

IMPLEMENTASI PENGGUNAAN ALGORITMA GREEDY BEST FIRST SEARCH UNTUK MENENTUKAN RUTE TERPENDEK DARI CILACAP KE YOGYAKARTA

Raden Dewa Saktia Purnama^{1*}, Faridatun Nisa², Tundo³, Khafid Nurohman⁴, Fakhurrofi⁵,
Lutfi Nugrahaini⁶, Dalail⁷

^{1,2,4,5,6,7}Informatika, Universitas Nahdlatul Ulama Al-Ghazali Cilacap

³Informatika, Universitas 17 Agustus 1945 Jakarta

Riwayat artikel:

Received: 27 Januari 2024

Accepted: 30 Maret 2024

Published: 2 April 2024

Keywords:

Shortest path finding;

Greedy algorithm;

Travel;

City.

Correspondent Email:

radendewasaktiapurnama17@gmail.com

Abstrak. Saat kita berencana untuk melakukan perjalanan, pertimbangan mengenai rute perjalanan menjadi hal yang umum dipertimbangkan. Oleh karena itu, perlu dipikirkan opsi perjalanan yang optimal dari satu tempat ke tujuan, terutama jika menuju destinasi wisata, agar perjalanan dapat diselesaikan dengan efisien. Selain itu, aspek waktu juga harus diperhitungkan agar tidak menghabiskan terlalu banyak waktu dalam perjalanan. Tujuan dari penelitian ini adalah untuk menemukan rute alternatif terpendek dari UNUGHA Cilacap (node 1) ke Titik Nol Kilometer Yogyakarta (node 83), dengan mempertimbangkan faktor jarak dan waktu menggunakan algoritma greedy. Untuk menentukan rute perjalanan dengan jarak terpendek dan waktu yang cepat dapat dilakukan dengan menggunakan solusi pendekatan algoritma greedy. Algoritma Greedy membentuk solusi langkah per langkah dan terdapat beberapa pilihan yang memberikan hasil terbaik dengan membuat pilihan optimum lokal pada setiap langkah sehingga diperoleh solusi optimum global. Metode pengumpulan data menggunakan teknik dasar studi literatur, observasi dan memahami permasalahan objek penelitian selanjutnya dilakukan alur penelitian untuk memudahkan dalam implementasi analisa pemilihan rute. Berdasarkan hasil dan pembahasan pada objek penelitian diperoleh pencarian rute terpendek dari node 1 menuju node 83 dengan jarak 345,8 kilometer merupakan alternatif terbaik dari 3 rute alternatif.

Abstract. When we plan to travel, it is common to consider the route. Therefore, it is necessary to think about the optimal travel options from one place to the destination, especially if it is to a tourist destination, so that the journey can be completed efficiently. In addition, the time aspect must also be taken into account so as not to spend too much time travelling. The purpose of this research is to find the shortest alternative route from UNUGHA Cilacap (node 1) to Yogyakarta Kilometer Zero Point (node 83), by considering the distance and time factors using the greedy algorithm. To determine the travel route with the shortest distance and fast time can be done by using the greedy algorithm approach solution. The Greedy Algorithm forms a step-by-step solution and there are several options that give the best results by making local optimum choices at each step so that a global optimum solution is obtained. The data collection method uses the basic techniques of literature study, observation and understanding the problems of the research object, then the research flow is carried out to facilitate the implementation of route selection analysis. Based on the results and discussion of the research object, it was found that searching for the shortest route from node 1 to node 83 with a distance of 345.8 kilometers was the best alternative out of 3 alternative routes.

1. PENDAHULUAN

Dalam era kemajuan teknologi informasi yang pesat, mencari rute terpendek menjadi

salah satu tantangan utama dalam pengembangan sistem navigasi. *Greedy Best First Search* (GBFS) merupakan algoritma

pencarian yang terbukti efektif dalam menentukan rute terpendek dalam berbagai konteks. Tujuan jurnal ini adalah untuk menyelidiki penerapan GBFS dalam menentukan rute terpendek dari Cilacap ke Yogyakarta, suatu tantangan yang relevan dalam konteks transportasi dan perjalanan.

Cilacap dan Yogyakarta merupakan dua kota penting di Indonesia yang memiliki jarak geografis yang signifikan. Pemilihan rute terpendek antara keduanya menjadi sangat penting, terutama untuk meminimalkan waktu perjalanan dan meningkatkan efisiensi transportasi. Algoritma GBFS, yang menggunakan pendekatan serakah, dapat memberikan solusi yang cepat dan memadai dalam mengatasi permasalahan semacam ini. Penerapan GBFS dalam pencarian rute antarkota tidak hanya memberikan solusi yang efisien, tetapi juga dapat diterapkan secara luas dalam berbagai aplikasi navigasi modern, termasuk aplikasi perjalanan online, sistem manajemen transportasi, dan lainnya. Keunggulan GBFS terletak pada kemampuannya untuk menjelajahi ruang pencarian dengan memberi prioritas pada langkah-langkah yang memiliki nilai heuristik paling rendah, sehingga dapat mencapai solusi dengan cepat.

Dalam jurnal ini, akan diuraikan langkah-langkah implementasi GBFS, dilakukan analisis performa algoritma dalam menentukan rute terpendek, dan dilakukan evaluasi keefektifannya dalam konteks perjalanan dari Cilacap ke Yogyakarta. Penelitian ini juga akan membahas potensi pengembangan lebih lanjut dan penerapan algoritma ini dalam skenario perjalanan dunia nyata. Oleh karena itu, diharapkan hasil penelitian ini dapat memberikan kontribusi yang signifikan dalam pengembangan teknologi navigasi dan transportasi di Indonesia [1].

2. TINJAUAN PUSTAKA

2.1. Rute

Rute adalah jarak atau arah yang harus ditempuh dan dilalui denganurut. Rute terpendek yaitu lintasan atau jalur perjalanan yang memiliki jjarak atau waktu tempuh paling singkat antara dua atau lebih lokasi dan titik tertentu. Dimana tujuannya adalah mencari jalur tercepat atau terpendek antara titik awal dan titik tujuan.[2]

2.2. Algoritma Greedy Best First Search

Algoritma Greedy merupakan suatu algoritma yang memiliki metode pendekatan untuk membangun solusi optimal dengan mencari rute terpendek dengan waktu yang optimum. [3]

Algoritma Greedy termasuk dalam kategori algoritma informed search (pencarian heuristik). Prinsip Greedy adalah mengambil keputusan terbaik pada saat terjadi masalah yang diharapkan keputusan tersebut menjadi solusi terbaik. Oleh karena itu, karena keputusan yang diambil tidak dapat diubah lagi. *Greedy Best First Search* (GBFS) memiliki sebuah fungsi evaluasi $f(n)$. Nilai fungsi evaluasi pada *Greedy Best First Search* bergantung pada nilai fungsi heuristik $h(n)$ itu sendiri. Fungsi heuristik $h(n)$ akan memberikan estimasi arah yang benar, sehingga pencarian jalur terpendek dapat sangat cepat [4]. Secara matematis fungsi evaluasi pada *Greedy Best First Search* dapat ditulis sebagai berikut:

$$f(n) = h'(n) \quad (1)$$

Dimana:

$f(n)$: fungsi evaluasi

$h'(n)$: estimasi biaya untuk sampai pada suatu tujuan mulai dari n .

Algoritma Greedy dapat memilih sebuah rute yang diambil dahulu untuk memberikan suatu alternatif lokal agar menghasilkan alternatif optimal dengan menyeluruh sehingga diperoleh rute terpendek dari Cilacap ke Yogyakarta. Selain itu *Algoritma Greedy* juga membuat beberapa pilihan terbaik yang didapat Ketika itu juga. Beberapa permasalahan optimasi dalam lingkup *Algoritma Greedy* berikut yaitu;

1. Himpunan Kandidat (C): Himpunan ini berisi sebuah elemen penilai solusi. Tiap Langkah kandidat diambil dari himpunannya.
2. Himpunan Solusi (S): Himpunan kandidat yang terpilih menjadi solusi dari permasalahan.
3. Fungsi Seleksi: Merupakan fungsi yang tiap langkahnya memilih kandidat terbaik untuk menuju solusi optimal.
4. Fungsi Kelayakan (Feasible): Dinyatakan fungsi kelayakan apabila telah berhasil

memeriksa kandidat untuk memberi solusi layak dan tidak melanggar kendala.

5. Fungsi Objektif: Fungsi yang berguna untuk memaksimalkan atau meminimalkan nilai solusi.[5]

2.3. Kelebihan dan Kekurangan

Algoritma Greedy Best First Search (GBFS) melakukan proses searching pada semua node berada pada level atau layer yang sama terlebih dahulu sebelum melanjutkan proses searching pada node di level berikutnya. Kelebihan *Algoritma Greedy*:

1. Kesederhanaan: *Algoritma Greedy* seringkali sederhana dan mudah dipahami, sehingga dapat diakses oleh pemogram dari semua tingkat keahlian. Hal ini melibatkan pengambilan pilihan yang optimal secara lokal pada setiap Langkah, dibandingkan mencoba menemukan solusi yang optimal secara global.
2. Efisiensi: *Algoritma Greedy* bisa sangat efisien dalam hal kompleksitas ruang dan waktu. Mereka biasanya melibatkan pengulangan masukan masalah satu kali, dan solusinya sering kali memiliki kompleksitas waktu linier atau logaritmik.
3. Fleksibilitas: *Algoritma Greedy* dapat diterapkan pada berbagai masalah, mulai dari penjadwalan, traversal grafik, hingga kompresi data. Mereka adalah alat serbaguna yang dapat digunakan dalam berbagai konteks.
4. Intuisi: *Algoritma Greedy* sering kali membuat pilihan intuitif yang mudah dipahami dan dijelaskan. Hal ini menjadikannya alat yang berguna untuk mengajarkan keterampilan pemecahan masalah dan memperkenalkan mahasiswa pada ilmu komputer.
5. Pendekatan: *Algoritma Greedy* dapat digunakan untuk menemukan solusi perkiraan terhadap masalah yang sulit atau tidak mungkin diselesaikan secara tepat. Meskipun solusi ini mungkin tidak optimal, namun dalam praktiknya masih berguna.

Kekurangan *Algoritma Greedy*:

1. Optimalitas Lokal: *Algoritma Greedy* membuat pilihan optimal secara lokal disetiap Langkah, tanpa mempertimbangkan konteks yang lebih luas. Artinya, hal tersebut mungkin tidak selalu menghasilkan solusi optimal secara global.
2. Kurangnya Pelacakan Mundur: Sekali *Algoritma Greedy* membuat keputusan, akan sulit atau tidak mungkin untuk mundur dan membatalkan keputusan tersebut. Hal ini dapat menyebabkan solusi yang kurang optimal atau bahkan hasil yang salah.
3. Sensitivitas Terhadap Masukan: *Algoritma Greedy* sering kali sensitif terhadap urutan data masukan. Mengubah masukan dapat menghasilkan keluaran yang sangat berbeda, sehingga sulit untuk menggeneralisasi algoritma ke konteks baru.
4. Penerapan Yang Terbatas: Meskipun *Algoritma Greedy* dapat digunakan untuk menyelesaikan berbagai macam masalah, ada beberapa jenis masalah yang tidak cocok untuk pendekatan ini. Secara khusus, permasalahan yang melibatkan banyak kendala atau tujuan yang saling bertentangan mungkin memerlukan teknik yang lebih canggih.
5. Kesulitan Dalam Analisis: Menganalisis kebenaran dan efisiensi *Algoritma Greedy* dapat menjadi tantangan, memerlukan pemahaman mendalam tentang masalah dan algoritma itu sendiri. Hal ini dapat mempersulit pengoptimalan algoritma atau mengidentifikasi potensi kesalahan.

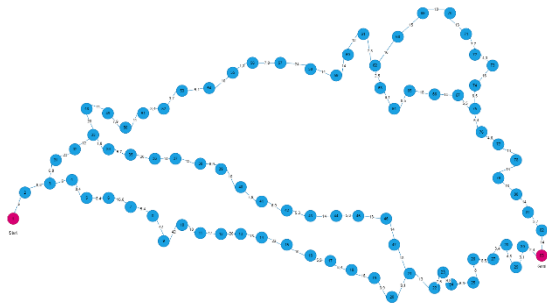
3. METODE PENELITIAN

3.1. Pengumpulan Data

Dari penelitian yang akan dilakukan, peneliti dapat menunjukkan suatu algoritma yang berfungsi untuk menghasilkan rute alternatif terpendek dengan menggunakan *Algoritma Greedy*. Teknik pengumpulan data yang akan dilakukan adalah sebagai berikut:

1. Studi Literatur: Dalam proses ini peneliti sebelumnya harus mempelajari dan memahami tentang *Algoritma Greedy*.
2. Dimana *Algoritma Greedy* dapat menganalisa dan mencari solusi secara cepat dalam menentukan rute alternatif terpendek.
3. Observasi : Observasi ini sutau kegiatan sebagai pengaplikasian suatu data yang sudah diambil dan bertujuan untuk memastikan data yang benar dan akurat.[6]

Berikut Gambaran rute alternatif yang terbentuk dari Cilacap ke Yogyakarta tampak pada Gambar 1.



Gambar 1 Rute Cilacap ke Yogyakarta

3.2. Analisis Data

Untuk menyimpulkan hasil penelitian tersebut maka dilakukan analisis data. Dimana data yang telah dipastikan valid dan akurat, selanjutnya akan dikumpulkan dan dikelompokkan untuk dianalisa menggunakan tahap analisa data dengan *Algoritma Greedy*. Berikut beberapa tahapannya:

1. Pengelompokan data: Dalam tahapan ini data yang telah terkumpul akan dikelompokkan untuk dilakukan tahapan terhadap langkah-langkah perhitungan *Algoritma Greedy* untuk menghasilkan rute alternatif terpendek.
2. Penerapan data: Dalam proses ini data yang siap akan diaplikasikan untuk diperhitungkan dalam *Algoritma Greedy*.

Data hasil perhitungan *Algoritma greedy* ini akan mendapatkan jarak dan waktu tempuh yang optimal dari tempat asal menuju tempat tujuan sehingga akan mempercepat jarak tempuh. Hasil dari penelitian ini bertujuan untuk melakukan observasi masalah yang dihadapi dan untuk menunjukkan atau

menemukan solusi rute alternatif terpendek dari Kota Cilacap menuju Kota Yogyakarta dengan menggunakan metode *Algoritma Greedy*. Sehingga diperoleh hasil penelitian yang dapat memberikan manfaat.

3.3. Perancangan Sistem

Dalam membangun sebuah sistem dengan menggunakan *Algoritma Greedy*, tahapan ini bertujuan untuk membantu kebutuhan pengguna dalam penggunaannya. Dalam penerapan ini peneliti membangun terlebih dahulu Diagram Flowchart (*Alur Algoritma Greedy*) dimana dalam membangun program ini terdiri dari beberapa perintah yaitu input data, hasil pengolahan, dan riwayat pengolahan.

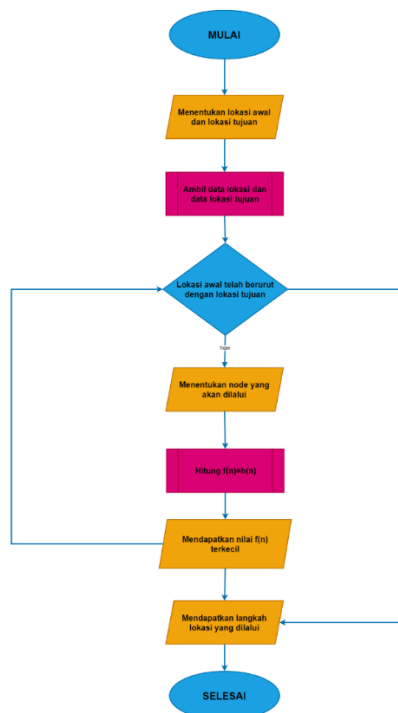
Untuk perancangan data yang didalamnya terdapat tahapan sebuah perintah inputan atau masukan dan output atau keluaran. Berikut beberapa tahapan-tahapannya:

1. Asal lokasi: Asal lokasi ini bertujuan sebagai inputan dimana pengguna akan memilih suatu titik lokasi yang telah ditentukan sistem yang keberadaan terdekat dengan asal lokasi pengguna.
2. Pemilihan rute alternatif terpendek: Pemilihan alternatif rute terpendek ini merupakan perhitungan dalam *Algoritma Greedy*, dimana akan menunjukkan sebuah alternatif rute terpendek yang akan ditempuh oleh pengguna.
3. Lokasi tujuan: Lokasi tujuan bertujuan sebagai output atau keluaran dimana pengguna akan memilih suatu titik lokasi tujuan yang telah ditentukan oleh sistem.[7]

3.3.1 Alur Greedy Best First Search

Untuk proses pencarian rute alternatif terpendek dengan metode *Greedy Best First Search* ditunjukkan gambar dibawah. Proses pencarian rute alternatif terpendek dimulai dari menentukan lokasi awal dan lokasi tujuan, penentuan lokasi tersebut diambil dari bantuan Google Maps. Selanjutnya pengguna menentukan beberapa rute yang ingin ditempuh. Saat telah menentukan beberapa rute yang telah ingin ditempuh, maka diambil data lokasi awal dan lokasi tujuan untuk dilanjutkan ke proses pencarian rute terpendek dengan metode *Greedy Best First Search*.

Proses pencarian dengan metode *Greedy Best First Search* dimulai dari pengecekan lokasi awal berurut atau terhubung dengan lokasi tujuan, jika lokasi awal sudah berurut atau terhubung dengan lokasi tujuan maka pencarian jarak terpendek dengan metode *Greedy Best First Search* telah selesai dan apabila lokasi awal tidak berurut atau tidak terhubung dengan lokasi tujuan maka akan dilakukan proses inialisasi semua node atau titik yang akan dilalui. Setelah menginisialisasi semua titik yang akan dilalui dilanjutkan dengan proses penghitungan nilai $h(n)$, Penghitungan nilai $h(n)$ dilakukan untuk mendapatkan nilai $f(n)$ terkecil. Semua proses perulangan akan dilakukan sampai lokasi awal telah berurut atau terhubung dengan lokasi tujuan yang terlihat pada Gambar 2.[8]



Gambar 2 Diagram Alir Greedy Best First Search

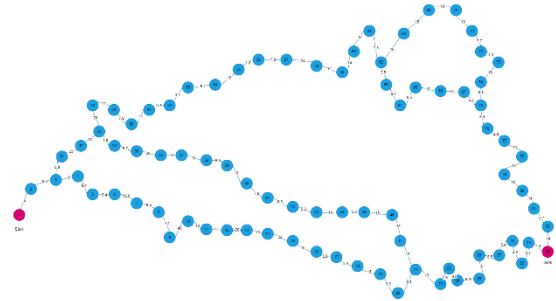
4. HASIL DAN PEMBAHASAN

Implementasi *Algoritma Greedy* dalam penyelesaian masalah dalam melakukan pencarian rute alternatif terpendek dari Cilacap ke Yogyakarta dibuat dalam skenario berikut:

4.1. Kasus : Rute Terpendek dari Cilacap ke Yogyakarta

Penamaan titik node kecamatan dari Cilacap ke Yogyakarta, Kesugihan (UNUGHA) disimbolkan dengan angka 1 sampai ke Titik Nol (Yogyakarta) dengan angka 83, sedangkan

titik node kecamatan lain yang dilalui disimbolkan dengan angka dan lingkaran (node), seperti terlihat pada Gambar 3 dan nama daerah untuk setiap node tampak pada Tabel 1.



Gambar 3 Rute Dari Cilacap ke Yogyakarta

Table 1 Daftar Nama Kecamatan

Node	Nama Kecamatan	$h'(n)$
1	Kesugihan (UNUGHA)	163
2	Maos	157
3	Adipala	154
4	Doplang	160
5	Kroya	151
6	Binangun	140
7	Nusawungu	132
8	Jetis	99
9	Kebumen	97
10	Ayah	125
11	Karangbolong	111
12	Suwuk	108
13	Tambakmulyo	106
14	Petanahan	97
15	Klirong	94
16	Buluspesantren	87
17	Ambal	79
18	Mirit	71
19	Grabag	65
20	Ngombol	55
21	Purwodadi	121
22	Temon	38
23	Wates	32
24	Pengasih	26
25	Sentolo	20
26	Sedayu	13
27	Gamping	4,5
28	Kasih	7,3
29	Sewon	4,3
30	Mergangsan	2,4
31	Jatilawang	165
32	Banyumas	171
33	Rawalo	159

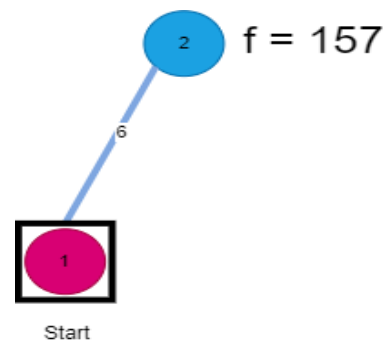
34	Cindaga	158
35	Kebasen	155
36	Sumpiuh	135
37	Gombang	118
38	Sruweng	102
39	Kebumen	97
40	Lundong	87
41	Prembun	82
42	Butuh	75
43	Kutoarjo	70
44	Purworejo	48
45	Banyuurip	63
46	Purwodadi	51
47	Bagelen	49
48	Sokaraja	160
49	Somagede	149
50	Susukan	149
51	Purwareja Klampok	142
52	Klampok	147
53	Kaliwinasuh	143
54	Mandiraja	138
55	Banjarnegara	141
56	Purwonegoro	143
57	Bawang	127
58	Sigaluh	103
59	Wonosobo	93
60	Sawangan	103
61	Selomerto	91
62	Kertek	84
63	Kalikajar	81
64	Sapuran	75
65	Kepil	67
66	Kaliangkrik	58
67	Magelang	45
68	Kledung	83
69	Temanggung	68
70	Parakan	77
71	Madureso	65
72	Bengkal	59
73	Secang	56
74	Mertoyudan	41
75	Mungkid	35
76	Muntilan	29
77	Salam	25
78	Sucen	14
79	Baledono	1
80	Tempel	18
81	Sleman	11
82	Ngaglik	14
83	Titik Nol (Yogyakarta)	0

Untuk melakukan penerapan Algoritma Greedy, dibuat diagram alir dengan data yang telah diambil. Gambar 2 merupakan urutan proses perjalanan berdasarkan *Algoritma Greedy* dengan skenario dari lokasi awal di titik 1 Kesugihan (UNUGHA) menuju ke lokasi tujuan di titik 83 (Titik Nol Yogyakarta). Untuk menentukan apakah rute yang dipilih hasil *Algoritma Greedy* merupakan yang paling optimal pada setiap langkahnya.[9], [10], [11].

4.2. Hasil Pengujian

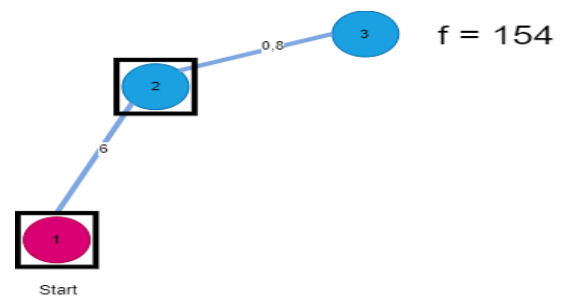
Contoh penyelesaian kasus mencari rute alternatif terpendek dari Cilacap ke Yogyakarta menggunakan *Algoritma Greedy* :[12]

1. Langkah 1



OPEN berisi satu simpul yaitu (1). Karena di OPEN hanya terdapat satu simpul, maka (1) terpilih sebagai best node yang akan dipindahkan ke CLOSED. Kemudian, bangkitkan suksesor dari (1) yaitu (2). Karena suksesor (2) belum pernah ada di OPEN maupun CLOSED, maka (2) dimasukan ke OPEN. Langkah pertama menghasilkan OPEN = (2) dan CLOSED = (1).

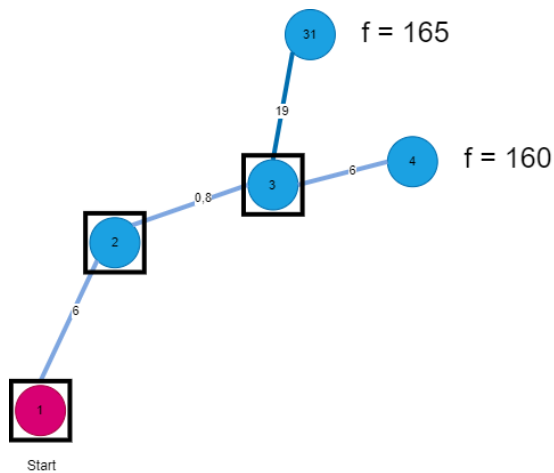
2. Langkah 2



Karena di OPEN hanya terdapat satu simpul, maka (2) terpilih sebagai best node dan dipindahkan ke CLOSED. Kemudian,

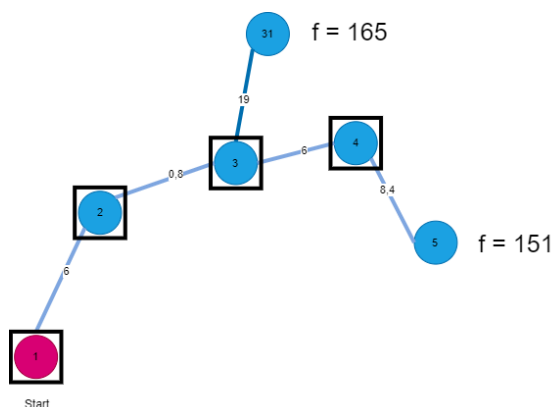
bangkitkan suksesor dari (2) yaitu (3). Karena suksesor (3) belum pernah ada di OPEN maupun CLOSED, maka (3) dimasukkan ke OPEN. Langkah kedua menghasilkan OPEN = (3) dan CLOSED (1, 2).

3. Langkah 3



Karena di OPEN hanya terdapat satu simpul, maka (3) terpilih sebagai best node dan dipindahkan ke CLOSED. Kemudian, bangkitkan suksesor dari (3) yaitu (4, 31). Karena suksesor (4, 31) belum pernah ada di OPEN maupun CLOSED, maka (4, 31) dimasukkan ke OPEN. Langkah ketiga menghasilkan OPEN = (4, 31) dan CLOSED (1, 2, 3).

4. Langkah 4

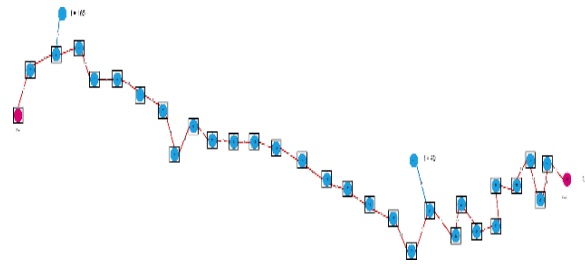


Karena Greedy Best First Search hanya menghitung biaya perkiraan, maka $f(n) = h(n)$, (4) dengan biaya terkecil (yaitu $f(4) = h(4) = 160$) terpilih sebagai best node dan dipindahkan ke CLOSED. Selanjutnya, suksesor (4) dibangkitkan, yaitu (5). Karena

di suksesor (5) belum pernah ada OPEN maupun CLOSED, maka (5) dimasukan ke OPEN. Langkah keempat menghasilkan OPEN (5, 31) dan CLOSED (1, 2, 3, 4).

Begitu seterusnya, sampai semua suksesor berada di dalam CLOSED, dimana semua hasil akhir berhenti pada langkah 31.

5. Langkah 31



Karena di OPEN terdapat tiga simpul yaitu [31, 47, 83], maka evaluasi ketiga simpul tersebut dan [83] terpilih sebagai best node karena biaya perkiraan paling kecil (yaitu $f = 83 = h[83] = 0$). Karena best node tersebut merupakan goal, berarti solusi telah ditemukan.

Rute yang dihasilkan ialah $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 22 \rightarrow 23 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30 \rightarrow 83$ (Kesugihan (UNUGHA Cilacap) \rightarrow Maos \rightarrow Adipala \rightarrow Doplang \rightarrow Kroya \rightarrow Binangun \rightarrow Nusawungu \rightarrow Jetis \rightarrow Kebumen \rightarrow Ayah \rightarrow Karangbolong \rightarrow Suwuk \rightarrow Tambakmulyo \rightarrow Petanahan \rightarrow Klirong \rightarrow Buluspesantren \rightarrow Ambal \rightarrow Mirit \rightarrow Grabag \rightarrow Ngombol \rightarrow Purwodadi \rightarrow Temon \rightarrow Wates \rightarrow Pengasih \rightarrow Sentolo \rightarrow Sedayu \rightarrow Gamping \rightarrow Kasihan \rightarrow Sewon \rightarrow Mergangsan \rightarrow Titik Nol (Yogyakarta)) dengan total jarak 345,8 km.

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

- Berdasarkan hasil pengujian penelitian diatas didapatkan rute terpendek dari UNUGHA Cilacap menuju ke Titik Nol Kilometer dengan menggunakan metode *Algoritma Greedy Best First Search* ialah $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18$

- 19 → 20 → 21 → 22 → 23 → 24 → 25 → 26 → 27 → 28 → 29 → 30 → 83 dengan jarak 345,8 kilometer.
- b. Adapun dalam praktek di kehidupan, Algoritma Greedy dapat digunakan untuk mencari alternatif jarak tempuh terpendek dalam safari destinasi wisata atau tujuan lainnya yang memerlukan analisa waktu optimal untuk mengunjungi beberapa kota seperti halnya dari UNUGHA Cilacap ke Titik Nol Kilometer Yogyakarta.

UCAPAN TERIMA KASIH

Alhamdulillahirobbil'alamin. Segala puji dan syukur penulis panjatkan kehadiran Allah SWT, karena dengan izin-Nya lah penulis dapat menyelesaikan penelitian ini. Shalawat dan salam semoga senantia tercurah kepada junjungan penulis, yakni Nabi besar Muhammad SAW.

DAFTAR PUSTAKA

- [1] Y. Afrianto Dan N. B. Ginting, "Implementasi Metode Greedy Best First Search Untuk Penjadwalan Perkuliahan (Studi Kasus: Fakultas Teknik Dan Sains Universitas Ibn Khaldun Bogor)." [Daring]. Tersedia Pada: [Http://Pkm.Uika-Bogor.Ac.Id/Index.Php/Prosiding/Index](http://Pkm.Uika-Bogor.Ac.Id/Index.Php/Prosiding/Index)
- [2] A. S. Syathirah, P. Daru Kusuma, Dan C. S. Ningsih, "Analisis Routing Pada Aplikasi Pemanggilan Darurat Menggunakan Algoritma Hill Climbing Dan Greedy."
- [3] F. Nova Arviantino, W. Gata, L. Kurniawati, Y. A. Setiawan, Dan D. Priansyah, "Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Masjid-Masjid Di Kota Samarinda," *Metik Jurnal*, Vol. 5, No. 1, Hlm. 8–11, Jun 2021, Doi: 10.47002/Metik.V5i1.188.
- [4] A. Dhiyaanisafa Goenawan, A. Faqih, Dan M. P. Pulungan, "2 Nd Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (Senafti) 21 Maret 2023-Jakarta," 2023.
- [5] N. F. Lakutu, S. L. Mahmud, M. R. Katili, Dan N. I. Yahya, "Algoritma Dijkstra Dan Algoritma Greedy Untuk Optimasi Rute Pengiriman Barang Pada Kantor Pos Gorontalo," *Euler: Jurnal Ilmiah Matematika, Sains Dan Teknologi*, Vol. 11, No. 1, Hlm. 55–65, Jun 2023, Doi: 10.34312/Euler.V11i1.18244.
- [6] S. Saifulloh Dan R. E. Fitriyani, "Analisa Pencarian Rute Tercepat Menuju Telaga Sarangan Menggunakan Algoritma Greedy."
- [7] U. Al Dkk., "Analisa Pencarian Rute Tercepat Menuju Tempat Wisata Pulau Kumala Kota Tenggarong Menggunakan Algoritma Greedy," *Jurnal Ilmiah Ilmu Komputer*, Vol. 7, No. 1, 2021.
- [8] A. Subagio, B. Rahayudi, Dan M. A. Fauzi, "Analisis Performansi Algoritma Greedy Best First Search Dan Dijkstra Pada Aplikasi Pencarian Jalur Pendorong Darah Terdekat," 2019. [Daring]. Tersedia Pada: [Http://J-Ptiik.Ub.Ac.Id](http://J-Ptiik.Ub.Ac.Id)
- [9] Algoritma Greedy
- [10] M. Rute Terbaik Dari Bolok Menuju Lasiana, "Penerapan Metode Pencarian Greedy Best First Search Dalam."
- [11] Y. Darnita Dan R. Toyib, "Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek... Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Instansi-Instansi Penting Di Kota Argamakmur Kabupaten Bengkulu Utara," 2019.
- [12] M. Furqan, Y. R. Nasution, Dan T. S. Nurdianti, "Penerapan Algoritma Greedy Untuk Menentukan Rute Terpendek Antar Klinik Gigi," *Csrid (Computer Science Research And Its Development Journal)*, Vol. 12, No. 3, Hlm. 170, Mar 2021, Doi: 10.22303/Csrid.12.3.2020.170-178.