
SIMULASI RUTE TERPENDEK LOKASI PARIWISATA DI NIAS DENGAN METODE BREADTH FIRST SEARCH DAN TABU SEARCH

¹Delima Zai, ²Haeni Budiati, ³Sunneng Sandino Berutu

Teknik Informatika, Fakultas Sains dan Komputer, Universitas Kristen Immanuel
Jalan Solo Km. 11 PO Box 4 YKAP Yogyakarta, ph: (0274) 496256-296247 fax: (0274) 496258
¹Delimanzai@gmail.com, ²heni@ukrimuniversity.ac.id, ³sandinoberutu@ukrimuniversity.ac.id

Abstrak

Kemampuan metode Breadth First Search melakukan pencarian secara melebar dengan mengunjungi dari suatu simpul atau titik kemudian mengunjungi semua titik-titik yang bertetangga untuk menemukan sebuah solusi sedangkan Tabu Search adalah suatu algoritma yang menuntun setiap tahapannya agar dapat menghasilkan fungsi tujuan yang paling optimum, Algoritma Breadth First Search dapat digabungkan dengan algoritma Tabu Search untuk mendapatkan jalur terpendek yang optimal berdasarkan jarak, dengan cara ketika Breadth First Search menemukan solusi maka tidak langsung berhenti, tetapi terus melakukan pencarian hingga dapat menemukan jalur terpendek dengan menggunakan prinsip dari Tabu Search. Penelitian ini membuat sebuah aplikasi pencarian rute terpendek lokasi pariwisata di Pulau Nias Berbasis web dengan metode Breadth First Search dan Tabu Search. Algoritma Breadth First Search merupakan salah satu algoritma untuk menentukan pencarian jalur terpendek, algoritma ini melakukan pencarian secara melebar dengan mengunjungi dari suatu simpul atau titik kemudian mengunjungi semua titik-titik yang bertetangga untuk menemukan solusi. Setelah melakukan modifikasi terhadap Algoritma Breadth First Search dan Tabu Search maka kedua metode dapat diimplementasikan pada simulasi rute terpendek lokasi pariwisata di pulau Nias untuk mencari jalur terpendek tempat wisata, tempat SPBU, tempat restoran, hotel, dan kantor pemerintahan.

Kata kunci --- algoritma BFS, Tabu Search, jalur terpendek, wisata.

Abstract

The ability of the method of Breadth First Search conducting a broad search by visiting on a node or point then visit all the points neighbors to find a solution, while Tabu Search is an algorithm that guide each stage in order to produce an objective function optimum, Algorithm Breadth First search can be combined with Tabu search algorithm to obtain the optimal shortest path based on the distance, by the way when Breadth First search find a solution then do not immediately stop, but continue to search to find the shortest path by using the principle of Tabu. Research that make a application search short path area tourism in Nias island based web with algorithm Breadth First Search dan Tabu Search. Breadth First Search is algorithm for search short path, this algorithm doing search be widen with visit from one point to another point so visit all point neighbor for find solution. After doing modifikasi on Breadth First Search Algorithm and Tabu Search, the two methods can be implemented in the shortest route simulation tourist locations on the island to find the shortest path tourist spots, where gas stations, where restaurants, hotels, and government offices.

Keyword --- Algorithm Breadth First, Tabu Search algorithms, short path, travelling.

1. PENDAHULUAN

Pemanfaatan teknologi untuk menunjang sektor pariwisata sangat dibutuhkan agar para wisatawan mudah mengakses informasi tempat wisata, khususnya di pulau Nias. Perkembangan teknologi di era globalisasi ini telah memberikan banyak pilihan teknologi dalam membantu manusia untuk pencarian suatu lokasi. Diantaranya menggunakan teknologi sistem informasi geografi. Oleh sebab itu pada penelitian ini dibuat sebuah aplikasi simulasi rute terpendek lokasi pariwisata di pulau Nias berbasis web dengan metode breadth first search & tabu search yang bertujuan untuk membantu para pengguna dalam menentukan jalur terpendek tempat wisata di pulau Nias. Salah satu algoritma yang digunakan untuk pencarian jalur terpendek yaitu algoritma BFS, namun BFS hanyalah solusi tersingkat dari *state* awal dan tidak menunjukkan jalur terpendek berdasarkan jarak dan biaya, sehingga untuk memperoleh jalur terpendek berdasarkan jarak yang lebih optimal maka bfs dimodifikasi dengan algoritma tabu search.

Beberapa penelitian sebelumnya menunjukkan bahwa algoritma *breadth first search* dapat diimplementasikan dalam beberapa aplikasi. Salah satu bentuknya berupa pada aplikasi penerapan search tree pada penyelesaian masalah penentuan jalur kota terpendek [1]. Aplikasi tersebut memanfaatkan konsep-konsep dari *breadth first search*.

Aplikasi pencarian jalur terpendek menggunakan algoritma modifikasi *Breadth First Search* dengan *Tabu Search* dapat diimplementasikan dan digunakan di pulau Nias.

2. MATERI DAN METODE PENELITIAN

2.1 Graph

Graph G didefinisikan sebagai pasangan himpunan (V, E) , dituliskan 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 (Munir, 2005).

Secara umum *graph* dapat didefinisikan sebagai kumpulan titik (*nodes* atau *vertices*) yang dihubungkan dengan garis (*arcs* atau *edges*).

Secara singkat *graph* dapat ditulis dengan notasi $G = (V, E)$ dimana

- V : sekumpulan dari titik (*nodes* atau *vertices*).
 : $\{v_1, v_2, v_3, v_4, v_5, \dots, v_n\}$
- E : sekumpulan garis yang menghubungkan titik ke titik yang lain (*arcs* atau *edge*).
 : $\{e_1, e_2, e_3, e_4, e_5, \dots, e_n\}$

2.1.1 Terminologi graph

Ada beberapa terminologi *graph* adalah sebagai berikut

1. Ketetanggaan (*adjacent*) Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung.
2. Bersisian (*incidency*) Untuk sembarang sisi $e = (v_i, v_j)$ dikatakan bahwa e bersisian dengan simpul (*vertex*) v_i dan v_j .
3. Derajat (*degree*) Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Notasinya adalah $d(v)$
4. Lintasan (*path*) Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam sebuah graf adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_n$.
5. Sirkuit (*Circuit*) Lintasan yang berawal dan berakhir di tempat (simpul) yang sama disebut sirkuit.
6. Graf berbobot (*weighted graph*) Apabila sisi-sisi pada *graph* disertai juga dengan suatu (atau beberapa) harga yang menyatakan secara unik kondisi keterhubungan tersebut maka *graph* tersebut disebut *graph* berbobot.

2.2 Jalur terpendek

Jalur terpendek merupakan suatu pencarian nilai variable yang dianggap dapat menghasilkan nilai yang maksimal. Jalur terpendek memiliki peranan penting dalam penyusunan system. Dengan jalur terpendek dapat diperoleh hal-hal yang memiliki nilai profit tinggi serta meminimalkan jarak. Banyak masalah yang berhubungan dengan pencarian jalur. Umumnya untuk pencarian jalur yang mengoptimumkan kinerja manajemen taksi. Menentukan rute terpendek merupakan pencarian dari beberapa simpul yang berhubungan. Simpul tersebut sering direpresentasikan dalam bentuk graf berbobot. Graf yang digunakan untuk mencari lintasan tersebut diberikan bobot atau nilai. Bobot disini berfungsi untuk menyatakan jarak.

Menurut Mudi Arsa (2010), untuk menemukan lintasan terpendek antara dua atau beberapa simpul lebih yang berhubungan. Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. Persoalan ini biasanya direpresentasikan dalam bentuk graf. Graf yang digunakan dalam pencarian lintasan terpendek atau shortest path adalah graf berbobot (weighted graph), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya.

Ada beberapa macam lintasan terpendek yaitu

1. Lintasan terpendek antara dua titik simpul tertentu (*a pair short path*).
2. Lintasan terpendek antara semua simpul tertentu (*all pair short path*).
3. Lintasan terpendek antara sebuah simpul tertentu ke semua simpul (*singel saource short path*).
4. Lintasan terpendek antara dua simpul yang melalui simpul tertentu (*intermediate short path*).

2.3 Algoritma

Apa yang dimaksud dengan algoritma, algoritma berarti solusi. Ketika orang berbicara mengenai algoritma di bidang matematika, maka yang dimaksud adalah solusi dari suatu masalah yang harus dipecahkan dengan menggunakan perhitungan matematika. Algoritma harus dibuat secara runut agar dapat dimengerti dan diterapkan. Analisis kasus sangat dibutuhkan dalam membuat sebuah algoritma, misalnya proses apa saja yang dibutuhkan untuk menyelesaikan masalah yang akan diselesaikan (Sukamto, 2010).

Menurut Rinaldi Munir (2005 : 176) "*Algoritma adalah urutan logis langkah-langkah penyelesaian masalah yang disusun secara sistematis*". Alur pemikiran dalam menyelesaikan suatu pekerjaan yang dituangkan secara tertulis. Yang ditekankan pertama adalah alur pikiran, sehingga algoritma seseorang dapat juga berbeda dari algoritma orang lain. Sedangkan penekanan kedua adalah tertulis, yang artinya dapat berupa kalimat, gambar, atau tabel tertentu.

Algoritma dapat dituliskan dalam berbagai notasi, misalnya dalam notasi kalimat-kalimat deskriptif. Dengan notasi kalimat deskriptif, deskripsi setiap langkah dijelaskan dengan bahasa sehari-hari secara jelas. Setiap langkah biasanya diawali dengan kata kerja seperti 'baca', 'hitung', 'masukan', 'bagi', 'ganti', dan sebagainya. Sedangkan pernyataan bersyarat dinyatakan dengan 'jika', 'maka', dan sebagainya.

2.4 Algoritma Breadth First Search

Breadth First Search adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara *preorder* yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya.

Algoritma ini memerlukan sebuah antrian q untuk menyimpan simpul yang telah dikunjungi. Simpul-simpul ini diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetangga dengannya. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan *tableboolean* untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali.

2.6.1 Cara kerja algoritma *breadth first search*

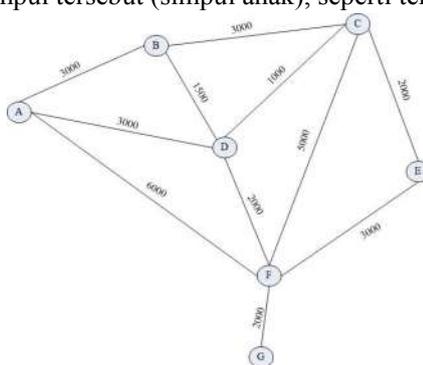
Dalam algoritma BFS, simpul anak yang telah dikunjungi disimpan dalam antrian. Antrian ini digunakan untuk mengacu simpul-simpul yang bertetangga dengan yang akan dikunjungi kemudian sesuai urutan pengantrian untuk memperjelas cara kerja algoritma BFS beserta antrian yang digunakannya.

Berikut langkah-langkah BFS

1. Masukkan simpul ujung (akar) ke dalam antrian.
2. Ambil simpul dari awal antrian, lalu cek apakah simpul merupakan solusi.
3. Jika simpul merupakan solusi, pencarian selesai dan hasil dikembalikan.
4. Jika simpul bukan solusi, masukkan seluruh simpul yang bertetangga dengan simpul tersebut (simpul anak) ke dalam antrian.

2.6.2 Contoh algoritma *breadth first search*

Algoritma *Breadth First Search* penelusuran setiap titik dilakukan pada simpul yang bertetangga dengan simpul tersebut (simpul anak), seperti terlihat pada gambar 1 berikut.



Gambar 1. Contoh *graph* untuk BFS dan *Tabu Search*

Misalnya menentukan jalur terpendek dengan *Breadth First Search* dari A ke E, maka berdasarkan gambar diatas kita dapat mengimplementasikan kedalam BFS untuk memperoleh jalur terpendek dengan langkah-langkah sebagai berikut.

Langkah 1

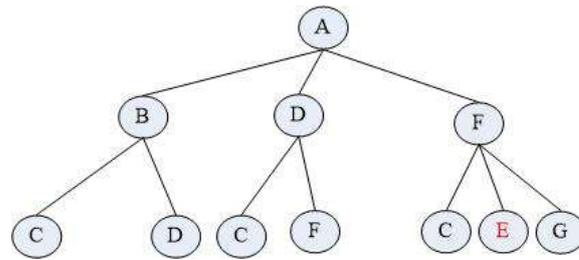
Kita ambil state awal adalah A, A memiliki pilihan jalur yaitu menuju B dan D dan F sehingga kita dapat representatif pohon pencarian.

Langkah ke 2

Kita melakukan pencarian jalur terpendek berdasarkan konsep dari *breadth first search* seperti yang dijelaskan sebelumnya

tujuan dari pencarian adalah menemukan titik E. Pencarian diawali dengan menelusuri titik pangkal yaitu titik A, karena titik A merupakan titik dengan level tertinggi maka penelusuran dilanjutkan dengan menjelajahi titik-titik pada level di bawahnya atau simpul yang bertetangga dengan simpul tersebut (simpul anak). Penelusuran selanjutnya adalah pada level kedua atau simpul yang bertetangga dengan simpul tersebut yaitu titik B, D dan F, Penelusuran selanjutnya adalah memasukkan simpul yang bertetangga dengan B yaitu C dan D Penelusuran selanjutnya adalah memasukkan simpul yang bertetangga dengan D yaitu C dan F, tahap selanjutnya memasukkan simpul yang bertetangga dengan F yaitu C dan E dan G.

dalam diagram ini titik E merupakan titik tujuan, maka dengan demikian proses pencarian berhenti, karena sudah menemukan solusi, seperti yang terlihat dalam gambar 2.



Gambar 2. Proses penyelesaian dengan metode *Breadth First Search* berdasarkan penelusuran diatas dengan metode *Breadth First Search* ditemukan jalur terpendek dari A menuju E adalah A,F,E.

2.5 Algoritma *tabu search*

Tabu Search pertama kali diperkenalkan oleh Glover pada tahun 1986. *Tabu Search* merupakan salah satu algoritma yang berada dalam ruang lingkup metode heuristik. Konsep dasar dari *Tabu Search* adalah suatu algoritma yang menuntun setiap tahapannya agar dapat menghasilkan fungsi tujuan yang paling optimum tanpa terjebak ke dalam solusi awal yang ditemukan selama tahapan ini berlangsung. Tujuan dari algoritma ini adalah mencegah terjadinya perulangan dan ditemukannya solusi yang sama pada suatu iterasi yang akan digunakan lagi pada iterasi selanjutnya.

Menurut Glover dan Laguna (1997) kata *tabu* atau “*taboo*” berasal dari bahasa tongan, suatu bahasa polinesia yang digunakan oleh suku aborigin pulau tonga untuk mengindikasikan suatu hal yang tidak boleh “disentuh” karena kesakralannya. Menurut kamus webster, *tabu* berarti larangan yang dipaksakan oleh kebudayaan sosial sebagai suatu tindakan pencegahan atau sesuatu yang dilarang karena berbahaya. Bahaya yang harus dihindari dalam *Tabu Search* adalah rute perjalanan yang tidak layak, dan terjebak tanpa ada jalan keluar.

Untuk menunjang sistematis dari tujuan *Tabu Search* digunakan dua macam *tools*, yaitu *adaptive memory* and *responsive exploration*. Keutamaan dari *adaptive memory* menuntun suatu prosedur yang mampu melakukan pencarian solusi dengan lebih ekonomis dan efektif. *Responsive exploration* lebih menekankan pada tahapan tiap proses yang harus dilalui selama proses pencarian itu berlangsung, di mana pada setiap tahapan tersebut mempunyai suatu variabel keputusan yang akan menuntun pada tahapan berikutnya sampai akhir proses pencarian dihentikan.

Struktur memori dalam *Tabu Search* menggunakan empat prinsip utama: *recency*, *frekuensi*, *quality* dan *influence*. *Recency* atau lebih lengkapnya *recency based memory* menjaga rekaman atau jejak solusi yang mengalami transformasi dan menyimpannya ke dalam suatu *short term memory* yang disebut *tabu list*. *Recency* menyediakan sebuah tipe informasi yang telah direkam oleh *recency based memory*. *Recency* dan frekuensi dapat saling melengkapi untuk membentuk suatu informasi permanen guna mengevaluasi pergerakan yang terjadi.

Quality menyatakan kemampuan untuk membedakan solusi terbaik yang dikunjungi selama pencarian atau iterasi berlangsung. *Influence* mempertimbangkan efek yang terjadi dari pemilihan solusi yang dipilih selama pencarian berlangsung, tidak hanya kualitas saja yang dipertimbangkan melainkan juga strukturnya

Glover dan Laguna (1997) mengatakan bahwa memori pada *Tabu Search* mempunyai dua sifat yaitu *explicit memory* dan *attributive memory*. *Explicit memory* menyimpan *complete solution* yang umumnya menghabiskan alokasi ruang memori dan waktu, sehingga untuk menghindari hal ini *complete solution* dikurangi sehingga hanya terdiri dari *elite solution* yang dikunjungi selama pencarian. *Attributive memory* menyimpan informasi tentang atribut dari solusi yang ditemukan yang mungkin dapat berubah dari satu solusi ke solusi lain. Sebagai suatu algoritma, *Tabu Search* mempunyai tahapan-tahapan dalam mencari solusi optimalnya.

2.6.1 Cara kerja algoritma *Tabu Search*

Langkah-langkah yang harus dikerjakan pada metode *Tabu Search* adalah sebagai berikut:

1. Menentukan titik awal.
2. Menentukan solusi alternatif yaitu dengan melakukan move (menukarkan) dua titik dalam solusi.
3. Mengevaluasi solusi-solusi alternatif dengan tabu list untuk melihat apakah kandidat solusi (solusi alternatif) tersebut sudah ada pada tabu list. Apabila solusi alternatif sudah ada dalam tabu list, maka solusi alternatif tersebut tidak akan dievaluasi lagi. Apabila solusi alternatif belum terdapat dalam tabu list, maka solusi alternatif tersebut disimpan dalam tabu list sebagai solusi alternatif terbaik.
4. Memilih solusi terbaik dari semua daftar calon solusi alternatif dan menetapkannya sebagai solusi optimum baru.
5. Memperbarui tabu list dengan memasukkan solusi optimum baru.
6. Apabila kriteria pemberhentian terpenuhi maka proses berhenti dan diperoleh solusi optimum. Jika tidak, proses kembali berulang dimulai dari langkah kedua.

2.6 Pencarian jalur terpendek modifikasi *Breadth First Search Dan Tabu Search*

Untuk memperoleh jalur terpendek berdasarkan jarak yang lebih optimal maka melakukan modifikasi pada algoritma *Breadth First Search* dengan algoritma *Tabu Search*. Cara kerja pencarian jalur terpendek dengan modifikasi algoritma BFS dan *Tabu Search* adalah sebagai berikut

1. Inisiasi yang sama dengan inisiasi BFS, yaitu penentuan simpul awal. Selain itu, tentukan jaraknya untuk dapat mengimplementasikan algoritma *Tabu Search*.
2. Melakukan *Breadth First Search* sampai ditemukan solusi, solusi tersebut disimpan.
3. Melakukan pengecekan pada setiap jarak yang dilalui, jika jarak tersebut melebihi atau sama dengan jarak sebelumnya maka proses pencarian akan berhenti.
4. Terus melakukan pencarian hingga mendapatkan jalur yang terpendek.

2.6.1 Contoh pencarian jalur terpendek modifikasi *breadth first search dan tabu search*

BFS merupakan pencarian jalur terpendek namun tidak optimal berdasarkan jarak oleh sebab itu perlu Modifikasi algoritma *Breadth First Search* dan *Tabu Search* untuk memperoleh jalur terpendek berdasarkan jarak, berikut contoh modifikasi algoritma BFS dan *tabu search* seperti yang terlihat pada gambar 4.

Misalkan menentukan jalur terpendek dari A ke E berdasarkan modifikasi *Breadth First Search* dengan *Tabu Search*, maka berdasarkan gambar 1, kita dapat mengimplementasikan kedalam BFS dan *Tabu Search* untuk memperoleh jalur terpendek dengan langkah-langkah sebagai berikut.

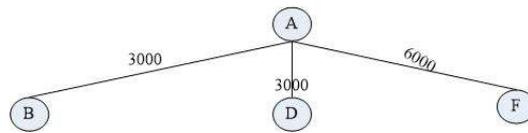
Langkah 1

Kita ambil state awal adalah A, A memiliki pilihan jalur yaitu menuju B dan D dan F sehingga kita dapat representasi pohon pencarian.

Langkah ke 2

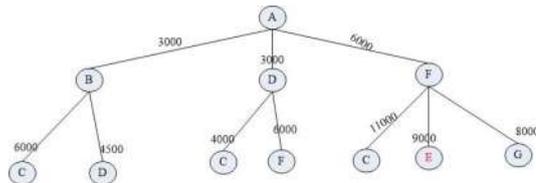
Kita melakukan pencarian jalur terpendek berdasarkan konsep dari *Breadth First Search* yang telah dimodifikasi dengan algoritma *Tabu Search*.

Tujuan dari pencarian adalah menemukan titik E. Pencarian diawali dengan menelusuri titik pangkal yaitu titik A, karena titik A merupakan titik dengan level tertinggi maka penelusuran dilanjutkan dengan menjelajahi titik-titik pada level di bawahnya atau simpul yang bertetangga dengan simpul tersebut (simpul anak), seperti yang terlihat pada gambar 3.



Gambar 3 proses pencarian jalur terpendek BFS dan *tabu Search*.

bahwa ini bukan merupakan solusi maka dilakukan pencarian lagi untuk menemukan tujuan atau solusinya.



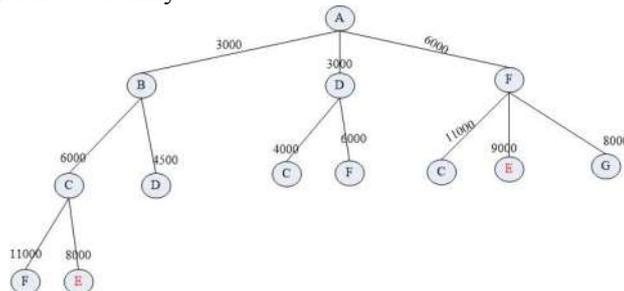
Gambar 3. proses pencarian jalur terpendek BFS dan *tabu Search* (lanjutan)

Langkah ke 3

Solusi ditemukan dengan jarak 9000, goal ini disimpan tetapi ini belum tentu merupakan jalur terpendek maka melakukan pencarian lagi untuk mendapatkan jalur terpendek.

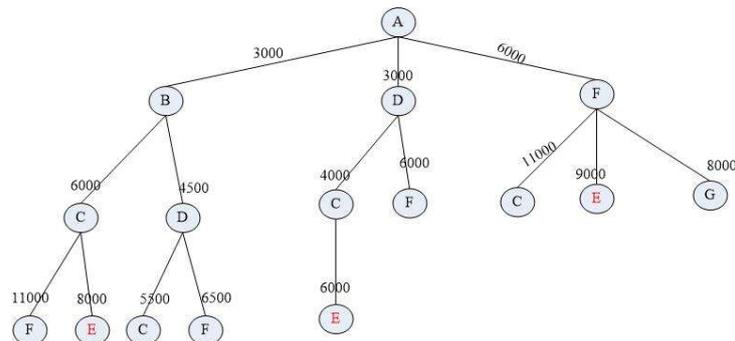
Langkah ke 4

Cek jarak untuk melihat *node* yang tidak bisa lagi dilanjutkan pencariannya karena melampaui atau sama dengan jarak sebelumnya.



Gambar 4. proses pencarian jalur terpendek BFS dan *tabu Search* (lanjutan)

Solusi ditemukan lagi dengan jarak 8000 maka jarak ini disimpan dan menggantikan jarak sebelumnya, tetapi ini belum tentu merupakan jalur terpendek maka melakukan pencarian lagi untuk mendapatkan jalur terpendek.



Gambar 4. proses pencarian jalur terpendek BFS dan *tabu Search* (lanjutan).

Solusi ditemukan lagi dengan jarak 6000 maka jarak ini disimpan dan menggantikan jarak sebelumnya, dengan demikian proses pencarian berhenti karena jarak yang akan lalu

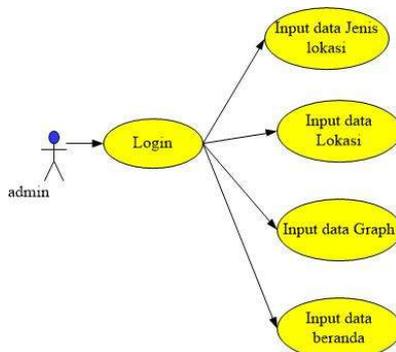
sudah melampaui atau sama dengan jarak sebelumnya. Maka berdasarkan proses penelusuran di atas dengan metode *Breadth First Search* dan *Tabu Search* ditemukan jalur terpendek dari A menuju E adalah A,D,C,E dengan jarak 6000.

2.7 Use case

Use case diagram adalah gambaran *graphical* dari beberapa atau semua aktor, *use case*, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use case* digunakan untuk menjelaskan bagaimana langkah-langkah yang seharusnya dikerjakan oleh system.



Gambar 5. use case diagram user



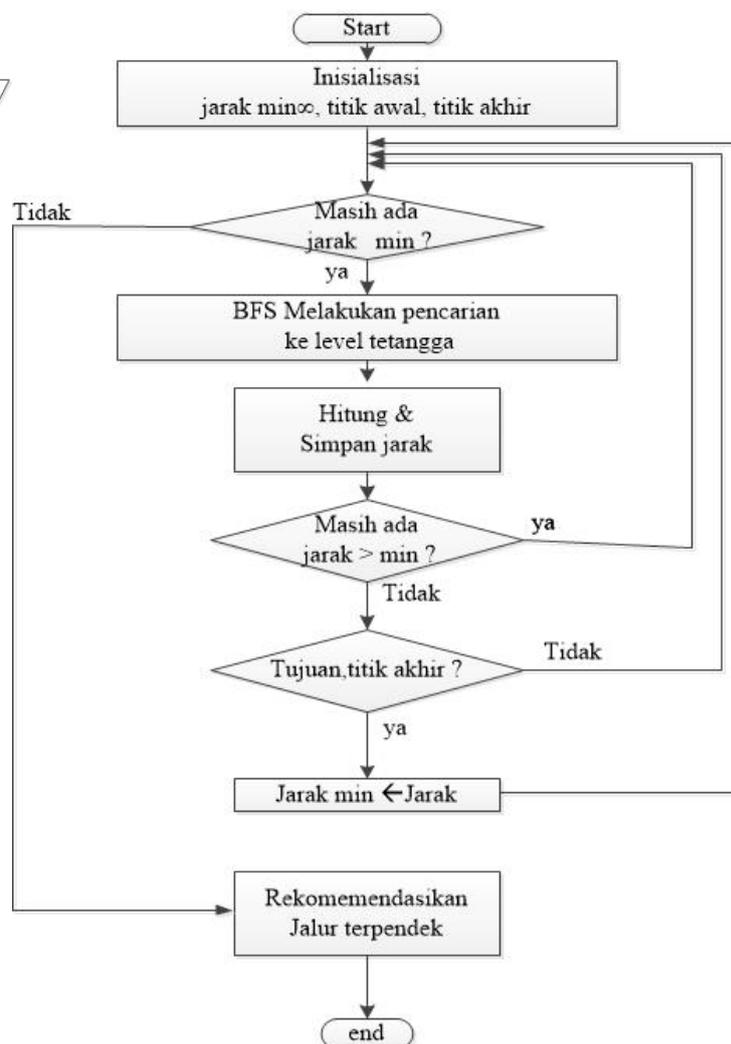
Gambar 6. use cae diagram admin

2.8 Flowchart

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. *Flowchart* merupakan langkah awal dalam pembuatan program. *Flowchart* digunakan untuk mempermudah penerjemahan alur aplikasi ke dalam bentuk program.



Gambar 7. Pencarian jalur terpendek BFS dan *Tabu Search*.

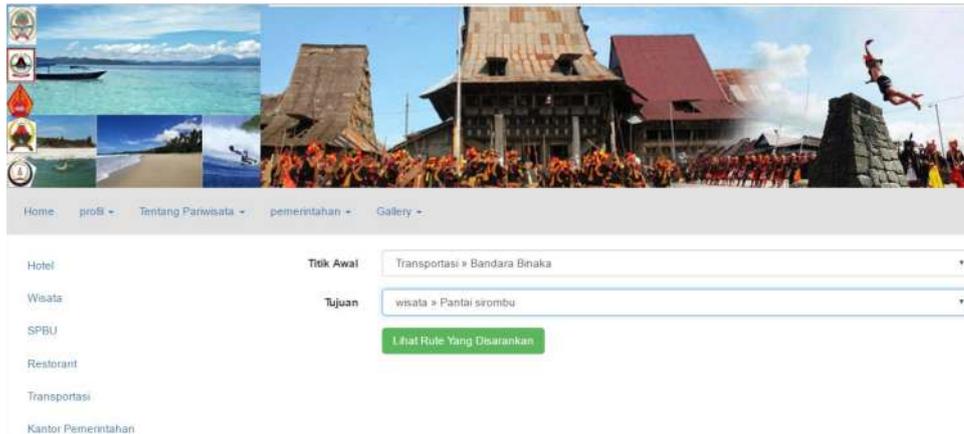


Gambar 8. Pencarian jalur terpendek dengan modifikasi algoritma BFS dan *Tabu Search*.

3. HASIL DAN PEMBAHASAN

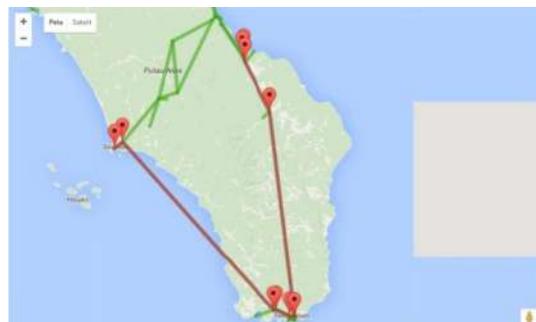
Model simulasi yang telah dibuat diuji untuk memastikan algoritma BFS dan modifikasi *Breadth First Search* dengan *Tabu Search* yang telah diimplementasikan berjalan dengan baik. Untuk melakukan pengujian maka *user* perlu memilih titik awal dan titik tujuan dan kemudian sistem akan menunjukkan jalur terpendek dan rekomendasi tempat yang dilalui seperti yang terlihat pada gambar berikut.

Halaman ini merupakan tampilan awal pencarian lokasi pariwisata dimana pengguna memilih titik awal dan kemudian memilih salah satu tujuan wisata yang akan dikunjungi.



Gambar 9. Halaman input titik awal dan tujuan.

Halaman berikut merupakan *ouput* dari sistem yang akan menunjukkan lokasi wisata dalam bentuk peta.



Gambar 10. Halaman *ouput* dari peta.

Halaman berikut merupakan *ouput* dari sistem yang akan merekomendasikan tempat-tempat yang dilewati berdasarkan algoritma *Breadth First Search*.

No.	Awal	Akhir	Jarak
1.	Bandara Binaka	Rumah makan pondok ayu	0.9 km
2.	Rumah makan pondok ayu	TB3	1.7 km
3.	TB3	Simpang idanogawo	15 km
4.	Simpang idanogawo	TB31	70.4 km
5.	TB31	TB 27	0.4 km
6.	TB 27	TB28	0.9 km
7.	TB28	Kantor Bupati Nias selatan	3.6 km
8.	Kantor Bupati Nias selatan	Simpang lagundri	0.75 km
9.	Simpang lagundri	simpang sirombu	78.7 km
10.	simpang sirombu	Pantai sirombu	3.1 km
JARAK TOTAL			175.45 km

Gambar 11. Halaman rekomendasi wisata yang dilewati dengan algoritma BFS.

Halaman berikut merupakan *ouput* dari sistem yang akan merekomendasikan tempat-tempat yang dilewati berdasarkan modifikasi algoritma *Breadth First Search* dan *Tabu Search*. Halaman ini juga merupakan halaman rekomendasi tempat wisata yang jaraknya lebih optimal.

Berdasar Jarak Terpendek

No.	Awal	Akhir	Jarak
1.	Bandara Binaka	Rumah makan pondok ayu	0.9 km
2.	Rumah makan pondok ayu	titik bantu4	0.101 km
3.	titik bantu4	Mega nasional Hotel	1.1 km
4.	Mega nasional Hotel	pantai togi laowo maru	0.35 km
5.	pantai togi laowo maru	Restorant bamboo house	0.4 km
6.	Restorant bamboo house	TB22	0.85 km
7.	TB22	Simpang miga	0.25 km
8.	Simpang miga	miga	1.1 km
9.	miga	TB33	14 km
10.	TB33	simpang mandehe1	17 km
11.	simpang mandehe1	simpang sirombu	16.5 km
12.	simpang sirombu	Pantai sirombu	3.1 km
JARAK TOTAL			55.651 km

Gambar 12. Halaman rekomendasi wisata yang dilewati dengan modifikas algortima BFS dan *Tabu Search*.

4. KESIMPULAN

Berdasarkan hasil implementasi rancangan dari simulasi rute terpendek lokasi pariwisata di pulau Nias berbasis web dengan metode *Breadth First Search* dan *Tabu Search* maka dapat disimpulkan bahwa aplikasi pencarian jalur terpendek menggunakan algoritma modifikasi *Breadth First Search* dengan *Tabu Search* dapat diimplementasikan dan digunakan di pulau Nias dan program simulasi yang dibuat mampu menunjukkan jalur terpendek yang optimal dengan modifikasi algoritma *Breadth First Search* dan *Tabu Search*.

5. SARAN

Saran yang diusulkan yaitu memperluas cakupan wilayah penelitian seperti meliputi rumah sakit, kantor-kantor penting lainnya sehingga dapat menjangkau kepentingan masyarakat dan para wisatawan dan perlu dilakukan penelitian untuk lintasan jalur terpendek dengan algoritma lain seperti *depth first search* dan *bround and branch* untuk membandingkan pencarian yang lebih optimal.

DAFTAR PUSTAKA

- [1]Aldisen Juniansyah. 2016. *aplikasi penentuan rute terpendek untuk bagian Pemasaran produk roti surya dengan metode Best First Search*. TeknikInformatika. Universitas Dahanse Bengkulu.
- [2]Andi Setya Wardana. 2015. *Penerapan algoritma Tabu Search untuk pencarian jalur terpendek dalam pengiriman barang*. TeknikInformatika. Universitas Widyatama Bandung. Bandung.
- [3]Chandra Maulana Dwi Kusumah. 2014. *penentuan jalur shuttle bus perusahaanOtobus*

-
- efisiensi yogyakarta menggunakan algoritma semut*. Pendidikan Matematika.Univeritas Negri Yogyakarta.
- [4]Fajar Rahmat Hidayat. 2010 .*penentuan Lintasan menggunakan algoritma breadth first search Studi kasus routing antar sentral trunk di pulau jawa*. Teknik Telekomunikasi. Semarang.
- [5]Feddy Setio Pribadi & Anggraini Mulwinda. 2010. *pencarian rute terpendek dengan menggunakan algoritma depthfirst, breath first dan hill climbing (study comparative)*, Semarang. Jurusan Teknik Elektro, Universitas Negeri Semarang.
- [6]Imron fauzi. 2011. *Penggunaan algoritma dijkstra dalam pencarian rute tercept dan rute terpendek(studi kasus pada jalan raya blok M dan kota*.Teknik Informatika. UIN Syaif Hidayatullah Jakarta.
- [7]I Wayan Eka Swastikayana. 2011.*sistem informasi geografis berbasis web untuk Pemetaan pariwisata kabupaten ganyar (studi kasus pada dinas Pariwisata kabupaten ganyar)*. Teknologi Industri. Universitas Pemangunan Nasional Veteran Yogyakarta.
- [8]Silva Rizkiah Widiyanto. 2011.*Aplikasi jalur terpendek dalam pencarian rute situs Pariwisata.Sistem Informasi* . Universitas Gunadarma.