

Permasalahan Jalur Terpendek pada Aplikasi Ojek Online *Go-Jek*

Alvin Junianto Lan 13514105
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika ITB
Bandung, Indonesia
Alvin.junianto@gmail.com

Abstrak—Algoritma yang ditemukan oleh Edsger Dijkstra merupakan algoritma yang sudah banyak digunakan untuk menentukan rute terpendek. Algoritma ini mengembangkan konsep algoritma *greedy* yang sudah ada sebelumnya sehingga permasalahan untuk menentukan rute terpendek menjadi jauh lebih efektif.

Pada makalah ini akan dibahas lebih lanjut implementasi nyata dari algoritma Dijkstra yang merupakan pengembangan dari algoritma Greedy pada aplikasi ojek online yang saat ini sedang ramai digunakan di Indonesia yaitu *Go-Jek*. Dalam aplikasi ini terdapat beberapa fungsi yang mengharuskan mencari jarak terpendek misalnya untuk menentukan armada yang akan menjemput dan jalan yang harus dilalui untuk mencapai lokasi tujuan. Dalam kasus-kasus tersebut Algoritma Dijkstra sebagai salah satu algoritma yang paling mangkus untuk permasalahan tersebut sangat mungkin digunakan guna memperoleh hasil yang maksimal dari penggunaan aplikasi tersebut.

Keywords—*uber; taxi; djikstra; rute terpendek;*

I. PENDAHULUAN

Kebutuhan akan transportasi umum yang murah, nyaman, cepat, mudah dan aman merupakan masalah klasik yang sudah sejak lama ada di Indonesia. Beberapa transportasi yang sudah ada saat ini dirasa masih belum dapat menciptakan rasa aman dan nyaman dengan harga yang murah bagi penggunanya, namun di era perkembangan teknologi yang sangat pesat ini konsep untuk memiliki kendaraan umum yang murah, aman dan nyaman serta mudah digunakan sangat mungkin untuk diciptakan. Dengan mengoptimalkan berbagai aspek seperti rute yang digunakan dan ketersediaan kendaraan dapat memotong sebagian biaya sehingga dapat diperoleh biaya yang efisien untuk penggunaan kendaraan umum tersebut.

Go-Jek adalah salah satu perusahaan yang berusaha mewujudkan konsep kendaraan umum yang murah, nyaman, cepat, mudah dan aman di Indonesia. Lalu lintas Indonesia yang terkenal padat dan seringkali terjadi kemacetan membuat kendaraan roda 2 dianggap cukup efisien untuk menghindari kemacetan tersebut sehingga perjalanan dapat memakan waktu lebih singkat, namun kendaraan umum roda 2 seperti ojek tradisional masih memiliki kekurangan seperti sisi kemandirian dan kenyamanannya. Selain itu kadang biaya menggunakan ojek tradisional dirasa tidak sesuai karena harga hanya ditentukan berdasarkan perkiraan tukang ojeknya saja dan

seringkali ojek sulit ditemukan dari tempat-tempat yang jauh dari pangkalan ojek. Melihat kondisi inilah Go-Jek memanfaatkan konsep ojek yang dimodernkan guna mencapai optimalisasi penggunaan ojek. Selain itu perusahaan yang berbasis aplikasi mobile ini membuat konsep yang lebih efisien dalam memanfaatkan kendaraan roda 2, Go-Jek menghubungkan calon penumpang dengan pengemudi sepeda motor terdekat yang bersedia mengantarkan penumpang ke tempat tujuannya sehingga waktu yang terbuang untuk menunggu ataupun perjalanan menuju pangkalan ojek terdekat dapat diminimalisir. Selain pemilihan pengemudi terdekat juga aplikasi Go-Jek memberikan rute terpendek yang dapat dilalui pengemudi untuk mengantarkan penumpangnya sampai ke tujuan, dengan demikian kekurangan ojek tradisional yang mengandalkan kemampuan setiap pengemudi untuk menentukan rute menuju tujuan yang seringkali kurang efektif sehingga memakan biaya bahan bakar dan waktu yang lebih banyak dapat dipangkas.



Gambar 1.1 . Logo Go-Jek

(Sumber : <http://www.go-jek.com/>)

Algoritma Dijkstra yang digunakan untuk menyelesaikan permasalahan mencari rute terpendek dapat diterapkan untuk menentukan pengemudi terdekat dan juga jalan yang akan dilalui pengemudi Go-Jek tadi.

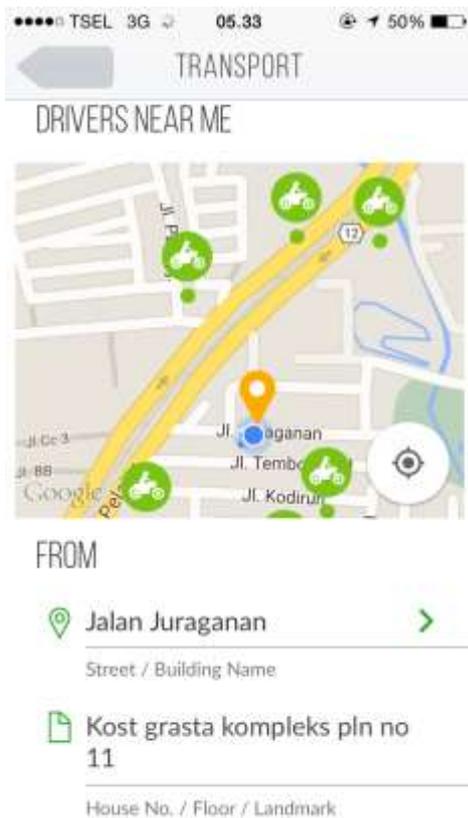
II. GO-JEK

A. Deskripsi

Go-Jek adalah perusahaan yang memperbaharui transportasi umum motor di Indonesia. Go-Jek bermitra

dengan para pengendara sepeda motor di berbagai daerah meliputi area JABODETABEK, Bandung, Bali & Surabaya dan menjadi solusi utama dalam pengiriman barang, pesan antar makanan, berbelanja dan berpergian di tengah kemacetan.

Dengan menggunakan aplikasi Go-Jek pengguna dapat memesan ojek untuk mengakses semua layanan yang disediakan. Beberapa fitur yang terdapat dalam aplikasi Go-Jek ini antara lain dapat memperoleh perkiraan harga penggunaan ojek, layanan 'Use my location' dapat digunakan untuk mengarahkan Driver ke tempat pengguna berada, teknologi berbasis lokasi yang terdapat dalam aplikasi akan mencari mencari pengemudi yang posisinya paling dekat dengan pengguna dan pengguna dapat melihat profil dari pengemudinya, mengirimkan sms dan juga meneleponnya.



Gambar 2.1 . Tampilan Aplikasi Go-Jek
(Sumber : <https://griagowes.com>)

B. Sejarah

Go-Jek lahir dari ide sang CEO dan Managing Director Nadiem Makarim yang mengaku seorang pengguna ojek. Ojek yang merupakan kendaraan motor roda dua ini memang transportasi yang sangat efektif untuk mobilitas di kemacetan kota. Dengan pengalamannya saat naik ojek di jalanan yang macet inilah ia kemudian menciptakan Go-Jek, sebuah layanan antar jemput dengan ojek modern berbasis pesan. PT Go-Jek Indonesia yang sudah melewati perjalanannya sejak tahun 2011 kini sudah memiliki 1.000 armada ojek yang tersebar di seluruh kawasan Jabodetabek. Selain itu, Go-Jek telah memperoleh

berbagai penghargaan dari berbagai komunitas bisnis maupun sosial. Di situs resminya disebutkan bahwa layanan Go-Jek adalah sebagai berikut :

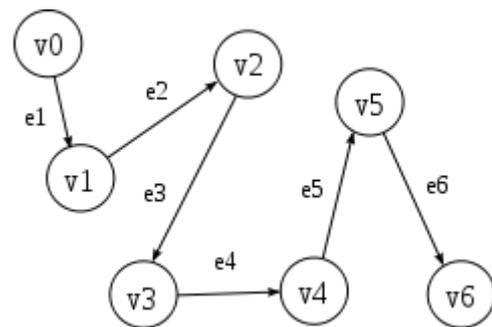
1. Jasa kurir (90 minute delivery anywhere in the city)
2. Jasa transportasi (transparent pricing, free shower cap and masker)
3. Jasa delivery makanan (delivering your favorite food under 60 minutes in Jabodetabek)
4. Jasa belanja dengan nominal di bawah 1 juta rupiah (shop for food, ticket, medicine, anything under Rp 1.000.000. We'll pay for it first).

Dari beberapa testimoni pengguna Go-Jek bisa ditarik satu kesimpulan yang sama bahwa mereka sangat puas dengan alasan yang hampir sama pula yaitu tarif yang jelas dan transparan selain tentu saja sistemnya yang mengakomodir gaya hidup modern saat ini yaitu penggunaan teknologi. Calon pengguna hanya perlu melakukan instalasi aplikasi Go-Jek di smartphone-nya untuk kemudian melakukan pemesanan sekaligus memantau status pemesanannya. Modern, transparan, dan profesional.

III. LANDASAN TEORI

A. Graf

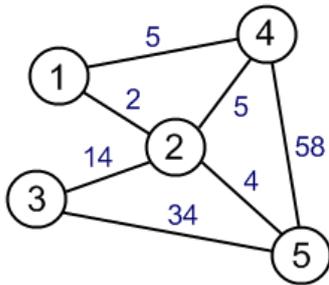
Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara satu objek dengan objek lainnya. Graf biasanya dituliskan sebagai himpunan simpul yang terhubung oleh sisi atau busur, umumnya digambarkan sebagai kumpulan titik-titik yang melambangkan simpul dan dihubungkan oleh garis-garis yang melambangkan sisi atau garis dengan panah yang melambangkan busur.



Gambar 3.1 . Contoh Graf
(Sumber : <http://danysatriokintoko.blogspot.co.id/>)

B. Graf Berbobot

Graf berbobot adalah graf yang setiap sisinya terdapat nilai yang merepresentasikan bobot untuk melalui sisi tersebut.



Gambar 3.2 . Graf Berbobot

(Sumber: web.cecs.pdx.edu/sheard/course/Cs163/Doc/Graphs.html)

C. Algoritma Greedy

Algoritma greedy adalah algoritma yang digunakan untuk menentukan hasil dari persoalan optimasi. Ada 2 macam persoalan optimasi yaitu maksimum dan minimum. Pada setiap langkah, pilihan yang menghasilkan nilai optimal paling besar (local optimal) diambil sebagai bagian dari optimal total (global optimal) dengan harapan bahwa total dari local optimal tersebut merupakan hasil/tidak jauh dari hasil optimal global sebenarnya.

Dua syarat algoritma greedy adalah :

- Greedy Choice Property: dari local optimal kita bisa mencapai global optimal tanpa harus mengganti/mempertimbangkan keputusan yang sudah diambil.
- Optimal Substructure Property: solusi optimal dari permasalahan dapat ditentukan dari subsolusi permasalahan tersebut.

Elemen-elemen algoritma greedy:

1. Himpunan kandidat, C.
2. Himpunan solusi, S
3. Fungsi seleksi (selection function)
4. Fungsi kelayakan (feasible)
5. Fungsi obyektif

D. Algoritma Lintasan Terpendek

Algoritma lintasan terpendek merupakan algoritma yang digunakan untuk menentukan lintasan yang diambil untuk pergi dari suatu titik ke titik lain. Diberikan graf berbobot $G = (V, E)$. Tentukan lintasan terpendek dari sebuah simpul asal a ke setiap simpul lainnya di G .

Ada 4 macam bentuk algoritma shortest path:

- a) Lintasan terpendek antara dua buah simpul tertentu (a pair shortest path).
- b) Lintasan terpendek antara semua pasangan simpul (all pairs shortest path).

- c) Lintasan terpendek dari simpul tertentu ke semua simpul yang lain (single-source shortest path).
- d) Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (intermediate shortest path). Asumsi yang kita buat adalah bahwa semua sisi berbobot positif.

E. Algoritma Dijkstra

Algoritma Dijkstra adalah algoritma pencarian graf yang memecahkan masalah jalur terpendek yang bersumber dari satu simpul untuk sebuah graf dengan bobot simpul tidak boleh negatif. Analisis dilakukan dengan cara memeriksa simpul dengan bobot terkecil dan memasukkannya ke dalam himpunan solusi dengan awal pencarian simpul asal membutuhkan pengetahuan tentang semua jalur dan bobotnya, sehingga dibutuhkan pertukaran informasi dengan semua simpul. Algoritma dijkstra memiliki sifat yang sederhana dan straightforward, sesuai dengan prinsip kerja greedy.

Algoritma dijkstra dapat menyelesaikan jalur terpendek dari sebuah titik asal dan titik tujuan dalam suatu graf berbobot. Jarak terpendek diperoleh dari dua buah titik jika total bobot dari semua simpul dalam jaringan graf adalah yang paling minimal.

Sebelum meneruskan lebih lanjut perlu diketahui beberapa notasi yang digunakan:

1. $l(i,j)$: panjang jarak antara simpul i ke simpul j
2. a : titik awal pencarian
3. d_{ai} : jarak yang dikenal terpendek dan bersifat permanen dari titik awal ke titik (i) dalam suatu jaringan.
4. q_i : titik terdahulu yang dikenal terpendek dari titik awal ke titik (i) dalam suatu jaringan.
5. c : titik terakhir yang telah pindah ke keadaan permanen.

Langkah prosedural algoritma dijkstra sebagai berikut :

1. Proses dimulai dari titik awal (a), dan titik yang satu-satunya permanen adalah titik a . Dan titik lain berlabel sementara yang diisi dengan inisial tak hingga.
2. Memeriksa cabang titik yang keluar dari titik terakhir yang bersifat permanen dengan persamaan dengan persamaan :

$$d_{ai} = \min[d_{ai}, d_{ac} - l(c,i)]$$

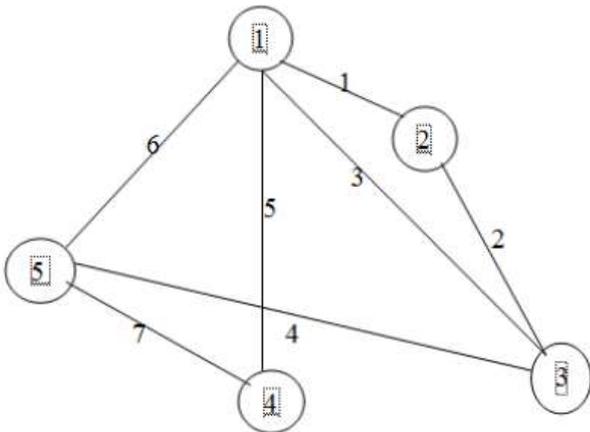
3. Menentukan titik mana yang akan lulus dari label sementara menjadi label permanen. Dengan cara membandingkan nilai titik dari hasil langkah kedua dan diambil nilai titik paling minimum. Kemudian untuk mengetahui nilai titik permanen terdahulu, masukkan kedalam persamaan:

$$[d_{ai} - l(i,j)] = \dots d_{ai}$$

4. Setelah mendapatkan titik dengan nilai paling minimum, maka titik yang bernilai minimum tersebut ditetapkan sebagai titik permanen berikutnya.

5. Jika masih terdapat titik yang belum berlabel permanen maka ulangi kembali dari langkah 2.

Di bawah ini diberikan contoh sebuah graf tak berarah yang terdiri dari 5 buah simpul dan 7 buah. Algoritma djikstra digunakan untuk mencari jarak terpendek dari sebuah titik ke titik lainnya pada graf tak berarah tersebut.



Gambar 3.3 . Soal Penjelasan Algoritma Dijkstra
(Sumber : http://eprints.dinus.ac.id/5180/1/P6-TI24-SEMANTIK-80_Farida_Ardiani-UII_Yogyakarta.pdf)

Berdasarkan contoh graf tak berarah di atas ditentukan titik awal pencarian adalah simpul 1 dengan tujuan yaitu simpul 4 dan akan dicari jarak terpendek yang dapat ditempuh dari simpul 1 untuk menuju simpul 4. Berikut ini tabel penjelasan graf menggunakan algoritma Dijkstra :

Tabel 3.1. Tabel Sisi

Sisi	Sisi Awal				
	1	2	3	4	5
	0	0	0	0	0
1	1	0	0	0	0
1-2	1	1	0	0	0
2-3	0	1	1	0	0
3-5	0	0	1	0	0
5-4	0	0	0	1	1

Tabel 3.2. Tabel Simpul

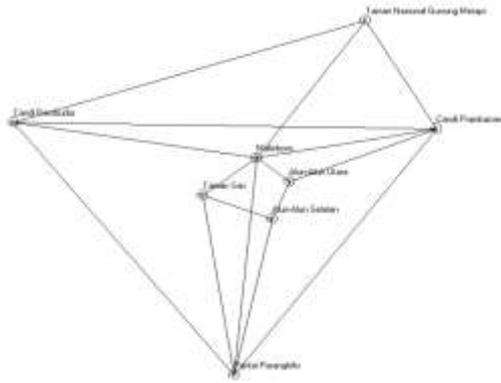
Simpul	l(i,j)				
	1	2	3	4	5
1	∞	∞	∞	∞	∞
2	1	∞	∞	∞	∞
3	3	2	∞	∞	∞
4	5	∞	∞	∞	7
5	6	∞	4	∞	∞

Penyelesaian algoritma djikstra jalur simpul 1 ke simpul 4 telah diselesaikan seperti pada penjelasan dan tabel di atas menurut perhitungan penelusuran graf sesuai langkah prosedural algoritma djikstra. Pada baris pertama semua successor di set 0 artinya untuk memberi nilai pada sumber titik rute yang akan dijadikan rute dan ketidakterbatasan untuk semua titik lain, yang menyatakan fakta bahwa tidak diketahui lintasan manapun.

Untuk selanjutnya karena simpul 1 sebagai sumber lintasan maka sudah pasti terpilih. Sehingga status set 0 berubah menjadi 1. Simpul 1 akan cek titik yang bertetangga langsung yaitu simpul 2, 3, 4 dan 5. Dari situ djikstra akan memilih yang mempunyai bobot terendah untuk menuju simpul selanjutnya. Terpilih simpul 2 dengan bobot 1, set status 0 berubah menjadi 1 dan seterusnya. Maka dari pencarian jarak terpendek di atas, didapat lintasan yang terpendek berdasarkan pencarian djikstra dari simpul 1 ke 4 adalah melalui simpul 1 langsung simpul 4 dengan bobot lintasan 5.

IV. PENERAPAN ALGORITMA DIJKSTRA PADA APLIKASI GO-JEK

Guna mencari pengemudi terdekat dan rute terdekat untuk tiba di tujuan pada aplikasi Go-Jek dengan cara yang lebih optimal dapat menerapkan Algoritma Dijkstra. Selain itu Go-Jek yang mengakomodir kerdaraan sepeda motor, maka nantinya akan sering ditemukan jalur yang tidak dapat dilewati oleh pengemudi misalnya jalan tol. Contoh penerapannya adalah sebagai berikut :



Gambar 4.1 Contoh Kasus

(Sumber : <http://eprints.dinus.ac.id/5180/1/P6-TI24-SEMANTIK-80> Farida Ardiani-UII Yogyakarta.pdf)

Tabel 4.1. Keterangan Lokasi Gambar 4.1

Simpul	Lokasi
1	Pantai Parangtritis
2	Malioboro
3	Alun-Alun Utara
4	Alun-Alun Selatan
5	Taman Sari
6	Candi Prambanan
7	Taman Nasional Gunung Merapi
8	Candi Borobudur

Tabel 4.2. Keterangan Jarak Lokasi Gambar 4.1

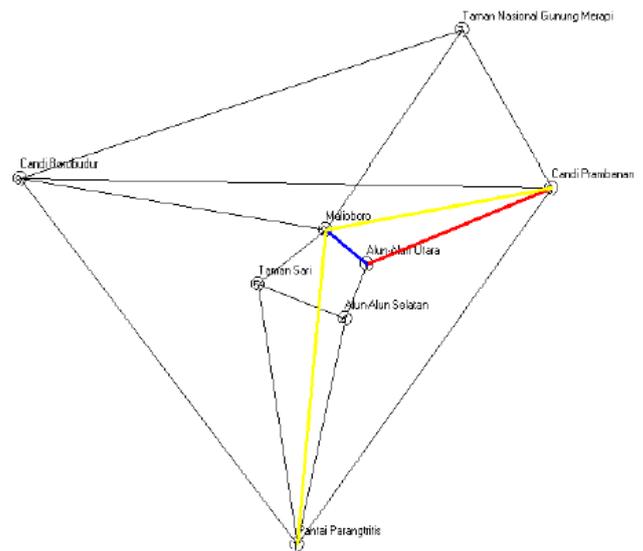
Sisi	Jarak (Km)
1-2	30
1-4	25
1-5	28
1-6	48
1-8	65
2-3	1
2-5	3
2-6	17
2-7	29
2-8	42
3-4	1

3-6	15
4-5	3
6-7	27
6-8	57
7-8	46

Misalkan Jalan dari Malioboro ke Alun-Alun Utara tidak dapat dilewati oleh motor. Posisi awal adalah simpul 1 (Pantai Parangtritis) dan tujuan adalah simpul 6 (Candi Prambanan).

Dengan menerapkan Algoritma Dijkstra sesuai proses yang dijelaskan pada bagian 3.E hasil dengan menghilangkan jalur 2-3 karena jalur tersebut tidak dapat dilewati maka akan diperoleh hasil seperti berikut :

Jarak terpendek = 57 km



Gambar 4.2. Solusi Contoh Kasus

(Sumber : <http://eprints.dinus.ac.id/5180/1/P6-TI24-SEMANTIK-80> Farida Ardiani-UII Yogyakarta.pdf)

Untuk mencari pengemudi terdekat nantinya yang dijadikan posisi awal adalah lokasi setiap pengemudi dan posisi akhir adalah lokasi pengguna, dan pengemudi yang dipilih nantinya adalah pengemudi dengan lokasi yang memiliki jalur terpendek paling kecil menuju lokasi pengguna.

Sedangkan untuk rute perjalanan, posisi awal adalah lokasi penjemputan pengguna dan tujuan adalah lokasi pengguna dan hasil penerapan Algoritma Dijkstra dapat langsung digunakan pengemudi untuk memilih jalur yang akan dilewati.

V. KESIMPULAN

Pada makalah ini dijelaskan penggunaan algoritma Dijkstra pada aplikasi Go-jek khususnya untuk fitur menentukan pengemudi terdekat dan rute yang akan ditempuh oleh pengemudi. Pada bagian analisis digunakan data mengenai lokasi yang sesungguhnya dari berbagai tempat di Indonesia dan kemudian Algoritma Dijkstra akan memperkirakan jarak terpendek dengan menghilangkan beberapa jalur jika terdapat jalur yang tidak dapat dilewati oleh armada Go-Jek.

Jika memperhatikan kekurangan Algoritma Dijkstra sebagai pencari jalur terpendek adalah tidak dapat mengatasi bobot sisi negatif maka penggunaan Algoritma ini pada Go-Jek dinilai sudah tepat karena pada aplikasi Go-Jek, bobot yang diambil untuk setiap sisi adalah jarak antara 2 tempat sehingga nilainya tidak mungkin negatif.

Namun akan ada kalanya jika terdapat jalur yang tidak dapat dilewati atau semacamnya dapat membuat hasil perhitungan dengan Algoritma Dijkstra menjadi tidak akurat. Maka dari itu diharapkan nantinya akan ada penelitian lebih jauh mengenai jalur yang tidak dapat dilalui pada Algoritma Dijkstra sehingga tetap memberikan hasil yang akurat.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah memampukan penulis untuk menyelesaikan makalah ini dengan baik. Selanjutnya penulis juga mengucapkan terima kasih kepada orang tua penulis serta rekan-rekan di sekitar penulis yang terus memberikan dukungan dan nasihat sehingga menjadi motivasi bagi penulis dalam menyelesaikan kewajiban dalam perkuliahan di ITB.

Penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir selaku dosen mata kuliah Strategi Algoritma yang telah memberikan tugas ini serta membekali

penulis dengan materi yang berkaitan dengan Algoritma yang digunakan dan dimanfaatkan dalam penulisan makalah ini. Tidak lupa penulis juga berterima kasih kepada pihak-pihak yang secara langsung maupun tidak telah membantu dalam merampungkan makalah ini.

REFERENCES

- [1] Munir, Rinaldi. Matematika Diskrit Edisi Kedua, Bandung : Informatika, 2001, hlm. 291, 294.
- [2] Ratnasari, Asti. "Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif Menggunakan Algoritma Dijkstra serta Estimasi Waktu Tempuh", Yogyakarta : Universitas Islam Indonesia. 2013.
- [3] <http://achmad-asrori.blogspot.co.id/2013/01/algoritma-dijkstra.html> diakses pada 7 Mei 2016 15.00 WIB
- [4] <http://www.go-jek.com/> diakses pada 8 Mei 2016 10.00 WIB
- [5] <http://web.cecs.pdx.edu/~sheard/course/Cs163/Doc/Graphs.html> diakses pada 8 Mei 2016 20.00 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Mei 2016



Alvin Junianto Lan - 13514105