



PAPER – OPEN ACCESS

Dynamic Programming dalam Penyelesaian Masalah Penjadwalan

Author : Fikri Akbar L dan Rosnani Ginting
DOI : 10.32734/ee.v2i3.751
Electronic ISSN : 2654-704X
Print ISSN : 2654-7031

Volume 2 Issue 3 – 2019 TALENTA Conference Series: Energy & Engineering (EE)



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

Published under licence by TALENTA Publisher, Universitas Sumatera Utara



Dynamic Programming dalam Penyelesaian Masalah Penjadwalan

Fikri Akbar L¹, Rosnani Ginting²

¹Jl. Bahagia, Padang Bulan, Kota Medan 20155, Indonesia

²Kampus USU, Jl. Almamater, Padang Bulan, Kota Medan 20155, Indonesia

¹f.akbarluthfi@gmail.com, ²rosnani_usu@yahoo.co.id

Abstrak

Traveling Salesman Problem adalah salah satu masalah untuk menemukan rute terpendek dari bepergian seorang salesman dari kota pertama dan kemudian ke kota tujuan dan akhirnya kembali ke kota pertama, tetapi satu kota hanya sekali dikunjungi. Ada beberapa algoritma untuk menyelesaikan masalah salesman keliling, seperti Greedy Algorithm, Artificial Bee Colony Algorithm, Algoritma Heuristics Insertion Termurah, Algoritma Genetika dan banyak lagi. Dalam tulisan ini, hanya dibahas algoritma serakah, algoritma heuristik penyisipan termurah, dan pemrograman dinamis. Setelah dibandingkan menggunakan contoh kasus dengan 5 kota dan diselesaikan dengan algoritma ketiga, rute terpendek sama tetapi cara penyelesaiannya berbeda. Mereka memiliki kelebihan dan kekurangan masing-masing, dan memiliki karakteristik masing-masing. Algoritma serakah lebih cocok bila digunakan untuk sejumlah kota yang tidak terlalu banyak karena prosesnya lebih sederhana, tetapi algoritma heuristik penyisipan termurah lebih cocok untuk kasus dengan lebih banyak kota yang prosesnya lebih rumit daripada algoritma serakah. Menghitung dalam pemrograman dinamis harus benar karena akan berpengaruh untuk hasil penghitungan berikutnya.

Kata Kunci: *Cheapest Insertion Heuristics, Greedy, Dynamic Programming, Travelling Salesman Problem.*

Abstract

Travelling Salesman Problem is one of problems to find shortest route from travelling a salesman from first city and then to destination cities and finally back to first city, but one city just only once visited. There are some algorithms to solving travelling salesman problem, such as Greedy Algorithm, Artificial Bee Colony Algorithm, Cheapest Insertion Heuristics Algorithm, Genetic Algorithm and many more. In this paper, only greedy algorithm, cheapest insertion heuristics algorithm and dynamic programming are discussed. After compared using an example case with 5 cities and solved by third of algorithm, shortest route is same but the way to solving is different. They have advantages and disadvantages of each, and has the characteristics of each. Greedy algorithm is more suitable when used for a number of cities that are not too much because the process is more simple, but cheapest insertion heuristics algorithm is more suitable to case with more city through the process more complicated than greedy algorithm. Counting in Dynamic programming must be right because it will be influence for the next counting result.

Keywords: *Cheapest Insertion Heuristics, Greedy, Dynamic Programming, Travelling Salesman Problem.*

1. Pendahuluan

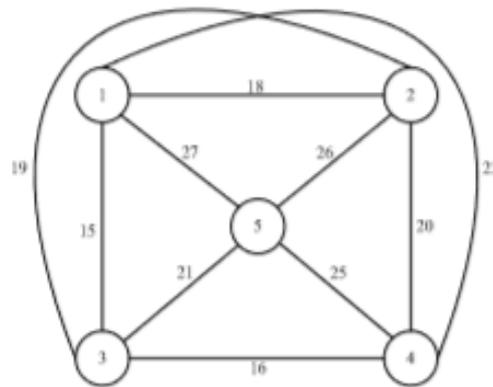
Ketika kita berpergian dari suatu tempat ke tempat lain maka kita mempertimbangkan waktu seefisien mungkin dan biayanya sehingga kita mencari jalur terpendek yang akan ditempuh. Permasalahan tersebut dikenal dengan nama Travelling Salesman Problem. Travelling Salesman Problem (TSP) adalah perjalanan seorang salesman menuju ke semua tempat yang akan dituju dan akhirnya kembali ke tempat awal dengan menggunakan jalur terpendek, akan tetapi semua tempat hanya boleh dilalui satu kali. Pada permasalahan Travelling Salesman Problem ini dicari sebuah solusi yaitu menghitung jalur terpendek atau rute perjalanan minimum dari semua lintasan. Terdapat beberapa algoritma yang digunakan untuk menemukan jalur terpendek dari travelling salesman problem tersebut. Algoritma yang digunakan diantaranya adalah Algoritma Greedy, Algoritma Artificial Bee Colony, Algoritma Cheapest Insertion Heuristics, Algoritma Genetika dan masih banyak yang lainnya. Masing-masing algoritma memiliki kelebihan dan kekurangannya. Suatu algoritma memiliki hasil yang berbeda-beda, karena belum tentu suatu algoritma yang memiliki optimasi yang tinggi untuk suatu kasus memiliki optimasi yang tinggi pula untuk kasus yang lain. Dengan demikian maka pada pembahasan kali ini akan dibandingkan hasil dari perhitungan dengan menggunakan algoritma greedy, algoritma cheapest insertion heuristics dan dynamic programming yang nantinya akan diketahui algoritma mana yang lebih baik untuk menyelesaikan kasus Travelling Salesman Problem yang ada. Travelling Salesman Problem adalah salah satu contoh persoalan optimasi yang mengaplikasi teori graf. Bentuk umum Travelling Salesman Problem pertama kali dipelajari oleh para matematikawan pada tahun 1930 yang diawali oleh Karl Menger di Vienna dan Harvard. Setelah itu permasalahan Travelling Salesman Problem dipublikasikan oleh Hassler Whitney dan Merrill Flood di Princeton. Menurut Dian (2013:2), Travelling Salesman Problem dikatakan ada dua jenis, yaitu: 1. Travelling Salesman Problem asimetris Pada Travelling Salesman Problem jenis ini, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1. Dengan n kota, besarnya ruang pencarian adalah $n! = (n - 1)!$ jalur yang mungkin n 2. Travelling Salesman Problem simetris Sedangkan pada Travelling Salesman Problem jenis simetris, biaya dari kota 1 ke kota 2 adalah sama dengan biaya dari kota 2 ke kota 1. Apabila dengan n kota, jumlah jalur yang mungkin adalah: $n! = (n-1)!$ jalur yang mungkin. $2n$ Menurut Horowitz dalam Dian (2008) Dynamic Programming adalah sebuah metode desain algoritma yang dapat digunakan ketika solusi untuk sebuah masalah yang mungkin terlihat sebagai hasil dari sebuah urutan keputusan[2]. Dynamic Programming dapat dikatakan pula sebagai metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah (step) atau tahapan (stage) sedemikian rupa sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Pada penyelesaian persoalan dengan metode ini: 1) Terdapat sejumlah berhingga pilihan yang mungkin. 2) Solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya. 3) Menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.

2. Metodologi Penelitian

Penelitian ini dilakukan dengan langkah-langkah sebagai berikut:

1. Menguraikan konsep algoritma greedy, algoritma cheapest insertion heuristics dan dynamic programming untuk mencari lintasan terpendek.
2. Mengimplementasikan algoritma greedy, algoritma cheapest insertion heuristics dan dynamic programming dalam suatu kasus.
3. Melakukan perbandingan dari algoritma greedy, algoritma cheapest insertion heuristics dan dynamic programming sesuai dari hasil penyelesaian kasus yang ada.
4. Membuat kesimpulan algoritma mana yang cocok untuk kasus tersebut. diantaranya adalah algoritma greedy, algoritma artificial bee colony, algoritma cheapest insertion heuristics, algoritma genetika dan masih banyak yang lainnya.

Namun pada pembahasan kali ini algoritma yang digunakan adalah algoritma greedy, algoritma cheapest insertion heuristics dan dynamic programming Ketiga algoritma tersebut diimplementasikan ke dalam sebuah contoh untuk mengetahui perbandingan dari ketiga algoritma tersebut. Untuk contoh kasusnya dapat dilihat pada gambar 2. Seperti terlihat pada gambar 2 terdapat 5 buah kota yang masing-masing tempat mempunyai jarak yang berbeda-beda.



Gambar 1. Skema Jalur

Jarak antara kota yang satu dengan kota yang lainnya dapat dilihat pada tabel 1 di bawah ini:

Tabel 1. Jarak antar Kota

Kota Asal	Kota Tujuan	Jarak
1	2	18
1	3	15
1	4	23
1	5	27
2	3	19
2	4	20
2	5	26
3	4	16
3	5	21
4	5	25

3. Hasil dan Pembahasan

A. Algoritma Greedy

Cara penyelesaian travelling salesman problem dengan algoritma greedy adalah:

1. Mulai dari sembarang tujuan, jika tempat asal telah ditetapkan maka mulailah dari tempat tujuan tersebut. Tempat asal ya adalah no.1
2. Kemungkinan solusi yang muncul adalah $(n-1)! = (5-1)! = 24$, maka kemungkinan jalurnya adalah:

Tabel 2. Kemungkinan Jalur

Langkah ke-	Jalur	Total Jarak
1	1-2-3-4-5-1	18+19+16+25+27 = 105
2	1-2-3-5-4-1	18+19+21+25+23 = 106
3	1-2-4-5-3-1	18+20+25+21+15 = 99
4	1-2-4-3-5-1	18+20+16+21+27 = 102
5	1-2-5-4-3-1	18+26+25+16+15 = 100
6	1-2-5-3-4-1	18+26+21+16+23 = 104
7	1-3-4-5-2-1	15+16+25+26+18 = 100
8	1-3-3-2-5-1	15+16+20+26+27 = 104
9	1-3-4-2-5-1	15+21+26+20+23 = 105
10	1-3-5-4-2-1	15+21+25+20+18 = 99
11	1-3-2-4-5-1	15+19+20+25+27 = 106
12	1-3-2-5-4-1	15+19+26+25+23 = 108
13	1-4-5-3-2-1	23+25+26+19+15 = 108
14	1-4-5-3-2-1	23+25+21+19+18 = 106
15	1-4-2-3-5-1	23+20+19+21+27 = 110
16	1-4-2-3-5-1	23+20+26+21+15 = 105
17	1-4-3-5-2-1	23+16+21+26+18 = 104
18	1-4-3-2-5-1	23+16+19+26+27 = 111
19	1-5-2-3-4-1	27+26+19+16+23 = 111
20	1-5-2-4-3-1	27+26+20+16+15 = 104
21	1-5-3-4-2-1	27+32+16+20+18 = 102
22	1-5-3-2-4-1	27+21+19+20+23 = 110
23	1-5-4-2-3-1	27+25+20+19+15 = 106
24	1-5-4-3-2-1	27+25+16+19+18 = 105

Berdasarkan hasil perhitungan dengan algoritma greedy diatas maka jalur yang terpendek ada dua buah kemungkinan jalur yaitu: 1-2-4-5-3-1 atau 1-3-5-4-2-1 dengan total jarak 99km.

B. Algoritma Cheapest Insertion Heuristics

Berdasarkan tahapan-tahapan pada algoritma cheapest insertion heuristics maka penyelesaian kasus pada gambar 2 tersebut adalah: 1. Penelusuran dimulai dari sebuah kota pertama yang dihubungkan dengan sebuah kota terakhir yaitu dari kota 1 ke kota 5 2. Dibuat sebuah hubungan subtour antara 2 kota tersebut : (1,5) □ (5,1) 3. Ganti salah satu arah hubungan (arc) dari dua kota dengan kombinasi dua arc, yaitu arc (i,j) dengan arc (i,k) dan arc (k,j), dengan k diambil dari kota yang belum masuk subtour dan dengan tambahan jarak terkecil. Jarak diperoleh dari:

Cij

Cik adalah jarak dari kota i ke kota k

Ckj adalah jarak dari kota k ke kota j

Cij adalah jarak dari kota i ke kota j

Maka dibuatlah tabel 3 yang menyimpan kota yang bisa disisipkan dalam subtour beserta tambahannya.

Tabel 3. Arc Penambahan Subtour ke1

Arc yang akan digunakan	Arc yang ditambahkan ke subtour	Tambahan Jarak
(1,5)	(1,2)-(2,5)	$C_{12}+C_{25}-C_{15} = 18+26-27=17$
(1,5)	(1,3)-(3,5)	$C_{13}+C_{35}-C_{15} = 15+21-27=9$
(1,5)	(1,4)-(4,5)	$C_{14}+C_{45}-C_{15} = 23+25-27=21$
(5,1)	(5,2)-(2,1)	$C_{52}+C_{21}-C_{51} = 18+26-27=17$
(5,1)	(5,3)-(3,1)	$C_{53}+C_{31}-C_{51} = 21+15-27=9$
(5,1)	(5,4)-(4,1)	$C_{54}+C_{41}-C_{51} = 25+23-27=21$

Berdasarkan tabel 3 diatas maka dapat diperoleh jarak yang terkecil adalah 9, apabila arc (1,5) diganti dengan arc (1,3) dan arc (3,5) atau arc (5,1) diganti dengan arc (5,3) dan arc (3,1). Karena terdapat dua jarak yang terkecil maka dari kemungkinan tersebut bisa dipilih salah satu, misalkan kemungkinan pertama maka subtour (1,5) □ (5,1) diganti menjadi (1,3) □ (3,5) □ (5,1).

Karena masih ada kota yang belum masuk, maka dibuat kembali seperti hasil dari tabel 4.

Tabel 4. Arc Penambahan Subtour ke2

Arc yang akan digunakan	Arc yang ditambahkan ke subtour	Tambahan Jarak
(1,3)	(1,2)-(2,3)	$C_{12}+C_{23}-C_{13} = 18+19-15=22$
(1,3)	(1,4)-(4,3)	$C_{14}+C_{43}-C_{13} = 23+16-15=24$
(3,5)	(3,2)-(2,5)	$C_{32}+C_{25}-C_{35} = 19+26-21=24$
(3,5)	(3,4)-(4,5)	$C_{34}+C_{45}-C_{35} = 16+25-21=20$
(5,1)	(5,2)-(2,1)	$C_{52}+C_{21}-C_{51} = 26+18-27=17$
(5,1)	(5,4)-(4,1)	$C_{54}+C_{41}-C_{51} = 25+23-27=21$

C. Dynamic Programming

Berdasarkan cara dynamicProgramming maka travelling salesman problem dapat diselesaikan dengan cara sebagai berikut:

Notasi-notasi yang digunakan adalah:

N = Banyaknya tahap

n = Indeks untuk tahap sekarang (n=1,2,...N)

Sn = Keadaan sekarang untuk tahap n

Snk = Keadaan k yang mungkin ditempuh pada tahap n (k ∈ bilangan bulat positif)

CSn = Biaya pada tahap n

S0 = Kota asal

Xn = Peubah keputusan untuk tahap n

X*n = Nilai Optimal

Xn (diketahui Sn) Fn(Sn, Xn) = Kontribusi tahap n, n+1, ..., N

Untuk tahapan berikutnya nilainya mengacu dari hasil tahapan ke-5. Maka untuk tahapan ke-4 jika dihitung dengan menggunakan cara dynamic programming dapat dilihat pada tabel di bawah ini:

Tabel 5. Perhitungan Dynamic Programming

X ₄	f ₄ (S ₂ X ₄) = C _{XX4} + f* ₅ (X ₄)				f* ₄ (s)	X* ₄
	S ₄₁	S ₄₂	S ₄₃	S ₄₄		
S ₃₁	-	15+19	23+20	27+26		
S ₃₂	18+19	-	23+16	27+21		
S ₃₃	18+20	15+16	-	27+25		
S ₃₄	18+26	15+21	23+25	-		

Atau jika nilai tersebut dihitung maka dapat dilihat pada tabel di bawah ini,

Tabel 6. Hasil Perhitungan Dynamic Programming

X ₄	f ₄ (S ₂ X ₄) = C _{XX4} + f* ₅ (X ₄)				f* ₄ (s)	X* ₄
	S ₄₁	S ₄₂	S ₄₃	S ₄₄		
S ₃₁	-	34	43	53	34	S ₄₂
S ₃₂	37	-	39	48	37	S ₄₁
S ₃₃	38	31	-	52	31	S ₄₂
S ₃₄	44	36	48	-	36	S ₄

Sesuai dengan hasil dari tahapan ke-4 maka diperoleh nilai f*₄(s) paling minimum adalah yang melalui S₄₂ dengan nilai f*₄(s) = 31. Dengan demikian pada tahapan ini yang diambil adalah rute S₄₂. Selanjutnya untuk tahapan ke-3 dapat dilihat pada tabel di bawah ini.

D. Perbandingan Algoritma Greedy dengan Algoritma Cheapest Insertion Heuristics

Setelah diketahui jalur terpendek yang dihasilkan dengan algoritma greedy dan algoritma cheapest insertion heuristics dari kasus diatas yaitu 99 Km dengan jalur 1-3-54-2-1 maka dapat diperoleh perbandingan kedua algoritma tersebut seperti pada Tabel 8.

Tabel 8. Perbandingan Algoritma Greedy dengan Algoritma Cheapest Insertion Heuristics

Algoritma	Greedy	Cheapest Insertion Heuristics	Dinamik
Kelebihan	<ul style="list-style-type: none"> - Pencarian lebih simple apabila kota tidak terlalu banyak - Dapat sampai tepat waktu menuju tujuan karena prinsip pencarian lintasan terpendek memakai fungsi seleksi dan itu berguna untuk menentukan jalan tesingkat untuk menuju suatu tempat 	<ul style="list-style-type: none"> - Algoritma ini masih stabil digunakan untuk kasus TSP dengan jumlah kota yang besar 	<ul style="list-style-type: none"> - Hasil perhitungannya optimal - Sangat tepat apabila diimplementasikan ke bahasa pemrograman pascal
Kekurangan	<ul style="list-style-type: none"> - Keputusan optimal di setiap sub langkah tidak menghantarkan kita ke solusi optimal global yang diinginkan - Kurang cocok digunakan apabila dengan jumlah kota yang besar 	<ul style="list-style-type: none"> - Prinsip pencariannya lebih rumit dan lebih lama dibandingkan algoritma greedy karena kemungkinan terjadinya perulangan perhitungan tambahan jarak (probabilitas edge yang akan digantikan) pada iterasi yang berbeda 	<ul style="list-style-type: none"> Hasil perhitungan selanjutnya sangat bergantung dari hasil perhitungan sebelumnya sehingga apabila hasil perhitungan sebelumnya salah maka hasil selanjtnya pun salah

Data yang diperlukan dalam penelitian ini adalah sebagai berikut:

- a) Data Primer : waktu pengerjaan suatu produk oleh unit produksi pada sepanjang lintasan jalur produksi.
- b) Data Sekunder: waktu set up mesin, latar belakang perusahaan, umur mesin, jumlah mesin, kapasitas mesin dan rencana produksi untuk suatu periode tertentu.

4. Kesimpulan dan Saran

Kesimpulan dari paper ini adalah:

1. Algoritma greedy, algoritma cheapest insertion heuristics dan dynami programming adalah algoritma yang bisa digunakan untuk pemecahan TSP.
2. Dari hasil penyelesaian dengan algoritma greedy, algoritma cheapest insertion heuristics dan dynamic programming maka rute terpendek adalah 99 Km dengan jalur 1-3-5-4-2-1
3. Untuk kasus pada paper ini algoritma greedy lebih simple cara penyelesaiannya karena jumlah kotanya masih sedikit akan tetapi apabila jumlah kotanya lebih banyak maka lebih baik menggunakan algoritma cheapest insertion heuristics.
4. Ketiga algoritma tersebut memiliki kelebihan dan kekurangan masing-masing.
4. Jumlah kota yang dikunjungi mempengaruhi pemilihan algoritma.

Referensi

- [1] Dian. 2013. Algoritma Optimasi Untuk Penyelesaian Travelling Salesman Problem (Optimization Algorithm For Solving Travelling Salesman Problem). Jurnal Transformatika. Jurusan Teknologi Informasi Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang. Volume 11, No.1, Juli 2013.
- [2] Ginting, Rosnani. 2016. *Penjadwalan Mesin*.
- [3] Kusriani dan Istiyanto, J.E.. 2007. *Penyelesaian Travelling Salesman Problem dengan Algoritma Cheapest Insertion Heuristic dan Basis Data*. Jurnal Informatika. Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra, Vol. 8, No. 2, Nopember 2007
- [4] Sinulingga, Sukaria. 2016. *Metode Penelitian*.