

Penerapan Algoritma Greedy

Penentuan jarak terdekat pada distribusi barang

Shah Raja Abdullah Al Turtusi 120140064

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Sumatera Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung,

E-mail (gmail): shah.120140064@student.itera.ac.id

Distribusi barang merupakan hal penting yang harus diperhatikan pada keberlangsungan sebuah bisnis. Semakin baik distribusi barang maka akan berdampak terhadap meningkatnya kepuasan pelanggan. Persoalan dalam menemukan jalur terpendek seiring dengan penghematan waktu yang tersingkat. Hal ini menjadi penting dalam kedinamisan masyarakat perkotaan. Jumlah rute yang ditempuh juga menjadi persoalan tersendiri untuk mencapai tempat tujuannya. Kita akan menentukan titik-titik manakah yang harus dilalui sehingga mendapatkan tempat tujuan dengan jarak terpendek dan penggunaan waktu yang tersingkat dengan menggunakan algoritma Dijkstra. Pencarian lintasan terpendek merupakan persoalan optimasi.

Keywords— Distribusi barang, optimasi waktu, algoritma greedy

I. PENDAHULUAN

Penentuan sebuah jalur terpendek merupakan hal yang penting dan dibutuhkan sehubungan dengan optimasi waktu yang digunakan serta beberapa penghematan di bidang lainnya. Dengan jalur terpendek yang dilalui, membuat pekerjaan lebih efektif, cepat dan dapat tentunya terjadi penghematan biaya. Jalur terpendek dapat diartikan sebagai nilai minimal dari suatu lintasan, yaitu jumlah nilai dari keseluruhan bentuk lintasan. Untuk membantu menentukan lintasan terpendek dapat memilih jalur yang terpendek dari tempat asal ke tujuan. Hal ini terkadang tidak dapat membantu secara maksimal di karenakan banyaknya jumlah jalan yang harus dipilih dan tidak dapat diperkirakan jarak tempuh pada jalur itu. Untuk itu diperlukan sebuah sistem untuk membantu melakukan pencarian lintasan terpendek yang dapat merepresentasikan data[5]. Data tersebut dapat disimpan, diolah dan disajikan dalam bentuk yang lebih sederhana sehingga memudahkan dalam menentukan lintasan terpendek. Algoritma Dijkstra yang juga merupakan algoritma Greedy bisa menentukan lintasan terpendek. Penelitian ini adalah membuat sebuah simulasi pencarian jalur terpendek dengan menggunakan Algoritma Dijkstra yang dapat membantu dalam pencarian jalur terpendek. Dengan skema yang telah ditetapkan dan digambarkan

II. LANDASAN TERORI

A. Pengertian Graph

Graph adalah struktur diskrit yang terdiri dari adanya simpul(vertex) dan adanya sebuah sisi (edge), graph adalah pasangan himpunan (V, E) dimana V merupakan sebuah himpunan yang tidak kosong dari sebuah vertex dan E adalah himpunan sisi yang menghubungkan sepasang simpul dalam graph tersebut [2].

B. Jalur lintasan terpendek (Shortest Path)

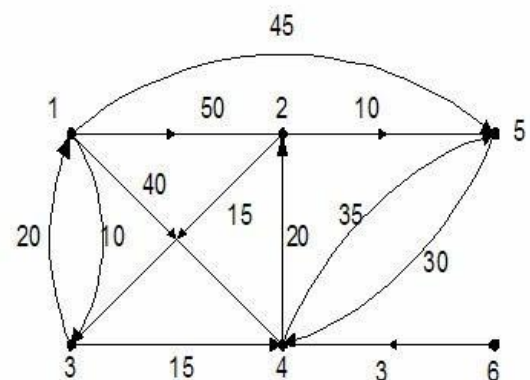
Kata Terpendek pada persoalan sebuah lintasan memiliki pengertian yaitu proses minimalisasi bobot pada sebuah lintasan graph. Beberapa jenis persoalan lintasan terpendek antara lain:

- Lintasan terpendek antara dua buah simpul.
- Lintasan terpendek antara semua pasangan simpul.
- Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
- Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu [1].

Dalam pemecahan persoalan tentang graf berbobot

$$G = (V, E)$$

dan sebuah simpul a . Penentuan lintasan terpendek dari a ke setiap simpul lainnya di G . Asumsi yang kita buat adalah bahwa semua sisi berbobot positif perhatikan Gambar 2.1 dibawah ini:



Gambar 1. Graph contoh persoalan lintasan terpendek. Source : <https://rahadikusuma.blogspot.co.id>

III. DEFINISI ALGORITMA

Kata algoritma (*algorithm*) diambil dari kata algorism yang berasal dari nama seorang ilmuwan dari Arab yang terkenal yaitu abu ja'far Muhammad Ibnu musa Al Khuwarizmi dan oleh orang barat dilafalkan menjadi Algorism. Dan pengertian algorism itu sendiri adalah sekumpulan instruksi atau perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis untuk penyelesaian suatu masalah [3].

1. Komponen Algoritma

Algoritma memiliki 5 komponen urutan yaitu *finiteness* (terbatas), *definiteness* (Kepastian), *input* (masukan), *output* (keluaran), dan *effectiveness* (efektivitas).

2. Notasi (Tata cara penulisan) Algoritma

Penulisan algoritma terkadang sulit untuk dipahami dan maksud dari algoritma tersebut. Selain itu juga sulit untuk menuliskan algoritmanya. Agar mempermudah bisa dilakukan notasi – notasi algoritma.

3. Aturan Penulisan Teks Algoritma

Algoritma merupakan deskripsi langkah - langkah penyelesaian dari sebuah permasalahan. Deskripsi tersebut bisa dituliskan ke dalam bentuk notasi apapun, yang penting mudah dibaca dan dipahami. Tidak ada notasi yang baku dalam penulisan teks algoritma. Tiap orang dapat membuat aturan penulisan dan algoritma sendiri. Namun, agar notasi algoritma dapat dengan mudah ditranslasi ke alam notasi bahasa pemrograman, maka sebaiknya notasi algoritma itu berkoresponden dengan notasi bahasa pemrograman secara umum.

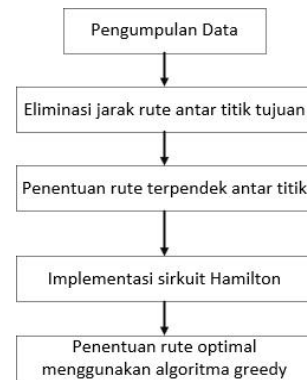
IV. ANALISIS MASALAH UMUM

UD XYZ adalah sebuah usaha dagang dan juga merupakan salah satu distributor di Surabaya yang bergerak di bidang distributor masker motor. Pada penelitian ini, penulis menggunakan algoritma *greedy* sebagai metode untuk menentukan solusi optimum rute. Optimum rute pada penelitian ini adalah rute terpendek. Algoritma *greedy* adalah algoritma yang mengikuti heuristik pemecahan masalah membuat pilihan optimal lokal pada setiap tahap dengan harapan menemukan optimal global. Jika algoritma *greedy* dapat memecahkan masalah, maka secara umum itu menjadi metode terbaik untuk menyelesaikan masalah itu karena algoritma *greedy* secara umum lebih efisien daripada teknik lain seperti pemrograman dinamis. Tapi algoritma *greedy* tidak selalu bisa diterapkan. Misalnya, masalah *Fractional Knapsack* dapat diselesaikan menggunakan *greedy*, tetapi 0-1 *knapsack* tidak dapat diselesaikan menggunakan *greedy*.

A. Metode

Terdapat beberapa langkah yang digunakan pada penelitian ini meliputi pengumpulan data, eliminasi jarak rute antar titik tujuan, penentuan rute terpendek antar titik, implementasi

sirkuit *hamilton*, dan penentuan rute optimal menggunakan algoritma *greedy*



Gambar 2. alur metologi penelitian

Tabel 1. Matriks Jarak Antar Titik Distribusi

	UD. XYZ	R1	R2	R3	R4	R5	R6	R7
UD. XYZ	0	600m	920m	880m	1200m	650m	550m	469m
R1	600m	0	670m	890m	860m	910m	750m	880m
R2	1300m	780m	0	650m	788m	865m	897m	886m
R3	860m	890m	932m	0	976m	677m	658m	1032m
R4	1200m	869m	800m	975m	0	994m	760m	750m
R5	650m	900m	845m	677m	882m	0	876m	455m
R6	650m	612m	671m	991m	701m	876m	0	987m
R7	500m	833m	921m	592m	699m	796m	907m	0

A) Implementasi sirkuit *Hamilton*.

Pada proses ini penentuan rute tiap titik di implementasikan dengan gambar sirkuit *Hamilton*. Model sirkuit *Hamilton* adalah sebuah aliran rute antar titik yang melewati tiap titik tepat satu kali kecuali untuk titik awal karena nantinya titik awal tersebut nantinya menjadi titik akhir [16]. Sirkuit *Hamilton* digunakan untuk menggambarkan beberapa alternatif alur rute dari awal di UD XYZ sampai Kembali lagi ke titik awal yaitu di UD XYZ.

B) Penentuan Rute Optimal Menggunakan Algoritma Greedy

Penentuan rute optimal menggunakan algoritma *greedy* yaitu dengan membandingkan beberapa alternatif rute dengan jarak yang sudah diketahui sesuai implementasi sirkuit *Hamilton*. Hasil alternatif rute tersebut dapat disebut sebagai solusi optimum lokal. Setelah penentuan solusi optimum lokal, selanjutnya yaitu menentukan jarak terpendek dari semua kemungkinan alternatif solusi yang disebut solusi optimum global. Implementasi algoritma *greedy* untuk pemilihan rute optimal dapat dijelaskan pada notasi *pseudocode* pada Gambar 2.

```

Deklarasi
  int J[n] //alternatif solusi
  lokal
  int hasil //jarak rute terpendek
hasil = 0
For i=1 to n do
  if (hasil < J[i]) then
    hasil = J[i]
  endif
end
end

```

Gambar 3. *Pseudocode* optimasi menggunakan algoritma *greedy*

Dari Gambar 2 dapat dilihat bagaimana algoritma *greedy* mengambil kemungkinan jarak terpendek dari beberapa alternatif solusi yang ada di lambangkan dalam notasi $J[n]$, dimana n adalah jumlah alternatif solusi. Kemudian setiap alternatif solusi tersebut akan dibandingkan dengan alternatif lain yang akan menghasilkan solusi optimum global yaitu jarak rute terpendek.

C) Penentuan Rute Optimal dari Tiap Titik

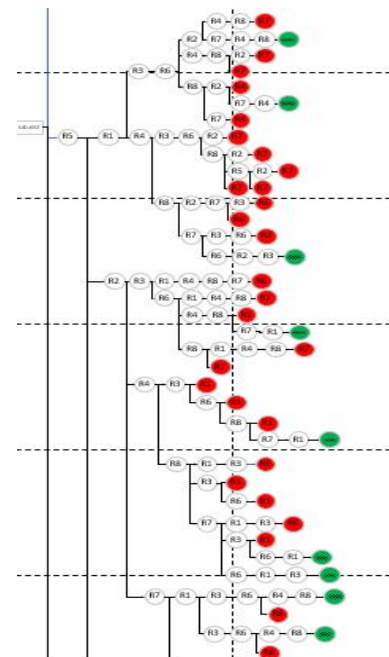
	UD. XY Z	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8
UD. XY Z	0	6 0 0 m	9 2 0 m	-	120 0m	6 5 0 m	5 5 0 m	4 6 9 m	4 9 8 m
R1	600 m	0	6 7 0 m	8 9 0 m	86 0m	-	-	-	-
R2	-	-	0	6 5 0 m	78 8m	-	-	8 8 6 m	-
R3	860 m	8 9	-	0	-	6 7	6 5	-	-

		0 m				7 m	8 m		
R4	120 0m	-	-	9 7 5 m	0	-	-	-	6 0 5 m
R5	650 m	9 0 0 m	8 4 5 m	6 7 7 m	88 2m	0	8 7 6 m	4 5 5 m	-
R6	-	6 1 2 m	6 7 1 m	-	70 1m	8 7 6 m	0	-	4 5 0 m
R7	-	8 3 3 m	-	5 9 2 m	69 9m	-	9 0 7 m	0	-
R8	498 m	8 1 0 m	6 6 3 m	6 5 5 m	-	5 8 9 m	-	4 5 0 m	0

Tabel 2. Matriks jarak setelah proses penyederhanaan rute

D) Implementasi Sirkuis Hamilton

Setelah dipetakan rute optimal pada tiap titik. Maka langkah selanjutnya adalah mengimplementasikan rute tersebut ke dalam graf dengan model sirkuit Hamilton.



Gambar 4. Implementasi Sirkuit Hamilton

E) Penerapan Rute Optimal Menggunakan Algoritma Greedy

Setelah melakukan percobaan kemungkinan rute yang dilalui, maka langkah selanjutnya adalah penentuan rute terpendek. Dari 30 alternatif rute, pada penelitian ini peneliti menampilkan 18 kemungkinan rute jarak terbaik. Alternatif tersebut dapat dilihat pada Tabel 3.

No.	Rute alternatif	Jarak
1	UD.XYZ-R5-R1-R3-R6-R2-R7-R4-R8- UD.XYZ	6457m
2	UD.XYZ-R5-R1-R3-R6-R8-R2-R7-R4- UD.XYZ	6996m
3	UD.XYZ-R5-R1-R4-R8-R7-R6-R2-R3- UD.XYZ	6553m
4	UD.XYZ-R2-R3-R6-R4-R8-R7-R1- UD.XYZ	5992m
5	UD.XYZ-R5-R4-R3-R6-R6-R8-R7-R1- UD.XYZ	6531m
6	UD.XYZ-R5-R2-R4-R8-R7-R3-R6-R1- UD.XYZ	5800m
7	UD.XYZ-R5-R2-R4-R8-R7-R6-R1-R3-UD.XYZ	6607m
8	UD.XYZ-R5-R2-R7-R1-R3-R6-R4-R8-UD.XYZ	6566m
9	UD.XYZ-R5-R2-R7-R3-R6-R1-R4-R8-UD.XYZ	6206m
10	UD.XYZ-R5-R2-R7-R3-R6-R4-R8-R1-UD.XYZ	6347m
11	UD.XYZ-R5-R2-R7-R3-R6-R8-R1-R4-UD.XYZ	6301m
12	UD.XYZ-R5-R2-R7-R4-R3-R6-R8-R1-UD.XYZ	6573m
13	UD.XYZ-R5-R2-R7-R4-R8-R3-R6-R1-UD.XYZ	5560m
14	UD.XYZ-R5-R2-R7-R6-R1-R4-R8-R3-UD.XYZ	6880m
15	UD.XYZ-R5-R2-R7-R6-R4-R8-R1-R3-UD.XYZ	7154m
16	UD.XYZ-R5-R2-R7-R6-R4-R8-R3-R1-UD.XYZ	6739m
17	UD.XYZ-R5-R2-R7-R6-R8-R1-R4-R3-UD.XYZ	7243m
18	UD.XYZ-R5-R2-R7-R6-R8-R3-R1-R4-UD.XYZ	7343m

Tabel 3. Rute Alternatif

V. PENUTUP

Algoritma *greedy* memungkinkan kita memilih hasil minimum dari beberapa alternatif solusi lokal. Dalam hal ini hasil minimum yang dimaksud adalah jarak rute terpendek yaitu pada rute alternatif nomor 13 dengan panjang rute **5560 meter**. Kelebihan dari algoritma *greedy* adalah kecepatan dalam menentukan rute optimal karena langsung mencoba semua peluang yang ada. Adapun kelemahan dari algoritma *greedy* adalah membutuhkan waktu yang sangat lama apabila titik – titik sebagai alternatif rute lebih banyak, karena pengecekan jarak antar titik dilakukan dengan mengecek satu per satu jarak tempuh antar titik. Dengan demikian algoritma *greedy* lebih cocok untuk jumlah alternatif titik yang sedikit sedangkan untuk titik yang lebih besar tidak cocok.

Dalam penelitian ini variabel yang digunakan hanya berupa jarak tempuh antar titik. Untuk alternatif lain seperti waktu, efisiensi bahan bakar dan kemacetan tidak dihitung. Kedepannya penelitian ini akan menambahkan beberapa variabel pendukung agar dapat digunakan dalam kondisi yang bervariasi.

REFERENCES

- [1] B.A. Pradhana.” Studi dan Implementasi Persoalan Lintasan Terpendek Suatu Graf Dengan Algoritma Dijkstra dan Algoritma Bellman-Ford”. Bandung. 2007.
- [2] J. Daud. “Studi Efektifitas Penggunaan jalan Kota Medan”. Jurnal Sistem Teknik Industri, No. 3 Vol. 6. 2005.
- [3] L. Fanani, E.M. Adams, S. Wicaksono, A. Satrio. 2012. “Rancang Bangun Aplikasi Web Pencarian Rute Terpendek Antar Kota Medan Algoritma Dijkstra”. Jurnal Basic Science And Techonology, 1(3),7-11, Malang. 2012.
- [4] Iryanto. “Pengantar Teori dan Aplikasi Dan Diagram DFD”. Medan: USU Press. 2003
- [5] R. Munir. “Deskripsi Sistem. Edisi ke-2”. Bandung: Informatika. 2003.
- [6] M.R. Rizaldi, “Pencarian Jalur Terpendek dalam GPS dengan Menggunakan Teori Graf”. Universitas Yogyakarta. 2007
- [7] R. Saputra. ”Sistem Informasi Pencarian Obyek Kota Medan Dengan Algoritma Dijkstra.” Jurnal Matematika, No. 1, Vol.14, Hal 19-24, April 2011.
- [8] <http://jurnal.untag-sby.ac.id/index.php/KONVERGENSI/article/view/5295> - diakses pada tanggal 29 Maret 2022