



IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PENENTUAN JALUR TERPENDEK STUDI KASUS JARAK TEMPAT KULIAH TERDEKAT

Muhammad Muharrom

Prodi Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika,
muhammad.muu@bsi.ac.id

Jl. Kamal Raya No. 18, Ringroad Barat, Cengkareng, Jakarta Barat
Telp. (021) 54376399, fax. (021) 54376298

Keywords:

Algoritma Dijkstra,
jarak terpendek

Abstract

For anyone who wants to travel to Jakarta, wherever his domicile is located, such as the Bekasi area which is one of the areas close to the Jakarta province, whether traveling to Jakarta about work or college must know which route is the fastest and shortest to go through and as much as possible perhaps avoiding the region with severe traffic jams, because traffic congestion in Jakarta has become a daily sight. Many steps have been taken by the government to overcome these bottlenecks, such as the construction of flyovers and underpasses, the operation of busway lanes, the implementation of clock trees in one and so on. However, traffic jams still often occur until now, therefore we need the active role of road users themselves to be able to overcome these bottlenecks. One of the most effective ways is to look for alternative routes that can be passed. Therefore, the author tries to make a case study of a trip from Ujungharapan-Bekasi to Menara Salemba (STMIK Nusa Mandiri) - Central Jakarta as an experiment using the method using the Dijkstra algorithm that can find the fastest and shortest paths. So that the journey from Bekasi to STMIK Nusa Mandiri will be faster and save time, energy and fuel costs.

Kata Kunci:

Algoritma Dijkstra,
penentuan jarak
terpendek

Abstrak

Bagi siapapun yang ingin bepergian menuju Jakarta, darimana pun domisili orang tersebut berada, seperti wilayah Bekasi yang memang salah satu wilayah yang berdekatan dengan provinsi Jakarta, entah bepergian menuju ke Jakarta perihal bekerja ataupun kuliah harus mengetahui rute mana yang tercepat dan terpendek untuk dilalui dan sebisa mungkin menghindari wilayah yang macetnya parah, karena kemacetan di Jakarta sudah menjadi pemandangan sehari-hari. Banyak langkah-langkah yang telah dilakukan oleh pemerintah untuk mengatasi kemacetan tersebut, seperti pembangunan fly over dan underpass, pengoperasian jalur busway, pemberlakuan jam tree in one dan sebagainya. Akan tetapi kemacetan tetap saja masih sering terjadi sampai saat ini, Oleh karena itu diperlukan peran aktif dari pengguna jalan sendiri untuk dapat mengatasi kemacetan tersebut. Salah satu cara yang paling efektif yaitu dengan mencari rute alternatif yang dapat dilalui. Oleh karena itu penulis mencoba membuat study kasus perjalanan dari Ujungharapan-Bekasi menuju Menara Salemba (STMIK Nusa Mandiri) - Jakarta Pusat sebagai percobaan penerapan metode dengan menggunakan algoritma Dijkstra yang dapat menemukan jalur tercepat dan terpendek. Sehingga perjalanan dari Bekasi menuju STMIK Nusa Mandiri akan lebih cepat dan menghemat waktu, tenaga dan biaya bahan bakar.

Pendahuluan

Kemacetan di Jakarta selalu terjadi setiap hari kerja. Jumlah kepadatan penduduk yang besar dan aktifitas yang beragam membuat kota ini tidak pernah sepi dari lalu lintas manusia. Mobilitas yang tinggi tersebut sering terkendala dengan minimnya sarana dan prasarana jalan yang sangat terbatas. Oleh karena itu diperlukan sebuah solusi yang tepat yang dapat memperlancar mobilitas tersebut. Salah satu solusi yaitu dengan menambah sarana dan prasarana jalan yang saat ini sangat terbatas. Akan tetapi hal ini sulit dilakukan selain karena tidak adanya lahan untuk perluasan jalan, tetapi juga karena hal ini membutuhkan dana yang sangat besar. Kita bisa ambil contoh banyak proyek semacam itu yang sampai saat ini masih terbengkalai. Solusi lain yang mungkin dilakukan adalah dengan menggunakan kecermatan kita sendiri memilih dan memanfaatkan sarana yang tersedia dengan semaksimal mungkin.

Ada beberapa Penelitian sebelumnya terkait dengan algoritma Dijkstra ini yaitu membahas tentang pemetaan lokasi, pencarian lokasi, dan penentuan jalur terpendek untuk mencapai lokasi SPBU tersebut. Hasil pengambilan data didapatkan 33 SPBU yang menjual premium dan bio solar di Kabupaten Jember. Pencarian SPBU terdekat dipengaruhi oleh kriteria, cost, dan reverse_cost. Dimana untuk jalan satu arah diberikan nilai reverse_cost sebesar 1000000, sehingga jalan ini tidak akan pernah dipilih. Algoritma dijkstra sangat sesuai dan mudah digunakan pada studi kasus di penelitian ini. [1]

Pada penelitian lain terkait dengan penentuan jarak terdekat Dengan menggunakan algoritma Dijkstra, lintasan terpendek dari FMIPA diperoleh dengan memilih minimum lokal atau akses dengan jarak terdekat dari setiap lokasi yang kemudian digabungkan menjadi sebuah kumpulan lintasan dari satu lokasi ke lokasi lainnya dengan jarak terpendek.[2]

Dalam hal ini penulis mengimplementasikan algoritma Dijkstra pada proses bepergian menuju Jakarta terutama kampus yang menjadi tujuan, alangkah baiknya jika kita mengetahui setiap rute ketempat tujuan kita, dan mana saja dari rute tersebut yang memiliki jalur yang terpendek sehingga kita mencapai tujuan dengan menghemat waktu, biaya serta bahan bakar kendaraan, sehingga penulis mencoba membuat study kasus tentang penerapan

algoritma dijkstra dalam menentukan jarak terpendek sehingga perjalanan kuliah dari Ujungharapan Bekasi menuju STMIK Nusa Mandiri Jakarta Pusat, akan lebih menghemat waktu dan juga biaya.

A. Perumusan Masalah

1. Bagaimana cara menemukan jalur terpendek dari Ujungharapan - Bekasi menuju Menara Salemba-Jakarta Pusat?.
2. Faktor-faktor apa saja yang harus diperhitungkan dalam mencari jalur tersebut?

B. Batasan Masalah

Batasan masalah yang dilakukan pada penelitian ini yaitu:

1. Penelitian ini hanya dilakukan di wilayah antara wilayah Ujungharapan Bekasi dan Menara Salemba Jakarta Pusat.
2. Objek pada penelitian ini hanya pada jalan raya, yaitu jalan umum yang dapat dilalui oleh kendaraan yang berukuran besar seperti Bus.
3. Penelitian ini hanya sebatas menemukan jalur tercepat yang dapat digunakan oleh pengguna.
4. Data kecepatan dan waktu tempuh pada penelitian ini menggunakan acuan kendaraan roda empat yang didapatkan dari Peta Provinsi DKI Jakarta dan google map.

C. Tujuan dan Manfaat Penelitian

Penelitian ini penulis lakukan dengan beberapa tujuan. Adapun tujuan tersebut sebagai berikut.

1. Menemukan jalur tercepat dari satu lokasi ke lokasi lainnya yang ada di Jakarta khususnya antara Ujungharapan dan Menara Salemba.
2. Menemukan faktor-faktor yang mempengaruhi waktu tempuh perjalanan menuju STMIK Nusa Mandiri Jakarta .
3. Penulis dapat menerapkan ilmu yang didapatkan ke dalam sebuah permasalahan yang ada di masyarakat.
4. Penulis dapat mempelajari secara mendalam mengenai algoritma Dijkstra dalam melakukan pencarian terhadap jalur terpendek.

Landasan Teori

A. Graf

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan dalam kehidupan sehari-hari sampai saat ini. Banyak persoalan pada dunia nyata yang sebenarnya merupakan representasi visual dari

graf. Contoh salah satu representasi visual dari graf adalah peta. Banyak hal yang dapat digali dari representasi tersebut, diantaranya adalah menentukan jalur terpendek dari satu tempat ke tempat lain, menggambarkan 2 kota yang bertetangga dengan warna yang berbeda pada peta, menentukan tata letak jalur transportasi, pengaturan jaringan komunikasi atau jaringan internet dan masih banyak lagi. Selain peta, masih banyak hal lain dalam dunia nyata yang merupakan representasi visual dari graf. Secara matematis graf didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (vertex atau node) dan E adalah himpunan sisi (edge) yang menghubungkan sepasang simpul [5].

Simpul (vertex) pada graf dapat disimbolkan dengan huruf, bilangan atau gabungan keduanya. Sedangkan posisi sisi-sisi yang menghubungkan antara simpul u dengan simpul v dinyatakan dengan pasangan yang terhubung (u, v) atau dinyatakan dengan lambang yaitu e_1, e_2, e_3 dan seterusnya. Dengan kata lain, jika e adalah sisi yang menghubungkan antara simpul u dengan simpul v , maka e dapat dituliskan dengan sebagai $e = (u, v)$.

Klasifikasi pada graf cukup luas dan global, klasifikasi tersebut bergantung pada faktor-faktor yang membedakannya (tergantung inputannya). Berdasarkan orientasi arah pada sisinya, maka secara umum graf dibedakan atas dua jenis sebagai berikut :

1. Graf tidak berarah (undirected graph)
Graf yang sisinya tidak memiliki orientasi arah disebut graf tidak berarah. Pada graf tidak berarah, urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi, $(u, v) = (v, u)$ adalah sisi yang sama.
2. Graf berarah (directed graph)
Graf yang setiap sisinya diberikan orientasi arah disebut graf berarah, pada graf berarah (u, v) dan (v, u) menyatakan dua buah sisi yang berbeda. Dengan kata lain dapat ditulis $(u, v) \neq (v, u)$.

B. Algoritma

Kata algoritma (algorithm) berasal dari kata algorism yang diambil dari nama penulis buku Arab yang terkenal, yaitu Abu Ja'far Muhammad ibnu Musa Al-Khuwarizmi (al-Khuwarizmi dibaca orang Barat menjadi algorism)[5].

Adapun pengertian algoritma adalah kumpulan instruksi atau perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis untuk penyelesaian suatu masalah. 5 komponen utama dalam algoritma yaitu finiteness, definiteness, input, output dan effectiveness. Sehingga dalam merancang sebuah algoritma ada tiga komponen yang harus ada yaitu:

1. Komponen masukan (input)
Komponen ini biasanya terdiri dari pemilihan variabel, jenis variabel, tipe variabel, konstanta dan parameter (dalam fungsi).
2. Komponen keluaran (output)
Komponen ini merupakan tujuan dari perancangan algoritma dan program. Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran. Karakteristik keluaran yang baik adalah keluaran yang benar menjawab permasalahan dan tampilan (interface) yang ramah.
3. Komponen proses (processing)
Komponen ini merupakan bagian utama dan terpenting dalam merancang sebuah algoritma. Dalam bagian ini terdapat logika masalah, logika algoritma (sintaksis dan semantik), rumusan, metode (rekursi, perbandingan, penggabungan, pengurangan dan lain-lain).

Adapun jika dilihat dari segi kondisinya, komponen algoritma terdiri dari dua kondisi yaitu:

1. Pre condition Pre condition adalah kondisi suatu program ketika algoritma siap dijalankan (sebelum dilaksanakan algoritma). Dengan kata lain pre condition merupakan kondisi awal dimana algoritma akan dijalankan. Pre condition dinyatakan dengan mendefinisikan input dari sebuah algoritma.
2. Post condition
Post condition merupakan kondisi setelah suatu algoritma selesai dijalankan. Post condition dinyatakan dengan mendefinisikan hasil (output) dari suatu algoritma.

C. Algoritma Dijkstra

Algoritma Dijkstra merupakan salah satu metode untuk mencari lintasan terpendek dari sebuah simpul ke semua simpul lainnya dalam graf yang hanya memiliki bobot positif. Secara formal, masalah lintasan terpendek semua

pasangan simpul adalah untuk mencari lintasan terpendek di antara semua pasang simpul $v_i, v_j \in V$ sedemikian sehingga $i \neq j$.

Dalam mencari lintasan terpendek dari suatu simpul ke semua pasangan simpul algoritma Dijkstra melalui sejumlah langkah yang menggunakan prinsip greedy. Prinsip greedy pada algoritma Dijkstra menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukkannya dalam himpunan solusi [5].

Dijkstra ditemukan oleh Edsger Wybe Dijkstra pada tahun 1959. Algoritma ini merupakan algoritma yang dapat memecahkan masalah pencarian jalur terpendek dari suatu graf pada setiap simpul yang bernilai tidak negatif. Dijkstra merupakan algoritma yang termasuk dalam algoritma greedy, yaitu algoritma yang sering digunakan untuk memecahkan masalah yang berhubungan dengan suatu optimasi. Dalam pencarian jalur terpendeknya algoritma Dijkstra bekerja dengan mencari bobot yang paling minimal dari suatu graf berbobot, jarak terpendek akan diperoleh dari dua atau lebih titik dari suatu graf dan nilai total yang didapat adalah yang bernilai paling kecil. Misalkan G adalah graf berarah berlabel dengan titik-titik $V(G) = \{v_1, v_2, \dots, v_n\}$ dan path terpendek yang dicari adalah dari v_1 ke v_n .

Algoritma Dijkstra dimulai dari titik v_1 . Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya. Langkah-langkah dalam menentukan lintasan terpendek pada algoritma Dijkstra yaitu:

1. Pada awalnya pilih node sumber sebagai node awal, diinisialisasikan dengan '1'.
2. Bentuk tabel yang terdiri dari node, status, bobot, dan predecessor. Lengkapi kolom bobot yang diperoleh dari jarak node sumber ke semua node yang langsung terhubung dengan node sumber tersebut.
3. Jika node sumber ditemukan maka tetapkan sebagai node terpilih.
4. Tetapkan node terpilih dengan label permanen dan perbaharui node yang langsung terhubung.
5. Tentukan node sementara yang terhubung pada node yang sudah terpilih sebelumnya dan merupakan bobot terkecil dilihat dari tabel dan tentukan sebagai node terpilih berikutnya.

6. Apakah node yang terpilih merupakan node tujuan?. Jika ya, maka kumpulan node terpilih atau predecessor merupakan rangkaian yang menunjukkan lintasan terpendek.

D. Matriks Ketetanggaan

Misalkan sebuah graf $G = (V, E)$ dengan jumlah simpul n . Matriks ketetanggaan G adalah matriks bujur sangkar dengan ukuran $n \times n$. atau $M = [m_{ij}]$, dengan $m_{ij} = 1$ jika simpul i dan j bertetangga, sebaliknya $m_{ij} = 0$ jika simpul i dan j tidak bertetangga [5].

E. Masalah Lintasan Terpendek (SPP)

Pencarian rute terpendek merupakan suatu masalah yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950. Pencarian rute terpendek ini telah diterapkan di berbagai bidang untuk mengoptimasi kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat jalannya suatu proses. Salah [3]

SPP adalah suatu persoalan untuk mencari lintasan antara dua atau lebih simpul pada graf berbobot yang gabungan bobot sisi graf yang dilalui berjumlah paling minimum. Persoalan ini juga merupakan suatu persoalan optimasi yang menggunakan graf berbobot, dimana bobot dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya[4].

Metode

Berikut adalah Teknik pengumpulan data yang digunakan dalam proses penelitian dan pembuatan aplikasi ini:

a. Observasi;

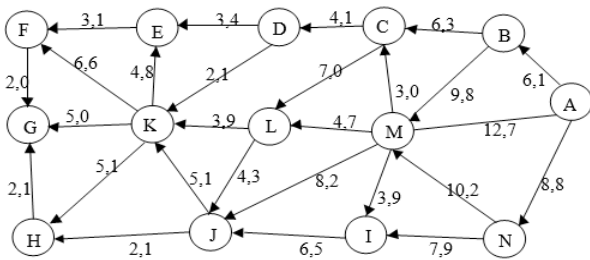
Observasi dilakukan penulis dengan cara mengamati secara langsung perjalanan yang dilalui kemudian dicek kembali pada google map jarak yang ditempuh. berupa jarak yang dimulai dari Ujungharapan Bekasi menuju menara salemba yang diperoleh berdasarkan pengukuran pada peta (Google Map).

b. Studi Pustaka;

Adalah tahapan yang penulis lakukan dalam mengumpulkan material atau bahan-bahan yang akan dipakai dalam merancang program dengan cara mengumpulkan melalui buku-buku literatur, jurnal dan internet yang sesuai dengan judul penelitian yaitu buku-buku yang terkait dengan materi yang akan dibahas.

Hasil dan Pembahasan

Penulis melakukan penelitian tentang jarak tempuh dari Ujungharapan menuju ke Salemba, ada banyak jalan yang bisa dilalui penulis jika ingin menuju ketempat tujuan tersebut, namun penulis hanya memberikan beberapa titik tujuan perlintasan yang bisa dilalui bisa dilihat dari graf yang dibuat oleh penulis berdasarkan titik tujuan dan panjang jarak (km) yang ditempuh. Berikut graf yang dibuat oleh penulis:



Keterangan :

- A. Jl. Ujungharapan
- B. Jl. Harapan Indah Bekasi
- C. Jl. Pasar Cakung
- D. Jl. Perintis Kemerdekaan
- E. Jl. Letjen R. Suprpto
- F. Jl. Stasiun Senen
- G. Jl. Salemba Raya (Menara Salemba)
- H. Jl. Matraman Raya
- I. Jl. Raya Kalimalang
- J. Jl. Stasiun Jatinegara
- K. Jl. Pemuda Rawamangun
- L. Jl. Klender
- M. Jl. I Gusti Ngurah Rai
- N. Jl. Metropolitan Mall Bekasi

Dapat kita lihat di gambar tersebut jalur mana saja yang dapat dilalui dan beserta jarak (km) yang ditempuh jika melalui jarak tersebut. Dan penulis merincikan sebagai berikut :

Kesimpulan dan Saran

Berdasarkan perumusan masalah dan serangkaian penelitian yang telah penulis lakukan. Maka dapat disimpulkan beberapa hal sebagai berikut :

1. Cara menemukan rute tercepat dan rute terpendek dari suatu lokasi ke lokasi yang lain, dalam hal ini berupa suatu persimpangan ke persimpangan yang lain adalah dengan menggunakan algoritma Dijkstra dengan input berupa persimpangan asal dan tujuan serta sebuah graf yang merepresentasikan node sebagai persimpangan dan simpul atau path sebagai jalur yang menghubungkannya.

Tabel 1. Daftar Lintasan Rumah ke Kampus

No	Lintasan	Jarak (KM)
1	A-B-C-D-E-F-G	25
2	A-B-M-C-D-E-F-G	31,5
3	A-B-M-C-D-K-H-G	32,3
4	A-B-M-C-L-K-H-G	37
5	A-B-M-L-K-H-G	31,7
6	A-B-C-L-K-E-F-G	33,2
7	A-M-C-D-E-F-G	28,3
8	A-M-L-K-H-G	28,5
9	A-M-C-L-K-H-G	33,8
10	A-M-C-D-K-H-G	29,1
11	A-M-L-K-E-F-G	31,2
12	A-N-I-J-H-G	27,4
13	A-N-M-L-K-H-G	34,8
14	A-N-I-J-K-H-G	35,5
15	A-M-I-J-H-G	25,1
16	A-M-L-J-H-G	25,9
17	A-M-J-H-G	25,1
18	A-M-L-K-G	26,3

Tabel 2. Matrix Ketetanggan

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	0	6,1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	12,7	8,8
B	6,1	0	6,3	∞	∞	∞	∞	∞	∞	∞	∞	∞	9,8	∞
C	∞	6,3	0	4,1	∞	∞	∞	∞	∞	∞	∞	7	3	∞
D	∞	∞	4,1	0	3,4	∞	∞	∞	∞	∞	2,1	∞	∞	∞
E	∞	∞	∞	3,4	0	3,1	∞	∞	∞	∞	4,8	∞	∞	∞
F	∞	∞	∞	∞	3,1	0	2	∞	∞	∞	6,6	∞	∞	∞
G	∞	∞	∞	∞	∞	2	0	2,1	∞	∞	5	∞	∞	∞
H	∞	∞	∞	∞	∞	∞	2,1	0	∞	2,1	5,1	∞	∞	∞
I	∞	∞	∞	∞	∞	∞	∞	∞	0	6,5	∞	∞	3,9	7,9
J	∞	∞	∞	∞	∞	∞	∞	2,1	6,5	0	5,1	4,3	8,2	∞
K	∞	∞	∞	2,1	4,8	6,6	5	5,1	∞	5,1	0	3,9	∞	∞
L	∞	∞	7	∞	∞	∞	∞	∞	∞	4,3	3,9	0	4,7	∞
M	12,7	9,8	3	∞	∞	∞	∞	∞	3,9	8,2	∞	4,7	0	10,2
N	8,8	∞	∞	∞	∞	∞	∞	∞	7,9	∞	∞	∞	10,2	0

2. Faktor-faktor yang dapat diperhitungkan dalam mencari jalur tercepat yaitu waktu tempuh dari suatu persimpangan ke persimpangan lainnya adanya pedagang kaki lima, volume kendaraan, lampu merah.

Berikut ini beberapa saran yang dapat dipergunakan untuk pengembangan penelitian dalam mencari rute tercepat dan terpendek.

1. Memperluas cakupan wilayah penelitian seperti meliputi kota Jakarta, Bogor, Depok, Tangerang dan Bekasi, sehingga dapat menjangkau kepentingan masyarakat yang lebih luas.

2. Menggunakan data yang real time, sehingga sangat akurat dalam menentukan rute tercepat pada saat sistem digunakan.

Referensi

- [1] Retnani WEY, Istiadi D, Roqib A. Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma DIJKSTRA (Studi Kasus di Kabupaten Jember). *J Nas Tek Elektro*. 2015;4(1):89. doi:10.25077/jnte.v4n1.132.2015
- [2] Salaki DT. Penentuan Lintasan Terpendek Dari Fmipa Ke Rektorat Dan Fakultas Lain Di Unsrat Manado Menggunakan Algoritma Dijkstra. *J Ilm Sains*. 2011;11(1):73. doi:10.35799/jis.11.1.2011.46
- [3] ArdyanS, Suyitno A, Mulyono. Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten. *UNNES J Math*. 2017;6(2):108-116.
- [4] Pradhana, B.A. 2009. Studi Dan Implementasi Persoalan Lintasan Terpendek Suatu Graf Dengan Algoritma Dijkstra Dan Algoritma Bellman-Ford.
- [5] Munir, Rinaldi. (2012). *Matematika Diskrit*. Bandung: Informatika.