Implementasi Algoritma Greedy dan Djikstra untuk Efektifitas Rute Pariwisata Populer di Borobudur

Candra Agustina
Program Studi Sistem Informasi Akuntansi
Universitas Bina Sarana Informatika
Jakarta, Indonesia
candra.caa@bsi.ac.id

Eka Rahmawati
Program Studi Sistem Informasi
Universitas Bina Sarana Informatika
Jakarta, Indonesia
eka.eat@bsi.ac.id

Abstract*—New tourist attractions in the Borobudur region always bring domestic and foreign tourists to visit. Besides visit the Borobudur Temple, now, the tourists can visit the other popular tourist attractions near the main destination. For example, in 2019, a new tourist attraction was created, it called Mata Langit. Many tourists will include all the interest object on the visit list. The more destinations that will be visited will increase the allocation of time to travel. Tourists must be careful in determining the route. The effective route is very important to manage the time. In the field of computer technology, several algorithms can help to determine the shortest route. Among them are the Greedy algorithm and Djikstra's algorithm. Both algorithms have different principles in processing data. Therefore to get the best results, it is necessary to compare the 2 algorithms. The first time, determined the 5 most popular attractions in Borobudur based on the number of visitors in the last three months. Then the data is processed using Greedy and Djikstra's algorithm. The winner is determined based on the shortest time owned by each route produced. The results obtained show that the Greedy algorithm is more effective in calculating the shortest route to visit popular tours in Borobudur.

Keywords—Shortest route; Greedy Algorithm; Djikstra Algorithm

Abstrak*—Munculnya objek wisata baru di wilayah Borobudur menjadi daya Tarik bagi wisatawan baik domestic maupun wisatawan mancanegara. Sebagai contoh padah tahun 2019 muncul objek wisata baru yaitu mata langit. Tentu saja, banyak dari wisatawan akan memasukkan objek tersebut ke dalam daftar kunjungan. Semakin banyak tujuan yang ingin dikunjungi tentu saja akan menambah alokasi waktu untuk berwisata. Wisatawan harus cermat dalam menentukan rute agar waktu yang dimiliki dapat digunakan secara efektif. Dalam bidang teknologi komputer dikenal beberapa algoritma yang dapat membantu untuk menetukan rute terpendek. Diantaranya algoritma Greedy dan Algotitma Djikstra. Kedua algoritma memiliki prinsip yang berbeda dalam mengolah data. Oleh karena itu untuk mendapatkan hasil terbaik perlu dilakukan perbandingan ke 2 algoritma tersebut. Pertama kali, ditentukan 5 objek wisata terpopuler di Borobudur berdasarkan jumlah pengunjung tiga bulan terakhir. Kemudian dari data tersebut diolah menggunakan algoritma Greedy dan Djikstra. Pemenang ditentukan berdasarkan waktu terpendek yang dimiliki oleh masing-masing rute yang dihasilkan. Hasil yang diperoleh menunjukan algoritma Greedy lebih efektif untuk menghitung rute terpendek untuk mengunjungi wisata populer di Borobudur.

Keywords—Rute terpendek; Algoritma Greedy; Algoritma Djikstra

PENDAHULUAN

Teknologi komputer dapat digunakan dalam berbagai bidang. Kehadirannya dapat mendukung manusia untuk melakukan banyak hal. Salah satunya adalah bidang pariwisata, teknologi komputer juga dapat dimanfaatkan baik oleh pelaku usaha pariwisata ataupun pengguna jasa wisata. Efektifitas waktu untuk menempuh rute suatu perjalanan wisata menjadi hal yang diperhitungkan. Pasalnya, dalam satu hari wisatawan ingin mengunjungi berbagai tempat wisata tanpa terlewat satu pun. Selain itu perhitungan waktu yang tepat juga dapat membantu wisatawan untuk menentukan estimasi waktu untuk berwisata. Namun, tidak seluruh wisatawan dapat meluangkan waktu yang cukup untuk berwisata. Sebagian besar dari mereka biasanya memanfaatkan waktu weekend yang sangat terbatas. Oleh karena itu, efektifitas rute sangat diperlukan agar setiap objek wisata populer dapat dikunjungi.

Penelitian untuk menentukan rute terdekat menggunakan algoritma Greedy telah dilakukan oleh Mahendra, Nuryanto, dan Burhanuddin menunjukan bahwa algoritma greedy dapat membantu mengoptimapkan sistem informasi pengiriman darah palang merah Indonesia di kota Semarang namun penelitian belum membandingkannya dengan algoritma Djikstra dan A*[1]. Penelitian juga dilakukan oleh Oktaviandi dkk yang menggunakan algoritma Greedy untuk menemukan rute terpendek namun penelitian ini juga belum membandingkannya dengan algoritma Djikstra dan A*[2]. Penggunaan Greedy Algorithm juga dilakukan pada penelitian Ammar yang menggunakan konsep greedy by weight dan konsep Greedy by Density untuk menyelesaikan kasus Knapsack Problem[3].

Penggunaan Algoritma Djikstra dilakukan sebagai dasar untuk pembuatan Aplikasi Pengiriman Pesanan Makanan[4]. Hasil yang diperoleh menunjukan efiensi dari penggunaan alogritma Djikstra. Efektifitas algoritma Djikstra juga ditunjukan pada penelitian untuk sistem pendukung keputusan bagi penentuan jalur terpendek pengiriman paket barang pada travel[5]. Selain itu, implementasi dari Algoritma Djikstra juga dilakukan dalam menemukan jarak terdekat dari lokasi pengguna ke tanaman yang di tuju berbasis android studi kasus di Kebun Raya Purwodadi[6]. Implementasi algoritma Djikstra juga dilakukan pada penentuan jarak terdekat pencarian outlet minimal store berbasis Android pada Area Jabodetabek[7].

DOI: https://doi.org/10/25047/jtit.v8i2.216

Selanjutnya, penelitian juga dilakukan oleh Risald, Antonio E. Mirino, dan Sutoyo. Dimana algoritma Djikstra dan Floyd Warshall digunakan untuk menentukan rute terdekat untuk menuju rumah sakit[14]. Algoritma Djikstra juga digunakan dalam pendekatan visi stereo untuk deteksi jalan[15]. Selanjutnya, penggunaan Algoritma Greedy dan Algoritma Djikstra pada penelitan ini akan digunakan dalam menentukan rute terpendek untuk menuju lima lokasi wisata populer di Borobudur.

ALGORITMA GREEDY

Algoritma Greedy adalah salah satu metode banyak digunakan sebagai pemecah masalah terkait dengan optimasi. Kata Greedy berasal dari bahasa inggris yang memiliki arti rakus, tamak atau serakah [8]. Konsep yang digunakan pada algoritma greedy yaitu perhitungan dengan step by step. Metode greedy menjadi algoritma yang sering digunakan untuk menyelesaikan masalah knapsack. Metode tidak selalu memberikan solusi optimal, namun dapat menghasilkan solusi optimal lokal yang mendekati solusi optimal global dengan waktu yang sesuai[5].

Persoalan optimasi dalam penggunaan metode Greedy yang didasarkan pada elemen-elemen biasanya terdapat pada beberapa hal seperti Himpunan kandidat yang berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya. Selanjutnya terdapat pada Himpunan solusi yang terdiri dari kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi adalah himpunan bagian dari himpunan kandidat.

Fungsi seleksi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Fungsi kelayakan (feasible) untuk memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendaara yang ada. Terakhir dadalah Fungsi obyektif untuk memaksimumkan atau meminimumkan nilai solusi. Adakalanya 2 optimum global belum tentu merupakan solusi optimum, namun dapat merupakan solusi sub-optimum atau pseudo- optimum[9].

ALGORITMA DJIKSTRA

Algoritma Dijkstra adalah sebuah algoritma yang dipakai dalam memecahkan permasalahan jarak terpendek (shortest path problem) untuk sebuah graf berarah (directed graph) dengan bobot-bobot sisi (edge weights) yang bernilai tak- negatif[10]. Penggunaan algoritma ini juga biasanya untuk menentukan dasar penetapan rute dalam suatu aplikasi. Dijkstra menjadi salah satu algoritma yang banyak digunakan untuk menentukan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Algoritma Dijkstra memiliki prinsip pencarian dengan dua lintasan yang paling kecil. Algoritma Dijsktra memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek[11].

Perhitungan dengan algoritma Dijkstra memerlukan skema umum untuk mencari jarak terpendek. Hal pertama yang dilakukan adalah dengan menyusun tiga list, yaitu list jarak (list 1), list simpul-simpul sebelumnya (list 2) dan list yang sudah dikunjungi (list 3), serta sebuah variabel untuk menampung simpul pada saat ini (current vertex). Dalam list jarak, diisi dengan nilai tak hingga, kecuali simpul awal yang diisi dengan nilai 0. Pada list 2, diisi dengan false dan list 3, diisi dengan null. Selanjutnya Current vertex diisi dengan simpul awal (start) dan current vertex ditandai sebagai simpul yang telah dikunjungi. Lakukan update list 1 dan 2 berdasarkan simpul- simpul yang dapat langsung dicapai dari current vertex. Kemudian update current vertex dengan simpul yang paling dekat dengan simpul awal[12].

Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari 1 terkecil. Titiktitik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iteras berikutnya[13].

Sebagai contoh:

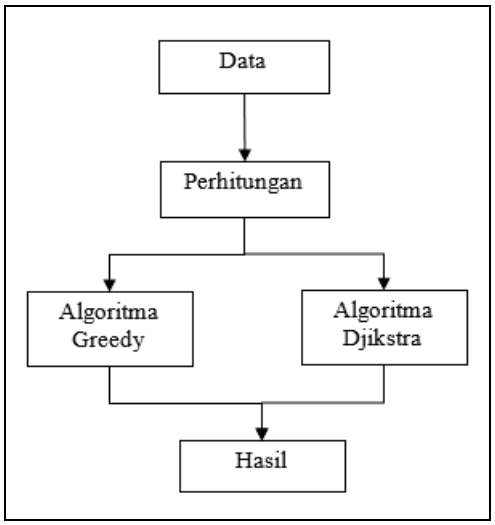
V(E) = {v1,v2,.....vn}
L = Himpunan titik ∈ V(E) yang terpilih dalam path terpendek
D(j) = Jumlah bobot dari path terpendek v1 ke v2
W(i,j) = Bobot dari titik vi ke titik vj
W*(1,j) = Jumlah bobot dari path terkecil dari vi ke titik vj

Maka dalam pencarian path terpendek menggunakan algoritma Djikstra akan dilakukan dengan langkah sebagai berikut:

a. L = { };
   V = {v1,v2,.....vn}.
b. Untuk i = 2,....., n, maka D (i) = W (l,i)
c. Selama vn ∈ L maka:
   - Pilih titik vk ∈ V-L dengan D (k) yang memilki nilai terkcil L = L ∪ {vk}
   - Untuk setiap vj ∈ V-L maka:

   Jika D (j) > D (k) lalu ubah D (j) dengan D (k) + W (k,j)
d. Untuk vj ∈ V, w*(l,j) = D (j)

METODE PENELITIAN



Gambar 1 Metode Penelitian

a. Data

   Data yang digunakan adalah top 5 wisata populer yang terdapat di Borobudur. Kelima tempat wisata tersbut adalah yang paling sering dikunjungi oleh para

DOI: https://doi.org/10/25047/jtit.v8i2.216

wisatawan.

b. Perhitungan
   Tahapan ini merupakan perhitungan untuk menentukan jarak terdekat dengan menggunakan 2 metode yaitu Greedy Algorithm dan Djikstra Algorithm.

c. Hasil

Hasil perhitugan dapat dijadikan pedoman untuk efektifitas waktu dalam menempuh rute wisata populer di Borobudur.

HASIL DAN PEMBAHASAN

Pada tahapan ini pengolahan data dilakukan dengan mengumpulkan tempat wisata populer yang sering dikunjungi oleh wisatawan. Tempat wisata populer yang sering dikunjungi oleh wisatawan diantaranya Candi Borobudur, Bukit Rhema, Punthuk Setumbu, Mata Langit, dan Junkyard. Waktu tempuh yang diperlukan dari beberapa tempat wisata tersebut tertuang pada Tabel 1.

Tabel 1 Data Waktu Tempuh Antar Lokasi Tempat Wisata Populer di Borobudur

	A	**B**	**C**	**D**	**E**
A	**0**	13	12	19	**9**
B	13	0	7	27	17
C	12	**7**	0	25	15
D	19	27	**25**	0	16
E	9	18	17	**16**	0

Keterangan:
A: Candi Borobudur
B: Bukit Rhema
C: Mata Langit
D: Junkyard
E: Punthuk Setumbu

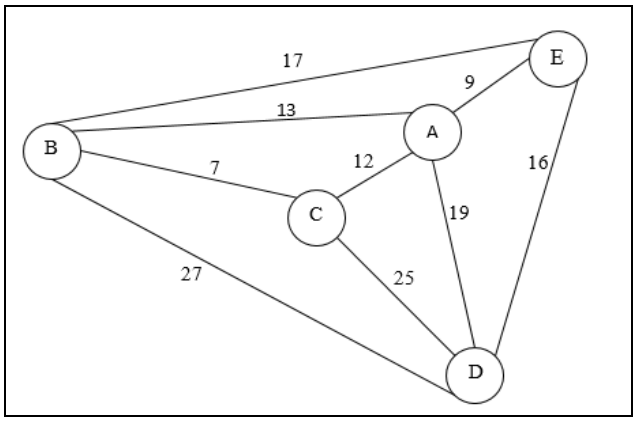
Pertama, perhitungan jalur terpendek dilakukan dengan metode Greedy. Perhitungan menggunakan algoritma Greedy dilakukan dengan menentukan rute dengan waktu tempuh terpendek dari setiap titiknya. Proses perhitungan dengan menggunakan metode Greedy terdapat pada Tabel 2.

Tabel 2. Perhitungan dengan Algoritma Greedy

	A	**B**	**C**	**D**	**E**
A	**0**	13	12	19	9
B	13	**0**	7	27	17
C	12	7	**0**	25	15
D	19	27	25	**0**	16
E	9	18	17	16	**0**

Model perhitungan pertama menjumlah waktu tempuh dan waktu kunjung wisata terlebih dahulu. Titik yang terpilih sebagai rute selanjutnya menjadi titik awal untuk melakukan perhitungan berikutnya. Perhitungan jarak terdekat pertama adalah dari A-E. Dapat dilihat bahwa jarak tempuh A-E merupakan jarak tempuh terpendek yaitu 9 menit. Kemudian, beralih pada garis E. Jarak terdekat selanjutnya adalah E-D dengan waktu tempuh 16 menit. Dari D, terdapat 2 kemungkinan untuk ke titik B atau C. Namun, jarak tempuh tercepat adalah menuju ke titik C dengan waktu 25 menit. Selanjutnya C-B menjadi titik terakhir dengan waktu tempuh 7 menit. Hasil perhitungan dari algoritma Greedy menunjukan waktu tempuh terpendek adalah dari titik A–E–D–C–B dengan total waktu tempuh 57 menit.

Selanjutnya diuji coba dengan menggunakan teknik djikstra. Pengujian dilakukan dengan menggambar graph terlebih dahulu untuk mendapatkan rute terpendek. Adapun graph untuk menuju ke lima lokasi wisata populer di Magelang terdapat pada Gambar 2.



Gambar 2 Graph Tempat Wisata Populer di Magelang

Perhitungan rute terpendek dimulai dari titik A untuk dapat melalui semua objek hingga sampai di titik E. Adapun proses perhitungannya terdapat pada Tabel 3.

Tabel 3. Perhitungan dengan Algoritma Djikstra

V	**A**	**B**	**C**	**D**	**E**
A	**0A**	13A	12A	19A	9A
E		13A	12A	16E	**9A**
D		13A	**12A**	16E	
C		**7C**		16E	
B				**16E**	

Titik awal dimulai dari A, dimana rute terpendek dari titik tersebut adalah menuju ke titik E. Kemudian dari titik E, dicari rute terpendek dimana terpendek berada di titik C. Dari titik C rute terpendek selanjutnya adalah di titik B, lalu titik terakhir adalah D. Jadi hasil dari perhitungan rute terpendek menggunakan Algoritma Djikstra adalah A-E-C-B-D dengan total waktu tempuh 58 menit.

Agar perbedaan antara perhitungan menggunakan metode Greedy dan metode Djikstra dapat terlihat, maka hasil perbadingan terdapat pada tabel 4.

Tabel 4. Hasil Perhitungan dengan Algoritma Greedy dan DJikstra

Algoritma	Rute-1	Rute-2	Rute-3	Rute-4	Rute-5	Waktu
Greedy	A	E	D	C	B	57 Menit
Djikstra	A	E	C	B	D	58 Menit

Algoritma Greedy menunjukan waktu tempuh yang lebih efisien dengan selisih 1 menit jika dibandingkan dengan Algoritma Djikstra. Perbedaan rute terlihat dari rute ke-3 hingga rute ke-5.

KESIMPULAN

Selisih waktu tempuh yang hanya 1 menit membuat Algoritma Greedy dan Djikstra memiliki tingkat akurasi yang hampir setara. Untuk pengambilan keputusan lokasi

mana terlebih dahulu yang akan dituju dapat ditentukan oleh wisatawan. Adapun hasil dari penelitian dapat digunakan sebagai pertimbangan untuk menentukan rute.

Dari penelitian yang dilakukan, Algoritma Greedy lebih efisien untuk digunakan dalam menentukan rute untuk lima tempat wisata populer di Borobudur.

REFERENSI

[1] Y. D. Mahendra, N. Nuryanto, and A. Burhanuddin, "Sistem Penentuan Jarak Terdekat Dalam Pengiriman Darah Di Pmi Kota Semarang Dengan Metode Algoritma Greedy," J. Komtika, vol. 2, no. 2, pp. 136–142, 2019.

[2] R. B. Oktaviandi, M. S. T. Hadi, A. G. Santoso, and N. El Maidah, "Perbandingan Algoritma Genetika dengan Algoritma Greedy Untuk Pencarian Rute Terpendek," INFORMAL Informatics J., vol. 3, no. 1, p. 6, 2019.

[3] M. Ammar, "Implementasi Algoritma Greedy Dalam Menyelesaikan Knapsack Problem Pada Jasa Pengiriman PT. Citra Van Titipan Kilat (TIKI) Kota Makassar," Axiomat. J. Mat. dan Apl., vol. 1, no. September, pp. 26–32, 2019.

[4] L. Affandi, R. Rismanto, and M. M. Firmansyah, "Aplikasi pengiriman pesanan makanan menggunakan algoritma djikstra," Tugas AKhir, pp. 1–10, 2017.

[5] P. S. Juwita, J. Halomoan, F. T. Elektro, and U. Telkom, "Menggunakan Algoritma Greedy Untuk Otomatisasi Rumah Design and Implementation of Power Management Using Greedy," vol. 4, no. 2, pp. 1512–1519, 2017.

[6] M. S. Yusuf, H. M. Az-zahra, and D. H. Apriyanti, "Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Lokasi Pengguna Ke Tanaman Yang Di Tuju Berbasis Android (Studi Kasus di Kebun Raya Purwodadi)," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 1, no. August, pp. 1779–1781, 2017.

[7] R. T. Shita and F. Tarigan, "Implementasi Algoritma Djikstra untuk menentukan Jarak Terdekat Pencarian outlet Minimal Store berbasis Android pada Area Jabodetabek," vol. 6, no. 3, pp. 140–146, 2018.

[8] D. Silvi and P. Dwiza, "Promosi Marketing Menggunakan Algoritma Genetika Dan," Informatika, vol. 3, no. September, pp. 299–313, 2016.

[9] Y. Darnita and R. Toyib, "Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Instansi-Instasi Penting Di Kota Argamakmur Kabupaten Bengkulu Utara," vol. 15, no. 2, 2019.

[10] J. V. Ginting and E. S. Barus, "Aplikasi Penentuan Rute Rumah Sakit Terdekat Menggunakan Algoritma Dijkstra," J. Mantik Penusa, vol. 2, no. 2, pp. 1–8, 2018.

[11] R. Dwi, Saputra and Ardana, "Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang," Skripsi Jur. Ilmu Komputer, Fak. Sains Dan Mat. Univ. Diponegoro, no. Snik, pp. 299–306, 2016.

[12] C. V. Esanata and F. T. Industri, "Penerapan metode djikstra sebagai penentuan rute terpendek distribusi pengiriman kantor jne pusat kabupaten jombang," vol. 3, no. 1, pp. 79–84, 2019.

[13] A. Z. Hasibuan, "Penentuan Alur Terpendek Pengiriman Barang PT.Kencana Link Nusantara Medan Dengan Algoritma Djikstra," J. Ris. Komput., vol. 3, no. 6, pp. 14–19, 2016.

[14] Risald, A. E. Mirino, and Suyoto, "Best routes selection using Dijkstra and Floyd-Warshall algorithm," Proc. 11th Int. Conf. Inf. Commun. Technol. Syst. ICTS 2017, vol. 2018-January, no. October, pp. 155–158, 2018, doi: 10.1109/ICTS.2017.8265662.

[15] Y. Zhang, Y. Su, J. Yang, J. Ponce, and H. Kong, "When dijkstra meets vanishing point: A stereo vision approach for road detection," IEEE Trans. Image Process., vol. 27, no. 5, pp. 2176–2188, 2018, doi: 10.1109/TIP.2018.2792910.

DOI: https://doi.org/10/25047/jtit.v8i2.216