

APLIKASI PEMESANAN OJEK ONLINE BERBASIS ANDROID MENGUNAKAN METODE DIJKSTRA

Bagus Rahmadhian¹, Yuri Ariyanto², Dyah Ayu Irawati³

^{1,2}Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang
¹bagus.rahmadhian@yahoo.com, ²yuri.bjn@gmail.com, ³dyah.ayu.irawati@gmail.com

Abstrak

Permasalahan yang terjadi pada tukang ojek sekarang adalah, sulitnya bersaing dengan tukang ojek berbasis online yang sedang marak pada saat ini, sehingga tukang ojek yang biasanya sering di pesan oleh masyarakat menjadi jarang ada pemesan dikarenakan masyarakat pengguna tukang ojek sekarang lebih memilih menggunakan jasa ojek online karena lebih efisien dan tidak ribet.

Pada penelitian ini dibuat system pemesanan ojek online berbasis android menggunakan metode Dijkstra untuk membantu para tukang ojek agar dapat bersaing dengan “Go-Jek” dan sejenisnya sehingga para tukang ojek ini tidak kehilangan pelanggannya dan dapat meningkatkan penghasilan. Kelebihan metode Dijkstra ini adalah dapat mencari rute terdekat dari titik awal menuju titik akhir dengan membandingkan nilai terkecil antar titik yang akan dijadikan jalur yang nantinya akan dilewati oleh tukang ojek agar sampai lebih cepat menuju tujuan.

Hasil pengujian yang dilakukan yaitu dengan menggunakan sampel data jalan dan pangkalan tukang ojek di sekitar Kota Malang daerah Jalan Soekarno-Hatta dan sekitarnya, lalu memberi titik atau marker pada peta google maps pada jalan dan pangkalan ojek yang akan di tandai. Lalu marker tersebut akan dihubungkan dan tersimpan sehingga jarak antar marker dapat diketahui yang nantinya akan dihitung jarak terdekat antar marker lalu diambil nilai terkecil dari jarak antar marker tersebut dan diperoleh rute mana saja yang akan dilewati agar sampai ke tempat tujuan lebih cepat.

Kata kunci : Ojek Online, Pencarian Rute Terdekat, Dijkstra

1. Pendahuluan

Ojek merupakan salah satu alat transportasi darat yang banyak diminati oleh masyarakat, terutama masyarakat menengah kebawah. Jasa ojek dapat digunakan oleh masyarakat dengan cukup mudah. Masyarakat dapat memesan jasa ojek dengan menghampiri tukang ojek terdekat secara langsung, atau menghubungi tukang ojek dengan sms atau telephone.

Saat ini salah satu penyedia layanan telekomunikasi yang berkembang pesat di Indonesia adalah Android. Pertumbuhan pemakaian smartphone berbasis android dari tahun ke tahun semakin meningkat. Kemacetan yang sering terjadi di kota besar seperti Malang Kota, membuat jasa tukang ojek semakin banyak diminati oleh masyarakat, karena tukang ojek dapat mencari jalur yang cepat menuju tujuan untuk dilalui. Namun pada saat ini, jasa tukang ojek semakin tergerus dengan banyaknya jasa ojek online seperti “Go-Jek” dan lainnya. Untuk itu diperlukan suatu system yang dapat membantu tukang ojek khususnya tukang ojek di Malang Kota agar dapat bersaing dengan ojek online seperti “Go-Jek” dan lain lainnya.

Hampir semua type smartphone berbasis android yang di distribusikan di Indonesia memiliki perangkat GPS (Global Positioning System) yang tertanam, yang memudahkan untuk mencari lokasinya saat itu. Dengan memanfaatkan fasilitas GPS yang membantu mencari lokasi pelanggan, maka dapat dikembangkan system untuk pemesanan tukang ojek online lewat smartphone android.

Pada penelitian ini, digunakan metode untuk memberikan informasi perhitungan jarak terdekat antara pelanggan dengan tukang ojek melalui perangkat mobile dengan menggunakan aplikasi google maps. Dengan adanya aplikasi, diharapkan dapat memudahkan pelanggan dan tukang ojek untuk mencari rute terdekat dari lokasi pemesan menuju lokasi tujuan pemesan dan rute terdekat dari lokasi tukang ojek menuju lokasi pemesan.

2. Metode Dijkstra

2.1 Konsep Dasar Dijkstra

Metode Dijkstra (Dinamai menurut penemunya, seorang ilmuwan komputer Edsger Dijkstra), adalah sebuah algoritma yang dipakai dalam memecahkan permasalahan jarak terpendek (Shortest path problem) untuk sebuah graf berarah (directed graph) dengan bobot-bobot sisi (edge weight) yang bernilai tak negative.. Himpunan

semua tepi disebut E. Bobot (weight) dari semua sisi dihitung dengan fungsi

$$w: E \rightarrow [0, \infty]$$

jadi $w(u,v)$ adalah jarak tak negative dari vertex u ke vertex v. Ongkos (cost) dari sebuah sisi dapat dianggap sebagai jarak antara dua vertex, itu jumlah jarak semua sisi dalam jalur tersebut. Untuk sepasang vertex s & t dalam V, algoritma ini menghitung jarak terpendek dari s ke t. Algoritma Dijkstra adalah suatu algoritma untuk menentukan jalur terpendek antar node dengan berdasar pada basis penghitungan “dari satu node menuju seluruh node”. Algoritma Dijkstra termasuk dalam jenis algoritma Link State, itu memperhatikan total jarak dan rute yang akan dilalui. Pertama, kita ketahui 2 verteks X (Source Vertex) dan Y (DestinationVertex) dan sisi yang menghubungkan mereka. Adapun rumusnya seperti berikut :

$$\text{Min}(\text{DestValue}, \text{MarkedValue} + \text{EdgeWeight})$$

Keterangan :

DestValue=Nilai dalam vertex tujuan.

MarkedValue= Nilai dalam vertex awal.

EdgeWeight= bobot dari sisi yang menghubungkan vertex.

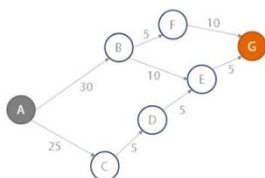
Contoh:

Jika DestValue=15,MarkedValue=7 dan EdgeWeight = 5

Min (15,12) = 12 karena 12 lebih kecil dari 15 (minimum

Berikut ini contoh kasus dari perhitungan Metode Dijkstra:

2.2 Contoh Kasus



Gambar 2.1 Contoh Graph

Pada penelitian kali ini penulis menggunakan Metode Dijkstra untuk mencari rute terpendek dari node A ke node G. Kolom marked berisikan vertex yang telah di marked, karena starting point di marked A maka diberi nilai 0. Sedangkan untuk kolom B-G diberi nilai ∞ karena nilainya belum di ketahui.

Marked	A	B	C	D	E	F	g
A	0	∞	∞	∞	∞	∞	∞

Tabel 2.1 Baris pertama

Kemudian cari vertex yang berhubungan dengan vertex A, yang berhubungan dengan vertex A adalah vertex B dan C. maka gunakan minimum formula untuk mencari nilai terkecil antara vertex B dan vertex C yang hasilnya adalah 25. Setelah

mendapat nilai terkecil antar vertex maka nilai terkecil tersebut di tandai dengan warna merah.

Marked	A	B	C	D	E	F	G
A	0	∞	∞	∞	∞	∞	∞
C	0	$\text{Min}(\infty, 0+30)$ 30	$\text{Min}(\infty, 0+25)$ 25	∞	∞	∞	∞

Tabel 2.2 Baris kedua

Setelah vertex c di tentukan sebagai vertex dengan nilai terkecil maka perhitungan dilanjutkan dengan mencari vertex yang terhubung dengan vertex C, pada gambar contoh graph, vertex C hanya berhubungan dengan vertex D. Setelah itu masukan minimum formula untuk mencari nilai terkecilnya.

Marked	A	B	C	D	E	F	G
A	0	∞	∞	∞	∞	∞	∞
C	0	$\text{Min}(\infty, 0+30)$ 30	$\text{Min}(\infty, 0+25)$ 25	∞	∞	∞	∞
D	0	30	25	$\text{Min}(\infty, 25+5)$ 30	∞	∞	∞

Tabel 2.3 Baris ketiga

Setelah menghitung nilai dari vertex D yang diketahui nilai terkecilnya adalah 30, maka dilanjutkan dengan menghitung vertex yang terhubung dengan vertex D yaitu vertex E. karena nilai vertex E (DestValue) belum diketahui maka dalam minimum formula diberi nilai ∞ .Lalu vertex D yang berhubungan dengan vertex E. Masukan perulangan dalam kolom EMin(DestValue ∞ , MarkValue dari D yaitu 30, EdgeWeight 5).

Marked	A	B	C	D	E	F	G
A	0	∞	∞	∞	∞	∞	∞
C	0	$\text{Min}(\infty, 0+30)$ 30	$\text{Min}(\infty, 0+25)$ 25	∞	∞	∞	∞
D	0	30	25	$\text{Min}(\infty, 25+5)$ 30	∞	∞	∞
E	0	30	25	30	$\text{Min}(\infty, 30+5)$ 35	∞	∞

Tabel 2.4 Baris keempat\

Setelah menghitung vertex E, maka sekarang menghitung vertex yang terhubung oleh vertex E yaitu vertex B dan vertex G. Saat ini mulai menghitung dari vertex B terlebih dahulu. Gunakan minimum formula masukan DestValue karena sudah diketahui yaitu 30 lalu masukan MarkValue dari E yaitu 30 dan Edge Weight dari vertex E ke B yaitu 10. Setelah dijumlahkan maka nilainya adalah 30 dan 40 lalu pilih nilai terkecil yaitu 30.

Marked	A	B	C	D	E	F	G
A	0	∞	∞	∞	∞	∞	∞
C	0	$\text{Min}(\infty, 0+30)$ 30	$\text{Min}(\infty, 0+25)$ 25	∞	∞	∞	∞
D	0	30	25	$\text{Min}(\infty, 25+5)$ 30	∞	∞	∞
E	0	30	25	30	$\text{Min}(\infty, 30+5)$ 35	∞	∞
B	0	$\text{Min}(30, 35+10)$ 30	30	30	35	∞	$\text{Min}(\infty, 35+5)$ 40

Tabel 2.5 Baris kelima

Tabel hasil semua perhitungan

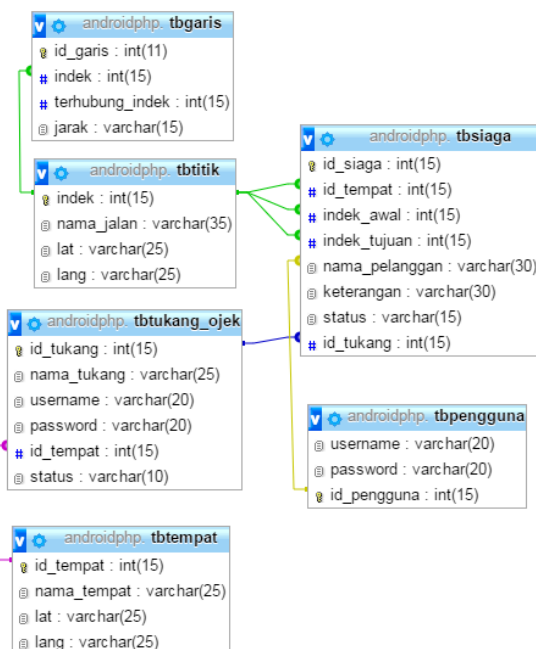
Marked	A	B	C	D	E	F	G
A	0	∞	∞	∞	∞	∞	∞
C	0	$\text{Min}(\infty, 0+30)$ 30	$\text{Min}(\infty, 0+25)$ 25	∞	∞	∞	∞
D	0	30	25	$\text{Min}(\infty, 25+5)$ 30	∞	∞	∞
E	0	30	25	30	$\text{Min}(\infty, 30+5)$ 35	∞	∞
B	0	$\text{Min}(30, 35+10)$ 30	30	30	35	∞	$\text{Min}(\infty, 35+5)$ 40
F	0	30	30	30	35	$\text{Min}(\infty, 30+5)$ 35	40
G	0	30	30	30	35	35	$\text{Min}(40, 35+10)$ 40

Tabel 2.6 Hasil semua perhitungan

3. Perancangan dan Implementasi

3.1.1 Perancangan Sistem

Desain database pada aplikasi ini adalah sebagai berikut:



Gambar 3.1 Desain Database

3.1.2 Alat Penelitian

Alat yang diperlukan dalam penelitian ini, antara lain: Kebutuhan software yang dibutuhkan adalah

- Bahasa pemrograman PHP dan Java
- MySql XAMPP
- Adobe Dreamweaver
- Android Studio
- Hosting Online

Sedangkan kebutuhan hardware yang dibutuhkan adalah

- 1 unit laptop / Personal Computer (PC).
- 1 unit printer.
- 2 unit handphone berbasis Android.

3.2 Arsitektur Sistem

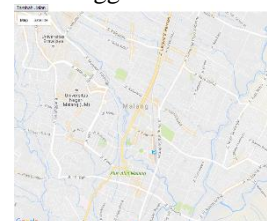


Gambar 3.2 Arsitektur Sistem

Pada diagram diatas menggambarkan siklus jalannya aplikasi, perangkat android melakukan request data menggunakan media internet untuk mengambil data yang berupa JSON dimana JSON ini sebelumnya di generate oleh PHP dari database MYSQL.

3.3 Implementasi Sistem

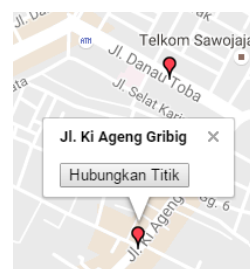
Implementasi sistem merupakan implementasi dari rancangan antarmuka sistem yang telah dibuat pada bab sebelumnya. Rancangan yang dibuat tersebut kemudian diimplementasikan untuk membangun aplikasi pemesanan ojek online berbasis android menggunakan metode Dijkstra.



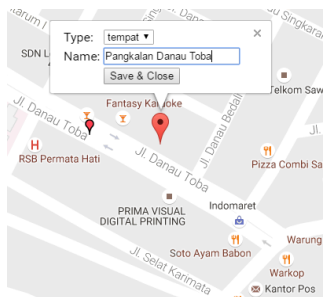
Gambar 3.4 Halaman Tampil Peta Google



Gambar 3.5 Tambah Data Jalan



Gambar 3.6 Hubungkan antar marker



Gambar 3.7 Tambah data pangkalan



Gambar 3.8 Data pangkalan tersimpan



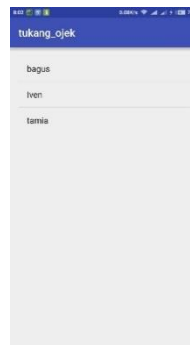
Gambar 3.9 Halaman Login Pengguna



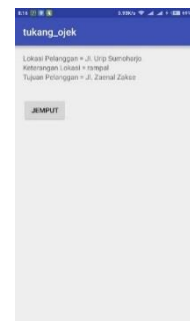
Gambar 3.10 Halaman Pemesanan Pengguna



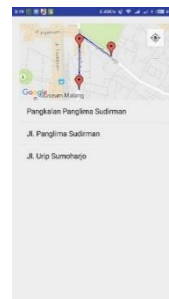
Gambar 3.11 halaman tunggu konfirmasi dan tampil rute



Gambar 3.11 Halaman Daftar Pesanan



Gambar 3.12 Halaman Tukang Ojek Memilih Pelanggan Beserta detail lokasi



Gambar 3.13 Halaman rute yang harus dilalui tukang ojek

4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada bab 1 hingga bab 6, maka dapat disimpulkan bahwa :

1. Sistem ini digunakan untuk membantu warga Kota Malang khususnya tukang ojek untuk solusi dalam memesan tukang ojek.
2. Sistem yang dibuat berhasil menentukan rute terpendek dari lokasi awal ke tujuan dan sesuai dengan perhitungan.
3. Sistem ini saat ini dibutuhkan oleh masyarakat terutama tukang ojek, terbukti

dari hasil 20 kuisisioner untuk tukang ojek dan 20 kuisisioner untuk koresponden 34 semua orang menjawab aplikasi ini dibutuhkan dan hanya 3 yang menjawab ragu ragu serta 3 orang menjawab tidak.

4.2 Saran

Saran yang diberikan untuk pengembangan sistem ini ke depannya adalah sebagai berikut :

1. Penelitian ini dapat dilanjutkan dengan metode yang lain sebagai perbandingan dalam mendapatkan hasil rute terdekat dari lokasi awal hingga ke tujuan, dan juga mencoba menampilkan marker jika aplikasi web yang berfungsi sebagai penambah jalan, garis, dan pangkalan dapat ditampilkan.
2. Perlu adanya penelitian lebih lanjut untuk pengimplementasian Metode Dijkstra pada permasalahan yang lain selain mencari jalan terdekat.

Daftar Pustaka:

- Fitria, Apri Triansyah oktober 2013. “Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan” The Informatics and Business Institute Darmajaya Bandar Lampung Indonesia 35142
- Garuda Ginting oktober 2013, Dosen Tetap Program Studi Teknik Informatika STMIK Budi Darma Medan. “Penerapan Metode Dijkstra Dalam Pencarian Jalur Terpendek Pada Lokasi Wisata Di Kabupaten Karo” Teknik Informatika STMIK Budi Darma Medan
- Stevian, Suryo Saputro 2013 “Perancangan Aplikasi Gis Pencarian Rute Terpendek Peta Wisata Di Kota Manado Berbasis Mobile Web Dengan Algoritma Dijkstra” Teknik Informatika – S1 Universitas Dian Nuswantoro Jl. Imam Bonjol No. 207 Semarang.
- Michael Purvis, Jeffrey Sambells, Cameron Turner, 2006. *Beginning Google Maps Applications with PHP and Ajax*. New York: Apress
- Gintoro, Iwan Wijaya Suharto, Febiyan Rachman, Daniel Halim 2010. “Analisis Dan Perancangan Sistem Pencarian Taksi Terdekat Dengan Pelanggan Menggunakan Layanan Berbasis Lokasi” Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010) Yogyakarta, 19 Juni 2010. Supardi, Yanuar, Ir., 2007. *Pemrograman Java dan MySQL*. Jakarta: Elex Media Komputindo.
- Wahadyo, Agus, Ir., 2013. *Android 4 untuk Pengguna Pemula Tablet & Handphone*. Jakarta Selatan: Mediakita
- Kartika Gunadi, Yulia, 2014. “perencanaan rute perjalanan di jawa timur dengan dukungan gis menggunakan metode dijkstra’s”: Universitas Kristen Petra Fakultas Teknologi Industri, Jurusan Teknik Informatika
- Gossett, Eric., 2009. *Discrete Mathematics With Proof*. Canada: John Wiley & Son
- Baddam, Arjun Reddy., 2013. “Shortest Path using Dijkstra’s and A* Algorithm”: Indiana State University, Computer Science
- Marcin, Kossakowsky., 2014. *Finding shortest path using Dijkstra’s algorithm and weighed directed graph*. <http://www.marcinkossakowski.com/finding-shortest-path-using-dijkstras-algorithm/>. <http://www.metode-algoritma.com/2013/02/algoritma-dijkstra.html/>.
- Rosen, Kenneth H., *Discrete Mathematics and Its Applications*, 3rd ed., New York, Mc Graw Hill, Inc., 199