

OPTIMALISASI RUTE WISATA DI YOGYAKARTA MENGGUNAKAN METODE TRAVELLING SALESMAN PERSON DAN ALGORITMA BRUTE FORCE

by Condro Wibawa

Submission date: 29-Jan-2023 09:45PM (UTC-0500)

Submission ID: 2002031193

File name: 59-65_Condro_Wibawa.docx (187.03K)

Word count: 2536

Character count: 14793

OPTIMALISASI RUTE WISATA DI YOGYAKARTA MENGGUNAKAN METODE TRAVELLING SALESMAN PERSON DAN ALGORITMA BRUTE FORCE

Condro Wibawa

Fakultas Ilmu Komputer dan Teknologi Industri/Sistem Informasi, condro_wibawa@staff.gunadarma.ac.id,
Universitas Gunadarma

ABSTRACT

Yogyakarta is one of the popular tourist destinations in Indonesia. There are many tourist sites that can be visited in the city of Yogyakarta, including Prambanan Temple, Breksi Cliff, Van den Berg Museum, and Malioboro. For tourists who are not familiar with these tourist sites, determining the route to be visited is important because it will affect travel costs, especially transportation costs. To solve this problem, the Traveling Salesman Person method and the Brute Force Algorithm are used. The Brute Force algorithm was chosen because of its simple characteristics so that it is easy for all people to understand. The results of the research resulted in the most optimal route with a total cost of Rp. 100,300.00 and the least optimal route with a total cost of Rp. 131,000.00.

Keywords: travelling salesman person, brute force algorithm, graph

ABSTRAK

Yogyakarta adalah salah satu destinasi wisata populer di Indonesia. Terdapat banyak lokasi wisata yang bisa dikunjungi di Kota Yogyakarta, diantaranya adalah Candi Prambanan, Tebing Breksi, Museum Van den Berg, dan Malioboro. Bagi wisatawan yang belum terbiasa dengan lokasi wisata tersebut, penentuan rute yang akan dikunjungi menjadi penting karena akan berpengaruh pada biaya wisata, khususnya biaya transportasi. Untuk memecahkan masalah ini digunakanlah metode Travelling Salesman Person dan Algoritma Brute Force. Algoritma Brute Force dipilih karena karakteristiknya yang sederhana sehingga mudah dipahami oleh semua kalangan. Hasil penelitian menghasilkan rute paling optimal dengan total biaya Rp 100.300.00 dan rute paling tidak optimal dengan total biaya Rp 131.000.00.

Kata Kunci: travelling salesman person, algoritma brute force, graph

1. PENDAHULUAN

Yogyakarta merupakan salah satu destinasi wisata populer yang ada di Indonesia dengan berbagai tempat wisata menarik seperti Malioboro, Candi Prambanan, Tebing Breksi, Museum Van Den Berg, dan lain-lain. Dengan banyaknya tempat wisata yang ada dan waktu kunjungan yang terbatas membuat wisatawan harus mencari cara agar dapat mengunjungi semua tempat wisata dengan lebih efektif dan murah.

Traveling Salesman Problem (TSP) adalah salah satu masalah yang paling terkenal dalam teori grafik dan kombinatorik. Masalah yang berkaitan dengan TSP pertama kali diperkenalkan pada 1800-an oleh ahli matematika Irlandia Sir William Rowan Hamilton dan oleh ahli matematika Inggris Thomas Penyngton Kirkman. Traveling Salesman Problem (TSP) memiliki tujuan untuk menemukan urutan rute atau jalur terpendek untuk dilalui dengan cara melewati setiap lokasi yang dituju dengan tepat sekali dan selanjutnya kembali lagi ke posisi awal [1][2]. Permasalahan TSP ini sesuai dengan permasalahan yang dimiliki, dimana wisatawan mengharapkan adanya rute yang efektif dan murah.

Untuk menyelesaikan permasalahan ini digunakan algoritma Brute Force. Algoritma bruteforce merupakan salah satu metode yang dapat digunakan untuk mencari solusi terbaik dari suatu masalah dengan mencoba semua kemungkinan yang ada. Dengan menggunakan algoritma bruteforce, diharapkan dapat ditemukan rute terbaik dan dapat mengurangi biaya yang dibutuhkan untuk mengunjungi lokasi wisata di Yogyakarta.

Pada penelitian ini lokasi wisata yang diteliti adalah Malioboro, Candi Prambanan, Tebing Breksi, dan Museum Van den Berg. Biaya yang diperhitungkan hanyalah biaya perjalanan/transportasi dimana setiap 10 Km dikonversi dengan 1 liter bensin. Adapun algoritma yang digunakan adalah algoritma Brute Force.

3 TINJAUAN PUSTAKA

2.1. Travelling Salesman Person

Traveling salesman problem (TSP) adalah salah satu masalah yang paling terkenal dalam teori grafik dan kombinatorik. Masalah yang berkaitan dengan TSP pertama kali diperkenalkan pada 1800-an oleh ahli matematika Irlandia Sir William Rowan Hamilton dan oleh ahli matematika Inggris Thomas Penyngton Kirkman. Traveling Salesman Problem (TSP) memiliki tujuan untuk menemukan urutan rute atau jalur terpendek untuk dilalui dengan cara melewati setiap lokasi yang dituju dengan tepat sekali dan selanjutnya kembali lagi ke posisi awal [1].

Secara sederhana, karakteristik dari permasalahan TSP adalah sebagai berikut :

- Perjalanan diawali dan diakhiri dari dan ke node/titik awal
- Setiap titik/node harus dikunjungi tepat satu kali
- Perjalanan tidak boleh kembali ke titik/node awal sebelum semua titik/node terlewati

Tujuan dari permasalahan ini adalah meminimumkan total jarak yang ditempuh salesman dengan mengatur urutan-urutan kota yang harus dikunjungi [2]

2.2. Graph

Graf G didefinisikan sebagai pasangan himpunan (V,E) , ditulis dengan notasi $G = (V,E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (vertices atau node) dan E adalah himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul [3].

Simpul pada graf dapat dinomori dengan huruf, seperti a,b,c,d, ... atau dengan bilangan asli 1,2,3 atau juga gabungan dengan keduanya. Sedangkan untuk sisi yang menghubungkan antara simpul u dan v dinyatakan dengan (u,v) atau dapat dinyatakan dengan lambang e_1,e_2,e_3, \dots dengan 1,2,3 adalah indeks. Dapat dikatakan bahwa jika e merupakan sisi yang menghubungkan simpul u dengan v , maka e dapat ditulis sebagai $e = (u,v)$.

Dalam aplikasinya, setiap simpul pada graf dapat dijadikan sebagai objek kehidupan, yaitu sebagai objek titik jaringan pesan atau komunikasi, lokasi penempatan kerja, titik kota, jalur transportasi dan lain sebagainya. Sedangkan untuk sisi graf dijadikan sebagai bobot jarak, waktu, biaya dan kendala lainnya. Dan juga busur (arcs) adalah yang menunjukkan hubungan atau relasi dari sepasang simpul.

2.3. Algoritma Brute Force

Algoritma *Brute Force* adalah sebuah pendekatan yang langsung (straightforward) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (problem statement) dan definisi konsep yang dilibatkan. Algoritma *Brute Force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas [4][5].

Karakteristik pada Algoritma *Brute Force* adalah sebagai berikut [9]:

1. Jumlah langkah yang di perlukan besar.
2. Dipakai sebagai dasar dalam menemukan suatu solusi yang lebih efisien atau kreatif.
3. Hampir semua masalah dapat diselesaikan dengan metode ini.
4. Digunakan sebagai dasar dalam perbandingan kualitas suatu algoritma.

Algoritma *Brute Force* banyak diaplikasikan pada aplikasi pencarian data berbasis *string*, seperti pada juml Implementasi Algoritma *Brute Force* Sebagai Mesin Pencari (Search Engine) Berbasis Web Pada Database [4], Implementasi Algoritma *Brute Force* Dalam Pencarian Menu Pada Aplikasi Pemesanan Coffee [6], dan Penerapan Algoritma *Brute Force* Pada Aplikasi Sidayko Berbasis Android [7].

Akan tetapi ada pula yang menggunakan algoritma *Brute Force* untuk memecahkan permasalahan jalur terpendek seperti pada jurnal Penerapan Algoritma A-star dan *Brute Force* pada Aplikasi Jakvel (Jakarta Travel) Berbasis Android [8].

3. METODOLOGI PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini bisa dilihat pada bagan berikut.



Gambar 1. Bagan Metodologi Penelitian

Adapun penjelasan lebih detail mengenai langkah-langkah tersebut adalah sebagai berikut.

3.1 Pengumpulan Data

Data yang akan digunakan dalam penelitian ini adalah data jarak antar lokasi wisata, jadwal buka tempat wisata, dan harga tiket masuk. Data jarak antar lokasi wisata diambil dari aplikasi Google Maps. Sedangkan data jadwal buka tempat wisata dan harga tiket didapatkan langsung dari informasi di lokasi tempat wisata.

3.2 Membuat Tabel Konversi Jarak

Data jarak yang sudah didapatkan di langkah pertama kemudian dikonversi menjadi biaya dalam rupiah. Adapun rumus perhitungannya adalah menggunakan perbandingan 1:10, dimana setiap 1 liter bahan bakar diasumsikan untuk menempuh 10 Km. Bahan bakar yang digunakan adalah jenis Pertalite dengan harga sesuai rilis Pertamina yaitu Rp 10.000,00 per liter.

3.3 Membuat Graph

Graph dibuat untuk memberikan gambaran rute yang bisa diambil, Sekaligus untuk mempermudah dalam menganalisa data.

3.4 Implementasi Algoritma *Brute Force*

Melalui graph yang sudah terbentuk, selanjutnya dilakukan implementasi algoritma *Brute Force*. Hasil perhitungan algoritma *Brute Force* adalah berupa data total biaya yang dibutuhkan. Total biaya meliputi biaya transportasi dan biaya tiket masuk.

3.5 Analisa hasil

Hasil implementasi di langkah sebelumnya dianalisa untuk ditentukan rute mana yang paling efektif dan efisien untuk diambil.

4. HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas hasil dari tiap langkah penelitian sesuai dengan langkah-langkah yang dibuat pada bagian metodologi.

4.1 Pengumpulan Data

Data yang digunakan pada penelitian ini meliputi data jarak antar lokasi wisata, informasi buka/tutup tempat wisata, dan harga tiket masuk lokasi wisata. Tiap tempat wisata dibuat simbol/kode seperti yang bisa dilihat pada Tabel 1. Data jarak didapatkan dari aplikasi Google Maps yang hasilnya bisa dilihat pada Tabel 2. Data informasi buka/tutup dan harga tiket masuk lokasi wisata masing-masing bisa dilihat pada Tabel 3 dan Tabel 4.

Tabel 1. Simbol/Kode Lokasi Wisata

Node	Lokasi Wisata
A	Stasiun Yogyakarta
B	Candi Prambanan
C	Tebing Breksi
D	Museum Van den Berg
E	Malioboro

Pada tabel 1 di atas setiap lokasi wisata diberi simbol *node* untuk mempermudah dalam pembuatan graph dan perhitungan.

Tabel 2. Jarak Antar Lokasi Wisata

Awal	Tujuan	Jarak (KM)
A	B	17,8
A	C	20,0
A	D	3,0
B	C	6,0
B	E	17,5
C	B	6,0
C	E	17,5
D	B	18,0
D	C	17,5
D	E	2,0
E	A	1,0

Data jarak pada tabel 2 di atas didapat dari aplikasi Google Maps. Misal, jarak antara node A (Stasiun Yogyakarta) dan node B (Candi Prambanan) adalah 17.8 Km.

Tabel 3. Informasi Buka/Tutup Lokasi Wisata

Node	Lokasi Wisata	Buka/Tutup
A	Stasiun Yogyakarta	Setiap Saat
B	Candi Prambanan	P/M
C	Tebing Breksi	P/M
D	Museum Van den Berg	P
E	Malioboro	Setiap Saat

P = Kunjungan Pagi (06.00 - 15.00)

M = Kunjungan Malam (15.00 - 24.00)

Setiap lokasi wisata memiliki jadwal buka/tutup masing-masing. Data pada tabel 3 di atas menunjukkan jam buka/tutup di setiap lokasi wisata. Lokasi wisata dengan simbol P berarti lokasi wisata pada pagi/siang hari pada kisaran jam 06.00-15.00. Sedangkan lokasi wisata dengan simbol M berarti buka pada sesi sore/malam pada kisaran jam 15.00-24.00. Stasiun Yogyakarta dan Maliboro menjadi lokasi yang buka setiap saat karena memang merupakan fasilitas publik. Selanjutnya pada tabel 4 adalah informasi mengenai harga tiket masuk di setiap lokasi wisata. Candi Prambanan merupakan lokasi wisata termahal dan Maliboro menjadi lokasi wisata termurah.

Tabel 4. Informasi Harga Tiket Lokasi Wisata

Node	Lokasi Wisata	Harga Tiket Masuk
A	Stasiun Yogyakarta	0
B	Candi Prambanan	48.000

C	Tebing Breksi	5.000
D	Museum Van den Berg	3.000
E	Malioboro	0

4.2 Membuat Tabel Konversi Jarak

Setelah didapatkan data jarak antar lokasi, pada langkah ini akan dibuat tabel konversi jarak ke dalam biaya/nilai rupiah. Adapun rumus konversi biaya adalah sebagai berikut :

$$\text{Biaya} = \text{Jarak} \times \frac{1}{10} \times \text{Rp } 10.000$$

Dari perhitungan tersebut didapatkanlah biaya transportasi untuk setiap node seperti yang ditunjukkan pada tabel 5 berikut.

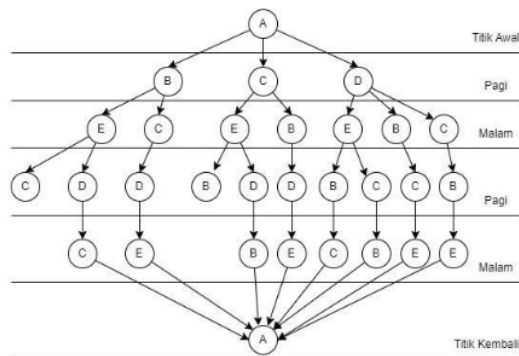
Tabel 5 Tabel Konversi Jarak ke Biaya

Awal	Tujuan	Jarak (KM)	Kebutuhan BBM(liter)	Biaya (Rp)
A	B	17,8	1,78	17.800
A	C	20,0	2,00	20.000
A	D	3,0	0,30	3.000
B	C	6,0	0,60	6.000
B	E	17,5	1,75	17.500
C	B	6,0	0,60	6.000
C	E	17,5	1,75	17.500
D	B	18,0	1,80	18.000
D	C	17,5	1,75	17.500
D	E	2,0	0,20	2.000
E	A	1,0	0,10	1.000

Tabel 5 menunjukkan jarak antar lokasi dan biaya yang diperlukan untuk dari satu titik ke titik lainnya. Sebagai contoh biaya transportasi dari titik A (Stasiun Yogyakarta) ke B (Candi Prambanan) adalah Rp 17.800. Biaya ini didapat dengan asumsi BBM yang digunakan adalah Pertalite dengan harga Rp 10.000,00.

4.3 Membuat Graph

Langkah selanjutnya adalah membuat graph. Graph dibuat untuk mempermudah analisa data khususnya dalam menentukan rute yang akan diambil.



Gambar 2. Graph Antar Lokasi Wisata

Graph di atas dibuat dengan memperhitungkan jam buka/tutup lokasi wisata. Sehingga lokasi wisata yang hanya buka di jam kunjungan pagi, hanya bisa dikunjungi saat pagi/siang hari saja. Penyusunan graph

juga diasumsikan lama kunjungan di tiap lokasi +/- 3 jam. Pada graph terdapat satu rute yang tidak memiliki akhir yaitu A-B-E-C, hal ini dikarenakan ketika ingin ke rute D, sudah masuk ke waktu sore sedangkan titik D (Museum Van den Berg) tidak buka di sesi malam.

4.4 Implementasi Algoritma Brute Force

Graph yang sudah dibuat digunakan untuk mempermudah dalam membuat rute. Rute dibuat dengan menarik garis dari root (node awal) sampai ke node akhir dari graph. Berikut adalah kemungkinan rute yang didapat dari graph.

1. A->B->E->D->C->A
2. A->B->C->D->E->A
3. A->C->E->D->B->A
4. A->C->B->D->E->A
5. A->D->E->B->C->A
6. A->D->E->C->B->A
7. A->D->B->C->E->A
8. A->D->C->B->E->A

Langkah selanjutnya adalah menghitung total biaya yang dibutuhkan. Total biaya merupakan penjumlahan dari biaya transportasi dan harga tiket. Total biaya untuk tiap rute bisa dilihat pada tabel 6 berikut ini.

Tabel 6 Total Biaya Tiap Rute

Rute	Biaya Transportasi	Total Harga Tiket	Total Biaya
A->B->E->D->C->A	A-B = 17.800 B-E = 17.500 E-D = 2.000 D-C = 17.500 C-A = 20.000 Total = 74.800	56.000	130.800
A->B->C->D->E->A	A-B = 17.800 B-C = 6.000 C-D = 17.500 D-E = 2.000 E-A = 1.000 Total = 44.300	56.000	100.300
A->C->E->D->B->A	A-C = 20.000 C-E = 17.500 E-D = 2.000 D-B = 18.000 B-A = 17.800 Total = 75.000	56.000	131.000
A->C->B->D->E->A	A-C = 20.000 C-B = 6.000 B-D = 18.000 D-E = 2.000 E-A = 1.000 Total = 47.000	56.000	103.000
A->D->E->B->C->A	A-D = 3.000 D-E = 2.000 E-B = 17.500 B-C = 6.000 C-A = 20.000 Total = 48.500	56.000	104.500
A->D->E->C->B->A	A-D = 3.000 D-E = 2.000 E-C = 17.500	56.000	102.300

	C-B = 6.000 B-A = 17.800 Total = 46.300		
A->D->B->C->E->A	A-D = 3.000 D-B = 18.000 B-C = 6.000 C-E = 17.500 E-A = 1.000 Total = 45.500	56.000	101.500
4 A->D->C->B->E->A	A-D = 3.000 D-C = 17.500 C-B = 6.000 B-E = 17.500 E-A = 1.000 Total = 45.000	56.000	101.000

4.5 Analisa Hasil

Hasil perhitungan biaya yang ditunjukkan pada tabel 6 menunjukkan bahwa rute dengan biaya paling murah adalah rute A->B->C->D->E->A (Stasiun Yogyakarta – Candi Prambanan – Tebing Breksi – Museum Van den Berg – Malioboro – Stasiun Yogyakarta) dengan total biaya Rp 100.300,00. Sedangkan rute dengan biaya paling mahal adalah rute A->C->E->D->B->A (Stasiun Yogyakarta – Tebing Breksi – Malioboro – Museum Van den Berg – Candi Prambanan - Stasiun Yogyakarta) dengan total biaya Rp 131.000,00.

4. KESIMPULAN

Menentukan rute perjalanan saat berwisata penting dilakukan, khususnya karena berkaitan dengan biaya yang dikeluarkan. Dalam penelitian ini, algoritma Brute Force terbukti mampu memberikan pilihan solusi rute perjalanan dengan biaya perjalanan paling optimal atau murah untuk kegiatan berwisata di beberapa lokasi wisata di Kota Yogyakarta. Rute Stasiun Yogyakarta – Candi Prambanan – Tebing Breksi – Museum Van den Berg – Malioboro – Stasiun Yogyakarta merupakan rute paling optimal dengan total biaya yang diperlukan yaitu Rp 100.300,00. Sedangkan rute Stasiun Yogyakarta – Tebing Breksi – Malioboro – Museum Van den Berg – Candi Prambanan - Stasiun Yogyakarta menjadi rute dengan total biaya termahal dengan total biaya Rp 131.000,00.

DAFTAR PUSTAKA

- [1]. Verenca Laila Okta, Muhammad Afif, Faisal Rautomo, "Implementasi Pendekatan Brute Force pada Penyusunan Jadwal Perjalanan Wisata Otomatis dengan Kombinasi Knapsack-Travelling Salesman Person", Seminar Informatika Aplikatif Polinema (SIAP), 2020.
- [2]. Augridita Prawidya, Bambang Pramono, L.M Bahtiar Aksar, "Travelling Salesman Problem untuk Menentukan Rute Terpendek Bagi Kurir Kota Kendari Menggunakan Algoritma Greedy Berbasis Android", Jurnal Jurusan Teknik Informatika, Fakultas Teknik Universitas Halu Oleo, Kendari, 3(1), 95-106, 2020.
- [3]. Munir, Rinaldi, "Algoritma Brute Force". Institut Teknologi Bandung, 2014.
- [4]. Widia Santoso, Bayu, Sundawa, Firdiansyah, Azhari, Muhammad, "Implementasi Algoritma Brute Force Sebagai Mesin Pencari (Search Engine) Berbasis Web Pada Database", Jurnal Sisfotek Global ISSN : 2088 – 1762 Vol. 6 No. 1, 2016.
- [5]. Candra Irawan, Mudafiq Riyan Pratama, "Perbandingan Algoritma Boyer-Moore dan Brute Force pada Pencarian Kamus Besar Bahasa Indonesia Berbasis Android". Jurnal Teknologi Informasi dan Rekayasa Komputer, 2(1), 54-60, 2020.
- [6]. Sinaga, Aditya, Nuraisana, "Implementasi Algoritma Brute Force Dalam Pencarian Menu Pada Aplikasi Pemesanan Coffee (Studi Kasus : Tanamera Coffee)", JIKOMSI [Jurnal Ilmu Komputer dan Sistem Informasi] E-ISSN : 2723-6129 Vol.4 No.1, pp 6-15, 2021.
- [7]. T.R. Alfin, H.S. Indyah, K. Sabitul, "Penerapan Algoritma Brute Force Pada Aplikasi Sidayko Berbasis Android", Jurnal MNEMONIC, Vol 5, No 1, 2022.
- [8]. A.Septi, Andrianingsih, F. Naif, "Penerapan Algoritma A-star dan Brute Force pada Aplikasi Jakvel (Jakarta Travel) Berbasis Android", Jurnal Media Informatika Budidarma, Vol 5, No 3, 2021.
- [9]. Wilson, Yuliant, Izzatul, "Analisis Penyelesaian Traveling Salesman Problem Dengan Metode Brute Force Menggunakan Graphic Processing Unit", e-Proceeding of Engineering, Vol 2, hal 1875-1876, 2015.

OPTIMALISASI RUTE WISATA DI YOGYAKARTA MENGGUNAKAN METODE TRAVELLING SALESMAN PERSON DAN ALGORITMA BRUTE FORCE

ORIGINALITY REPORT

15%

SIMILARITY INDEX

16%

INTERNET SOURCES

8%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

1	text-id.123dok.com Internet Source	7%
2	Andre Pratama Adiwijaya. "IMPLEMENTASI MULTI DATABASE SEKMA PADA INFRASTRUKTUR BERBASIS MICROSERVICE", Jurnal Teknik dan Science, 2022 Publication	3%
3	adoc.pub Internet Source	2%
4	journals.upi-yai.ac.id Internet Source	2%
5	jurnal.unprimdn.ac.id Internet Source	2%

Exclude quotes Off

Exclude bibliography On

Exclude matches < 2%