PENERAPAN ALGORITMA DJIKSTRA PADA PENCARIAN JALUR TERPENDEK

Ely Stya Arga[1], Gun Gun Firmansyah[2], Khairul Imam[3], Muchammad Fauzi[4]
[1,2,3,4]Teknik Industri, Fakultas Teknik, Universitas Widyatama
*Email : ely.arga@widyatama.ac.id

ABSTRAK : Perkembangan zaman menuntut adanya metode aktivitas yang efektif dan efisien. Salah satunya adalah metode untuk penentuan jalur. Aktivitas manusia yang dinamis dan dituntut untuk melakukan perpindahan membutuhkan suatu metode yang membantu dalam penentuan jalur terpendek. Penentuan jalur terpendek memiliki alasan untuk menghemat bahan bakar, waktu dan tenaga yang diberikan. Salah satu metode yang dapat digunakan untuk pencarian jarak terdekat adalah dengan menggunakan algoritma Dijkstra. Algoritma Dijkstra adalah algoritma yang menerapkan graph berarah dan berbobot, dimana jarak antar titik adalah bobot dari tiap panah tersebut). Algoritma ini akan mencari jalur dengan cost yang paling minimum antara titik yang satu dengan titik yang lainnya. Selain itu, algoritma Dijkstra juga bisa digunakan untuk menghitung total biaya atau cost dari lintasan terpendek yang sudah terbentuk.
Kata kunci: Algoritma Djikstra, Greedy, Jalur terpendek, Optimasi Waktu

ABSTRACT: The development of the times demands an effective and efficient method of activity. One of them is the method for determining the path. Dynamic human activities and being required to make displacement require a method that helps in determining the shortest path. Determining the shortest path has a reason to save fuel, time and power. One method that can be used for closest distance search is to use Dijkstra's algorithm. Dijkstra algorithm is an algorithm that applies a graph of direction and weight, where the distance between points is the weight of each arrow. This algorithm will look for the path with the minimum cost between one point and another. In addition, Dijkstra's algorithm can also be used to calculate the total cost or cost of the shortest trajectory that has been formed.
Keywords: Djikstra Algorithm, Greedy, Shortest Path, Time Optimization

PENDAHULUAN

Perkembangan zaman menuntut adanya metode aktivitas yang efektif dan efisien. Salah satunya adalah metode untuk penentuan jalur. Aktivitas manusia yang dinamis dan dituntut untuk melakukan perpindahan membutuhkan suatu metode yang membantu dalam penentuan jalur terpendek. Penentuan jalur terpendek memiliki alasan untuk menghemat bahan bakar, waktu dan tenaga yang diberikan (Ardiani, 2011). Metode tersebut akan memberikan hasil jalur yang akan dipilih untuk dilewati agar sampai pada tujuan.

Salah satu metode pencarian jarak terdekat adalah dengan menggunakan algoritma Dijkstra. Algoritma Dijkstra adalah algoritma yang menerapkan graph berarah dan berbobot, dimana jarak antar titik adalah bobot dari tiap panah tersebut (Lubis, 2009). Algoritma ini akan mencari jalur dengan cost yang paling minimum antara titik yang satu dengan titik yang lainnya. Selain itu, algoritma Dijkstra juga bisa digunakan untuk menghitung total biaya atau *cost* dari lintasan terpendek yang sudah terbentuk (Fakhri, 2008). Alasan dipilihnya algoritma Dijkstra adalah algoritma ini adalah salah satu jenis dari algoritma greedy. Karena algoritma Dijkstra beroperasi secara menyeluruh terhadap alternatif fungsi yang ada, dan dihasilkan lintasan terpendek dari semua node (Lubis 2009).

Selain itu algoritma Dijkstra dapat menyelesaikan beberapa kasus jalur terpendek, antara lain: pencarian jalur terpendek antara dua buah simpul *(a pair shortest path)*, pencarian jalur terpendek antara semua pasangan simpul *(all pairs shortest path)*, pencarian jalur terpendek dari simpul tertentu ke semua simpul yang lain *(single-source shortest path)* dan pencarian jalur terpendek antara dua buah simpul yang melalui beberapa simpul tertentu *(intermediate shortest path)* (Andriani, 2011).

Pada penelitian ini akan dilakukan simulasi pencarian jalur terpendek dengan menggunajan algoritma djikstra

STUDI KEPUSTAKAAN

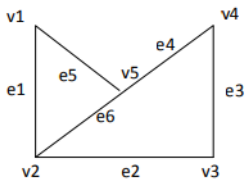
Pada jurnal ini akan dibahas mengenai penerapan algoritma djikstra untuk menentukan jalur terpendek.

2.1. Graph

Graph merupakan himpunan dari benda-benda yang disebut simpul (vertex atau node) yang terhubung oleh sisi (edge) atau busur (arc). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan simpul) yang dihubungkan oleh garis-garis (melambangkan sisi) atau garis berpanah (melambangkan busur). Suatu graf G didefenisikan sebagai pasangan himpunan (V,E), di mana V = himpunan yang berisikan simpul pada graf tersebut { , } dan E adalah himpunan sisi yang menghubungkan simpul-simpul { , } atau dapat ditulis dengan notasi G = (V,E).

Berdasarkan arah sisinya, graph terbagi menjadi dua jenis, antara lain:

1. Graf Tak Berarah
   Graf Tak Berarah (Undirected graf) merupakan graf yang memiliki setiap sisi tidak berarah.

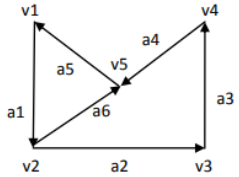


Gambar 2.1 Graf Tak Berarah

Graf pada gambar 2.2 menunjukkan graf tidak berarah dengan himpunan simpul, V(G) = { , } dan himpunan sisi E(G) {e1, e2, e3, e4, e5, e6} yaitu pasangan terurut dari {(v1,v2), (v2,v3), (v3,v4), (v4,v5), (v5,v1), (v5,v2)}.

2. Graf Berarah

Graf berarah (directed graf) merupakan graf yang memiliki setiap sisi berarah titik awal dari suatu sisi disebut simpul awal (initial vertex) sedangkan titik akhir dari suatu sisi disebut simpul akhir (terminal vertex). Loop pada graf adalah sisi yang simpul awal dan simpul akhirnya sama.



Gambar 2.2 Graf Berarah atau Digraf

Digraf pada Gambar 2.1 menunjukkan graf berarah dengan himpunan simpul, V(G) = { , } dan himpunan busur A(G) = {a1,a2, a3, a4, a5, a6} yaitu pasangan terurut dari {(v1,v2), (v2,v3), (v3,v4), (v4,v5), (v5,v1), (v2,v5)}.

2.2. Lintasan (*Path*)

Lintasan adalah hubungan antar titik atau node dalam sebuah graf. Suatu lintasan yang berawal dan berakhir pada node yang sama, maka disebut lintasan tertutup (close path), jika node awal dan node akhir dari lintasan tersebut berbeda, disebut lintasan terbuka (open path).

2.2.1. Lintasan Terpendek (*Shortest Path*)

Lintasan terpendek adalah lintasan yang memiliki total bobot minimum untuk mencapai suatu tempat dari tempat tertentu. Rute terpendek diartikan sebagai lintasan yang mempunyai biaya terkecil suatu rute dari node awal ke node tujuan dalam sebuah jaringan (Siswanto, 2013). Lintasan terpendek dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot, yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan, waktu, biaya dan sebagainya.

Terdapat beberapa jenis lintasan terpendek yakni.

1. Lintasan terpendek antara dua buah simpul.
2. Lintasan terpendek antara semua pasangan simpul.
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
4. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu.

2.3. Algoritma Djikstra

Algoritma djikstra merupakan algoritma yang digunakan untuk memecahkan permasalahan jalur terpendek *(shortest path problem)* pada sebuah graf berarah *(directed graf)* atau graf tidak berarah *(undirected graf)* dengan bobot-bobot sisi *(edge weights)* yang memiliki nilai tidak negatif. Node dari graf melambangkan titik-titik tujuan sedangkan bobot sisi *(edge weights)*

melambangkan jalur diantara titik tersebut, maka algoritma djikstra dapat digunakan untuk menemukan jalur terpendek antara titik tersebut.

Dalam menentukan jalur terpendek dari suatu graf oleh algoritma dijkstra akan didapatkan jalur yang terbaik, karena pada waktu penentuan jalur yang akan dipilih, akan dianalisis bobot dari node yang belum terpilih, lalu dipilih node dengan bobot yang terkecil. Jika ternyata ada bobot yang lebih kecil melalui node tertentu maka bobot akan dapat berubah. Algoritma dijkstra akan berhenti ketika semua node sudah terpilih. Sehingga akan ditemukan jalur terpendek dari seluruh node, tidak hanya untuk node dari asal dan node tujuan tertentu saja.

Langkah-langkah yang dilakukan untuk menentukan lintasan terpendek sebagai berikut:

1. Inisiasi node asal (V1) dan node tujuan (V2)
2. Buat 2 buah list, open list dan closed list. Keduanya tidak ada data atau kosong, dan formatnya {node, bobot, node induk}.
3. Masukkan V1 ke open list.
4. Pilih 1 node dengan bobot terkecil pada open list, tambahkan kedalam closed list.
5. Cari node yang bertetangga langsung dari node sebelumnya, yang masuk terakhir dalam closed list. Tambahkan bobot dengan node yang terkait, apabila sudah ada dalam closed list abaikan.
6. Apabila dalam open list terdapat node yang sudah ada bandingkan, lalu cari yang terkecil, dan perbaharui. Bila ternyata jumlah bobotnya sama dalam node yang sama, maka abaikan.
7. Apakah data dalam open list kosong? jika belum ulangi langkah 4.
8. Dalam closed list cari V2, telusuri jalur berdasarkan node induk sampai mengacu ke node asal (V1), dan balikkan urutan node.
9. Lintasan terpendek ditemukan bersama bobotnya

METODE PENELITIAN

Pencarian jalur terpendek merupakan permasalahan untuk menentukan lintasan terpendek antara dua buah vertex pada graph berbobot yang memiliki gabungan nilai jumlah bobot pada edge graph yang dilalui dengan jumlah yang paling minimum.

3.1. Langkah penyelesaian Algoritma Djikstra

Penentuan jalur terpendek pada algortima djikstra membutuhkan beberapa data antara lain, data asal, data akhir, dan proses pengambilan data antara node dan jarak. Diskripsi matematis untuk grafik dapat diwakili $G = \{V. E\}$, yang berarti sebuah grafik (G) didefenisikan oleh satu set simpul (Vertex = V) dan koleksi Edge (E). Terdapat beberapa entitas yang digunakan pada penentuan *shortest path* (Harahap, 2011) antara lain.

Tabel 3.1. Entitas pada Shortest Path

Entitas	Keterangan
Titik Asal	Node yang dipilih sebagai vertex asal

Titik Tujuan	Node yang dipilih sebagai vertex akhir
Jalur Shortest Path	Hasil perhitungan shortest path

Algoritma Dijkstra bekerja dengan membuat jalur ke satu simpul optimal pada setiap langkah. Jadi pada langkah ke n, setidaknya ada n node yang sudah kita tahu jalur terpendek. Langkah-langkah algoritma Dijkstra dapat dilakukan dengan langkah-langkah berikut:

Gambar 3.1 Flowchart Shortest Path

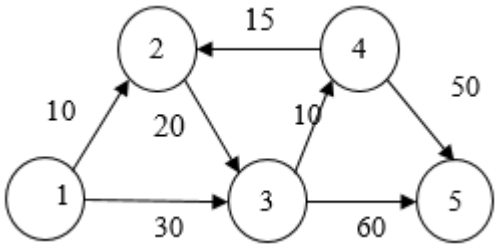
Pada proses penghitungan rute terpendek terdapat dua macam proses yaitu proses pemberian label dan proses pemeriksaan node. Metode pemberian label adalah metode untuk memberikan identifikasi pada setiap node dalam jaringan. Proses pemberian label berjalan seiring dengan proses scanning (pemeriksaan). Proses pemeriksaan node adalah proses membandingkan jarak antara node awal s dengan node i melalui node j sebagai node lain dalam suatu jaringan.

HASIL DAN PEMBAHASAN

Setelah dilakukan analisa masalah umum pemecahan penentuan jalur terpendek dengan menggunakan algoritma djikstra. Selanjutnya dilakukan analisis pada setiap tahapan dan penerapan algoritma djikstra.

4.1 Analisis Penerapan Algoritma Djikstra

Berikut merupakan contoh graph yang akan diselesaikan dengan penerapan algoritma Dijkstra.



Gambar 4.1 Contoh Model Algorima Djikstra

Keterangan pada algoritma djikstra

U_i merupakan jarak terpendek dari titik 1 ke titik i. $D_{ij} \geq 0$ merupakan panjang dari (i,j). Sehingga label pada titik j didefisikan menjadi.

$[U_{ij}] = (U_i + D_{ij}, i)$, dij ≥ 0. Dengan dua jenis label yakni permanen dan sementara. Label sementara dapat diganti bila ada label lain yang memiliki rute yang lebih pendek. Jika tidak ada maka status label tetap menjadi permanen. Berikan label dengan notasi (G).

$$G = (V,E)$$

Berikut merupakan iterasi untuk menemukan jalur terpendek antara titik 1 ke titik 2.

Tabel 4.1 Iterasi 1

Iterasi	Titik	Label	Status
0	1	[0,-]	Permanen

Tabel 4.2 Iterasi 2

Iterasi	Titik	Label	Status
1	1	[0,-]	Permanen
	2	[0,+100,1]	Sementara
	3	[0+30,1]	Sementara

Tabel 4.2 Iterasi 2

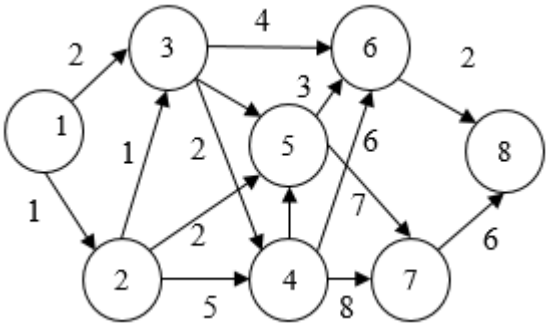
Iterasi	Titik	Label	Status
2	1	[0,-]	Permanen
	2	[100,1]	Sementara
	3	[30,1]	Permanen
	4	[40,3]	Permanen
	5	[90,3]	Sementara

Tabel 4.3 Iterasi 3

Iterasi	Titik	Label	Status
3	1	[0,-]	Permanen
	2	[55,4]	Permanen
	3	[30,1]	Permanen
	4	[40,3]	Permanen
	5	[90,3]	Sementara

Berdasarkan dari hasil iterasi ddidapatkan untuk jumlah biaya dengan rute terdekat antara 1 ke 2 yakni 30+10+15 = 55 dengan rute terpendek 1-3-4-2. Apabila dari iterasi belum didapatkan rute terpendek, maka iterasi akan dilakukan hingga didapatkan nilai optimal.

Contoh graph lain untuk penerapan penentuan jalur terpendek.



Gambar 4.2 Contoh Model Algorima Djikstra

1. Berikan rute untuk jarak terpendek 1-8

Tabel 4.4 Iterasi 0

Iterasi	Titik	Label	Status
0	1	[0,-]	Permanen

Tabel 4.5 Iterasi 1

Iterasi	Titik	Label	Status
1	2	[1,1]	Permanen
	3	[2,1]	Sementara

Tabel 4.6 Iterasi 2

Iterasi	Titik	Label	Status
2	3	[2,2]	Permanen
	4	[6,2]	Sementara
	5	[3,2]	Sementara

Tabel 4.7 Iterasi 3

Iterasi	Titik	Label	Status
3	4	[4,3]	Sementara
	5	[3,3]	Permanen
	6	[6,3]	Sementara

Tabel 4.8 Iterasi 4

Iterasi	Titik	Label	Status
4	6	[6,5]	Permanen
	7	[10,5]	Sementara

Tabel 4.9 Iterasi 5

Iterasi	Titik	Label	Status
5	8	[8,6]	Permanen

Berdasarkan hasil iterasi didapatkan rute terpendek dari 1-8 yakni 1-2-3-4-6-8 dengan total biaya 1+1+1+3+2=8.

KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan terdapat beberapa kesimpulan yakni.

1. Algortma Djikstra dilakukan dengan cara menentukan node awal, node akhir, dan keterangan nilai jarak sebagai pembanding antar rute.
2. Metode Djikstra digunakan untuk menentukan jalur terpendek dengan proses pemberian label berjalan seiring dengan proses scanning (pemeriksaan). Proses pemeriksaan node adalah proses membandingkan jarak antara node awal s dengan node i melalui node j sebagai node lain dalam suatu jaringan.

DAFTAR PUSTAKA

Ardiani, Farida. 2011. "*Penentuan Jarak Terpendek dan Waktu Tempuh Menggunakan Algoritma Dijkstra dengan Pemrograman Berbasis Objek*." Skripsi Universitas Islam Negeri Sunan Kalijaga.

Fakhri. 2008. "*Penerapan Algoritma Dijkstra dalam Pencarian Solusi Maximum Flow Problem.*" Makalah IF2251 Strategi Algoritmik.

Harahap & Khairina. "*Pencarian Jalur Terpendek dengan Algoritma Dijkstra*." Jurnal & Penelitian Teknik Informatika, No. 2, Vol.2, Hal 19-24, April 2011.

Lubis, Henny Syahriza. 2009. "*Perbandingan Algoritma Greedy dan Dijkstra untuk Menentukan Lintasan Terpendek*." Skripsi Universitas Sumatera Utara.