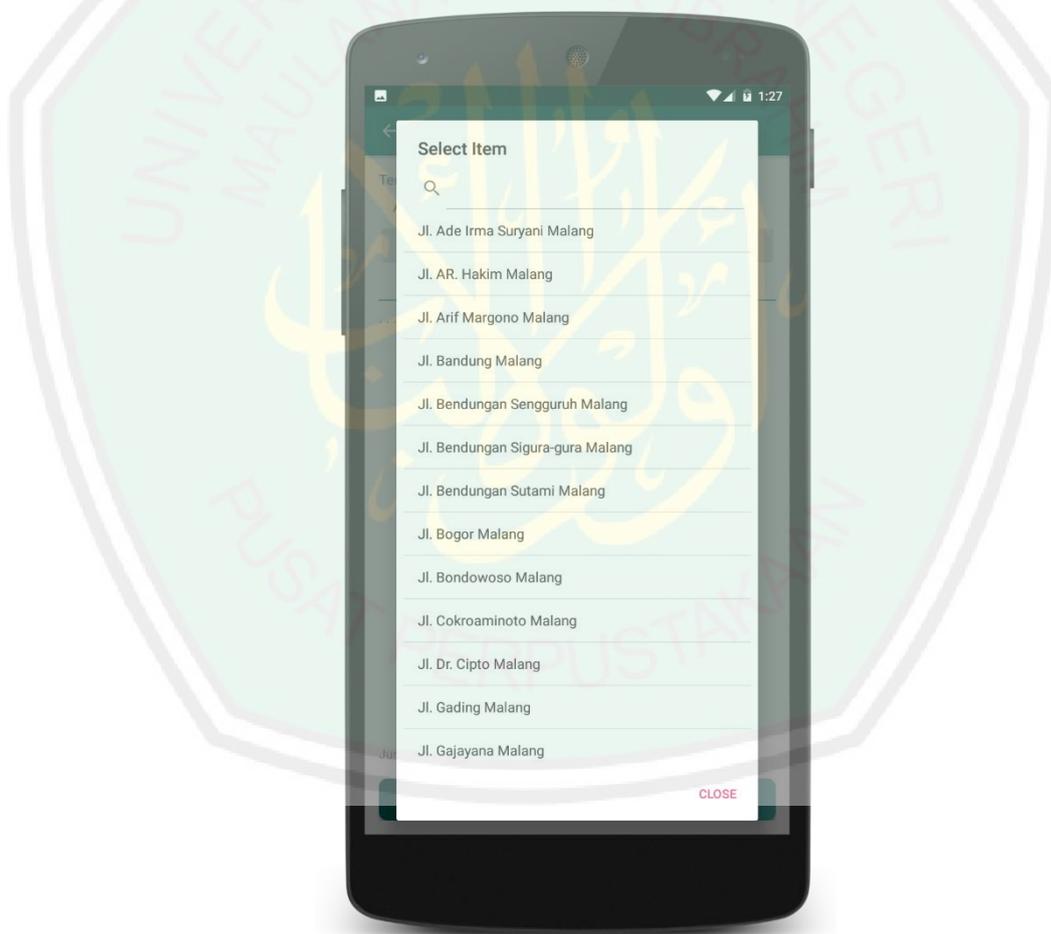
```
MATCH (n) RETURN n.nama AS nama
```

Gambar 4.4 *Cypher Query* Mengambil Nama Lokasi

Cypher Query tersebut mengembalikan nilai-nilai dari *node* (n) berupa nama lokasi yang tersimpan dalam properti “nama” menjadi sebuah data tabel

sehingga bisa dipanggil kembali kedalam aplikasi menggunakan fungsi `StatementResult` dan `Record` dan kemudian dimasukkan datanya menjadi sebuah `array`.

Data-data lokasi yang sudah tersimpan, kemudian ditampilkan menjadi sebuah *Popup Spinner Android* yang memiliki fitur pencarian sehingga memudahkan pengguna untuk mencari lokasi yang dikehendaki. Untuk *Spinner* dalam aplikasi yang dihasilkan ditampilkan dalam Gambar 4.5.



Gambar 4.5 Tampilan *Spinner* Yang Menampilkan Lokasi

Spinner di Android tersebut diberikan fitur pencarian sehingga memudahkan pengguna dalam menentukan lokasi, baik lokasi awal maupun lokasi

tujuan. Misal, untuk menentukan lokasi yang mengandung kata “batu” maka cukup menuliskan kata tersebut untuk menampilkan lokasi-lokasi yang terkait dengan kata kunci yang dikehendaki. Untuk hasil pencarian ditampilkan dalam Gambar 4.6 dibawah ini.



Gambar 4.6 Hasil Fitur Pencarian Dengan Kata Kunci "Batu"

Fitur *Spinner* untuk menampilkan lokasi-lokasi yang tersedia ini ditampilkan untuk menentukan lokasi awal keberangkatan maupun lokasi tujuan.

Setelah menentukan lokasi awal dan lokasi tujuan dari perjalanan, maka pengguna cukup mengklik tombol “Cari Rute Angkutan” untuk mengetahui alternatif rute mana saja yang bisa menyelesaikan perjalanan tersebut. Jika

ditemukan, maka pada halaman ini akan ditampilkan hasil dari alternatif mana saja yang tersedia. Mekanisme pengambilan alternatif rute yang dilakukan ditampilkan pada Gambar 4.7 sebagai berikut.

```

read lokasiAwal
read lokasiTujuan
read statementResult
while (statementResult has next)
  for (list.size())
    stringBuilder = append "Alternatif Ke: "
    while (node has next)
      read namaLokasi
      if (relationship has next)
        read namaTrayek
        if (trayekSesudah !=
            trayekSebelum && trayekSebelum != null)
          stringBuilder = append
            "Turun di namaLokasi, pindah
            menggunakan trayekSesudah"
        if (trayekSesudah ==
            trayekSebelum)
          stringBuilder = append
            "Menggunakan trayekSesudah"
        trayekSebelum = trayekSesudah

```

Gambar 4.7 Pseudocode Menemukan Alternatif Rute

Setelah ditemukan berbagai macam alternatif rute yang bisa menyelesaikan perjalanan, tidak semua data tersebut dimasukkan kedalam array untuk selanjutnya diproses lebih lanjut, namun perlu di saring dahulu supaya data lebih mendekati sesuai yang diharapkan dalam hal ini adalah batasan masalah yang menyebutkan maksimal trayek yang akan dihitung adalah 2 (dua) kali transfer. Proses penyaringan ini dilakukan dengan tahap-tahap pada Gambar 4.8 sebagai berikut.

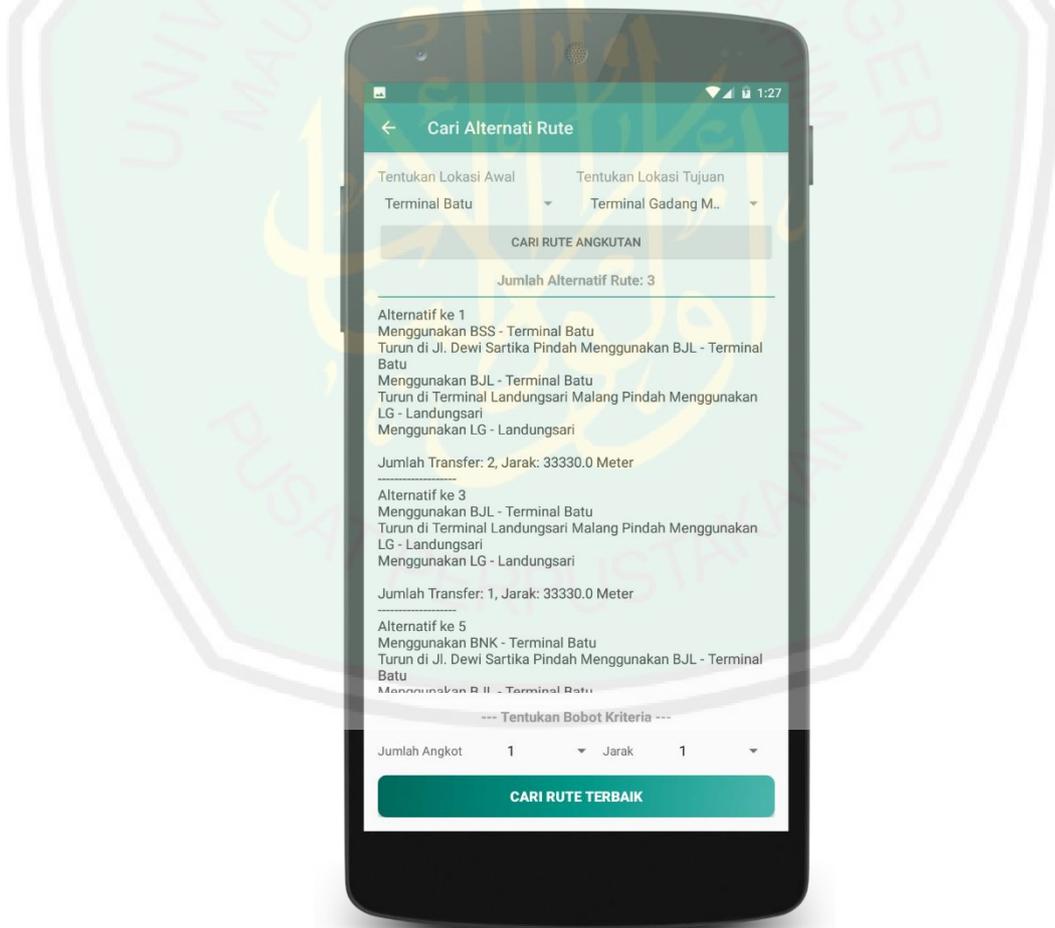
```

read jumlahTransfer
if (jumlahTransfer <= 2)
    arrayAlternatif = add "stringBuilder"

```

Gambar 4.8 *Pseudocode* Membatasi Jumlah Maksimal Transfer

Berikut ini adalah simulasi dari perjalanan dari Terminal Batu menuju Terminal Gadang Malang yang menghasilkan 3 (tiga) alternatif rute yang memenuhi syarat (maksimal 2 kali transfer) ditampilkan pada Gambar 4.9 berikut ini.



Gambar 4.9 Alternatif Rute Yang Dihasilkan dari Terminal Batu Menuju Terminal Gadang Malang

Dalam halaman ini, ketika tombol “Cari Rute Angkutan” ditekan, selain untuk menampilkan semua alternatif rute yang bisa menyelesaikan perjalanan juga memiliki fungsi untuk menentukan pembobotan untuk semua alternatif baik dari segi jumlah transfer maupun dari segi jarak yang nantinya akan digunakan untuk perhitungan alternatif rute mana yang terbaik menggunakan metode AHP.

Untuk pembobotan dari segi jumlah transfer, digunakan fungsi yang membuat menuliskan sebuah array yang berguna untuk menentukan bobot setiap alternatif. Seperti yang sudah dijelaskan bahwa dalam penelitian ini menggunakan pilihan jumlah alternatif dibatasi maksimal hingga 2 (dua) kali transfer, yang berarti paling maksimal memiliki 3 pilihan bobot yaitu 0 (nol) kali transfer yang berarti tidak perlu melakukan perpindahan angkutan, 1 (satu) kali transfer yang berarti memerlukan 2 angkutan, serta 2 (dua) kali transfer yang berarti memerlukan 3 angkutan berbeda. Untuk detail pembobotan dijelaskan dalam Tabel 4.2 dibawah.

Tabel 4.2 Tabel Pembobotan Alternatif Trayek Berdasarkan Jumlah Transfer

No	Jumlah Transfer	Nilai Pembobotan
1	0	7
2	1	5
3	2	3

Untuk memasukan data-data yang tersebut pada Tabel 4.2 kedalam program, menggunakan langkah-langkah pada Gambar 4.10 sebagai berikut.

```

read jumlahTransfer
if (jumlahTransfer == 0)
    pembobotanJumlahTransfer = 7
else if (jumlahTransfer == 1)
    pembobotanJumlahTransfer = 5
else if (jumlahTransfer == 2)
    pembobotanJumlahTransfer = 3

```

Gambar 4.10 *Pseudocode* Pembobotan Alternatif Berdasarkan Jumlah Transfer

Dalam penentuan pembobotan yang dilakukan berdasarkan jarak tidak bisa seperti sebelumnya yang berdasarkan jumlah trayek, karena memang jarak tidak pasti jumlahnya dari setiap alternatif rute yang dihasilkan.

Pertama yang perlu dilakukan setelah semua jarak total per alternatif ditemukan, adalah di cek apakah jarak masing-masing alternatif sama dengan menggunakan langkah-langkah sebagai pada Gambar 4.11 berikut.

```

read arrayJarak
for (arrayJarak.length)
    if arrayJarak[0] != arrayJarak[i]
        return false
else return true

```

Gambar 4.11 *Pseudocode* Penentuan Nilai Array Jarak

Langkah-langkah diatas akan menghasilkan atau mengembalikan nilai *true* jika memang semua jarak alternatif yang dihasilkan hasilnya sama dan akan mengembalikan nilai *false* jika salah satu atau lebih data jarak alternatif-alternatif ada yang berbeda. Jika nilai dari setiap alternatif sama, maka untuk pembobotanpun akan memiliki nilai yang sama. Jika diantara setiap alternatif jaraknya berbeda, maka perlu ditentukan bobot masing-masing berbeda pula. Jarak yang lebih pendek

tentu memiliki bobot yang lebih besar karena dianggap lebih efektif. Data jarak setiap alternatif trayek yang dihasilkan sebelumnya sudah dilakukan pengurutan berdasarkan dari jarak yang terdekat sehingga ketika masuk kedalam array awal, index yang pertama merupakan jarak yang terpendek atau sudah sepantasnya memiliki bobot yang terbesar dari yang lainnya yang berbeda. Untuk langkah-langkah penentuan bobot alternatif yang berdasarkan jarak, dilakukan langkah-langkah pada Gambar 4.12 sebagai berikut.

```

isi = 7.0
arrayPembobotanJarak = add "isi"
for (arrayJarak.length)
    temp = arrayJarak[i]
    temp2 = arrayJarak[i + 1]
    if temp == temp2
        arrayPembobotanJarak = add "isi"
    if temp != temp2
        isi = isi - 1
        arrayPembobotanJarak = add "isi"

```

Gambar 4.12 *Pseudocode* Pembobotan Alternatif Berdasarkan Jarak

Langkah tersebut akan menentukan nilai pembobotan berdasarkan jarak yang nantinya dipakai untuk perhitungan menggunakan AHP. Jika terdapat perbedaan dalam setiap jarak di alternatif yang tersedia, maka alternatif pertama merupakan jarak yang terpendek sehingga mendapat pembobotan paling tinggi yaitu 7, lalu ketika jarak dari alternatif selanjutnya sama juga diberi nilai 7, jika berbeda maka nilai pembobotan akan diberikan dengan nilainya dikurangi 1, begitu seterusnya.

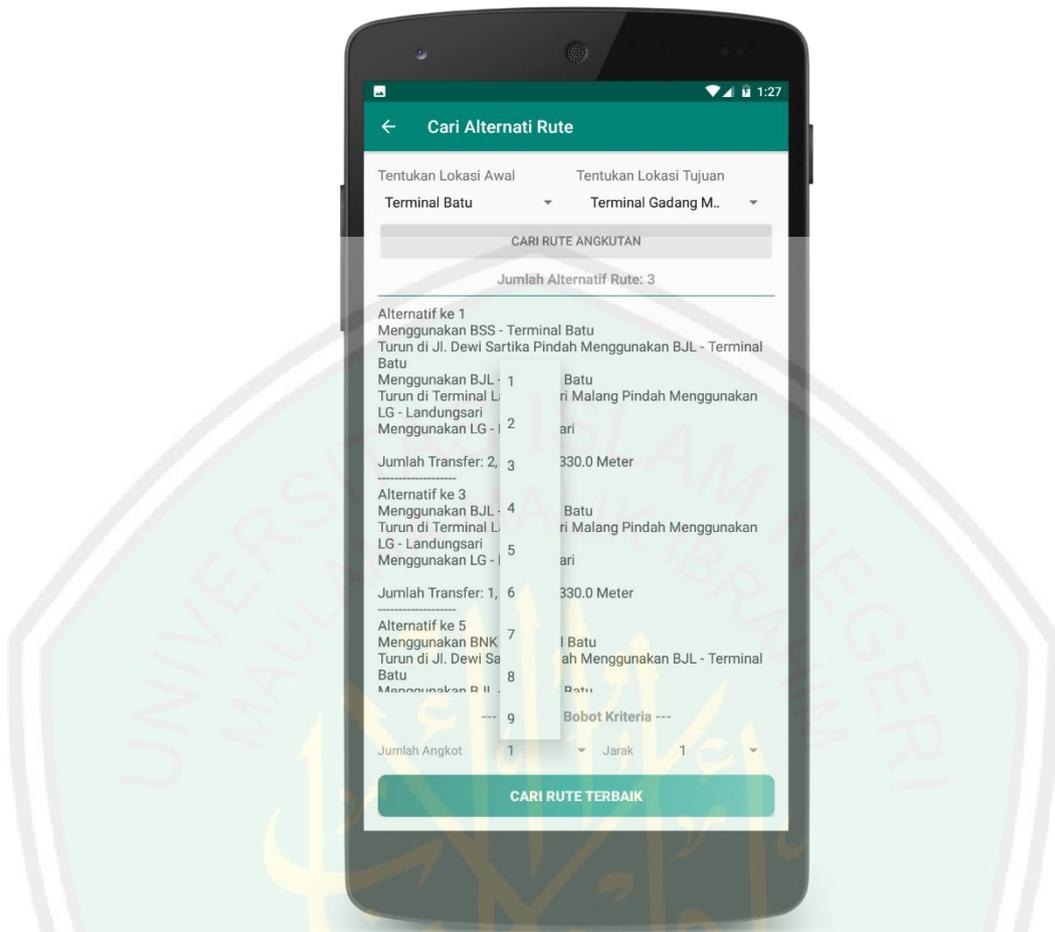
Di halaman ini, pengguna juga bisa menentukan nilai pembobotan kriteria yang ditentukan untuk jumlah transfer maupun jarak. Menurut Thomas L Saaty, prosedur penilaian perbandingan dalam AHP adalah seperti ditampilkan pada Tabel 4.3 berikut.

Tabel 4.3 Nilai Pembobotan Menurut Saaty

No	Nilai Pembobotan	Definisi
1	1	Kedua elemen / alternatif sama pentingnya
2	3	Elemen A sedikit lebih esensial dibandingkan dengan elemen B (<i>Moderate</i>)
3	5	Elemen A lebih esensial dari elemen B (<i>Strong</i>)
4	7	Elemen A jelas lebih esensial dari elemen B (<i>Very Strong</i>)
5	9	Elemen A mutlak lebih esensial dari elemen B (<i>Very Strong</i>)
6	2,4,6,8	Nilai antara dua perimbangan yang berdekatan

Untuk menentukan bobot kriteria, pengguna bisa dengan memilih nilai pembobotan untuk jumlah transfer dan jarak yang mana yang hendak menjadi prioritas. Dalam pemberian bobot kriteria ini, pengguna dapat dengan mudah karena sudah disediakan *Spinner* dalam aplikasi yang menyimpan nilai antara 1 sampai dengan 9 sesuai yang dijelaskan pada Tabel 4.3. Untuk tampilan yang dihasilkan seperti pada Gambar 4.13.

Setelah menentukan nilai pembobotan kriteria, maka pengguna cukup menekan tombol “Cari Rute Terbaik” untuk menemukan rute mana yang terbaik berdasarkan nilai-nilai yang didapatkan dan ditentukan sebelumnya. Dalam proses pencarian rute terbaik menggunakan AHP ada beberapa tahapan yang perlu diselesaikan.



Gambar 4.13 Penentuan Nilai Pembobotan Kriteria

Langkah selanjutnya adalah diperlukan fungsi untuk menghitung pembobotan matriks *Eigen* kriteria ditampilkan pada Gambar 4.14 berikut:

```

read arrayKriteria
for (declare i = 0, i < arrayKriteria.length, i++)
    for (declare j = 0, j < arrayKriteria.length, j++)
        matrikKriteria[j][i] = arrayKriteria[j] /
        arrayKriteria[i]

```

Gambar 4.14 *Pseudocode* Pembobotan Matriks Eigen Kriteria

Setelah itu menghitung jumlah kolom perbaris untuk menentukan matirks normalisasi dari pembobotan tersebut dengan langkah-langkah pada Gambar 4.15.

```

read arrayKriteria
read matrikKriteria
for (declare i = 0, i < arrayKriteria.length, i++)
    for (declare j = 0, j < arrayKriteria.length, j++)
        JKP = JKP + matrikKriteria[j][i]
arrayJKP = add "JKP"
JKP = 0

```

Gambar 4.15 *Pseudocode* Menghitung Jumlah Kolom Perbaris

Dalam menghitung matrik normalisasi, matrik yang tadi sudah didapatkan dibagi dengan JKP yang sudah didapatkan juga. Untuk langkah-langkahnya ditampilkan pada Gambar 4.16 sebagai berikut:

```

read arrayKriteria
read matrikKriteria
for (declare i = 0, i < arrayKriteria.length, i++)
    for (declare j = 0, j < arrayKriteria.length, j++)
        matrikKriteriaNormalisasi[j][i] =
            matrikKriteria[j][i] / arrayJKP[i]

```

Gambar 4.16 *Pseudocode* Pembagian Matrik Dengan JKP

Untuk menentukan nilai *Eigen* ternormalisasi, maka selanjutnya dilakukan dengan langkah-langkah perkalian matrik yang dilakukan dengan langkah-langkah dalam Gambar 4.17 berikut:

```

read arrayKriteria
read matrikKriteriaNormalisasi
for (declare i = 0, i < arrayKriteria.length, i++)
    for (declare j = 0, j < arrayKriteria.length, j++)
        eigenNormalisasi = eigenNormalisasi +
        matrikKriteriaNormalisasi[i][j]
    eigenNormalisasi = eigenNormalisasi /
    arrayKriteria.length
    arrayEgienKriteria = add "eigenNormalisasi"
    eigenNormalisasi = 0

```

Gambar 4.17 *Pseudocode* Menentukan Nilai Eigen Ternormalisasi

Untuk menentukan nilai *Eigen* dari perbandingan setiap alternatif baik dari segi jumlah transfer maupun jarak juga sama dengan langkah-langkah diatas. Selanjutnya, jika nilai *Eigen* kriteria, nilai *Eigen* perbandingan alternatif dari segi jumlah ransfer dan jarak sudah didapatkan, maka selanjutnya adalah menghitung bobot akhir yang nantinya menjadi sebuah penentu bahwa suatu alternatif tersebut merupakan alternatif terbaik berdasarkan nilai-nilai yang sudah ditentukan. Untuk langkah menghitung bobot akhir, dilakukan dengan cara pada Gambar 4.18.

```

read arrayEigenKriteria
read matrikEigenJumlahAngkotJarak
read jumlahAlternatif
for (declare i = 0, i < jumlahAlternatif, i++)
    for (declare j = 0, j < 1, j++)
        matrikBobotAkhir[i][j] = 0
        for (declare k = 0, k < 2, k++)
            matrikBobotAkhir[i][j] =
            matrikBobotAkhir[i][j] +
            matrikJumlahAngkotJarak[i][k] *
            arrayEigenKriteria[k]

```

Gambar 4.18 *Pseudocode* Menentukan Bobot Akhir

4.2.4 Halaman Rute Terbaik

Jika sudah ditemukan alternatif mana yang memiliki bobot terbanyak yang berarti bahwa alternatif tersebut merupakan yang terbaik berdasarkan perhitungan AHP dan nilai-nilai yang sudah ditentukan, di halaman “Rute Terbaik” ditampilkan satu alternatif rute yang terpilih tersebut. Gambar 4.19 berikut merupakan tampilan dari halaman Rute Terbaik.



Gambar 4.19 Halaman Rute Terbaik Menampilkan Alternatif Terpilih

Di bagian footer halaman tersebut juga menampilkan nilai-nilai *Eigen* dan nilai bobot akhir yang sudah ditemukan sebelumnya sebagai catatan. Di halaman ini juga ada informasi terkait lokasi awal dan lokasi tujuan yang sudah ditentukan pengguna sebelumnya agar tidak lupa dan lebih jelas informasi yang disampaikan.

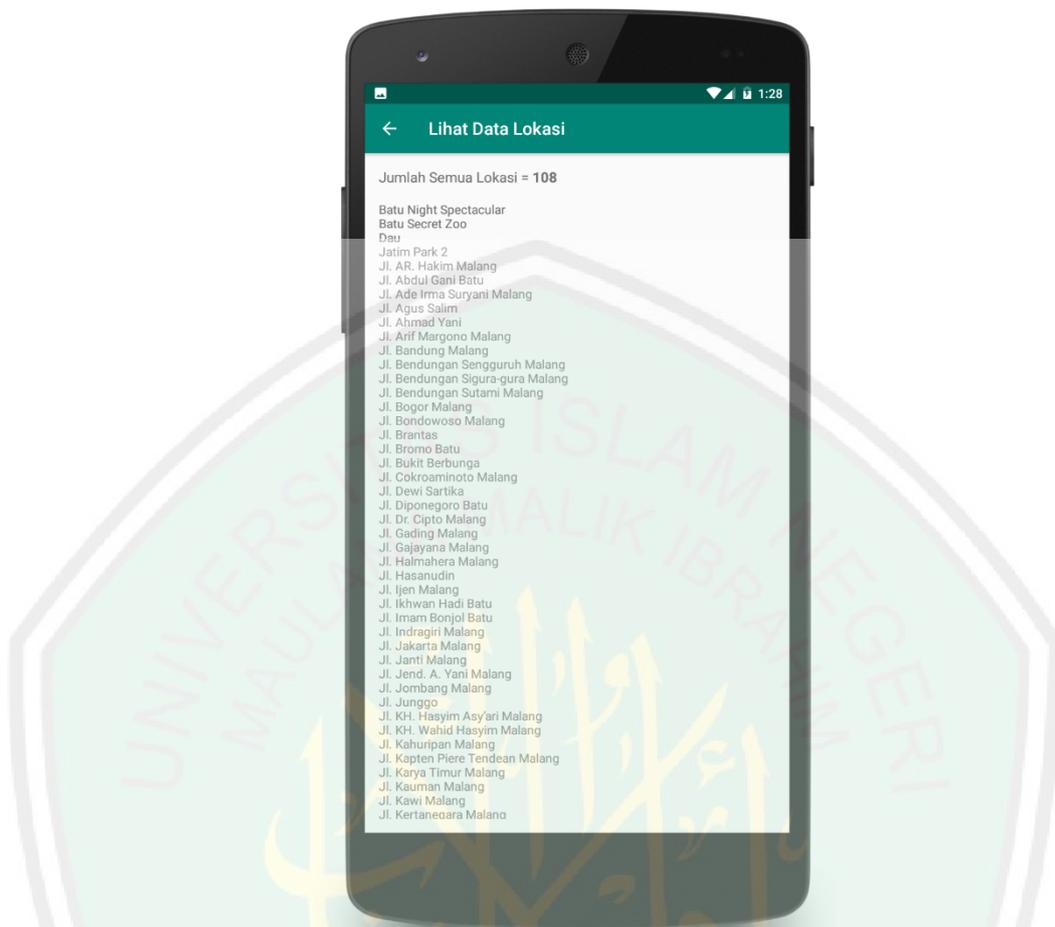
4.2.5 Halaman Lihat Data Lokasi

Dalam halaman Lihat Data Lokasi ditampilkan semua lokasi yang sudah tersedia dalam *Graph Database* Neo4j. Ini berguna untuk melihat secara langsung semua lokasi jika pengguna ingin mengetahuinya. Untuk memanggil data lokasi dari *Graph Database* dilakukan dengan menggunakan *Cypher Query* pada Gambar 4.20 sebagai berikut:

```
MATCH (n) RETURN n.nama AS nama ORDER BY nama
```

Gambar 4.20 *Cypher Query* Mengambil Data Lokasi

Cypher Query ini akan mengembalikan nama-nama lokasi yang direpresentasikan menjadi sebuah *node* dalam *Graph Database* Neo4j untuk selanjutnya dimasukkan kedalam halaman tersebut. Gambar 4.21 berikut merupakan tampilan dari halaman “Lihat Data Lokasi”. Di halaman Lihat Data Lokasi ini juga ditampilkan berapa jumlah lokasi yang tersedia dalam *Graph Database* Neo4j.



Gambar 4.21 Halaman Lihat Data Lokasi

4.2.6 Halaman Semua Rute

Halaman Semua Rute merupakan halaman yang berguna untuk menampilkan trayek-trayek yang tersedia dalam *Graph Database Neo4j*. Seperti yang sudah dijelaskan bahwa trayek-trayek yang ada direpresentasikan menjadi sebuah relasi yang saling terhubung dari satu lokasi menuju lokasi lainnya.

Setiap ada perpindahan dari satu lokasi ke lokasi lainnya bisa memiliki satu atau lebih relasi yang merupakan trayek yang bisa digunakan sehingga ada banyak sekali nama trayek yang sama, karena dalam satu perjalanan penuh suatu trayek bisa melewati berbagai macam tempat yang diantara lokasi-lokasi tersebut

menyimpan informasi trayek masing-masing, sedangkan dalam halaman ini dibutuhkan satu nama trayek saja dan tidak perlu ada pengulangan nama trayek yang sama. Untuk melakukan pemanggilan nama trayek, maka dilakukan dengan menggunakan *Cypher Query* pada Gambar 4.22. Fungsi *DISTINCT* berguna untuk hanya menampilkan satu kali saja jika ada data yang sama, sehingga tidak ada duplikasi data yang tidak perlu dalam halaman tersebut. Gambar 4.23 berikut merupakan tampilan dari halaman ini.

```
MATCH ()-[r]->() WITH r.nama_trayek as nama_trayek UNWIND
nama_trayek AS x WITH DISTINCT x RETURN x as nama_trayek2
```

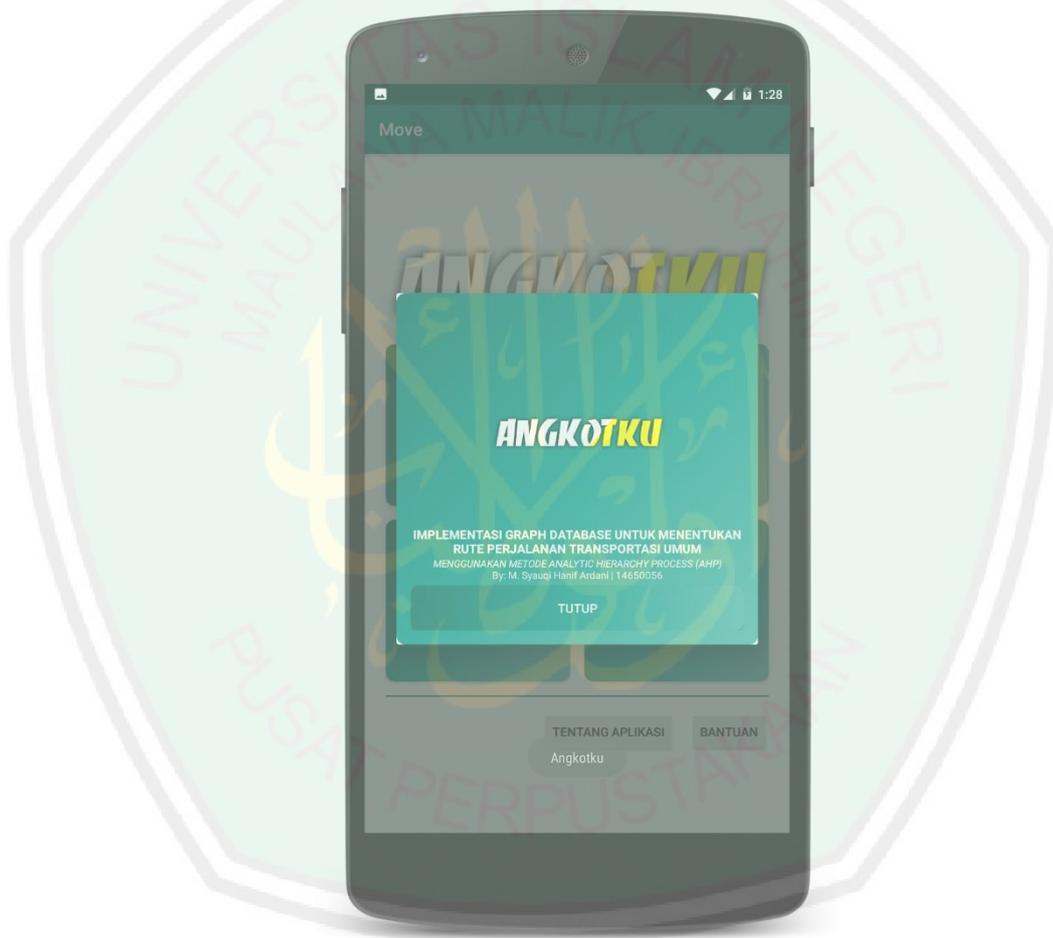
Gambar 4.22 *Cypher Query* Mengambil Data Semua Trayek Yang Tersedia



Gambar 4.23 Halaman Lihat Data Lokasi

4.2.7 Dialog Tentang Aplikasi

Dialog Tentang Aplikasi adalah sebuah Popup didalam aplikasi yang menampilkan informasi terkait dengan aplikasi ini. Ini merupakan sebuah dialog atau juga bisa disebut Popup bukan halaman baru, sehingga penampilannya tetap di halaman utama. Berikut Gambar 4.24 merupakan tampilan dari dialog ini.

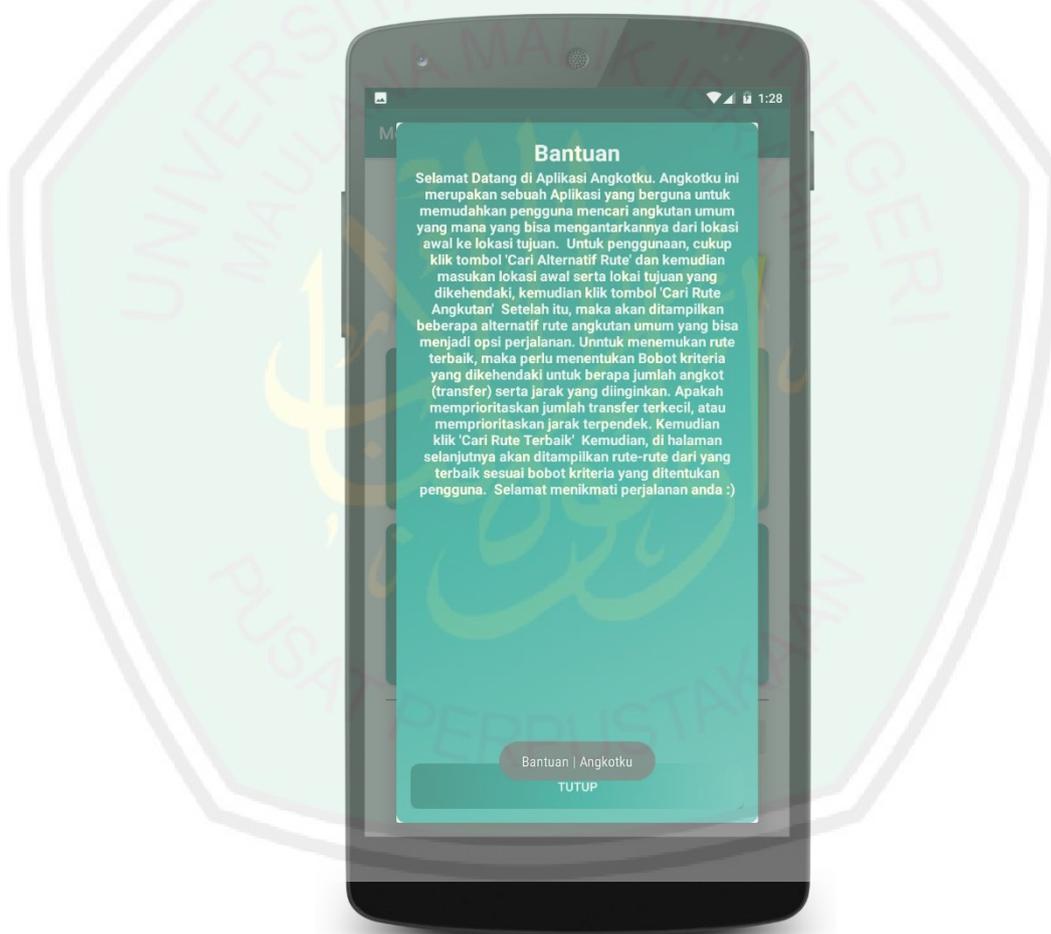


Gambar 4.24 Dialog Tentang Aplikasi

Jika di halaman utama pengguna menekan tombol “Tentang Aplikasi” maka akan muncul dialog seperti yang ditampilkan pada Gambar 4.11 serta sebuah Toast atau credits secara sementara yang bertuliskan nama aplikasi “Angkotku”.

4.2.8 Dialog Bantuan

Dialog Bantuan berisikan sedikit panduan untuk pengguna menggunakan aplikasi Angkotku ini. Dalam halaman utama, jika pengguna menekan tombol “Bantuan” maka akan muncul dialog Bantuan serta Toast bertuliskan “Bantuan”. Dialog ini berisikan text yang bisa dibaca oleh pengguna. Berikut Gambar 4.25 merupakan tampilan dari dialog bantuan.



Gambar 4.25 Dialog Bantuan

Untuk text bantuan menggunakan redaksi sebagai berikut ini:

“Selamat Datang di Aplikasi Angkotku. Angkotku ini merupakan sebuah Aplikasi yang berguna untuk memudahkan pengguna mencari angkutan umum

yang mana yang bisa mengantarkannya dari lokasi awal ke lokasi tujuan. Untuk penggunaan, cukup klik tombol 'Cari Alternatif Rute' dan kemudian masukan lokasi awal serta lokasi tujuan yang dikehendaki, kemudian klik tombol 'Cari Rute Angkutan'. Setelah itu, maka akan ditampilkan beberapa alternatif rute angkutan umum yang bisa menjadi opsi perjalanan. Untuk menemukan rute terbaik, maka perlu menentukan Bobot kriteria yang dikehendaki untuk berapa jumlah angkot (transfer) serta jarak yang diinginkan. Apakah memprioritaskan jumlah transfer terkecil, atau memprioritaskan jarak terpendek. Kemudian klik 'Cari Rute Terbaik'. Kemudian, di halaman selanjutnya akan ditampilkan rute-rute dari yang terbaik sesuai bobot kriteria yang ditentukan pengguna. Selamat menikmati perjalanan anda :)”

4.3 Pengujian Fungsionalitas

Pengujian fungsionalitas digunakan untuk menguji apakah fungsi-fungsi dari aplikasi yang dibangun sudah dapat dijalankan dengan benar dan sesuai yang diharapkan. Pengujian ini menggunakan metode Blackbox. Detail hasil pengujian ditampilkan dalam Tabel 4.4, Tabel 4.5 berikut.

Tabel 4.4 Pengujian Fungsionalitas Aplikasi

No	Halaman	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1	<i>Splash Screen</i>	Awal membuka aplikasi	Menampilkan <i>Splash Screen</i> selama 3 detik dan menuju halaman utama	Sesuai
2	Halaman utama	Menekan tombol cari alternatif rute	Menampilkan halaman cari alternatif rute	Sesuai
3	Halaman utama	Menekan tombol lihat data lokasi	Menampilkan halaman lihat data lokasi	Sesuai

Tabel 4.5 Pengujian Fungsionalitas Aplikasi Lanjutan

No	Halaman	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
4	Halaman utama	Menekan tombol lihat semua rute	Menampilkan halaman lihat semua rute	Sesuai
5	Halaman utama	Menekan tombol tentang aplikasi	Menampilkan dialog tentang aplikasi	Sesuai
6	Halaman utama	Menekan tombol bantuan	Menampilkan dialog bantuan	Sesuai
7	Halaman cari alternatif rute	Menekan tombol <i>Spinner</i> tentukan lokasi awal	Menampilkan daftar lokasi-lokasi yang tersedia untuk dijadikan lokasi awal	Sesuai
8	Halaman cari alternatif rute	Menekan tombol <i>Spinner</i> tentukan lokasi tujuan	Menampilkan daftar lokasi-lokasi yang tersedia untuk dijadikan lokasi tujuan	Sesuai
9	Halaman cari alternatif rute	Mencari lokasi dengan menggunakan filter	Menampilkan lokasi-lokasi yang sudah tersaring dengan kata kunci yang digunakan pengguna	Sesuai
10	Halaman cari alternatif rute	Menekan tombol cari rute angkutan	Menampilkan daftar alternatif-alternatif trayek yang bisa menyelesaikan perjalanan pengguna sesuai dengan lokasi awal dan tujuan yang telah ditentukan	Sesuai
11	Halaman cari alternatif rute	Menekan tombol <i>Spinner</i> bobot kriteria jumlah angkot	Menampilkan <i>Spinner</i> daftar angka pembobotan jumlah angkot antara 1 sampai 9	Sesuai
12	Halaman cari alternatif rute	Menekan tombol <i>Spinner</i> bobot kriteria jarak	Menampilkan <i>Spinner</i> daftar angka pembobotan jarak antara 1 sampai 9	Sesuai
13	Halaman cari alternatif rute	Menekan tombol cari rute terbaik	Menampilkan halaman rute terbaik sesuai dengan nilai-nilai yang sudah ditentukan	Sesuai

Berdasarkan dengan pengujian fungsionalitas yang sudah dilakukan didapatkan bahwa aplikasi yang sudah dibangun dapat berjalan sesuai dengan yang diharapkan.

4.4 Perhitungan Akurasi

Pada penelitian ini dilakukan pengujian akurasi untuk mengetahui berapa tingkat akurasi yang dihasilkan dari langkah-langkah implementasi Graph Database untuk menentukan rute transportasi umum. Untuk perhitungan akurasi menggunakan *Precision* pada persamaan 1, kemudian *Recall* pada persamaan 2 dan *Accuracy* pada persamaan 3. Pengukuran efektivitas suatu sistem temu kembali dapat dilakukan dengan perhitungan terhadap nilai perolehan (*recall*), nilai ketepatan (*precision*), dan jatuhnya semu (*fallout*) (Tague-Sutcliffe, 1992; Conlon dan Conlon, 1996). Namun, diantara metode tersebut, perhitungan ketepatan merupakan cara yang paling umum digunakan (Su, 1992; Tague-Sutcliffe, 1992). *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.

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

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

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

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 namun lebih dari 2x transfer. TN (*True Negative*) semua relasi

yang ada selain alternatif yang dihasilkan. FN (*False Negative*) adalah rute yang tidak bisa menyelesaikan perjalanan namun keluar di hasil pencarian. Keterangan juga ditampilkan pada Tabel 4.6 dibawah ini.

Tabel 4.6 Keterangan Nilai-nilai Perhitungan Akurasi

No	Variabel	Keterangan
1	TP (<i>True Positif</i>)	Jumlah Alternatif Maksimal 2x Transfer
2	FP (<i>False Positif</i>)	Jumlah Seluruh Alternatif - Jumlah Alternatif Maksimal 2x Transfer
3	TN (<i>True Negative</i>)	Jumlah Seluruh Relasi - Jumlah Seluruh Alternatif
4	FN (<i>False Negative</i>)	Jumlah Yang Keluar Namun Salah

Seperti yang sudah didapatkan di Bab 3, bahwa jumlah total relasi yang ada sebanyak 94. Untuk mengetahui tingkat akurasi yang digunakan, dilakukan beberapa kali simulasi perjalanan dengan data parameter berbeda yang dipilih secara random. Simulasi dilakukan sebanyak 15 kali. Setiap simulasi kemudian dihitung *Precision*, *Recall*, serta *Accuracy* masing-masing untuk kemudian dicari berapa rata-ratanya. Untuk detail percobaan yang dilakukan ditampilkan pada Tabel 4.7. Setiap simulasi yang dijalankan dengan nilai masukan parameter lokasi awal serta tujuan yang berbeda, menghasilkan nilai alternatif yang berbeda pula sehingga nilai akurasi yang dihasilkanpun memiliki nilai masing-masing. Untuk nilai *Precision*, *Recall* dan *Accuracy* dari masing-masing simulasi ditampilkan pada Tabel 4.8.

Tabel 4.7 Daftar Simulasi Yang Dilakukan Serta Alternatif Yang Didapatkan

No	Nama Lokasi Awal	Nama Lokasi Tujuan	Jumlah Keseluruhan Alternatif	Jumlah Alternatif Maksimal 2x Transfer
1	Terminal Batu	Terminal Gadang Malang	6	3
2	Terminal Arjosari Malang	Terminal Gadang Malang	8	4
3	Terminal Arjosari Malang	Terminal Landungsari Malang	1	1
4	Jl. Bendungan Sutami Malang	Terminal Landungsari Malang	42	8
5	Terminal Landungsari Malang	Jl. Bendungan Sutami Malang	1	1
6	Terminal Arjosari Malang	Jl. Sumbersari Malang	2	2
7	Jl. Sumbersari Malang	Jl. Surabaya Malang	84	8
8	Dau	Jl. Sumbersari Malang	42	2
9	Jl. Sengkaling	Jl. Letjend Sutoyo Malang	8	5
10	Terminal Landungsari Malang	Jl. Merdeka Selatan Malang	2	2
11	Jl. Gading Malang	Jl. Bogor Malang	40	8
12	Jl. Gading Malang	Jl. Dr. Cipto Malang	8	5
13	Jl. Bondowoso Malang	Terminal Arjosari Malang	32	8
14	Jl. Bondowoso Malang	Terminal Landungsari Malang	41	8
15	Terminal Arjosari Malang	Jl. Gading Malang	4	4

Tabel 4.8 Nilai *Precision*, *Recall* dan *Accuracy* Dari Setiap Simulasi

No	Nama Lokasi Awal	Nama Lokasi Tujuan	Precision	Recall	Accuracy
1	Terminal Batu	Terminal Gadang Malang	0.5000	1.0000	0.9681
2	Terminal Arjosari Malang	Terminal Gadang Malang	0.5000	1.0000	0.9574
3	Terminal Arjosari Malang	Terminal Landungsari Malang	1.0000	1.0000	1.0000
4	Jl. Bendungan Sutami Malang	Terminal Landungsari Malang	0.1905	1.0000	0.6383
5	Terminal Landungsari Malang	Jl. Bendungan Sutami Malang	1.0000	1.0000	1.0000
6	Terminal Arjosari Malang	Jl. Sumbersari Malang	1.0000	1.0000	1.0000
7	Jl. Sumbersari Malang	Jl. Surabaya Malang	0.0952	1.0000	0.1915
8	Dau	Jl. Sumbersari Malang	0.0476	1.0000	0.5745
9	Jl. Sengkaling	Jl. Letjend Sutoyo Malang	0.6250	1.0000	0.9681
10	Terminal Landungsari Malang	Jl. Merdeka Selatan Malang	1.0000	1.0000	1.0000
11	Jl. Gading Malang	Jl. Bogor Malang	0.2000	1.0000	0.6596
12	Jl. Gading Malang	Jl. Dr. Cipto Malang	0.6250	1.0000	0.9681
13	Jl. Bondowoso Malang	Terminal Arjosari Malang	0.2500	1.0000	0.7447
14	Jl. Bondowoso Malang	Terminal Landungsari Malang	0.1951	1.0000	0.6489
15	Terminal Arjosari Malang	Jl. Gading Malang	1.0000	1.0000	1.0000

Dari hasil akurasi diatas dengan perhitungan yang sudah ditampilkan, dihasilkan bahwa *Graph Database* dapat menemukan rute perjalanan transportasi umum. Setiap data percobaan dengan nilai parameter lokasi awal dan tujuan yang berbeda, memiliki nilai akurasi yang berbeda-beda. Selanjutnya nilai rata-rata dari semua simulasi yang sudah dilakukan ditampilkan pada Tabel 4.9.

Tabel 4.9 Nilai *Precision*, *Recall* dan *Accuracy* Akhir

No	Variabel	Nilai Rata-Rata	Nilai Presentase
1	<i>Precision</i>	0.5486	54.86 %
2	<i>Recall</i>	1.0000	100 %
3	<i>Accuracy</i>	0.8213	82.13 %

4.5 Integrasi Islam

Aplikasi ini membantu masyarakat untuk mengetahui rute angkutan umum yang akan digunakan ketika bepergian. Masyarakat pengguna angkot, terutama pendatang terkadang tidak mengetahui rute trayek angkotan sehingga diharuskan menanyakan kepada orang lain. Dalam hal ini, salah satu narasumber ketika diwawancarai mengutarakan bahwa dia pernah ditipu oleh oknum supir angkot karena ketidak tahuannya angkot mana yang akan digunakan.

Dalam Islam, penuturan berita yang tidak benar sangat dilarang syariat. Berita bohong dapat menyebabkan pelaku yang mengucapkan mendapatkan dosa, serta korban menjadi dirugikan. Dalam Al-Qur'an surat Al Hujarat ayat 6 Allah berfirman:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْحِحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ

Artinya: *“Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu.”*

Dalam kasus yang dialami narasumber diatas mendapatkan informasi yang tidak benar karena memang sebelumnya pengguna tidak mengetahui hendak memakai transportasi apa yang digunakan. Pengguna menanyakan kepada supir angkot, namun mungkin karena suatu alasan, supir angkot tersebut memberikan informasi yang tidak benar agar pengguna mau menaiki angkotnya. Maka, diperlukan sebuah informasi yang dapat diakses pengguna terkait dengan rute angkot untuk mengetahui rute yang bisa dilalui sesuai dengan data yang ada.



BAB V

PENUTUP

5.1 Kesimpulan

Dalam penggunaan angkot, pengguna diharuskan untuk mengikuti rute trayek yang sudah ditentukan. Hal ini dikarenakan angkot tidak bisa mengganti jalur sesuai dengan permintaan pengguna, karena sudah ada ketentuan trayek yang berlaku bagi masing-masing angkot. Pengguna juga harus menentukan hendak memakai angkot dengan kode trayek yang mana ketika hendak menggunakan angkot.

Penggunaan Graph Database untuk menentukan rute angkot dapat dilakukan dengan cara kombinasi *Cypher Query* dengan kemudian dilakukan perhitungan menggunakan *Analytical Hierarchy Process (AHP)* untuk menentukan rute terbaik. Implementasi Graph Database ini dapat menyelesaikan persoalan pencarian rute angkot yang ada. Hasil dari penelitian ini menemukan bahwa Implementasi Graph Database serta dilakukan perhitungan dengan AHP mendapatkan nilai *Precision* yang berarti bahwa data yang dihasilkan antara alternatif satu dengan yang lain memiliki tingkat kemiripan sebesar 54.86 % , nilai *Recall* yang menandakan jumlah data seharusnya yang dihasilkan sebesar 100%, serta nilai *Accuracy* yang menandakan tingkat akurasi sebesar 82.13 %.

5.2 Saran

Dalam penelitian ini ada beberapa kekurangan yang semoga bisa diselesaikan oleh kajian-kajian selanjutnya. Salah satunya terkait dengan penyaringan data alternatif rute angkutan umum yang masih dilakukan bukan di

dalam *Graph Database* tetapi melalui kode program dengan bahasa pemrograman Java. Diharapkan kajian selanjutnya bisa menerapkan penyaringan data langsung melalui *Cypher Query*.

Pada penelitian ini juga belum mengimplementasikan API dari penyedia maps seperti Google Maps. Pendataan lokasi masih dengan cara manual yaitu mengambil titik lokasi dari *Google Maps* kemudian dimasukkan kedalam *Graph Database Neo4j* menggunakan *Cypher Query* sehingga input data terbatas dari data yang memang sudah dimasukkan. Penggunaan API diharapkan bisa diaplikasikan dalam penelitian selanjutnya, sehingga penentuan lokasi bisa lebih dinamis.

Aplikasi yang membantu masyarakat menemukan rute angkot ini akan lebih berguna jika mendapat dukungan dari berbagai pihak. Pembuatan aplikasi semata tidak akan begitu berguna jika tidak disertai dengan pembenahan nyata di lapangan. Pemerintah kota dan supir angkot sangat berperan juga untuk menumbuhkan animo masyarakat dalam menggunakan angkot. Maka diharapkan transportasi publik angkot ini bisa dibenahi dari berbagai macam sisi.

DAFTAR PUSTAKA

Abdussakir, Nilna Niswatin Azizah, Fifi Framelia Nofandika. 2009. *Teori Graf*. Malang: UIN Maliki Press.

Aji Sasongko, Indah Fitri Astuti, Septya Maharani Maharani. 2017. *Pemilihan Karyawan Baru Dengan Metode Ahp (Analytic Hierarchy Process)*. Di Jurnal Informatika Mulawarman, Vol. 12, No. 2 September 2017.

Aminudin, Muhammad. 2018. *Sopir Angkot di Malang Demo, ini yang Dilakukan Dishub*. <https://news.detik.com/>. Diakses pada 13 Agustus 2018.

Bruggen, Rik Van. 2013. *Finding the Shortest Path through the Park*. <https://neo4j.com/>. Diakses pada 14 September 2018.

Gross, Jonathan L, Jay Yellen. 2006. *Graph theory and its applications*. Florida: Chapman and Hall/CRC.

Lestari, Nisaa Putri. 2016. *Uji Recall And Precision Sistem Temu Kembali Informasi Opac Perpustakaan Its Surabaya*: Universitas Airlangga. Vol. 5, No. 3.

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.

Panji Wisnu, Wirawan, Djalal. 2017. *Kajian Implementasi Graph Database pada Rute Bus Rapid Transit*: Universitas Diponegoro, Vol. 3, No. 3.

Panji Wisnu, Wirawan, Djalal, Khadijah. 2016. *Pemodelan Graph Database Untuk Moda Transportasi Bus Rapid Transit*: Universitas Diponegoro, Vol. 10, No. 2.

Rabbani, Imdad. 2017. *Mobile App Pencarian Rute Transportasi Umum Kota Malang dengan Algoritma Best-Path Planning*. Malang, Jawa Timur, Indonesia: Fakultas Sains dan Teknologi. Universitas Islan Negeri Maulana Malik Ibrahim Malang.

Robinson, Ian, Jim Webber, Emil Eifrem. 2015. *Graph Databases*. Sebastopol: O'Reilly Media, Inc.

Sidik, Fajar. 2018. *Pengguna Perangkat Mobile di Indonesia Semakin Tinggi, Ini Datanya!*. <http://industri.bisnis.com/>. Diakses pada 10 Oktober 2018.

Wikipedia. 2018. *Directed acyclic Graph*. <https://en.wikipedia.org/>. Diakses pada 11 Oktober 2018.

Novianti, Setiawan, Kusumawardani. *Peningkatan Nilai Recall dan Precision pada Penelusuran Informasi Pustaka Berbasis Semantik*. Konferensi Nasional Sistem & Informatika 2015.