

Pemilihan Rute Terbaik Pariwisata di Solo dengan *Dynamic Programming*

Haris Salman Al-Ghifary/ 13517052

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13517052@itb.ac.id

Abstract— Makalah ini akan menjelaskan tentang pengoptimasian rute perjalanan pariwisata di kota Solo dengan rute terbaik yang dapat ditemukan. Dalam permasalahan ini digunakan Traveling Salesman Problem atau TSP untuk menyelesaikan persoalan ini, Traveling Salesman Problem sendiri adalah suatu permasalahan di dalam dunia komputer dimana sebuah graf tak berarah berbobot menentukan lintasan terpendek yang dapat dilalui dan akan kembali lagi ke simpul awal. Maka penentuan rute terbaik yang ditempuh dapat menggunakan algoritma Dynamic Programming.

Keywords—Traveling Salesman Problem, Dynamic Programming, rute, optimasi

I. PENDAHULUAN

Dalam kehidupan ini rekreasi merupakan sesuatu yang sangat penting untuk dilakukan dalam rangka menghilangkan kejenuhan dan kepenatan. Oleh karena itu pariwisata selalu menjadi primadona bagi orang untuk melakukan aktivitas saat hari libur tiba. Apalagi di negara Indonesia yang dimana destinasi pariwisata nya telah terkenal sampai ke penjuru dunia. Ada Bali, Raja Ampat, Bunaken, Bromo, Solo dan masih banyak yang lainnya. Sektor pariwisata juga merupakan lumbung uang bagi masyarakat dan pemerintah karena tiap tahunnya destinasi terkenal yang ada di Indonesia dapat menarik ribuan bahkan jutaan pengunjung. Untuk kota Solo sendiri pariwisata menjadi suatu sektor penting yang dikelola dengan serius oleh pemerintah kota Solo. Biasanya para turis domestik maupun mancanegara senang menaiki Bus Pariwisata atau yang dikenal dengan bus Gatotkaca, transportasi ini sangat diminati oleh pengunjung karena kenyamanan serta murah.

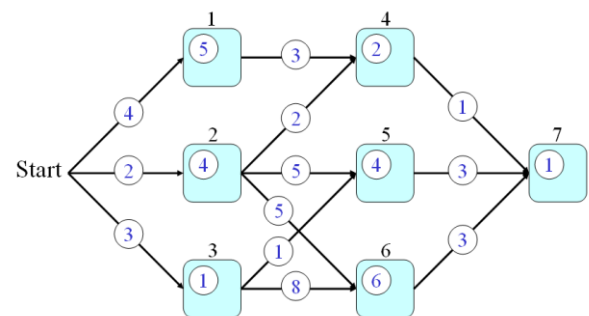
Transportasi dalam bidang pariwisata merupakan faktor penting dalam keberhasilan sektor ini, transportasi haruslah nyaman dan terjangkau bagi para pengunjung. Maka untuk menunjang hal tersebut perlu dilakukan pemangkusan biaya yang dikeluarkan oleh pemerintah kota agar dapat keuntungan sebesar mungkin. Dalam hal ini untuk pemangkusan biaya maka hal yang paling ideal adalah mengurangi jarak tempuh yang dilalui oleh bus Gatotkaca, sehingga dalam mengelilingi kota Solo dapat dilalui dengan rute yang mangkus sehingga biaya untuk bahan bakar dapat berkurang. Pemangkusan rute tidak hanya akan memangkas biaya yang dikeluarkan pemerintah kota Solo, akan tetapi juga dapat menghemat waktu baik pengunjung maupun pihak yang berwenang.

Permasalahan diatas merupakan salah satu contoh dari dunia komputer yaitu Traveling Salesman Problem atau TSP. Permasalahan rute terbaik dapat menggunakan pendekatan graf sehingga akan memiliki struktur yang optimal. Dalam penyelesaian rute terbaik ini sebenarnya dapat diselesaikan dalam berbagai cara, namun menurut saya algoritma yang akan saya gunakan merupakan yang paling mangkus.

II. DASAR TEORI

A. Dynamic Programming

Program dinamis merupakan metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan atau stage, sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan. Istilah program dinamis muncul karena perhitungan solusi menggunakan tabel-tabel [1].

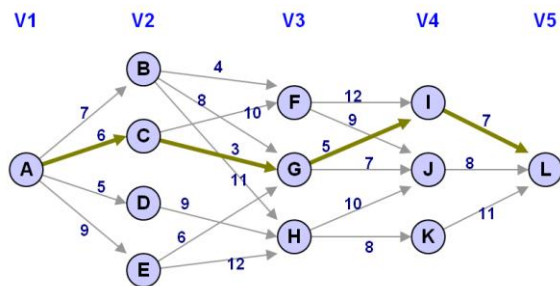


Terkadang program dinamis dan greedy disamakan bagi sebagian orang yang belum paham sekali dengan kedua algoritma tersebut. Ada perbedaan mendasar kenapa program dinamis berbeda dengan algoritma greedy. Pada algoritma greedy suatu permasalahan hanya akan menghasilkan satu solusi saja, sedangkan algoritma program dinamis dapat menghasilkan lebih dari satu solusi. Hal ini dapat terjadi karena dalam algoritma program dinamis menggunakan serangkaian keputusan berjenjang yang akan dipakai terus sampai solusi optimal ditemukan.

Karakteristik penyelesaian persoalan dengan program dinamis:

1. Terdapat sejumlah berhingga pilihan yang mungkin,

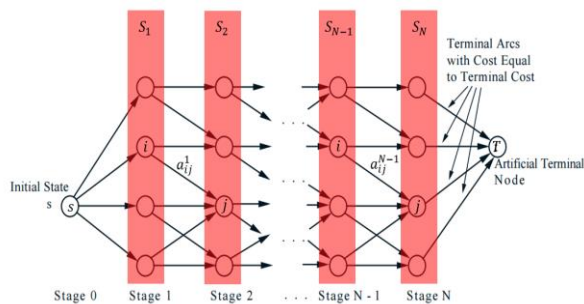
- Solusi pada setiap tahap dibangun dari hasil solusi tahap sebelumnya,
- Kita menggunakan persyaratan optimasi dan kendala untuk membatasi sejumlah pilihan yang harus dipertimbangkan pada suatu tahap.
- Biaya pada setiap tahap meningkat secara teratur seiring dengan bertambahnya jumlah tahapan.
- Keputusan terbaik pada suatu stage merupakan keputusan yang bersifat independent yang berarti tidak bergantung pada keputusan sebelumnya.



Dalam penyelesaian persoalan dengan algoritma program dinamis dapat menggunakan dua cara yaitu:

- Program dinamis maju (Forward)

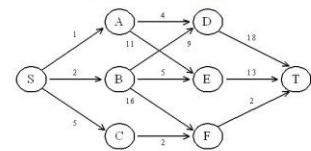
Program dinamis akan bergerak dari depan yaitu dari tahap 1, terus maju ke tahap 2, 3 dan seterusnya sampai tahap n. Runtutan variable adalah $x_1, x_2, x_3, \dots, x_n$.



- Program dinamis mundur (Backward)

Program dinamis mundur akan bergerak dari belakang yaitu dari tahap n, terus mundur ke tahap n-1, n-2 dan seterusnya sampai tahap 1. Runtutan variable adalah $x_n, x_{n-1}, x_{n-2}, \dots, x_1$.

Backward Approach (Forward Reasoning)

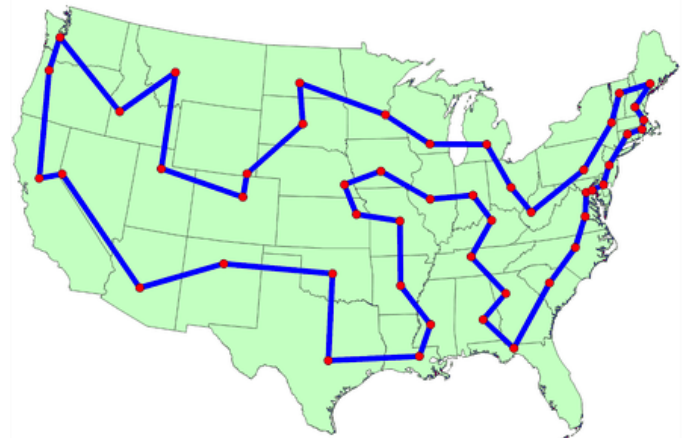


$$\begin{aligned}
 & d(S, A) = 1 \\
 & d(S, B) = 2 \\
 & d(S, C) = 5 \\
 & d(S, D) = \min\{d(S, A) + d(A, D), d(S, B) + d(B, D)\} \\
 & \quad = \min\{1 + 4, 2 + 9\} = 5 \\
 & d(S, E) = \min\{d(S, A) + d(A, E), d(S, B) + d(B, E)\} \\
 & \quad = \min\{1 + 11, 2 + 5\} = 7 \\
 & d(S, F) = \min\{d(S, A) + d(A, F), d(S, B) + d(B, F)\} \\
 & \quad = \min\{2 + 16, 5 + 2\} = 7
 \end{aligned}$$

B. Traveling Salesman Problem

Traveling salesman problem atau yang biasa dikenal dengan TSP merupakan sebuah permasalahan yang ada pada dunia komputer. Permasalahan ini biasanya digunakan untuk menentukan rute terbaik yang dapat ditempuh dari suatu lokasi ke lokasi lainnya dan akan kembali ke lokasi awal. Biasanya permasalahan ini digunakan untuk optimasi biaya yang dapat dikeluarkan oleh suatu pihak.

Masalah pengoptimalan dengan TSP biasanya dapat ditemui dalam bidang transportasi. Pihak yang berurusan dengan transportasi pasti menginginkan perjalanan yang cepat dan tentunya murah. Oleh karena itu solusi yang dapat ditemukan oleh algoritma ini akan sangat membantu bagi perusahaan maupun orang yang berada dalam bidang transportasi. TSP biasanya digunakan dengan representasi graf tak berarah dengan sisinya memiliki bobot yang melambangkan jarak antar lokasi.



Karakteristik persoalan dengan TSP:

- Perjalanan dimulai dari suatu lokasi dan akan ke lokasi lain namun akan berakhir di lokasi awal
- Semua lokasi harus dikunjungi tepat satu kali dalam perjalanan.
- Perjalanan tidak akan kembali ke lokasi awal jika belum mengunjungi semua lokasi yang ada.
- Tujuan algoritma ini adalah pengoptimalan jarak tempuh yang dilalui.

III. PEMILIHAN RUTE TERBAIK DENGAN DYNAMIC PROGRAMMING

Untuk pengimplementasian dalam kehidupan nyata, algoritma dynamic programming akan menggunakan salah satu variasinya yaitu menggunakan Traveling Salesman Problem. Akan dilakukan optimasi pemilihan rute pariwisata yang berada di kota Solo yaitu dengan bus Gatotkaca.

Bus Gatotkaca merupakan bus pariwisata yang ada di kota Solo. Pengambilan nama gatotkaca sendiri berasal dari salah satu cerita wayang yaitu Gatotkaca yang terkenal akan tulang kawat dan balung besinya. Bus Gatotkaca sendiri merupakan bus tingkat yang telah diresmikan sekitar dua tahun lalu, bus ini disediakan oleh pemerintah kota Solo untuk memfasilitasi turis yang datang ke Solo. Bus ini sendiri diresmikan oleh walikota solo yaitu FX Hadi Rudyatmo di depan balai Kota Solo. Harga sewa untuk dapat menaiki bus ini adalah 2 juta per tiga jam, yang mana dinilai terlalu mahal, meskipun fasilitas dari bus ini sangat bagus. Untuk mengatasi masalah ini, maka harus dilakukan optimasi baik biaya maupun waktu tempuh agar semua pengunjung ataupun pihak pengelola sama-sama untung. Bus ini memiliki desain yang terkesan bernuansa Jawa, didasari oleh corak batik dan sejenisnya.

Adapun bus wisata Werkudara yang mana bus ini memiliki harga yang relatif lebih murah dibandingkan saudaranya Gatotkaca. Untuk tarif sewanya sendiri sekitar 800 ribu per tiga jam atau 20 ribu per orang. Untuk rutenya sendiri dapat memilih sesuai keinginan pengunjung, namun biasanya pengunjung meminta turun dilokasi yang terkenal Namanya.



Baik kedua bus pasti memiliki tujuan perjalanan yang hampir sama. Kita asumsikan pengunjung ingin mengunjungi 4 destinasi wisata terkenal di Solo antara lain Museum batik Keris, Keraton Solo, Pasar Klewer, dan Benteng Vastenburg. Titik tujuan dapat kita asumsikan dengan simpul A1, A2, A3, A4 dengan sisinya memiliki bobot berbeda sesuai jarak antar lokasinya.

- A1 : Balaikota Solo
- A2 : Keraton Solo
- A3 : Pasar Klewer
- A4 : Benteng Vastenburg
- A5 : Taman Sriwedari

Jarak antar lokasi dapat direpresentasikan dengan adjacency matriks dibawah ini.

0	9	6	7	2
9	0	7	3	9
6	7	0	4	2
7	3	4	0	5
2	8	2	5	0

Dengan basis dan rekurens sesuai algoritma dynamic programming :

$$f(i, \emptyset) = c_i, 1 \leq i \leq n$$

(basis)

$$f(i, S) = \min\{c + f(j, S - \{j\})\}$$

(rekurens) (2)

Tahap 1:

$$f(2, \emptyset) = c_{21} = 9;$$

$$f(3, \emptyset) = c_{31} = 6;$$

$$f(4, \emptyset) = c_{41} = 7;$$

Tahap 2:

$$f(2, \{3\}) = \min\{c_{23} + f(3, \emptyset)\} = \min\{7 + 6\} = \min\{13\} = 13$$

$$f(3, \{2\}) = \min\{c_{32} + f(2, \emptyset)\} = \min\{7 + 9\} = \min\{16\} = 16$$

$$f(4, \{2\}) = \min\{c_{42} + f(2, \emptyset)\} = \min\{3 + 9\} = \min\{12\} = 12$$

$$f(2, \{4\}) = \min\{c_{24} + f(4, \emptyset)\} = \min\{3 + 7\} = \min\{10\} = 10$$

$$f(3, \{4\}) = \min\{c_{34} + f(4, \emptyset)\} = \min\{4 + 7\} = \min\{11\} = 11$$

$$f(4, \{3\}) = \min\{c_{43} + f(3, \emptyset)\} = \min\{4 + 6\} = \min\{10\} = 10$$

$$f(2, \{5\}) = \min\{c_{25} + f(5, \emptyset)\} = \min\{9 + 2\} = \min\{11\} = 11$$

$$f(3, \{5\}) = \min\{c_{35} + f(5, \emptyset)\} = \min\{2 + 2\} = \min\{4\} = 4$$

$$f(4, \{5\}) = \min\{c_{45} + f(5, \emptyset)\} = \min\{5 + 2\} = \min\{7\} = 7$$

$$f(5, \{2\}) = \min\{c_{52} + f(2, \emptyset)\} = \min\{8 + 9\} = \min\{17\} = 17$$

$$f(5, \{3\}) = \min\{c_{53} + f(3, \emptyset)\} = \min\{2 + 6\} = \min\{8\} = 8$$

$$f(5, \{4\}) = \min\{c_{54} + f(4, \emptyset)\} = \min\{5 + 7\} = \min\{12\} = 12$$

Tahap 3:

$$f(i, S) = \min\{c + f(j, S - \{j\})\}$$

dari hasil $S = 2$ dan $i \neq 1, 1 \notin S$ dan $i \notin S$.

Diperoleh:

$$f(2, \{3, 4\}) = \min\{c_{23} + f(3, \{4\}), c_{24} + f(4, \{3\})\}$$

$$\begin{aligned}
&= \min\{7 + 11, 3 + 10\} \\
&= \min\{18, 13\} = 13 \\
f(2, \{3, 5\}) &= \min\{c_{23} + f(3, \{5\}), c_{25} + f(5, \{3\})\} \\
&= \min\{7 + 4, 12 + 8\} \\
&= \min\{11, 20\} = 11 \\
f(2, \{4, 5\}) &= \min\{c_{24} + f(4, \{5\}), c_{25} + f(5, \{4\})\} \\
&= \min\{10 + 7, 12 + 12\} \\
&= \min\{17, 24\} = 17 \\
f(3, \{2, 4\}) &= \min\{c_{32} + f(2, \{4\}), c_{34} + f(4, \{2\})\} \\
&= \min\{7 + 10, 4 + 12\} \\
&= \min\{17, 16\} = 16 \\
f(3, \{2, 5\}) &= \min\{c_{32} + f(2, \{5\}), c_{35} + f(5, \{2\})\} \\
&= \min\{7 + 11, 2 + 17\} \\
&= \min\{19, 19\} = 19 \\
f(3, \{4, 5\}) &= \min\{c_{34} + f(4, \{5\}), c_{35} + f(5, \{4\})\} \\
&= \min\{4 + 7, 5 + 12\} \\
&= \min\{11, 17\} = 11 \\
f(4, \{2, 3\}) &= \min\{c_{42} + f(2, \{3\}), c_{43} + f(3, \{2\})\} \\
&= \min\{3 + 13, 4 + 16\} \\
&= \min\{16, 20\} = 16 \\
f(4, \{2, 5\}) &= \min\{c_{42} + f(2, \{5\}), c_{45} + f(5, \{2\})\} \\
&= \min\{3 + 11, 5 + 17\} \\
&= \min\{14, 22\} = 14 \\
f(4, \{3, 5\}) &= \min\{c_{43} + f(3, \{5\}), c_{45} + f(5, \{3\})\} \\
&= \min\{4 + 4, 5 + 8\} \\
&= \min\{8, 13\} = 8 \\
f(5, \{2, 3\}) &= \min\{c_{52} + f(2, \{3\}), c_{53} + f(3, \{2\})\} \\
&= \min\{8 + 13, 2 + 16\} \\
&= \min\{21, 18\} = 18 \\
f(5, \{2, 4\}) &= \min\{c_{52} + f(2, \{4\}), c_{54} + f(4, \{2\})\} \\
&= \min\{8 + 10, 5 + 12\} \\
&= \min\{18, 17\} = 17 \\
f(5, \{3, 4\}) &= \min\{c_{53} + f(3, \{4\}), c_{54} + f(4, \{3\})\} \\
&= \min\{2 + 11, 5 + 10\} \\
&= \min\{13, 15\} = 13
\end{aligned}$$

Tahap 4:

$$\begin{aligned}
f(2, \{3, 4, 5\}) &= \min\{c_{23} + f(3, \{4, 5\}), c_{24} + f(4, \{3, 5\}), \\
&c_{25} + f(5, \\
&\{2, 3\})\} \\
&= \min\{7 + 11, 3 + 8, 9 + 18\}
\end{aligned}$$

$$\begin{aligned}
&= \min\{18, 11, 27\} = 11 \\
f(3, \{2, 4, 5\}) &= \min\{c_{32} + f(2, \{4, 5\}), c_{34} + f(4, \{2, 5\}), \\
&c_{35} + f(5, \\
&\{2, 4\})\} \\
&= \min\{7 + 17, 4 + 14, 2 + 17\} \\
&= \min\{24, 18, 19\} = 18 \\
f(4, \{2, 3, 5\}) &= \min\{c_{42} + f(2, \{3, 5\}), c_{43} + f(3, \{2, 5\}), \\
&c_{45} + f(5, \\
&\{2, 3\})\} \\
&= \min\{3 + 11, 4 + 19, 5 + 18\} \\
&= \min\{14, 23, 23\} = 14 \\
f(5, \{2, 3, 4\}) &= \min\{c_{52} + f(2, \{3, 4\}), c_{53} + f(3, \{2, 4\}), \\
&c_{54} + f(4, \\
&\{2, 3\})\} \\
&= \min\{8 + 13, 2 + 16, 5 + 16\} \\
&= \min\{21, 18, 21\} = 18
\end{aligned}$$

Akhirnya diperoleh :

$$\begin{aligned}
f(1, \{2, 3, 4, 5\}) &= \min\{c_{12} + f(2, \{3, 4, 5\}), c_{13} + f(3, \{2, \\
&4, 5\}), c_{14} + \\
&f(4, \{2, 3, 5\}), c_{15} + f(5, \{2, 3, \\
&4\})\} \\
&= \min\{9 + 11, 6 + 18, 7 + 14, 2 + 18\} \\
&= \min\{20, 24, 21, 20\} = 20
\end{aligned}$$

Berdasar hasil dari tahapan-tahapan yang telah diperoleh oleh program dinamis dapat dilihat bahwa jarak minimum yang dapat ditempuh oleh bus Werkudara dan Gatotkaca adalah 20 km dengan dua rute yaitu :

1. Rute pertama

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 1$$

Rute tersebut dimulai dari Balaikota Solo $\rightarrow$ Keraton Solo $\rightarrow$ Benteng Vastenburg $\rightarrow$ Pasar Klewer $\rightarrow$ Taman Sriwedari $\rightarrow$ Balaikota Solo.

2. Rute kedua

$$1 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

Rute tersebut dimulai dari Balaikota Solo $\rightarrow$ Taman Sriwedari $\rightarrow$ Pasar Klewer $\rightarrow$ Benteng Vastenburg $\rightarrow$ Keraton Solo $\rightarrow$ Balaikota Solo

IV. IMPLEMENTASI

Untuk merealisasikan program dinamik ini perlu diadakan pengkajian lebih lanjut oleh pihak pemerintah kota Solo. Apakah hal ini mendesak atau tidak berdasar fakta-fakta yang telah ada di lapangan. Namun penulis sendiri berkeinginan agar makalah ini dapat membantu pengoptimalan biaya dan waktu yang ada dalam pelaksanaan bus wisata yang ada di kota Solo. Jika hal ini dapat

diimplementasikan, penulis membayangkan untuk pengimplementasian nya adalah dengan dibangun device khusus yang berada di kokpit supir guna menentukan rute selanjutnya yang ada, untuk kedepannya perlu juga untuk memperhitungkan kemacetan yang ada di kota Solo.

V. KESIMPULAN

Dynamic programming merupakan algoritma untuk menyelesaikan permasalahan optimasi yang berjalan lebih mangkus daripada algoritma lain seperti divide and conquer dan algoritma greedy. Dengan algoritma ini kita dapat menemukan beberapa alternatif solusi optimal yang mana itu akan membuat lebih banyak pilihan dalam menghadapi situasi dunia nyata.

Pada permasalahan makalah ini pula, pengoptimalan rute terbaik pariwisata di kota Solo dapat mempertimbangkan jarak tempuh antar destinasi wisata yang ada, hal ini membuat biaya yang dikeluarkan serta waktu yang dihabiskan menjadi lebih baik dan optimal. Algoritma dynamic programming sendiri dapat diterapkan dalam berbagai hal terutama dalam hal optimasi karena kemangkusan dari algoritma dynamic programming itu sendiri.

REFERENCES

- [1] Munir, Rinaldi. Strategi Algoritmik bab Dynamic Programming. Bandung. Informatika ITB.
- [2] <http://surakarta.go.id/?p=10182>
- [3] <http://pariwisatasolo.surakarta.go.id/wisata/keliling-solo-naik-bus-tingkat-werkudara>
- [4] <https://www.codechef.com/wiki/tutorial-dynamic-programming>

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, 25 April 2019



Haris Salman Al-Ghifary - 13517052