

# Algoritma Dijkstra Sebagai Dasar Pencarian Rute Tercepat pada Aplikasi Berbasis GPS “Waze”

Alson Cahyadi 13514035  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
alson.cahyadi@students.itb.ac.id

**Abstrak**— Dalam dunia digital di mana teknologi semakin hari berkembang semakin pesat ini, orang-orang secara perlahan menggantikan peta analog dengan peta berbasis internet. Peta berbasis internet ini jugalah yang merupakan batu loncatan untuk aplikasi-aplikasi navigasi berbasis GPS untuk menjadi pusat lampu sorot masyarakat yang membutuhkan arahan untuk mencapai tempat tujuan yang kurang familiar, dan sulit untuk dicapai. Makalah ini akan menjelaskan tentang sebuah aplikasi navigasi, Waze, dan algoritma pencarian jalur tercepatnya yang berlandaskan algoritma Dijkstra.

**Kata Kunci** – Dijkstra Algorithm, Navigasi, Shortest-Path algorithm, Waze.

## I. PENDAHULUAN

Saat ini penggunaan GPS untuk mengetahui koordinat keberadaan diri sendiri sudah ada di hampir semua *smartphone*. Akurasi GPS yang tinggi dan waktu updatenya yang *realtime* membukakan jalan aplikasi-aplikasi berbasis peta online untuk menawarkan fitur berupa fitur navigasi. Fitur navigasi yang ditawarkan mengoptimalkan perjalanan yang dibutuhkan berdasarkan waktu perjalanan.

Dengan fitur navigasi ini, user dapat berpergian ke tempat yang bahkan sama sekali tidak user ketahu. Informasi input yang dibutuhkan oleh system navigasi ini hanyalah nama tempat yang ingin dituju, dan dengan algoritmanya fitur navigasi ini akan menunjukkan rute tercepat menuju tempat tersebut dengan waktu tempuh paling cepat, dan rute paling mangkus.

Salah satu aplikasi yang menyediakan fitur navigasi terbaik yang telah banyak diketahui orang-orang terutama masyarakat Indonesia adalah Waze.

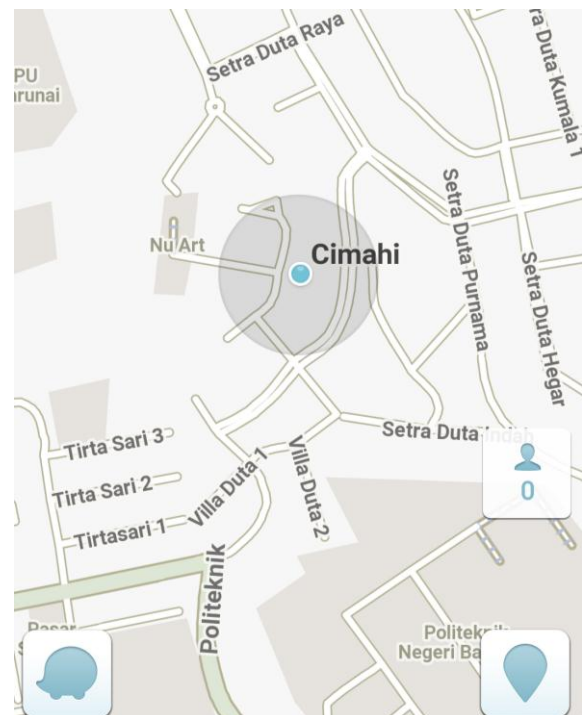
## II. WAZE



Gambar 2.1, Logo Waze dan tagline-nya.

Sumber: <http://cdn.slashgear.com/wp-content/uploads/2015/03/waze.jpg>

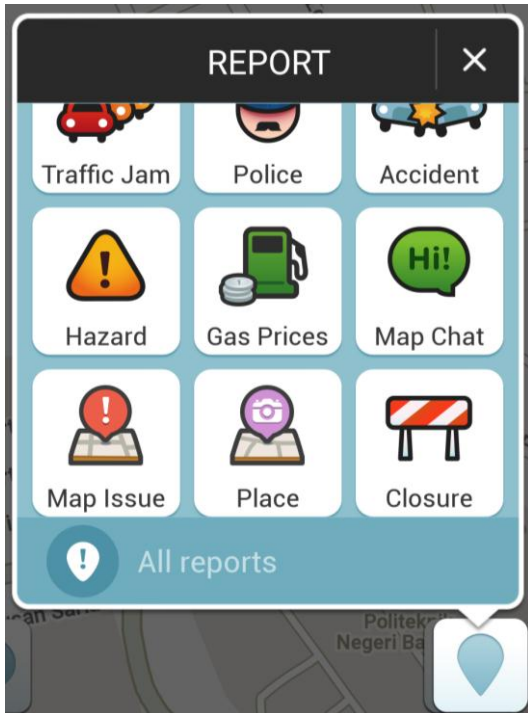
Waze merupakan sebuah piranti lunak navigasi bebas bayar untuk perangkat *mobile* seperti *smartphone* atau tablet, maupun PC yang memiliki fitur GPS. Sampai saat ini, Waze mendukung perangkat dengan system operasi iOS, Android, Windows Mobile, Symbian dan Blackberry.



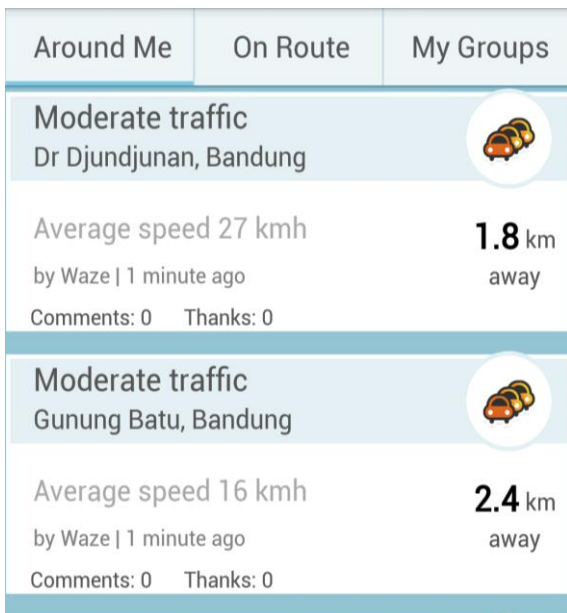
Gambar 2.2, Layour peta Waze

Sumber: Galeri penulis

Waze dapat diunduh dari negara mana saja di dunia termasuk Indonesia, tetapi peta dasar untuk negara Indonesia belum tersedia, sehingga kontribusi *user* sangat diutamakan dalam pengembangan peta Indonesia ini.



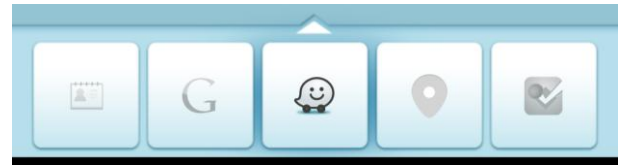
Gambar 2.3, Kontribusi yang *user* dapat berikan  
Sumber: Galeri penulis



Gambar 2.4, Hasil laporan *user* lain  
Sumber: Galeri penulis

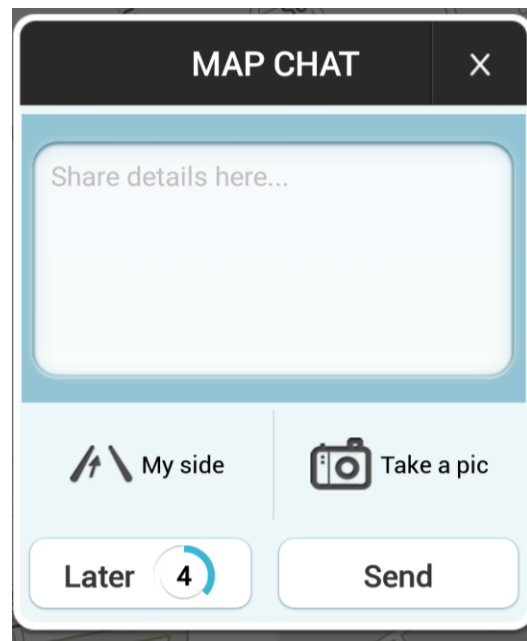
Waze berbeda dengan piranti lunak navigasi pada umumnya, dikarenakan peran *user* yang sangat besar dalam berkontribusi untuk informasi-informasi mengenai kecelakaan, kemacetan jalan, polisi, maupun bahaya berdasarkan kondisi nyata yang dilaporkan para penggunanya. *User* juga dapat melakukan pemutakhiran

peta, pemberian nomor rumah/bangunan, maupun penandaan lokasi secara pribadi dan langsung, seperti yang ada pada gambar 2.3. Gambar 2.4 menunjukkan contoh laporan jalan macet yang telah dilaporkan *user-user* yang ada di jalan raya. Informasi ini selanjutnya digunakan waze dalam algoritma pencarian rute teroptimalnya. Jalanan yang lebih macet akan menyebabkan estimasi waktu sampai yang lebih lama daripada jalan lancar.



Gambar 2.5, Pilihan search engine yang ditawarkan  
Sumber: Galeri penulis

Keunggulan Waze yang lain adalah kemampuannya untuk mencari tempat tujuan di search engine yang berbeda, seperti Google Maps, Foursquare, Contacts, dan lain-lain, seperti yang ada pada gambar 2.5.



Gambar 2.6, Layout map chat  
Sumber: Galeri penulis

Waze juga mempunyai fasilitas *chat* yang memberikan poin untuk setiap kegiatan yang dilakukan *user* seperti menjelajah peta, kontribusi informasi, pemutakhiran peta dan peristiwa khusus lainnya. Dengan fasilitas ini, bukan hanya piranti navigasi, tetapi Waze juga adalah jejaring sosial dengan permainan online yang layoutnya ada pada gambar 2.6.

### III. GRAF BERARAH

Graf berarah digunakan pada implementasi piranti navigasi karena adanya kemungkinan jalan yang satu arah di lapangan. Adapun penjelasan graf berarah sbb:

Definisi Graf adalah:

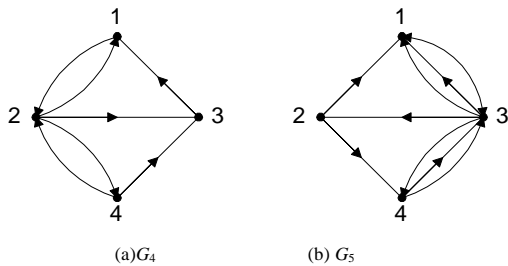
Graf  $G = (V, E)$ , yang dalam hal ini:

$V$  = himpunan tidak-kosong dari simpul-simpul  
 $= \{ v_1, v_2, \dots, v_n \}$

$E$  = himpunan sisi yang menghubungkan sepasang simpul

$= \{ e_1, e_2, \dots, e_n \}$

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Dua buah graf pada Gambar 3.1 adalah graf berarah.



**Gambar 3.1** (a) graf berarah, (b) graf-ganda berarah

Sumber: Slide Kuliah Rinaldi Munir (Graf 2015)

Tabel jenis-jenis graf dapat dilihat pada Tabel 1

**Tabel 1** Jenis-jenis graf [ROS99]

Jenis	Sisi	Sisi ganda dibolehkan?	Sisi gelang dibolehkan?
Graf sederhana	Tak-berarah	Tidak	Tidak
Graf ganda	Tak-berarah	Ya	Tidak
Graf semu	berarah	Tidak	Ya
Graf berarah	Tak-berarah	Ya	Ya
Graf-ganda berarah	berarah		

Sumber: Slide Kuliah Rinaldi Munir (Graf 2015)

### IV. ALGORITMA DIJKSTRA

#### A. Sejarah dan Penjelasan Algoritma

Algoritma Dijkstra merupakan algoritma yang dikemukakan oleh seorang ilmuwan komputer bernama

Edsger W. Dijkstra pada tahun 1956 dan dipublikasikan tiga tahun kemudian.

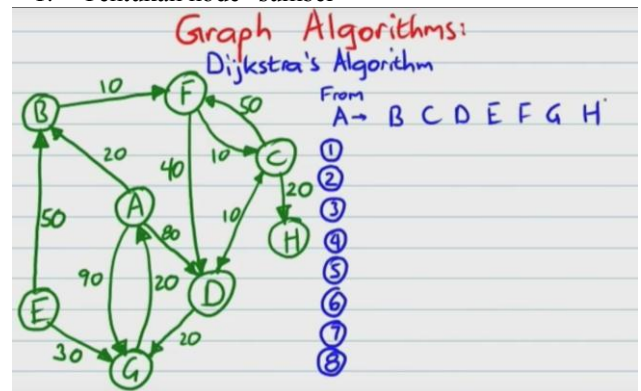
Algoritma ini adalah sebuah algoritma yang digunakan untuk menemukan rute terpendek antara node-node dalam suatu graf yang dapat merepresentasikan jalan, jaringan, dan sebagainya.

Algoritma Dijkstra yang asli memiliki tujuan untuk menemukan rute tercepat antara node asal dan node destinasi, sedangkan Algoritma Dijkstra yang lebih dikenal mencari rute optimal antara node asal dan semua node lainnya yang ada di graf, yang menghasilkan pohon rute terpendek.

#### B. Penerapan Algoritma Dijkstra pada Graf Berarah

Berikut adalah langkah-langkah implementasi algoritma Dijkstra pada graf berarah, beserta contohnya.

1. Tentukan node "sumber"

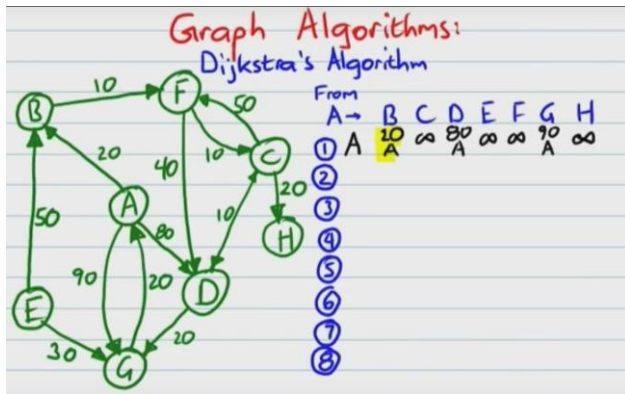


Gambar 3.1

Sumber: <https://www.youtube.com/watch?v=8Ls1RqHCOPw>

Tentukan node sumber, yang merepresentasikan posisi awal. Pada kasus ini, node sumbernya adalah A.

2. Gambar tabel pencarian  
 Gambarkan tabel  $N-1 \times N$  sesuai dengan gambar 3.1, dimana  $N$  adalah jumlah node yang ada pada graf, dengan node-node tujuan sebagai kolomnya, dan angka 1 sampai  $N$  sebagai barisnya.
3. Mulai pencarian rute

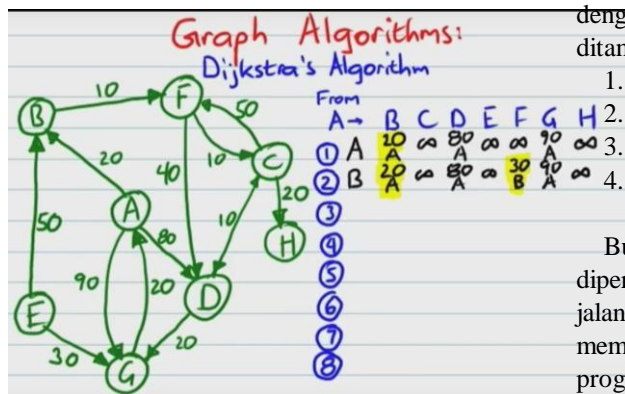


Gambar 3.2  
 Sumber: <https://www.youtube.com/watch?v=8Ls1RqHCOPw>

Pencarian rute dilakukan dengan cara traversasi baris dari 1 sampai N. Pada baris pertama, dipilih node sumber sebagai patokan. Beban yang ditempuh node sumber untuk menjelajah ke tetangganya ditulis di bawah kolom node tetangga tujuannya, seperti tertera pada Gambar 3.2.

Huruf "A" di bawah besar beban merupakan node yang menjadi media untuk berpindah ke node tujuan. Dalam kasus ini, untuk traversasi ke-1, B diraih dengan A sebagai medianya, begitu pula dengan node D dan G. Bila node bukan merupakan tetangga node sumber, tuliskan simbol tak hingga di bawah kolom node.

Node dengan beban terkecil diberi tanda sebagai penanda bahwa node tersebut telah dikunjungi.



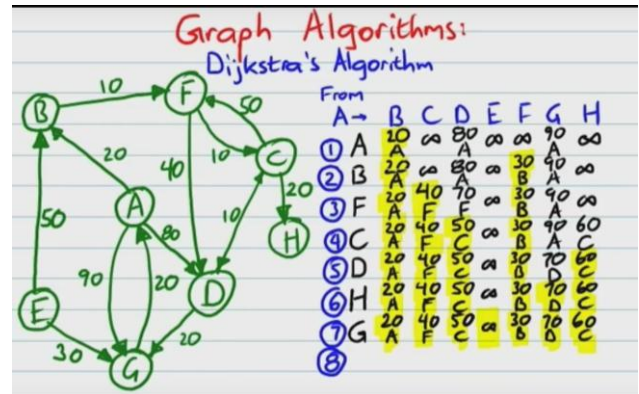
Gambar 3.3  
 Sumber: <https://www.youtube.com/watch?v=8Ls1RqHCOPw>

Untuk iterasi ke-2, node patokan sekarang berpindah kepada node yang terakhir dikunjungi, dalam kasus ini adalah node B.

Beban tetangga yang dapat dikunjungi node patokan ditambahkan dengan beban node yang telah dikunjungi sebelumnya. Bila hasil penambahan lebih kecil daripada angka yang telah dicatat sekarang, maka angka yang lama diganti dengan angka yang baru, dengan media yang baru, dalam khusus ini, beban F menjadi 30,

dengan media B.

Hal ini dilakukan terus menerus hingga menghasilkan pohon rute terpendek seperti yang dapat dilihat pada gambar 3.4.



Gambar 3.4  
 Sumber: <https://www.youtube.com/watch?v=8Ls1RqHCOPw>

### V. IMPLEMENTASI ALGORITMA DIJKSTRA PADA SISTEM NAVIGASI WAZE

Pada implementasinya dalam sistem navigasi, algoritma Dijkstra tidak secara gamblang dipakai untuk mencari rute teroptimal. Walaupun algoritma Dijkstra dipakai sebagai landasan teori pencarian rute teroptimal, banyak algoritma-algoritma yang ditambahkan pada algoritma tsb sehingga runtime yang tadinya 10 detik bila dijalankan dengan algoritma Dijkstra saja, dapat dijalankan hanya dengan satu millisecond<sup>[1]</sup>. Beberapa proses *speedup* yang ditambahkan adalah sbb:

1. Priority Queues
2. Bidirectional Search
3. Geometric Goal Directed Search (A\*)
4. Heuristics

Bukan hanya dengan proses *speedup*, pencarian juga dipercepat secara drastis dengan pengeksploitasi hierarki jalan-jalan yang menjadi sisi. Hal ini dilakukan dengan cara memproses hierarki yang sudah ditentukan sendiri sebelum program mencari rute (*preprocessing*). Hal ini memungkinkan untuk membuat algoritma semakin mangkus karena jalan yang jauh dari start tidak akan diperhitungkan oleh algoritma, sehingga mengakibatkan peningkatan performa dan akurasi hasil.

Dengan kombinasi antara proses-proses di atas dengan *preprocessing* hierarchy, maka dapat dirancang suatu algoritma yang kompleks namun memiliki runtime yang jauh lebih baik daripada algoritma Dijkstra biasa.

### V. KESIMPULAN

Pengaplikasian algoritma Dijkstra dalam system pencarian rute teroptimal pada fitur navigasi piranti lunak



Waze sangat penting. Modifikasi terhadap algoritma Dijkstra menyebabkan runtime yang jauh lebih cepat daripada algoritma Dijkstra yang biasa. Sistem pencarian rute teroptimal masih terus berkembang dan patut untuk diapresiasi.

## VI. UCAPAN TERIMAKASIH

Penulis mengucapkan terimakasih kepada Tuhan YME yang dengan berkat-Nya penulis dapat menyelesaikan makalah Matematika Diskrit ini. Penulis juga mengucapkan terimakasih kepada Bapak Rinaldi Munir selaku dosen mata kuliah Matematika Diskrit, untuk semua pelajaran yang telah diberikan, terutama pelajaran graf yang menjadi dasar penulisan makalah ini.


## REFERENSI

- [1] <http://stackoverflow.com/questions/430142/what-algorithms-compute-directions-from-point-a-to-point-b-on-a-map>  
diakses 10 Desember 2015 21.00 WIB
- [2] <https://www.youtube.com/watch?v=8Ls1RqHCOPw>  
diakses 10 Desember 2015 20.00 WIB
- [3] Peter Sanders and Dominik Schultes  
Universit'at Karlsruhe (TH), 76128 Karlsruhe, Germany,  
"Engineering Fast Route Planning Algorithms".
- [4] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO,  
private communication, May 1995.
- [5]

## 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 Desember 2015



ALSON CAHYADI  
13514035