Aplikasi Travelling Salesman Problem (TSP) dengan menggunakan Algoritma Branch & Bound untuk Menentukan Rute Tercepat Perjalanan Destinasi Wisata Pulau di Sumatra Barat

Akifa Nabil Ufairah 13519179
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13519179@std.stei.itb.ac.id

***Abstrak*—Indonesia sebagai negara kepulauan memiliki banyak sekali destinasi wisata pulau yang sangat indah, salah satunya destinasi wisata pulau di daerah Sumatra Barat. Banyaknya pulau-pulau yang indah di daerah Sumatra Barat membuat seringkali wisatawan ingin mengunjungi pulau-pulau tersebut dalam sekali kunjungan. Namun, waktu untuk liburan yang terbatas juga sering menjadi kendala dalam hal ini. Oleh karena itu, pada makalah ini akan dimplementasikan algoritma Djikstra untuk mencari rute terefektif pada destinasi wisata pulau di Sumatra Barat sehingga wisatawan dapat menikmati keindahan dari pulau-pulau di Sumatra Barat dalam waktu yang seefektif mungkin.**

***Kata Kunci*—Algoritma Branch & Bound, Graf, Pulau, Rute, Sumatra Barat, Travelling Salesman Problem, Wisata**

I. PENDAHULUAN

Sebagai sebuah negara kepulauan, Indonesia memiliki banyak destinasi wisata pulau dengan daya tarik yang beranekaragam. Sumatra Barat sebagai salah satu provinsi di Sumatra yang berbatasan langsung dengan laut juga memiliki berbagai pulau-pulau yang sangat indah. Hingga kini, wisata pulau di Sumatra Barat semakin berkembang sehingga menarik banyak wisatawan yang berkunjung ke Sumatra Barat untuk melakukan perjalananan wisata pulau ini.

Banyaknya pilihan pulau yang masih dalam satu Kawasan dan keindahan yang beragam disuguhkan setiap pulau ini membuat banyak wisatawan ingin mengunjungi beberapa pulau sekaligus dalam sekali kunjungan. Di samping itu, wisatawan juga biasanya memiliki waktu libur yang terbatas. Karena itu sebaiknya perlu direncanakan terlebih dahulu pulau-pulau yang ingin dikunjungi serta rute perjalanan mulai dari titik naik alat transportasi laut hingga urutan dari pulau-pulau yang ingin dikunjungi sehingga waktu perjalanan dapat diusahakan lebih efektif.

Pada makalah ini, penulis akan menerapkan pencarian rute tercepat dalam perjalanan wisata pulau di Sumatra Barat dengan menerapkan Travelling Salesman Probem (TSP) dengan bantuan algoritma Branch & Bound. Dengan perencanaan rute sebelum perjalanan wisata, wisatawan akan dapat mengunjungi pulau-pulau yang diinginkan dengan lebih efektif dari segi waktu.

II. LANDASAN TEORI

A. Teori Graf

a. Definisi

Graf dapat didefinisikan sebagai berikut,

$$G = (V,E)$$

dimana G adalah sebuah graf yang memenuhi kondisi:

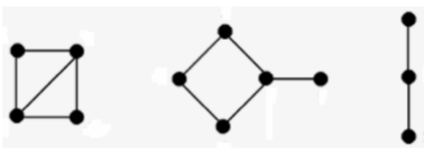
- V adalah sebuah himpunan tidak kosong dari simpul-simpul (*vertices*)
- E adalah sebuah himpunan sisi (edges) yang menghubungkan sepasang simpul.

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek-objek tersebut. Contohnya graf yang menyatakan peta jaringan jalan raya penghubung beberapa kota di Provinsi Jawa Timur.

b. Jenis-jenis graf

Berdasarkan ada tidaknya sisi gelang (sisi yang menghubungkan dua simpul yang sama) atau sisi ganda (terdapat lebih dari satu sisi menghubungkan dua simpul yang sama), graf terbagi menjadi dua jenis:

1. Graf sederhana (*simple graph*), yaitu graf yang tidak memiliki sisi ganda dan sisi gelang.



Gambar 1 : Contoh graf sederhana

(sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf)

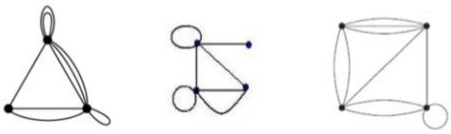
2. Graf tak-sederhana (*unsimple-graph*), yaitu graf yang memiliki sisi ganda atau sisi gelang. Graf tak-sederhana dapat diklasifikasikan lagi sebagai berikut,

a) Graf ganda (*multi-graph),* yaitu graf yang memiliki sisi ganda.

Gambar 2 : Contoh graf ganda

(sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf)

b) Graf semu (*pseudo-graph*), yaitu graf yang memiliki sisi gelang.

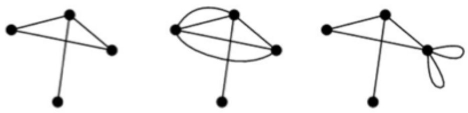


Gambar 3 : Contoh graf ganda

(sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf)

Berdasarkan orientasi arah pada sisi, graf juga diklasifikasikan menjadi dua jenis:

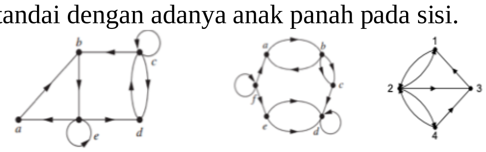
1. Graf tak-berarah (*undirected graph*), yaitu graf dengan sisi tidak memiliki orientasi arah,



Gambar 4 : Contoh graf tak-berarah

(sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf)

2. Graf berarah (*directed-graph* atau *digraph*), yaitu graf yang tiap sisi memiliki orientasi arah—ditandai dengan adanya anak panah pada sisi.

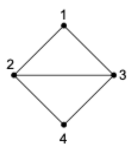


Gambar 5 : Contoh graf berarah

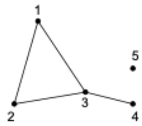
(sumber:http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf)

c. Terminologi Graf

1. Ketetanggaan (*Adjacent*)
   Dua simpul dikatakan 'bertetangga' jika terdapat sebuah sisi yang membuat kedua simpul terhubung langsung.



   Pada graf di atas, simpul 2 bertetangga dengan simpul 1,3, dan 4 sedangkan simpul 4 bertetangga dengan simpul 2 dan 3.
2. Bersisian (*Incidency*)
   Untuk sembarang sisi e = (v_i,v_j) dapat dikatakan e bersisian dengan simpul v_i atau simpul v_j. Contoh pada graf berikut,



   Sisi (3,4) bersisian dengan simpul 3 dan simpul 4, sedangkan tidak ada sisi yang bersisian denga simpul 5.
3. Simpul Terpencil (*Isolated Vertex*)
   Simpul terpencil adalah simpul yang tidak mempunyai sisi yang bersisian dengan simpul tersebut. Contoh graf pada poin bersisian, simpul 5 disebut simpul terpencil.
4. Graf Kosong (*empty graph* atau *null graph*), yaitu graf yang himpunan sisinya merupakan himpunan kosong (N_n). Contoh : Graf N_3 berarti adalah sebuah graf kosong dengan 3 simpul.
5. Derajat (*degree*)
   Derajat sebuah simpul berarti jumlah sisi bersisian pada sebuah simpul (dinotasikan dengan d(v)). Setiap sisi bersisian dihitung sebagai satu derajat, kecuali untuk sisi gelang, derajat dihitung dua. Khusus graf berarah, derajat simpul dibedakan lagi menjadi derajat masuk (d_{in}(v)) dan derajat keluar (d_{out}(v)) sesuai dengan arah panah sisi yang bersisian dilihat dari node terkait.
   Berdasarkan Lemma Jabat Tangan, jumlah derajat semua simpul pada suatu graf adalah genap, yaitu dua kali jumlah sisi pada graf tersebut.
6. Lintasan (*path*), yaitu barisan berselang-seling antara simpul dan sisi sebagai berikut v_0, e_1, v_1, e_2, v_2, …, v_{n-1}, e_n, v_n. yang menghubungkan simpul v_0 menuju simpul v_n pada graf G melalui sisi-sisi e_1= (v_0 ,v_1), e_2= (v_1 ,v), …, e_n = (v_{n-1} ,v_n), yaitu sisi-sisi pada graf G. Selain itu juga dikenal panjang lintasan yang menyatakan banyak sisi yang dilewati pada lintasan.
7. Sirkuit (*circuit*) atau Siklus (*cycle*), yaitu lintasan yang berawal dan berakhir pada simpul yang sama. Selain itu juga dikenal panjang sirkuit yang menyatakan banyak sisi yang dilewati pada lintasan.
8. Keterhubungan (*connected*)
   Dua simpul v_i dan v_j dikatakan terhubung jika terdapat lintasan dari v_i dan v_j . Sedangkan graf dikatakan terhubung jika terdapat lintasan untuk setiap pasang simpul v_i *dan* v_j pada himpunan simpul graf tersebut. Jika tidak, maka graf tersebut disebut graf tak-terhubung (*disconnected graph*). Untuk graf berarah, keterhubungan dapat diperiksa dengan menghapus arah pada graf tersebut dan meninjau keterhubungan seperti graf tak berarah.
9. Upagraf (*subgraph*), Komplemen Upagraf, dan Komponen (*connected component*)
   Misal G = (V,E) adalah sebuah graf. G1 = (V_1, E_1) dikatakan upagraf dari G jika $V_1 \subseteq$

V dan $E_1 \subseteq E$. Sedangkan komplemen upagraf G1 terhadap graf G adalah graf G2 = (V_2, E_2) sedemikian sehingga $E_2 = E - E_1$ dan V_2 adalah himpunan simpul yang bersisian dengan sisi-sisi pada E_2. Jumlah maksimum upagraf terhubung pada graf G disebut sebagai komponen graf.

10. Upagraf Merentang (*spanning subgraph*)

    Upagraf G_1 = (V_1, E_1) adalah upagraf merentang dari G jika $V_1 = V$ (yaitu G_1 mengandung semua simpul dari G).

11. Cut-set

    Cut-set dari suatu graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung sehingga cut-set selalu akab menghasilkan graf dengan dua komponen.

12. Graf Berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberi sebuah harga (bobot). Contohnya adalah graf ang merpresentasikan peta dari suatu lokasi dimana setiap sisi memiliki bobot yang menyatakan jarak antar dua titik.

d. Graf Khusus

Terdapat beberapa graf khusus sebagai berikut :

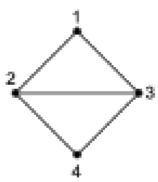
1. Graf Lengkap, yaitu graf sederhana dengan tiap simpulnya memiliki sisi ke semua simpul lainnya. Graf lengkap dengan n simpul dilambangkan dengan Kn dengan jumlah sisi adalah n(n-1)/2.
2. Graf Lingkaran, yaitu graf sederhana dengan derajat tiap simpul adalah dua. Graf lingkaran dengan n simpul dilambangkan dengan Cn.
3. Graf Teratur, yaitu graf dengan setiap simpul berderajat sama. Graf teratur dengan n simpul dan derajat tiap simpul r sering disebut sebagai graf teratur derajat r dengan jumlah sisi adalah nr/2.

e. Representasi Graf

Berikut beberapa contoh representasi graf.

1. Matriks Ketetanggaan (adjacency matrix)

   Setiap indeks elemen baris dan kolom menyatakan idetitas node yang ada pada graf sedangkan elemen pada baris ke-i dan kolom ke-j akan menyatakan banyaknya sisi dari simpul i ke simpul j. Pada graf berbobot elemen pada matriks sering digunakan untuk menyatakan nilai bobot dari sisi yang menghubungkan simpul i dan simpul j.

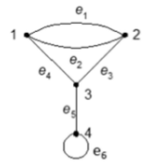


$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Gambar 6 : Matriks ketetanggan sebagai representasi graf

2. Matriks Bersisian (incidency matrix)

   Pada representasi ini, indeks elemen baris menyatakan identitas node yang terdapat pada graf dan indeks elemen kolom menyatakan idetitas edge yang ada pada graf. Setiap elemen matriks baris ke-i dan kolom ke-j akan bernilai 1 jika node dan sisi pada baris dan kolom tersebut bersisian, 0 jika tidak.



$$\begin{array}{c c} & \begin{array}{cccccc} e1 & e2 & e3 & e4 & e5 & e6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Gambar 7 : Matriks bersisian sebagai representasi graf

B. Travelling Salesman Problem (TSP)

Travelling Salesman Problem merupakan salah satu permasalahan yang berkaitan dengan aplikasi dari teori graf. TSP menyelesaikan persoalan mencari jarak minimum yang mengunjungi seluruh node pada sebuah graf dan kembali lagi ke kota asal keberangkatan. Persoalan ini erat kaitannya dengan sirkuit Hamilton. Sirkuit Hamilton merupakan sirkuit pada graf yang memastikan hanya melewati setiap node pada graf satu kali, kecuali node asal.

Setiap graf lengkap dengan n simpul selalu memiliki sirkuit Hamilton sejumlah $\frac{(n-1)!}{2}$. Pada dasarnya, TSP merupakan sebuah persoalan dengan kompleksitas algoritma penyelesaian yang besar. Contohnya, menggunakan Algoritma Brute Force, yaitu dengan mencari semua kemungkinan sirkuit hamilton dan membandingkan setiap total bobot sehingga didapatkan sirkuit hamilton dengan bobot minimum, kompleksitas yang diperoleh dapat dinyatakan dengan O(n!). Namun hingga kini, telah terdapat berbagai algoritma penyelesaian TSP yang mencoba memberikan solusi yang lebih efektif dari segi kompleksitas waktu dibandingkan menggunakan algoritma Brute Force.

C. Algoritma Branch & Bound

Branch & Bound merupakan sebuah algoritma yang pada umumnya dimanfaatkan untuk persoalan optimasi, yaitu dengan meminimalkan atau memaksimalkan suatu fungsi objektif, tanpa melanggar batasan dari persoalan. Algoritma ini akan menghitung suatu batasan pada simpul untuk menentukan apakah simpul layak atau tidak untuk di-*expand.* Dalam menyelesaikan persoalan dengan algoritma Branch & Bound digunakan pohon ruang status.

Pada algoritma Branch & Bound setiap simpul diberi suatu nilai *cost* dilambangkan dengan c(i), yang menyatakan nilai taksiran lintasan terkecil ke simpul berstatus tujuan dengan melalui simpul berstatus i. Kemudian, simpul berikutnya yang akan diekspansi adalah simpul yang memiliki cost yang paling kecil (*least cost search*).

Berikut adalah Algoritma Branch & Bound secara umum :

1. Simpul keberangkatan dijadikan sebagai akar pohon ruang status. Masukkan simpul akar ke dalam antrian Q. Jika simpul akar adalah solusi maka proses selesai.
2. Jika Q kosong, maka tidak ada solusi dan proses selesai.
3. Jika Q tidak kosong, pilih dari antrian Q simpul i dengan cost c(i) paling kecil. Jika terdapat lebih dari satu simpul dengan cost terkecil, pilih satu secara sembarang.
4. Jika simpul i yang dipilih merupakan simpul solusi, berarti solusi telah ditemukan dan proses selesai. Namun jika bukan simpul solusi, ekspansi seluruh anak dari simpul tersebut. Apabila simpul i tidak memiliki anak, kembali ke poin 2.
5. Pada setiap anak j dari simpul i yang telah dibangkitkan, hitung c(j) dan masukan semua anak ke dalam Q
6. Ulangi poin 2.

D. Implementasi Algoritma Branch & Bound pada TSP menggunakan Matriks ongkos-tereduksi (*Reduced Cost Matrix)*

Pada implementasi Branch & Bound dalam pemecahan TSP dapat digunakan bantuan Matriks ongkos-tereduksi. Matriks ongkos-tereduksi (*Reduced Cost Matrix}* adalah matriks adjacency dari graf yang sudah dilakukan proses reduksi sehingga setiap kolom dan barisnya mengandung paling sedikit satu buah nol dan semua elemen lainnya non-negatif. Pada kasus ini, matriks *adjacency* graf berisi elemen yang menyatakan bobot dari tiap sisi. Jika tidak terdapat sisi yang menghubungkan dua node, set elemen matriks dengan nilai tak hingga.

Misalkan sebuah matriks R berikut merupakan matriks *adjacency* sebuah graf yang ingin dicari Matriks ongkos tereduksinya

$$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}$$

Untuk membuat setiap baris dan kolom memiliki paling sedikit satu 0, kurangi setiap baris pada matriks adjacency dengan elemen terkecil pada baris tersebut.

$$\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix} \begin{matrix} R_1 - 10 \\ R_2 - 2 \\ R_3 - 2 \\ R_4 - 3 \\ R_5 - 4 \end{matrix} \begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix}$$

Kemudian lakukan hal yang sama pada setiap kolom matriks. Setiap nilai terkecil dari baris dan kolom dijumlahkan dan dicatat sebagai cost dari simpul akar.

$$\begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix} \begin{matrix} \\ \\ C_1 - 1 \\ C_3 - 3 \\ \\ \end{matrix} \begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix} = A$$

Total semua pengurang = 10 + 2 + 2 + 3 + 4 + 1 + 3 = 25 $\longrightarrow$ Cost simpul akar

Misalkan A adalah sebuah matriks tereduksi untuk simpul R sedangkan S adalah anak dari simpul R sehingga sisi (R,S) pada pohon ruang status berkoresponden dengan sisi (i,j) pada perjalanan. Jika S bukan simpul daun, maka matriks bobot tereduksi untuk simpul S dapat dihitung sebagai berikut :

1. Set semua nilai pada baris i dan kolom j menjadi tak hingga untuk mencegah tidak ada lintasan keluar dari simpul i atau masuk pada simpul j.
2. Ubah A(j,1) menjadi tak hingga untuk mencegah penggunaan sisi (j,1) yang berakibat lintasan balik ke simpul akar.
3. Reduksi lagi setiap baris dan kolom pada matriks A yang bukan elemen bernilai tak hingga.
4. Jika r merupakan total dari nilai pengurang, set nilai batas untuk simpul S sebagai berikut:
$$c(S) = c(R) + A(i,j) + r$$
5. Hasil reduksi menghasilkan matriks B yang akan dilakukan lagi.

Proses di atas dilakukan pada semua simpul anak yang memungkinkan. Setelah selesai, lanjutkan dengan pemilihan simpul aktif dengan *cost* minimum sebagaimana telah dijabaarkan pada sub-bagian sebelumnya.

E. Wisata Pulau di Sumatra Barat

Sumatra Barat merupakan salah satu provinsi di Indonesia yang terletak pada bagian barat pulau Sumatra. Geografis dari daerah Sumatra Barat yang memiliki banyak pulau-pulau yang indah menjadi salah satu destinasi wisata yang menarik dari provinsi ini. Kawasan di Sumatra Barat yang terkenal dengan pulau-pulaunya yang indah adalah Kawasan Mandeh. Pada Kawasan ini terdapat beberapa pulau kecil dengan beragam keindahan pemandangan yang disajikan oleh tiap pulaunya.

Gambar 8 Pulau-Pulau di Kawasan Mandeh

Berikut beberapa pulau-pulau yang terkenal di Kawasan ini :

1. Pulau Pasumpahan

Pulau yang berada di Kecamatan Bungus, Teluk Kabung ini memiliki obyek wisata pantai pasir putih dengan terumbu karang yang masih terjaga. Perairan yang jernih dan panorama bawah air yang masih terjaga ini membuat wisatawan biasanya tertarik untuk melakukan *snorkeling* di pulau ini.

Gambar 9 Pulau Pasumpahan

2. Pulau Pamutusan

Pulau yang seakan-akan terlihat terbelah dua saat air laut pasang ini memiliki pemandangan bawah laut yang juga sayang untuk dilewatkan. Wisawatan yang menyelam di perairan pulau ini akan disuguhkan dengan ikan warna-warni yang berenang keluar-masuk terumbu karang. Selain itu, di pulau ini pengunjung juga bisa melakukan hiking dan menikmati keindahan pasir putih dan air biru dilengkapi dengan pohon-pohon hijau dari puncak bukit.

Gambar 10 Pulau Pamutusan

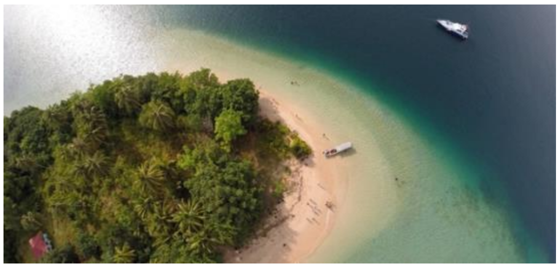
3. Pulau Cubadak

Pulau Cubadak juga tak kalah menariknya dengan pulau-pulau sebelumnya. Disini wisatawan bisa menikmati aktifitas berselancar, berlayar, snorkeling, hingga ski air. Di pulau ini juga terdapat berbagai penginapan dengan sentuhan tradisional.

Gambar 1 Pulau Cubadak

4. Pulau Setan

Pulau Setan merupakan pulau kecil tak berpenghuni dengan keindahan laut biru, pasir putih, dan ombak yang tenang. Pulau ini juga menjadi salah satu pulau yang menarik berbagai wisatawan untuk berkunjung.



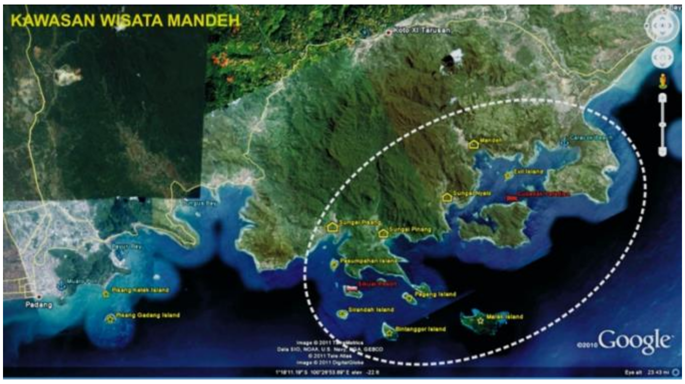
Gambar 2 Pulau Cubadak

Sumber : https://posmetropadang.co.id/pesona-wisata-pulau-setan-tawarkan-keindahan-pantai-yang-menakjubkan/

III. PEMBAHASAN

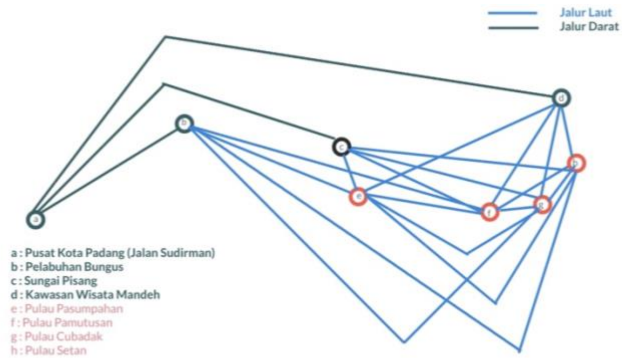
A. Pemodelan Masalah

Berikut adalah peta pulau-pulau yang berada pada Kawasan Mandeh, Sumatra Barat.



Gambar 3 Peta Kawasan Mandeh

Peta di atas pada graf dapat digambarkan sebagai berikut :

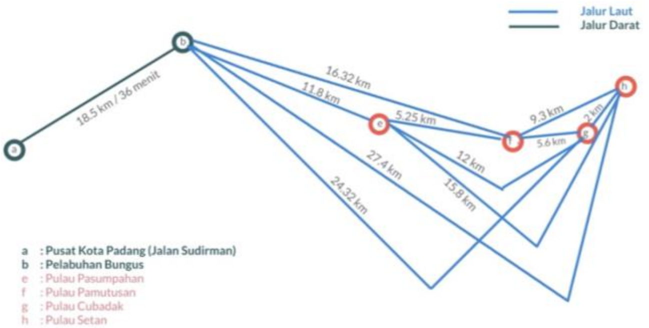


Simpul a merupakan titik keberangkatan wisatawan, yaitu dari Pusat Kota Padang, Sumatra barat. Kemudian terdapat 3 titik yang merupakan pilihan tempat sewa alat transportasi laut (simpul b,c, dan d). Pada makalah ini, hanya diambil 4 pulau yang paling banyak dikunjungi oleh wisatawan (ditandai dengan simpul berwarna merah), yaitu Pulau Pasumpahan, Pulau Pammutusan, Pulau Cubadak, dan Pulau Setan.

Pada perjalanan wisata pulau, biasanya wisatawan menggunakan kendaraan darat pribadi terlebih dahulu hingga sampai ke titik awal keberangkatan perjalanan laut dan harus kembali lagi ke titik tersebut. Persoalan ini mirip dengan TSP karena disini akan dicari sirkuit Hamilton minimum yang dapat ditempuh oleh wisatawan dalam perjalanan laut.

C. Aplikasi Algoritma Branch & Bound pada penentuan rute tercepat perjalanan destinasi wisata pulau di Sumatra Barat.

Karena pada kasus ini terdapat 3 pilihan node yang merepresentasikan lokasi sewa alat transportasi laut, graf dapat dipecah menjadi 3 berdasarkan titik keberangkatan awal perjalanan laut yang dipilih. Berikut adalah kasus pertama jika wisatawan memilih untuk mengawali perjalanan laut di Pelabuhan Bungus :



Matriks adjacency dari graf di atas dapat diperoleh sebagai berikut :

$$R = \begin{bmatrix} \infty & 11.8 & 16.32 & 24.32 & 27.4 \\ 11.8 & \infty & 5.25 & 12 & 15.8 \\ 16.32 & 5.25 & \infty & 5.6 & 9.3 \\ 24.32 & 12 & 5.6 & \infty & 2 \\ 27.4 & 15.8 & 9.3 & 2 & \infty \end{bmatrix}$$

Ubah kedalam bentuk matriks ongkos-tereduksi dan hitung *cost* untuk simpul akar.

$$R = \begin{bmatrix} \infty & 0 & 4.52 & 12.52 & 15.6 \\ 6.55 & \infty & 0 & 6.75 & 10.55 \\ 11.07 & 0 & \infty & 0.35 & 4.05 \\ 22.32 & 10 & 3.6 & \infty & 0 \\ 25.4 & 13.8 & 7.3 & 0 & \infty \end{bmatrix}$$

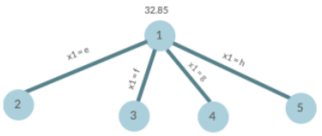
$$Reduksi\ baris:\ R1 - 11{,}8\, R2 - 5.25, R3 - 5.25, R4 - 2, R5 - 2$$

$$M1 = \begin{bmatrix} \infty & 0 & 4.52 & 12.52 & 15.6 \\ 0 & \infty & 0 & 6.75 & 10.55 \\ 4.52 & 0 & \infty & 0.35 & 4.05 \\ 15.77 & 10 & 3.6 & \infty & 0 \\ 18.85 & 13.8 & 7.3 & 0 & \infty \end{bmatrix}$$

$$Reduksi\ kolom:\ C1 - 6{,}55$$

Cost simpul akar / c(1) = 11 + 5.25 + 5.25 + 2 + 2 + 6.55 = 32.85

Lanjutkan dengan menambahkan simpul anak dari simpul akar, yaitu simpul-simpul yang dapat dituju dari simpul akar dan cari cost dari masing-masing simpul.



Ekspansi tiap simpul anak satu per satu. Set semua nilai pada baris ke-1 dan kolom ke-2 dan elemen baris ke-2 kolom ke-1 menjadi tak hingga, lanjutkan dengan proses reduksi dan perhitungan cost c(2).

$$M2 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 6.75 & 10.55 \\ 4.53 & \infty & \infty & 0.35 & 4.05 \\ 25.77 & \infty & 3.6 & \infty & 0 \\ 28.85 & \infty & 7.3 & 0 & \infty \end{bmatrix} R3 - 0{,}35, C1 - 4.17 \rightarrow\ M2 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 6.75 & 10.55 \\ 0 & \infty & \infty & 0 & 3.7 \\ 11.6 & \infty & 3.6 & \infty & 0 \\ 14.68 & \infty & 7.3 & 0 & \infty \end{bmatrix}$$

$$c(2) = 32.85 + A(1,2) + r = 32.85 + 0 + (0.35 + 4.17) = 37.37$$

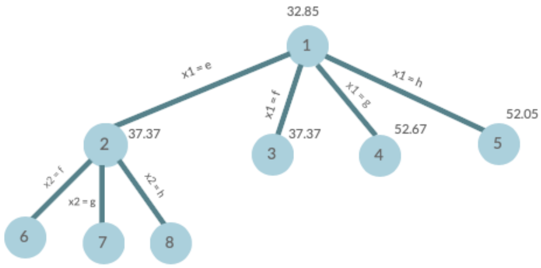
Lakukan hal yang sama dengan simpul 3, 4, dan 5 dan diperoleh Matriks ongkos-tereduksi dan cost masing-masing sebagai berikut

$$M3 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 6.75 & 10.55 \\ \infty & 0 & \infty & 0.35 & 4.05 \\ 15.77 & 10 & \infty & \infty & 0 \\ 18.85 & 13.8 & \infty & 0 & \infty \end{bmatrix}, M4 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 0 & \infty & 10.55 \\ 4.52 & 0 & \infty & \infty & 4.05 \\ \infty & 10 & 3.6 & \infty & 0 \\ 11.55 & 6.55 & 0 & \infty & \infty \end{bmatrix},$$

$$M5 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 0 & 6.75 & 10.55 \\ 4.52 & 0 & \infty & 0.35 & \infty \\ 12.17 & 6.4 & 0 & \infty & \infty \\ \infty & 13.8 & 7.3 & 0 & \infty \end{bmatrix}$$

$$c(3) = 37.37, c(4) = 52.67, c(5) = 52.05$$

Pilih dari simpul aktif yang ada (simpul 2,3,4,5) yang memiliki cost minimum, karena pada kasus ini terdapat dua simpul yang memiliki cost minimum (simpul 2 dan 3), pilih salah satu sembarang.



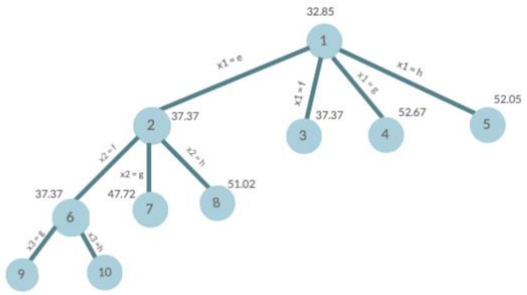
Lakukan hal yang sama dengan simpul 2,3,4, dan 5 pada simpul 6,7, dan 8 sehingga diperoleh

$$M6 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & 3.7 \\ 11.6 & \infty & \infty & \infty & 0 \\ 14.68 & \infty & \infty & 0 & \infty \end{bmatrix}, M7 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & 3.7 \\ 11.6 & \infty & 0 & \infty & 0 \\ 14.68 & \infty & 3.7 & 0 & \infty \end{bmatrix},$$

$$M8 = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & \infty \\ 8.6 & \infty & 0 & \infty & \infty \\ 14.68 & \infty & 7.3 & 0 & \infty \end{bmatrix}$$

$$c(6) = 37.37, c(4) = 47.72, c(5) = 51.02$$

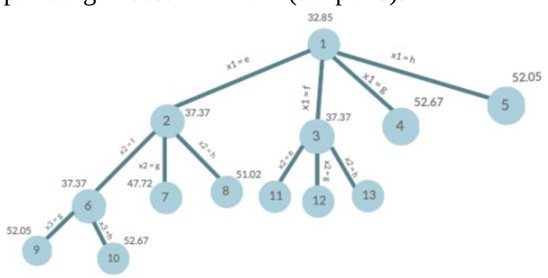
Pilih kembali dari simpul aktif yang ada (simpul 3,4,5,6,7,8) yang memiliki cost minimum. Karena pada kasus ini terdapat dua simpul yang memiliki cost minimum (simpul 6 dan 3), pilih salah satu simpul sembarang.



Lanjutkan dengan perhitungan cost simpul 9 dan 10 sama seperti perhitungan cost pada simpul-simpul sebelumnya dengan matriks ongkos-tereduksi. Diperoleh hasil :

$$c(9) = 52.05, c(10) = 52.67$$

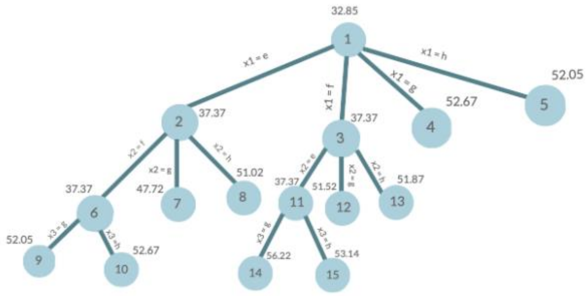
Dari simpul aktif saat ini (simpul 3,4,5,7,8,9,10) pilih kembali simpul dengan cost minimum (simpul 3).



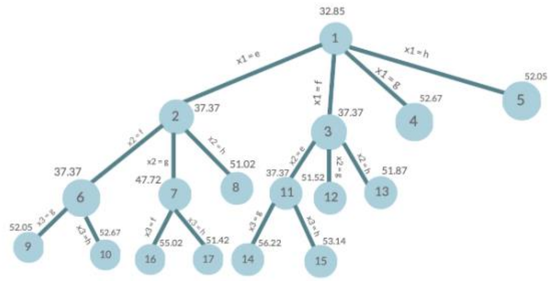
Dari perhitungan cost pada simpul 11,12, dan 13 diperoleh :

$$c(11) = 37.37, c(12) = 51.52, c(13) = 51.87$$

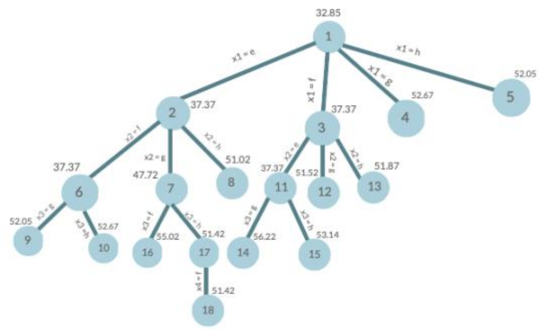
Dari pilihan simpul aktif saat ini (simpul 3,4,5,7,8,9,10,11,12,13) pilih kembali simpul dengan cost minimum (simpul 11). Lanjutkan dengan menghitung cost anak dari simpul 11 sehingga diperoleh pohon status berikutnya :



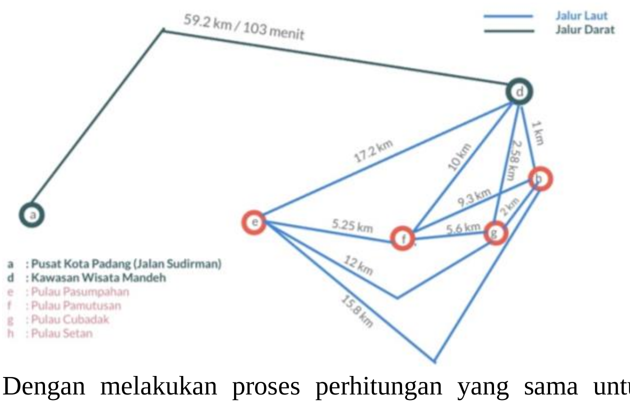
Pilih lagi simpul aktif dengan cost terkecil, yaitu simpul 7 dengan cost 47.72. Lakukan perhitungan cost simpul anak sehingga diperoleh pohon status sebagai berikut:



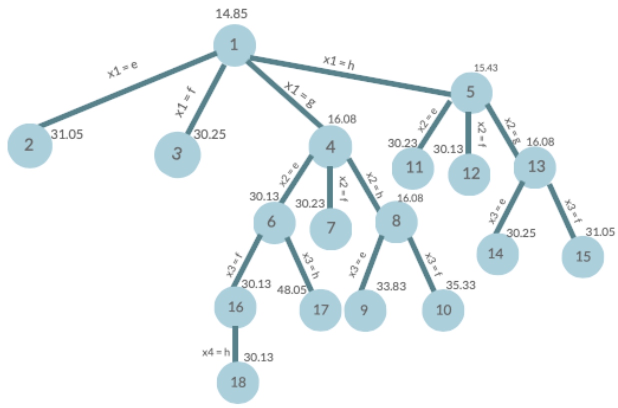
Simpul aktif dengan cost terkecil saat ini adalah simpul 17 dengan cost 51.42. Lakukan perhitungan cost simpul anak dari simpul 17



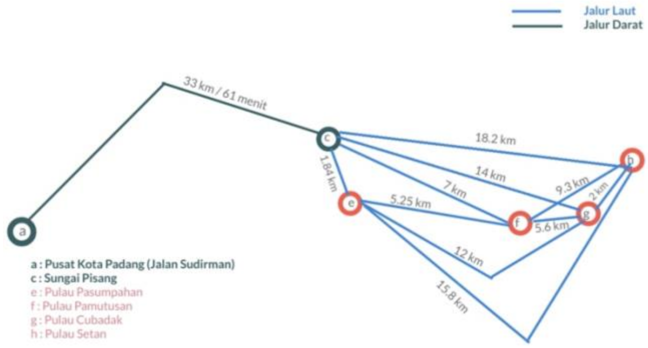
Simpul 18 merupakan simpul solusi sehingga diperoleh solusi sirkuit Hamilton dengan bobot jarak minimum pada persoalan ini adalah Pelabuhan bungus (node b)- Pulau Pasumpahan (e) – Pulau Cubadak (g) – Pulau Setan (h) – Pulau Pamutusan (f) – Pelabuhan bungus (node b) dengan total jarak 51.42 km.



Dengan melakukan proses perhitungan yang sama untuk menemukan sirkuit hamilton minimum pada persoalan di atas diperoleh pohon status akhir sebagai berikut



Diperoleh sirkuit Hamilton minimum perjalanan laut dengan titik naik alat transportasi laut di Kawasan Wisata Mandeh adalah 30.13 km yaitu melalui lintasan Kawasan Wisata Mandeh (simpul d) – Pulau Cubadak (simpul g) – Pulau Pasumpahan (simpul e) – Pulau Pamutusan (simpul f) – Pulau Setan (simpul h) – Kawasan Wisata Mandeh (simpul d).



Dengan proses yang sama juga diperoleh sirkuit Hamilton minimum perjalanan laut dengan titik naik alat transportasi laut di Sungai Pisang adalah 32.4 km, yaitu melalui lintasan Sungai Pisang (simpul c) – Pulau Pasumpahan (simpul e) – Pulau Cubadak (simpul g) – Pulau Setan (simpul h) – Pulau Pamutusan (simpul f) - Sungai Pisang (simpul c).

Setelah mendapat total panjang lintasan yang ditempuh pada perjalanan laut, untuk mendapat rute tercepat jika wisatawan berangkat dari Pusat Kota Padang, ini dapat dihitung dengan mempertimbangkan jarak dan perbedaan kecepatan alat transportasi darat dan laut. Kecepatan alat transportasi laut yang biasa digunakan adalah sekitar 30 km/jam sehingga jika diasumsikan semua kecepatan alat transportasi laut yang digunakan sama dengan 30 km/jam diperoleh waktu perjalanan laut jika berangkat dari Pelabuhan Bungus $\frac{51.42\ km}{30\ km/jam} = 1.714\ jam = 102.84\ menit$, jika berangkat dari Kawasan Mandeh $\frac{30.13\ km}{30\ km/jam} = 1.0043\ jam = 60.26\ menit$, sedangkan jika berangkat dari Sungai Pisang $\frac{32.4\ km}{30\ km/jam} = 1.08\ jam = 64.8\ menit$. Sehingga perkiraan total waktu yang akan ditempuh dalam perjalanan dari Kota Padang hingga balik lagi ke Kota Padang untuk masing-masing rute :

- Rute dengan titik naik alat transportasi laut di Pelabuhan Bungus : (36 + 102.84 + 36) menit = 174.84 menit
- Rute dengan titik naik alat transportasi laut di Kawasan Mandeh : (103 + 60.26 + 103) menit = 266.26 menit
- Rute dengan titik naik alat transportasi laut di Sungai

Pisang : (61 + 64.8 + 61) menit = 186.8 menit

Dari hasil ini diperoleh rute tercepat adalah rute dengan titik naik alat transportasi laut di Pelabuhan Bungus.

V. KESIMPULAN

Pencarian rute tercepat dalam perjalanan destinasi wisata pulau di Kawasan Mandeh Sumatra Barat merupakan salah satu aplikasi dari Travelling Salesman Problem (TSP). Terdapat berbagai algoritma yang dapat menyelesaikan permasalahan TSP, salah satunya yaitu Branch & Bound. Algoritma ini dapat mengoptimasi rute yang dipilih sehingga diperoleh rute tercepat jika berangkat dari Kota Padang adalah yaitu rute dengan titik naik alat transportasi laut di Pelabuhan Bungus dan Lintasan Perjalanan laut Pelabuhan bungus (node b)- Pulau Pasumpahan (e) – Pulau Cubadak (g) – Pulau Setan (h) – Pulau Pamutusan (f) – Pelabuhan bungus (node b).

VI. UCAPAN TERIMAKASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa berkat-Nya penulis dapat menyelesaikan makalah ini dengan baik, Penulis juga berterimakasih kepada keluarga,serta teman-teman yang telah memberikan dukungan dalam menyelesaikan makalah ini. Penulis juga tak lupa berterimakasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. , Ibu Fariska Zakhralativa Ruskanda S.T.,M.T, dan Ibu Dr. Nur Ulfa Maulidevi S.T., M.Sc. selaku dosen mata kuliah matematika diskrit yang telah memberikan banyak ilmu dalam perkuliahan. Terakhir, penulis memohon maaf apabila dalam penulisan makan ini terdapat kesalaha baik disengaja maupun tidak disengaja. Penulis berharap makalah ini dapat berguna bagi banyak orang.

REFERENSI

[1] Munir,Rinaldi.2015.Graf.http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf. (Diakses pada tanggal 3 Desember 2020)

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Branch-&-Bound-Bagian-1.pdf (Diakses pada tanggal 3 9 Desember 2020)

[3] Richard Neapolitan and Kumarss Naimipour. 2015. Foundations of Algorithms, Fifth Edition. Jones and Bartlett Publishers, Inc., USA. (Diakses pada tanggal 9 Desember 2020)

[4] https://travel.kompas.com/read/2014/08/09/152200527/Pasumpahan.Pulau.Terindah.di.Sumatera.Barat (Diakses pada tanggal 6 Desember 2020)

[5] https://www.nativeindonesia.com/pantai-dan-pulau-sikuai/ (Diakses pada tanggal 6 Desember 2020)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2020

Akifa Nabil Ufairah 13519179