

Implementasi Algoritma Dijkstra Dalam Penentuan Jalur Terpendek Di Yogyakarta Menggunakan GPS Dan Qt Geolocation

Blasius Neri Puspika⁽¹⁾
22074290@students.ukdw.ac.id

Antonius Rachmat C⁽²⁾
anton@ti.ukdw.ac.id

Erick Kurniawan⁽³⁾
erick@ukdw.ac.id

Abstract

With the development of information technology, the map is no longer in the form of sheet or book. Currently there is a digital map services already invested in mobile devices. Google Maps is one of the leading providers of online digital map which can be accessed using the Application Programming Interface (API) is available using tools such as Qt. Qt is a C++ framework which provides a library to get the location using a GPS device to the library QtGeolocation. By implementing Dijkstra's algorithm, the problem of determining the shortest path towards a desired location of a user's location can be overcome. This study discusses how to implement the algorithm to find the shortest path Dijkstra in Yogyakarta-based mobile devices.

Kata Kunci: Dijkstra, GPS, Geolocation

1. PENDAHULUAN

Peta merupakan suatu alat yang digunakan untuk mengetahui suatu lokasi. Peta pada umumnya dibuat dalam bentuk media cetak baik berupa lembaran ataupun berupa buku. Peta dalam bentuk media cetak dapat digunakan untuk mengetahui lokasi dimana pengguna berada dan juga dapat digunakan untuk mengetahui lokasi suatu tujuan yang akan dituju. Meskipun dapat mengetahui lokasi tujuan yang akan dituju, pengguna tidak dapat menentukan jalur terpendek untuk menuju ke lokasi tersebut dari lokasi dimana pengguna berada.

Dengan perkembangan teknologi informasi, peta tidak lagi berupa lembaran ataupun buku. Saat ini terdapat layanan peta secara digital yang sudah ditanamkan pada perangkat bergerak. Kelebihan dari peta ini adalah memudahkan pengguna dalam mencari lokasi suatu tempat. Pengguna juga dapat mengetahui posisi pengguna berada dengan menggunakan teknologi GPS (*Global Position System*). Meskipun dapat mengetahui posisi

⁽¹⁾Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, Yogyakarta

⁽²⁾Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, Yogyakarta

⁽³⁾Sistem Informasi, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, Yogyakarta

pengguna dan lokasi tujuan dengan lebih mudah, pengguna tetap tidak dapat menentukan jalur terpendek untuk menuju ke lokasi tersebut.

Salah satu cara untuk dapat menentukan jalur terpendek adalah dengan menginterpretasikan peta kedalam suatu graf. Dalam graf, terdapat metode yang dapat digunakan untuk menentukan jalur terpendek, Salah satu metode yang digunakan untuk pencarian jalur terpendek adalah Algoritma Dijkstra. Algoritma ini digunakan dalam graf berarah dimana setiap titik dihubungkan oleh sisi yang memiliki bobot. Dengan memperhitungkan bobot pada setiap sisi, algoritma ini dapat digunakan untuk menentukan jalur terpendek dari suatu titik ke titik akhir tujuan.

Berdasarkan latar belakang yang telah diuraikan diatas, maka pokok permasalahan dalam penelitian ini adalah bagaimana keefektifan dari implementasi penggunaan teknologi GPS dan *QT Library Geolocation* dalam menentukan suatu lokasi pada perangkat bergerak telepon seluler dengan sistem operasi Symbian untuk menentukan jalur terpendek menuju ke lokasi yang ditentukan menggunakan Algoritma Dijkstra.

Tujuan dari penelitian ini adalah mengimplementasikan Algoritma Dijkstra untuk mencari jalur terpendek dari suatu lokasi ke lokasi tujuan pada peta. Adapun ruang lingkup dari penelitian adalah sebagai berikut:

- a. Program hanya akan mencari jalur terpendek dari suatu graf berarah yang telah ditentukan dan ditampilkan secara visual pada peta.
- b. Hanya akan menampilkan satu jalur terpendek.
- c. Lokasi awal ditentukan berdasarkan lokasi pengguna berada menggunakan teknologi *GPS* dan lokasi tujuan ditentukan oleh pengguna berupa nama jalan dan beberapa nama rumah makan yang terdapat di kota Yogyakarta.

2. LANDASAN TEORI

2.1 Algoritma Dijkstra

Algoritma Dijkstra merupakan suatu algoritma disebut juga sebagai *single-source shortest path* yang digunakan dalam menentukan jalur terpendek dari simpul sumber menuju simpul tujuan berdasarkan bobot pada sisi. Bobot pada sisi dapat berupa jarak, waktu, biaya, ataupun bobot yang lainnya. Algoritma Dijkstra bekerja dengan cara mengunjungi semua semua simpul-simpul yang terdapat pada graf dengan dimulai pada simpul sumber. Secara berulang algoritma ini akan memilih simpul-simpul terdekat dan menghitung total bobot semua sisi yang dilewati untuk mencapai simpul tujuan.

Secara singkat algoritma Dijkstra dapat dijelaskan sebagai berikut (Gross And Yallen, 1998):

Masukkan : Graf berbobot.

Proses :

1. Inisialisasi verteks.
2. Inisialisasi jarak antar verteks.
3. Tentukan verteks awal (s) dan verteks tujuan (t).
4. Beri label permanen = 0 ke verteks awal (s) dan label sementara = ∞ ke verteks lainnya.
5. Untuk setiap verteks V_t yang belum mendapat label permanen, mendapatkan label sementara = $\min \{ \text{label lama } V_t, (\text{label lama } V_t + D_{st}) \}$
6. Cari harga minim diantara semua verteks yang masih berlabel sementara.
7. Jadikan verteks minimum yang berlabel sementara menjadi verteks dengan label permanen, jika lebih dari satu verteks dipilih sembarang.
8. Ulangi langkah 5 sampai 7 hingga verteks tujuan mendapat label permanen.
9. Simpan hasil perhitungan.
10. Tampilkan hasil pencarian.

2.2 Global Positioning System

Global Positioning System (GPS) merupakan suatu sistem koordinat yang dapat digunakan untuk menentukan koordinat suatu lokasi berdasarkan posisi bujur, lintang serta ketinggiannya. Pada awalnya, teknologi ini dikembangkan oleh Departement Pertahanan Amerika Serikat. Namun pada saat ini teknologi ini dapat digunakan oleh masyarakat luas.

Untuk dapat mengetahui lokasi dari penerima terlebih dahulu perlu diketahui jarak antara satelit dengan penerima. Jarak tersebut dapat diketahui dengan mengetahui waktu untuk transmisi sinyal yang dipancarkan oleh satelit hingga diterima oleh penerima. Jarak tersebut dapat dihitung menggunakan rumus sebagai berikut.

$$S = c \cdot \Delta t$$

Dimana:

S = jarak yang dihitung.

c = merupakan cepat rambat sinyal di udara.

Δt = selang waktu yang dibutuhkan gelombang dari satelit ke perangkat GPS

A-GPS merupakan metode penghitungan lokasi penerima hasil pengembangan dari metode penghitungan yang sudah ada. A-GPS merupakan metode penghitungan lokasi yang memiliki tingkat keakuratan yang tinggi. A-GPS menggabungkan antara teknologi GPS dengan jaringan telepon seluler dalam menentukan suatu lokasi

A-GPS terdiri dari 3 bagian yaitu:

- a. Perangkat nirkabel yang sudah terdapat penerima GPS yang telah terintegrasi
- b. A-GPS server yang mengacu pada penerima GPS yang secara bersamaan terhubung dengan satelit yang sama dengan penerima GPS
- c. Infrastruktur jaringan nirkabel

Sinyal GPS memiliki beberapa kelemahan seperti lemahnya sinyal karena pengaruh atmosfer atau tidak mampunya sinyal menembus benda yang bersifat tebal dan keras seperti gedung yang dapat berpengaruh pada penghitungan lokasi penerima. Dengan bantuan jaringan perangkat nirkabel, kelemahan yang terdapat pada GPS dapat diatasi dengan menggunakan metode ini.

2.3 Qt

Qt adalah suatu *framework* aplikasi C++ untuk membuat aplikasi grafis menggunakan pendekatan “*write once compile anywhere*” (Blanchette & Summerfield, 2008). Dengan menggunakan Qt, aplikasi dapat dikembangkan pada berbagai macam *platform* seperti Windows, Mac OS X, Linux, Solaris, HP-UX, dan berbagai macam versi Unix yang berbasiskan X11. Selain itu, aplikasi tersebut juga dapat dikembangkan pada *platform embedded* Linux, Symbian, dan Windows CE. Nokia sebagai salah satu produsen telepon seluler asal Finlandia, pada tahun 2008 mengakuisis Trolltech dan meluncurkan produk *Software Development Kit* (SDK) dengan nama Nokia Qt SDK. Dalam Nokia Qt SDK ini terdapat *Application Programming Interface* (API) bernama *Qt Mobility* yang memungkinkan pengembang aplikasi pada perangkat bergerak khususnya perangkat bergerak yang menggunakan platform Symbian. Berikut ini merupakan *Application Programming Interface* (API) pada QT Mobility.

Kelas pada pustaka Location yang digunakan adalah

1. QGeoPositionInfo

QGeoPositionInfo adalah kelas yang digunakan untuk mengumpulkan informasi pada posisi global, arah, dan kecepatan pada satuan waktu tertentu.

Fungsi yang digunakan yang merupakan anggota dari kelas QGeoPositionInfo adalah fungsi `coordinate() const:QGeoCoordinate`. Fungsi ini mengembalikan koordinat dari posisi perangkat bergerak. Cara penggunaan fungsi ini adalah membuat attribute bertipe QGeoCoordinate.

2. QGeoPositionInfoSource

QGeoPositionInfoSource merupakan suatu kelas abstrak yang digunakan untuk mengetahui pemutakhiran posisi dimana perangkat bergerak berada.

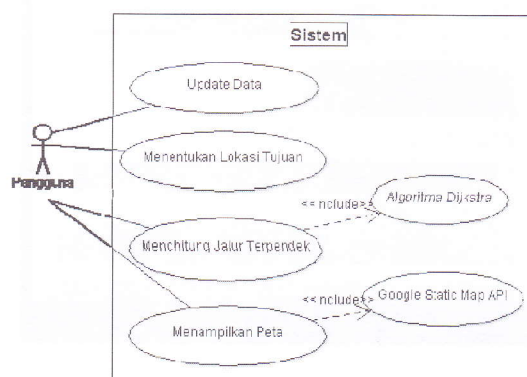
Fungsi yang digunakan yang merupakan anggota dari kelas QGeoPositionInfo antara lain QGeoPositionInfoSource::createDefaultSource (QObject*Parent). Fungsi ini membuat dan mengembalikan posisi sumber yang diberikan oleh *parent* yang membaca lokasi data pada sistem. Fungsi ini akan mengembalikan nilai 0 jika sistem tidak memiliki posisi sumber. Fungsi berikutnya adalah updateInterval:int. Fungsi ini dapat menentukan interval dalam melakukan pemutakhiran posisi. Nilai inputan pada fungsi ini dalam *mili detik*. Jika nilai dari fungsi ini tidak diset maka pemutakhiran posisi hanya dilakukan seperlunya saja.

3. PERANCANGAN SISTEM

3.1. Rancangan Sistem

Alur kerja sistem ini dimulai dengan sistem membaca semua berkas basis data yang berisi informasi data persimpangan serta jalan. Setelah itu, sistem melakukan request kepada satelit untuk mengetahui lokasi dari pengguna. Setelah lokasi pengguna didapat, pengguna dapat memilih lokasi tujuan berupa lokasi rumah makan yang dituju. Dari lokasi tujuan tersebut, sistem melakukan perhitungan jalur terpendek untuk mengetahui node-node mana saja yang dilewati. Setiap node memiliki posisi latitude dan longitude. Sistem kemudian melakukan request peta menggunakan *Google Static Map API*. *Server Google* akan memberikan respon berupa gambar peta statis berformat JPG yang kemudian ditampilkan oleh sistem.

3.2. Use Case Diagram



Gambar 1. Use Case Diagram

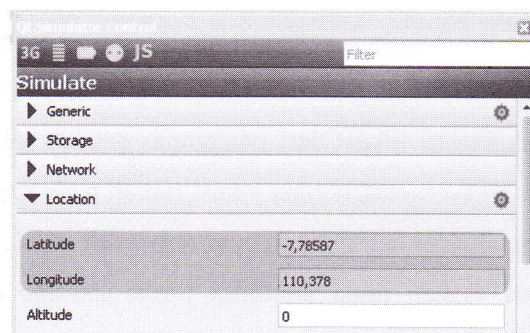
Pada gambar diatas terlihat bahwa sistem ini hanya memiliki satu aktor utama yaitu pengguna. Pengguna dapat melakukan *update* data, menentukan lokasi tujuan berupa *combo box* yang berisi lokasi mana saja yang telah terdapat pada basis data sistem, menghitung jalur terpendek dari lokasi pengguna menuju lokasi yang sudah ditentukan

menggunakan Algoritma Dijkstra, dan menampilkan hasil perhitungan jalur terpendek pada peta menggunakan *Google Static Map API*.

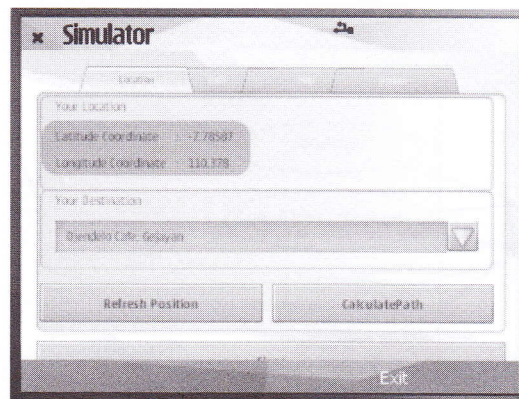
4. HASIL DAN PEMBAHASAN

4.1 Pengujian Qt Geolocation

Pada sistem ini, lokasi awal dari pencarian jalur terpendek menggunakan lokasi diaman pengguna berada. Untuk mendapatkan suatu lokasi, dibutuhkan perangkat GPS yang dapat menentukan lokasi pengguna berdasarkan koordinat garis lintang dan garis bujur. Dengan menggunakan pustaka yang disediakan oleh Qt berupa Qt Geolocation, posisi pengguna berdasarkan koordinat garis lintang dan garis bujur dapat ditentukan.



Gambar 2. Simulasi koordinat lintang dan bujur (*highlight* merah) pada simulator QtCreator

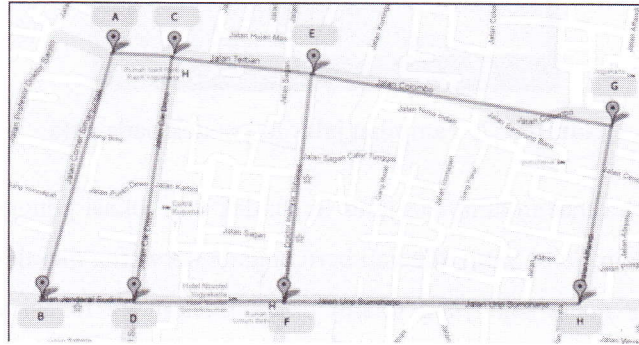


Gambar 3. Tampilan posisi koordinat lintang dan bujur (*highlight* merah) pada sistem

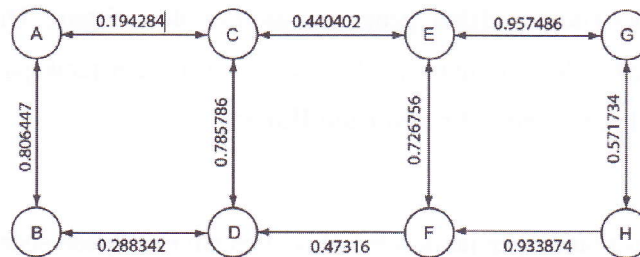
Berdasarkan pengujian sistem yang penulis lakukan pada perangkat bergerak, Qt Geolocation mampu menentukan koordinat lintang dan bujur pengguna dengan tingkat akurasi sekitar 10 meter. Hal ini diakibatkan karena adanya *multipath* yaitu keadaan dimana sinyal memantul suatu objek terlebih dahulu sebelum diterima oleh GPS *receiver*.

4.2 Pengujian Jalur Terpendek

Pada pengujian jalur terpendek ini, penulis menggunakan peta yang lebih kecil dari pada yang terdapat pada sistem dimana peta ini terdiri dari delapan *vertex* dan sepuluh *edge*. Berikut ini adalah gambar peta yang digunakan dalam pengujian ini beserta data masing-masing *vertex*.

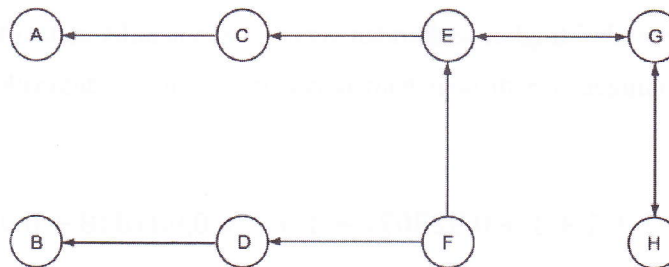


Gambar 4. Gambar peta yang digunakan pada penelitian

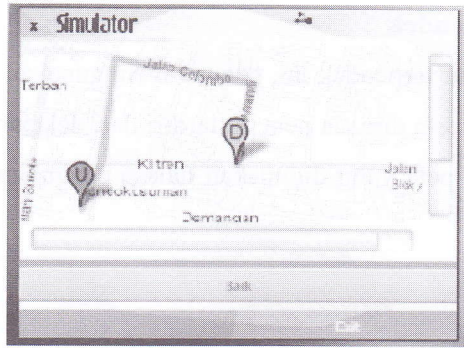


Gambar 5. Graf representasi dari peta yang akan digunakan untuk pengujian

Pada pengujian ini, lokasi dari koordinat garis lintang dan koordinat garis bujur yang digunakan sebagai lokasi pengguna yaitu -7.78587 pada garis lintang dan 110.378 pada garis bujur. Lokasi tersebut merupakan lokasi dari *Universitas Kristen Duta Wacana* dimana lokasi ini akan menjadi acuan titik awal untuk semua pengujian yang dilakukan. Pada pengujian ini, lokasi tujuan yang akan digunakan adalah *vertex H* yang merupakan lokasi dari Perempatan Jalan Solo.



Gambar 6. Hasil dari perhitungan jalur terpendek menggunakan Algoritma Dijkstra



Gambar 7. Tampilan jalur terpendek pada peta

Gambar 6 merupakan tampilan jalur terpendek dari lokasi pengguna menuju lokasi tujuan yang ditentukan. Jalur yang dilewati berupa garis berwarna merah muda dan Gambar 7 merupakan *vertex* mana saja yang dilalui untuk menuju ke lokasi yang dituju sesuai perhitungan jalur terpendeknya yaitu (F) Perempatan Galeria - (E) Perempatan Sagan – (G) Pertigaan Gejayan – (H) Perempatan Jalan Solo.

Pengujian berikutnya adalah pengujian akurasi dari sistem. Ditentukan sepuluh lokasi tujuan dan perbandingan antara jarak sebenarnya dengan jarak pada sistem dimana titik awal berlokasi di *Universitas Kristen Duta Wacana*.

Tabel 1.

Tabel hasil perbandingan jarak sebenarnya dengan jarak pada sistem

No	Tujuan	Akurasi
1	Djendelo Cafe	97,7778%
2	Coto Makassar Kota Baru	100%
3	Hartz Chicken	100%
4	Lecker Resto	100%
5	Angkringan KR	100%
6	Mc Donald Sudirman	92,3077%
7	Marfo	100%
8	Sego Macan	100%
9	Soto Kadipiro	98,1818%
10	EsSidoSemi,KotaGede	98,6842%

Untuk mengetahui tingkat keakuratan sistem maka penulis melakukan penjumlahan semua nilai perbandingan dan di bagi banyaknya data sehingga didapatkan hasil sebagai berikut:

$$0,977778 + 1 + 1 + 1 + 1 + 0,923077 + 1 + 1 + 0,981818 + 0,986842$$

$$10$$

$$= \frac{9,86952}{10} = 0,986952 = 98,6952\%$$

5. KESIMPULAN

Kesimpulan yang dapat diambil oleh penulis dari penelitian mengenai Implementasi Algoritma Dijkstra dalam Penentuan Jalur Terpendek Di Yogyakarta menggunakan GPS dan Qt Library Geolocation adalah:

- a) Berdasarkan pengujian terhadap Qt Library Geolocation untuk mendapatkan koordinat lokasi dari pengguna, Qt Library Geolocation mampu untuk menentukan lokasi pengguna dengan tingkat akurasi GPS sekitar 10 meter dari posisi aslinya.
- b) Penggunaan Algoritma Dijkstra dalam menentukan jalur terpendek dari lokasi pengguna menuju lokasi yang ditentukan mampu menghasilkan solusi jalur terpendek dengan tepat,
- c) Berdasarkan pengujian yang telah dilakukan, hasil akhir dari perhitungan jalur terpendek sistem dengan jarak aslinya memiliki tingkat keakuratan 98,6952% yang berarti jarak hasil perhitungan sistem hampir mendekati jarak aslinya.

Daftar Pustaka

- _____. *Qt Mobility White Paper*, Diakses 1 Oktober 2011, dari World Wide Web: <http://qt.nokia.com/files/pdf/qt-mobility-whitepaper-1.0.0>
- _____. *Qt Reference Documentation Signals and Slot*, Diakses 1 Oktober, dari World Wide Web: <http://doc.qt.nokia.com/latest/signalsandslots.html>
- _____. *Qt Reference Documentation Widgets and Layout*, Diakses 1 Oktober, dari World Wide Web: <http://doc.qt.nokia.com/latest/widgets-and-layouts.html>
- _____. (2005). *An Introduction to Using a Garmin GPS*. Kansas: Garmin.
- _____. (2008). *GPS Beginner's Guide*. Kansas: Garmin.
- Blanchette, J., & Summerfield, M. (2008). *C++ GUI Programming with Qt 4, Second Edition*. Massachusetts: Trolltech Press.
- Djuknic, G. M., & Richton, R. E. (2001, February). Geolocation and Assited GPS. *Computer*, pp. 123-125.
- Kurniawan, Erick., (2011). *Membangun Aplikasi Mobile dengan Qt SDK*. Yogyakarta: Andi.
- Larsen, J., & Clausen, J. (2009). *Supplementary Notes to Networks and Integer Programming*. Denmark: Kongens Lyngby.
- Rachmah, N. F. (2008). *Aplikasi Algoritma Dijkstra dalam Pencarian Lintasan Terpendek Graf*. Bandung: Institut Teknologi Bandung.
- Rischpater, R., & Zucker, D. (2010). *Beginning Nokia Apps Development*. New York: Apress.
- Siang, Jong Jek. (1997) *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Andi: Yogyakarta.
- Thelin, J. (2007). *Foundation of Qt Development*. New York: Apress.
- Wallis, W. (2007). *A Beginner's Guide to Graph Theory*. Boston: Birkhäuser.
- Wang, S.-z., & Gao, Y.-h. (2008). *A Google-Based Dynamic Route Guidance Algorithm and Its Implementations*. Beijing: High School Attached to Tsinghua University.
- Yulianto, W., Nurafianto, S. W., Damar, H. W., & Purnama, J. (2007). *Implementation of Dijkstra Algorithm to Track Location In a Mall*. Tangerang: Swiss German University.
- Zogg, J.-M. (2002). *GPS Basics*. Switzerland: u-blox.