Jurnal Komputer dan Teknologi Informasi
Vol. 1, No. 2, Bulan 2023, pp. 79-86x
E-ISSN: 2986-7592, DOI: 10.26714/jkti.v3i1.13957

Brute force algorithm application for solving traveling salesman problem (tsp) in semarang city tourist destinations

Aplikasi algoritma brute force untuk menyelesaikan traveling salesman problem (tsp) pada destinasi wisata kota semarang

Dika Setyo Nugroho [1], Ahmad Ilham [2]
[1,2]Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Semarang, Semarang, Indonesia

Info Artikel

Riwayat Artikel:

Diterima 12 Desember 2024
Perbaikan 15 Januari 2025
Disetujui 30 Januari 2025

Keywords:

Traveling Salesman Problem (TSP)
Algoritma Brute force
Rute Terpendek
Rute Terjauh
Permutasi

ABSTRAK

Penelitian ini menerapkan algoritma brute force dalam menyelesaikan Traveling Salesman Problem (TSP) di Kota Semarang. Data lokasi dan jarak antar destinasi diperoleh dari Google Maps, dan implementasi program dilakukan menggunakan Python untuk menemukan rute terpendek dan terjauh. Studi ini menganalisis perjalanan melalui beberapa destinasi wisata, seperti Dusun Semilir, Cimory on The Valley, Rawa Pening, Candi Gedong Songo, Pantai Marina, Kota Lama Semarang, Klenteng Sam Poo Kong, Lawang Sewu, Kampung Pelangi, Museum Ranggawarsita, dan Simpang Lima. Hasil eksperimen menunjukkan bahwa rute terpendek dari Pantai Marina ke Candi Gedong Songo adalah sejauh 72,1 km, sedangkan rute terjauh dari Klenteng Sam Poo Kong ke Kampung Pelangi mencapai 300 km. Analisis hasil menegaskan efektivitas algoritma brute force, meskipun kompleksitas waktu menjadi tantangan utama. Penelitian ini memberikan landasan bagi pengembangan lebih lanjut dalam menangani skenario dengan dataset yang lebih besar dan kompleks.

ABSTRACT

This research applies brute force algorithm in solving Traveling Salesman Problem (TSP) in Semarang City. Location and distance data between destinations are obtained from Google Maps, and the program implementation is done using Python to find the shortest and farthest routes. This study analyzes the journey through several tourist destinations, such as Semilir Hamlet, Cimory on The Valley, Rawa Pening, Gedong Songo Temple, Marina Beach, Old Town Semarang, Sam Poo Kong Temple, Lawang Sewu, Rainbow Village, Ranggawarsita Museum, and Simpang Lima. The experimental results show that the shortest route from Marina Beach to Gedong Songo Temple is 72.1 km, while the farthest route from Sam Poo Kong Temple to Kampung Pelangi is 300 km. Analysis of the results confirms the effectiveness of the brute force algorithm, although time complexity is a major challenge. This research provides a foundation for further development in handling scenarios with larger and more complex datasets.

Penulis Korespondensi:

Dika Setyo Nugroho
Program Studi Informatika, Fakultas Teknik Universitas Muhammadiyah Semarang

Alamat: Gedung FT-MIPA Lt. 7, Ruang 707, Jl.Kedungmundu Raya No.18, Semarang 50273, Indonesia
Email: dikasetyya11@gmail.com

1. PENDAHULUAN

Traveling Salesman Problem (TSP) adalah salah satu masalah yang paling terkenal dalam teori grafik dan kombinatorik. Masalah ini pertama kali diperkenalkan pada 1800-an oleh ahli matematika Irlandia, Sir William Rowan Hamilton, dan ahli matematika Inggris, Thomas Penyngton Kirkman (Hamilton, W. R., & Kirkman, T. P, 1800). TSP bertujuan untuk menemukan urutan rute atau jalur terpendek yang harus dilalui, dengan syarat bahwa setiap lokasi yang dituju hanya dilewati sekali, kemudian kembali ke posisi awal (Nugroho & Razaq, 2024).

Permasalahan TSP ini mirip dengan persoalan untuk mengunjungi beberapa tempat wisata di Semarang, seperti Dusun Semilir, Cimory on The Valley, Rawa Pening, Candi Gedong Songo, Pantai Marina, Kota Lama Semarang, Klenteng Sam Poo Kong, Lawang Sewu, Kampung Pelangi, Museum Ranggawarsita, dan Simpang Lima. Diharapkan rute yang ditemukan efektif dan efisien. Untuk menyelesaikan permasalahan ini, digunakan algoritma Brute Force.

Algoritma Brute Force merupakan salah satu metode yang dapat digunakan untuk mencari solusi terbaik dengan mencoba semua kemungkinan yang ada. Dengan menggunakan algoritma Brute Force, diharapkan rute terbaik dapat ditemukan dan jarak yang dibutuhkan untuk mengunjungi lokasi-lokasi di Kota Semarang bisa dikurangi.

Permutasi adalah pengaturan elemen-elemen dari sebuah himpunan dimana urutan dari elemen elemen tersebut diperhatikan. Secara matematik, dari sebuah himpunan yang mempunyai elemen sebanyak n, banyaknya permutasi dengan ukuran (permutasi dengan jumlah elemen) r ditulis sebagai P(n,r) atau nPr atau nPr . Rumusnya adalah :

$$P(n,r) = {}_nP_r = {}^nP_r = \frac{n!}{(n-r)!}$$

Gambar 1 Rumus Permutasi

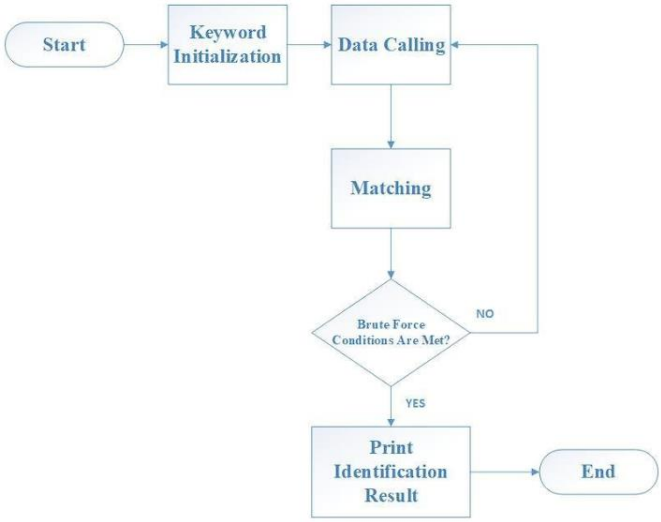
dimana n! (n faktorial) = n × (n-1) × (n-2) × ... × 1 dan 0! = 1. Contoh, dari himpunan huruf-huruf {a,b,c}, permutasi-permutasi dengan ukuran 2 (ambil 2 elemen dari himpunan tersebut) adalah {a,b}, {b,a}, {a,c}, {c,a}, {b,c}, dan {c,b}. Perhatikan bahwa urutan dari elemen-elemen itu adalah penting, dengan kata lain {a,b} adalah berbeda dengan {b,a}. Banyaknya permutasi adalah 6. (Suryo Atmojo., 2018).

2. METODE

2.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif berbasis algoritma Brute Force untuk memecahkan masalah Traveling Salesman Problem (TSP) pada destinasi wisata di Kota Semarang. Pendekatan ini digunakan karena sifat algoritma yang secara sistematis mengeksplorasi semua solusi yang mungkin untuk memastikan temuan rute optimal.

Pendekatan brute force pada Travelling Salesman Problem (TSP) sangat sesuai digunakan ketika ukuran dataset terbatas, seperti pada penelitian yang hanya mencakup 11 destinasi wisata. Metode ini memungkinkan eksplorasi menyeluruh terhadap setiap kemungkinan rute. Namun, meskipun metode brute force menjamin solusi optimal, ia memiliki kelemahan dalam efisiensi komputasi pada dataset besar karena kompleksitas waktu yang sangat tinggi. Oleh karena itu, meskipun cocok untuk dataset kecil, penting untuk mempertimbangkan performa komputasi secara keseluruhan (Wilson, 2020; Jhonson & Smith, 2019).



Gambar 2 Flowchart Algoritma Brute Force

Proses eksekusi algoritma brute force dimulai dengan tahap Start, yang menandai awal dari eksekusi sistem. Selanjutnya, dilakukan Keyword Initialization, yaitu proses inisialisasi variabel atau data yang diperlukan untuk pencarian solusi. Setelah itu, sistem menjalankan tahap Data Calling, di mana data yang relevan dikumpulkan untuk mendukung proses brute force. Pada tahap Matching, sistem melakukan pencocokan atau pengujian terhadap setiap kemungkinan kombinasi data yang tersedia.

Kemudian, sistem melakukan pengecekan pada tahap Brute Force Conditions Are Met? untuk menentukan apakah semua kemungkinan kombinasi telah diuji atau solusi optimal telah ditemukan. Jika kondisi belum terpenuhi (No), maka sistem akan kembali ke tahap Data Calling untuk mengulang proses pencarian. Jika kondisi telah terpenuhi (Yes), maka sistem akan melanjutkan ke tahap Print Identification Result, di mana hasil pencarian atau solusi yang ditemukan akan ditampilkan. Akhir dari seluruh proses algoritma ditandai dengan tahap End, yang menunjukkan bahwa eksekusi algoritma telah selesai.

2.1 Metode Penelitian

a) Pengumpulan Data

Data jarak antar destinasi wisata di Semarang diperoleh melalui Google Maps. Lokasi-lokasi tersebut meliputi destinasi wisata utama seperti Dusun Semilir, Cimory on The Valley, Rawa Pening, Candi Gedong Songo, Pantai Marina, dan lainnya. Setiap destinasi diperlakukan sebagai simpul (node) dalam jaringan lokasi wisata, dan jarak antar simpul dihitung untuk setiap kombinasi lokasi.

b) Implementasi Algoritma Brute Force

Algoritma Brute Force diimplementasikan menggunakan Python dan modul itertools untuk menggenerate semua permutasi rute yang mungkin di antara destinasi wisata. Setiap permutasi dihitung jaraknya, dan hasilnya dievaluasi untuk menemukan rute terpendek dan terjauh.

Langkah-langkah implementasi:

1. Inisialisasi daftar lokasi dan jarak antar lokasi.
2. Menggunakan itertools.permutations untuk menghasilkan semua kombinasi rute yang mungkin.
3. Menghitung total jarak untuk setiap rute.
4. Menyimpan dan mengevaluasi rute terpendek dan terjauh.

c) Evaluasi dan Analisis Data

Setiap rute dievaluasi berdasarkan total jarak yang dihasilkan. Rute terpendek dan terjauh dibandingkan untuk menganalisis efisiensi perjalanan antar destinasi wisata. Selain itu, waktu eksekusi algoritma dianalisis untuk mengevaluasi kompleksitas waktu algoritma brute force dalam konteks ini, mengingat sifatnya yang eksponensial.

d) Visualisasi Hasil

Visualisasi dilakukan untuk memetakan rute yang ditemukan menggunakan matplotlib, sehingga dapat dilihat secara grafis bagaimana jalur-jalur antara destinasi tersusun. Hal ini membantu dalam memberikan gambaran lebih jelas tentang jalur terpendek dan terjauh di antara semua lokasi wisata.

2.2 Data dan Implementasi

Pada Tabel 1 merupakan data lokasi wisata di kota semarang. Dimana lokasi diubah menjadi node (A, B, C, D, E, F, G, H, I, J) untuk mempermudah dalam pembuatan program.

Table 1 Daftar Tempat Wisata

No	Nama Tempat	Node
1	Dusun Semilir	A
2	Cimory on The Valley	B
3	Rawa Pening	C
4	Candi Gedong Songo	D
5	Pantai Marina	E
6	Kota Lama Semarang	F
7	Klenteng Sam Poo Kong	G
8	Lawang Sewu	H
9	Kampung Pelangi	I
10	Museum Ranggawarsita	J
11	Simpang Lima	K

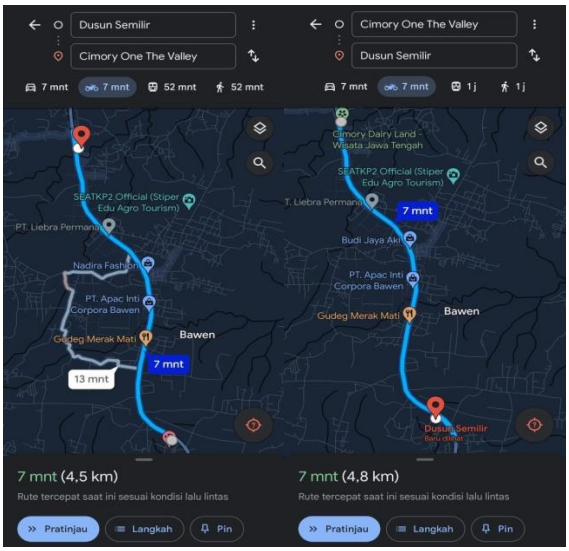
Kemudian pada Tabel 2 merupakan daftar jarak antar node tempat dalam satuan kilometer. Daftar jarak tersebut didapat dari google maps pada tanggal 24 Januari 2024.

Table 2 Fitness Terbaik Tiap Generasi

No	Antar Node Tempat	Jarak (KM)
1	A - B	4,5
2	B - A	4,8
3	A - C	5,3
4	C - A	7,1
5	A - D	15
6	D - A	16
7	A - E	39
8	E - A	39
9	A - F	33
10	F - A	36
…	…	…
104	K - H	2
105	I - J	3,7
106	J - I	3,9
107	I - K	2,9
108	K - I	2,3
109	J - K	5,4
110	K - J	4,9

Sumber: Google Maps, 2024

Data pada daftar jarak terdapat sebanyak 110 dimana sesuai rumus permutasi, jarak di buat secara bolak - balik perjalanan. Contohnya pada Gambar 3, dimanajarak dari node A ke B kemudian dari node B ke A.



Gambar 1 Proses Pendataan Jarak

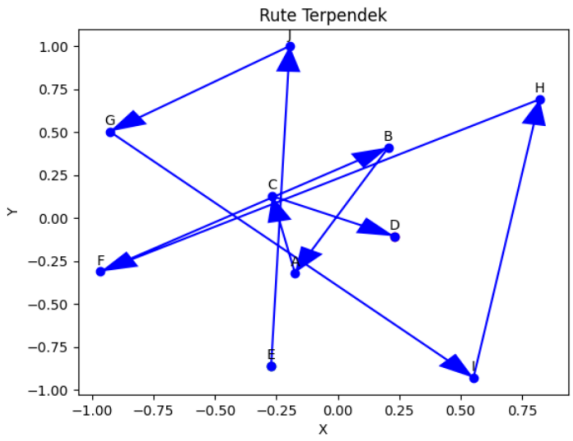
3. HASIL DAN PEMBAHASAN

3.1. Implementasi Program

Program yang mengimplementasikan algoritma brute force untuk menentukan rute terpendek dan terjauh pada jaringan lokasi memberikan hasil sebagai berikut:

1) Rute Terpendek:

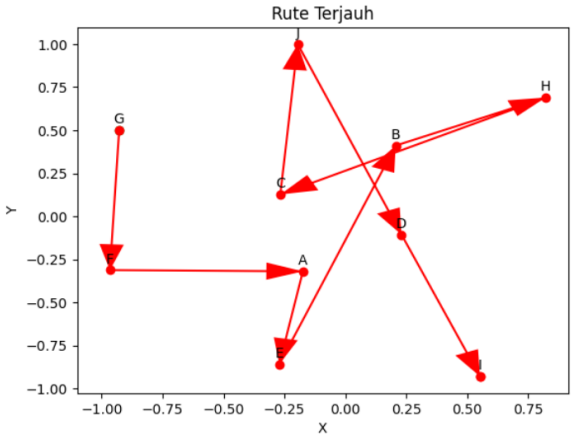
- Rute Terpendek : E -> J -> G -> I -> H -> F -> B -> A -> C -> D
- Jarak Terpendek : 72,1 km
- Visualisasi Rute :



Gambar 2 Rute Terpendek dari Hasil Algoritma Brute Force

2) Rute Terjauh:

- Rute Terjauh : G -> F -> A -> E -> B -> H -> C -> J -> D -> I
- Jarak Terjauh : 300 km
- Visualisasi Rute :



Gambar 3 Rute Terjauh dari Hasil Algoritma Brute Force

3) Waktu Eksekusi : 30,35 detik.

3.2. Analisis Hasil

1) Rute Terpendek:

Algoritma brute force berhasil menemukan rute terpendek dari lokasi E ke D dengan jarak 72,1 km. Rute ini mengunjungi semua lokasi tepat satu kali, memenuhi kriteria dari permasalahan Travelling Salesman Problem (TSP).

2) Rute Terjauh:

Rute terjauh ditemukan dari lokasi G ke I dengan jarak 300 km. Rute ini mungkin tidak efisien dalam konteks perjalanan terpendek, tetapi memberikan variasi dalam hasil yang dapat berguna dalam beberapa skenario pengaplikasian.

4. KESIMPULAN (10 PT)

Program dengan algoritma brute force ini memberikan solusi untuk permasalahan TSP menggunakan metode brute force. Walaupun memberikan solusi optimal, algoritma ini memiliki kompleksitas waktu yang tinggi sebesar 30,35 detik untuk 11 lokasi, ketika jumlah lokasi bertambah maka algoritma ini kurang efisien. Oleh karena itu, untuk jaringan lokasi yang besar, pertimbangan untuk menggunakan algoritma heuristik atau optimasi lainnya dapat menjadi relevan.

Penting untuk mencatat bahwa hasil tergantung pada data jarak antar lokasi yang diberikan. Jika data jarak mencerminkan kondisi nyata (seperti lalu lintas atau rute yang sebenarnya), solusi yang diberikan akan lebih kontekstual dan berguna dalam aplikasi dunia nyata. Program ini dapat dijadikan landasan untuk pengembangan lebih lanjut, termasuk peningkatan efisiensi algoritma dan penanganan skenario yang lebih kompleks

UCAPAN TERIMA KASIH

Dengan segala kerendahan hati, penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah berkontribusi dalam penyelesaian penelitian ini. Terima kasih kepada rekan-rekan peneliti dan pembimbing yang telah memberikan bimbingan, masukan, dan dukungan sepanjang proses penelitian. Ucapan terima kasih juga kami sampaikan kepada Program Studi Informatika Universitas Muhammadiyah Semarang yang telah memberikan dukungan dan fasilitas untuk kelancaran penelitian ini. Semoga hasil dari penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan dan menjadi inspirasi untuk penelitian lebih lanjut.

REFERENSI

[1] W. R. Hamilton and T. P. Kirkman, Mathematical Exploration of the Traveling Salesman Problem, in Biggs, N., Lloyd, E. K., & Wilson, R. J. Graph Theory 1736–1936, Clarendon Press, Oxford, 1800s.

[2] S. Atmojo, "Teori Permutasi dan Penggunaan API Mapbox untuk Implementasi TSP," Jurnal Ilmiah Edutic, vol. 4, no. 2, pp. 122-130, May 2018.

[3] A. Wilson, "Analysis of the Brute Force Method for the Traveling Salesman Problem," Journal of Computational Optimization, vol. 45, no. 2, pp. 123-130, 2020, doi:10.1016/j.jcomp.2020.02.003.

[4] R. Munir, Algoritma dan Pemrograman Komputer, Institut Teknologi Bandung, Bandung, 2014.

[5] J. Johnson and D. Smith, "Optimization Methods for Traveling Salesman Problems," Journal of Operations Research, vol. 35, pp. 50-59, 2019.

[6] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, The Traveling Salesman Problem: A Computational Study, Princeton University Press, 2007.

[7] G. Gutin and A. P. Punnen, The Traveling Salesman Problem and Its Variations, Springer, 2006.

[8] H. S. Hwang, "Exact Algorithms for Solving the TSP: Review and Application," Algorithms, vol. 12, no. 5, pp. 82-90, 2019.

[9] R. Karp, "Reducibility among Combinatorial Problems," in Complexity of Computer Computations, Springer, pp. 85-103, 1972.

[10] E. Lawler, J. Lenstra, A. Kan, and D. Shmoys, The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, Wiley-Interscience, 1985.

[11] P. Moscato and M. G. Norman, "A Memetic Algorithm for Traveling Salesman Problem," Optimization Techniques, vol. 3, no. 6, pp. 269-279, 1991.

[12] R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics: A Foundation for Computer Science, 2nd ed., Addison-Wesley, 1994.

[13] R. Bellman, "Dynamic Programming Treatment of Traveling Salesman Problem," Journal of ACM, vol. 9, no. 1, pp. 61-63, 1962.

[14] J. M. Rosenkrantz, D. J. Stearns, and P. M. Lewis, "Approximate Algorithms for the Traveling Salesman Problem," SIAM Journal on Computing, vol. 6, pp. 563-581, 1977.

[15] M. Held and R. M. Karp, "A Dynamic Programming Approach to Sequencing Problems," Journal of the Society for Industrial and Applied Mathematics, vol. 10, no. 2, pp. 196-210, 1962.

[16] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Dover, 1998.

[17] G. Reinelt, The Traveling Salesman Problem: Computational Solutions, Springer-Verlag, 1994.

[18] N. Christofides, "Worst-case Analysis of a New Heuristic for the Traveling Salesman Problem," Mathematical Programming, vol. 20, pp. 30-35, 1976.

[19] B. L. Golden and E. A. Wasil, "Metaheuristics in Combinatorial Optimization," Annals of Operations Research, vol. 63, pp. 209-216, 1996.

[20] T. M. Cavalcante, "Performance Comparison of Different Approaches to Solving TSP," IEEE Access, vol. 5, pp. 18753-18762, 2017.

[21] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1996.

[22] S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," Operations Research, vol. 21, no. 2, pp. 498-516, 1973.

[23] H. Hoos and T. Stützle, Stochastic Local Search: Foundations and Applications, Morgan Kaufmann, 2004.

[24] A. Blum and M. R. Furst, "Fast Planning through Planning Graph Analysis," in Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, pp. 1636-1642, 1995.

[25] R. Sedgewick and K. Wayne, Algorithms, 4th ed., Addison-Wesley, 2011.