

Penerapan Algoritma Branch and Bound dalam Menyelesaikan *Travelling Salesman Problem* untuk Menentukan Rute Halalbihalal Saat Hari Raya Idulfitri

Muhammad Azhar Faturahman - 13519020

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): m.azhar.faturahman@gmail.com

Abstrak—Halalbihalal merupakan salah satu tradisi masyarakat Indonesia saat merayakan hari raya Idulfitri. Kegiatan ini biasanya dilakukan dengan cara berkeliling mengunjungi setiap rumah kerabat. Rute perjalanan melakukan halalbihalal ini dapat dimodelkan dengan persoalan *Travelling Salesman Problem*. Salah satu metode untuk menyelesaikan persoalan tersebut dapat menggunakan algoritma *Branch and Bound* dengan teknik matriks ongkos-tereduksi.

Kata kunci—*branch and bound*; *travelling salesman problem*; *matriks ongkos-tereduksi*; *halalbihalal*; *idulfitri*

I. PENDAHULUAN

Hari raya Idulfitri atau Lebaran merupakan perayaan umat Islam yang jatuh pada tanggal 1 Syawal setelah selesai menjalankan ibadah puasa selama bulan Ramadan. Di Indonesia, perayaan hari raya Idulfitri biasanya diwarnai dengan acara halalbihalal, yaitu kegiatan maaf-memaafkan yang dilakukan oleh sekelompok orang seperti keluarga, teman, ataupun tetangga. Di beberapa daerah, kegiatan halalbihalal biasanya diadakan dengan cara berkeliling mengunjungi setiap rumah kerabat. Karena perlu mengunjungi banyak tempat, maka diperlukan rute perjalanan terbaik untuk mengunjungi tempat-tempat tersebut agar kegiatan halalbihalal dapat dilaksanakan dengan efisien, baik dari aspek biaya maupun waktu dan semua rumah kerabat dapat dikunjungi selama hari raya Idulfitri.

Persoalan menentukan rute perjalanan melakukan halalbihalal mirip dengan persoalan *Travelling Salesman Problem* (TSP), yaitu menentukan rute perjalanan dengan jarak terpendek yang dilakukan oleh seorang pedagang sehingga ia melalui setiap kota tepat hanya sekali dan kembali lagi ke kota asal keberangkatan. Pada kasus ini, pedagang adalah seseorang yang melakukan perjalanan untuk melakukan halalbihalal, sedangkan kota adalah kediaman yang ia singgahi, kota asal keberangkatan dapat diasumsikan sebagai rumah seseorang tersebut. Ada banyak cara untuk menyelesaikan persoalan TSP, salah satunya adalah dengan menggunakan algoritma Branch and Bound. Penggunaan algoritma Branch and Bound dalam menyelesaikan persoalan TSP pun ada bermacam teknik,

namun teknik yang digunakan dalam kasus ini akan menggunakan cara *reduced-cost matrix*.



Gambar 1. Ilustrasi Halalbihalal (Sumber: <https://umroh.com/blog/wp-content/uploads/2020/03/hikmah-silaturahmi-source-shutterstock1.jpg> diakses pada 8 Mei 2021)

II. LANDASAN TEORI

A. Algoritma Branch and Bound

Algoritma *Branch and Bound* (BnB) adalah salah satu algoritma yang dapat digunakan untuk menyelesaikan persoalan optimasi. Persoalan optimasi ialah persoalan untuk mencari nilai minimum atau maksimum dari suatu fungsi objektif yang tidak melanggar batasan dari persoalan tersebut. Umumnya persoalan yang melibatkan algoritma branch and bound direpresentasikan sebagai sebuah graf atau pohon. Proses pencarian nilai optimum dilakukan menggunakan pohon ruang status dengan setiap simpul memiliki bobot tertentu.

Pada prinsipnya, algoritma *Branch and Bound* terdiri atas 2 hal sebagai berikut :

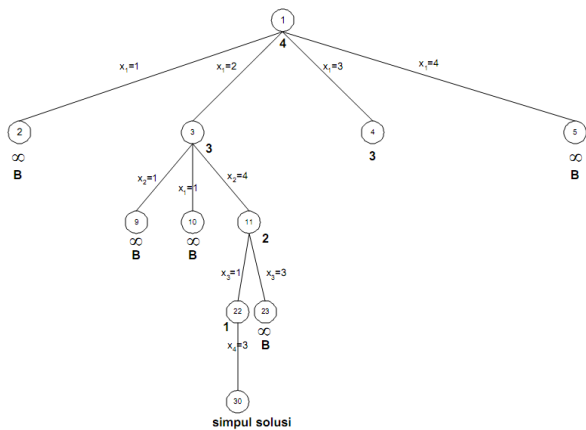
- Algoritma akan membangkitkan simpul-simpul anak pada pohon ruang status dari simpul yang sedang

diperiksa. Setiap simpul anak memiliki variasi tertentu dari simpul orang tuanya dan juga memiliki sebuah bobot yang merepresentasikan biaya yang dibutuhkan untuk mencapai simpul anak dari simpul asal. Proses ini dinamakan **branching**.

- b. Algoritma akan melakukan pemangkasan pada jalur atau simpul yang dianggap tidak lagi mengarah pada status yang diinginkan. Pemangkasan ini dilakukan menggunakan fungsi pembatas, sehingga proses ini disebut sebagai **bounding**.

Algoritma *Branch and Bound* merupakan kombinasi antara algoritma *Breath First Search* (BFS) dan *Least Cost Search*. Algoritma BFS adalah algoritma pencarian pada pohon ruang status dengan proses pembangkitan simpul-simpul lainnya secara "melebar", yaitu semua simpul tetangga secara terurut. Algoritma BFS menggunakan struktur data *queue* untuk menyimpan simpul-simpul yang sedang dibangkitkan dengan prinsip *First In First Out* (FIFO). *Least Cost Search* merupakan prinsip proses pencarian pada pohon ruang status dengan cara membangkitkan simpul anak dari sebuah simpul yang memiliki bobot paling kecil.

Pada algoritma *Branch and Bound*, pembangkitan simpul mengikuti beberapa aturan tertentu, yang paling umum dipakai adalah menggunakan *best-first rule*. Setiap simpul yang dibangkitkan diberi sebuah *cost* atau bobot tertentu, misalnya $\hat{c}(i)$, yang merupakan nilai taksiran lintasan termurah ke simpul status tujuan melalui simpul status i . Simpul berikutnya yang akan diekspans tidak lagi berdasarkan urutan pembangkitan, melainkan simpul yang memiliki *cost* yang paling minimum (untuk kasus minimasi).



Gambar 2. Pohon ruang status dalam pencarian solusi menggunakan algoritma *Branch and Bound* (Sumber: Diktat Kuliah Strategi Algoritma ITB)

Gambar 2 mengilustrasikan pohon ruang status yang digunakan dalam proses pencarian solusi menggunakan algoritma *Branch and Bound*. Proses pencarian ini dimulai dari simpul 1 atau simpul akar, kemudian dibangkitkan simpul 2, 3, 4, dan 5. Simpul 2 dan simpul 5 dimatikan karena telah melebihi Batasan atau *constraint* dan dianggap tidak akan mencapai solusi. Pada simpul 3 memiliki bobot paling kecil sehingga dilakukan ekspans untuk membangkitkan simpul anak-anaknya. Proses ini dilakukan terus menerus hingga

mencapai simpul solusi atau tidak ada lagi simpul yang dapat diekspans.

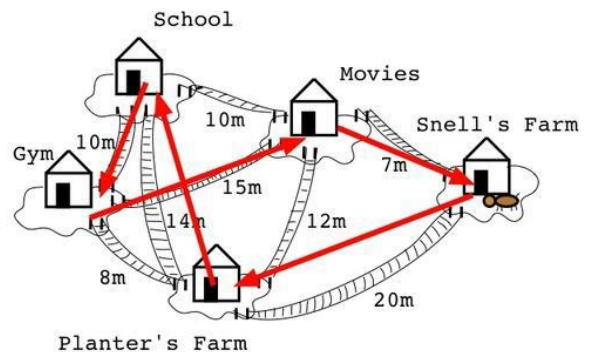
Algoritma global *Branch and Bound* sebagai berikut :

1. Masukkan simpul akar ke dalam antrian Q. Jika simpul akar merupakan solusi, maka solusi telah ditemukan. Stop.
2. Jika Q kosong, tidak ada solusi. Stop.
3. Jika Q tidak kosong, keluarkan simpul yang berada di antrian Q dengan nilai *cost* terkecil. Jika terdapat beberapa, ambil secara acak.
4. Jika simpul yang dikeluarkan merupakan simpul solusi, maka solusi telah ditemukan. Stop. Jika simpul yang dikeluarkan bukan simpul solusi, maka bangkitkan anak-anak simpulnya. Jika simpul yang dikeluarkan tidak mempunyai anak, kembali ke langkah 2.
5. Untuk setiap anak dari simpul yang dikeluarkan, hitung *cost* dari simpul-simpulnya dan masukkan ke dalam antrian berdasarkan urutan *cost* yang telah dihitung.
6. Kembali ke langkah 2.

B. Travelling Salesman Problem (TSP)

Persoalan *Travelling Salesman Problem* atau TSP adalah sebagai berikut: Diberikan suatu daftar kota dan jarak antar kota diketahui, tentukan rute perjalanan terpendek yang mana setiap kota dikunjungi tepat sekali dan kembali lagi ke kota asal keberangkatan. TSP adalah salah satu persoalan optimasi yang paling banyak diteliti dan merupakan persoalan yang penting dalam ilmu computer. Persoalan ini termasuk ke dalam jenis *NP-Hard Problem*.

Persoalan ini biasanya dimodelkan sebagai suatu graf lengkap dengan n buah simpul. Setiap simpul menyatakan sebuah kota, kemudian terdapat sisi yang menghubungkan antar simpul yang merepresentasikan jalan yang menghubungkan kota-kota. Setiap sisi memiliki bobot tersendiri yang menyatakan biaya untuk berpindah dari satu kota ke kota lain, umumnya biaya yang digunakan adalah jarak antar kota.



Gambar 3. Ilustrasi persoalan *Travelling Salesman Problem* (Sumber: Diktat Kuliah Strategi Algoritma ITB)

Persoalan *Travelling Salesman Problem* dapat dipandang sebagai suatu persoalan menemukan sirkuit Hamilton yang memiliki bobot paling minimum pada suatu graf. Ada banyak algoritma untuk menyelesaikan persoalan ini, seperti menggunakan cara *Brute Force* atau menggunakan algoritma *Branch and Bound*.

C. Menyelesaikan persoalan TSP menggunakan algoritma *Branch and Bound*

Terdapat dua cara untuk menyelesaikan persoalan TSP menggunakan algoritma *Branch and Bound*, yaitu dengan cara menggunakan matriks ongkos-tereduksi (*reduced cost matrix*) dari atau menggunakan bobot minimum tur lengkap. Pada kasus ini, penulis akan menggunakan metode matriks ongkos-tereduksi.

Sebuah matriks dikatakan tereduksi jika setiap kolom dan barisnya mengandung paling sedikit satu buah nol, dan semua elemen lainnya non-negatif.



Gambar 4. Reduksi matriks M menjadi matriks tereduksi M' (Sumber: *Diktat Kuliah Strategi Algoritma ITB*)

Metode ini menggunakan matriks ketetanggaan yang menyatakan bobot sisi antar 2 buah simpul yang bertetangga.

Langkah-langkah penyelesaiannya adalah sebagai berikut:

1. Mereduksi matriks dan menyimpan semua nilai pengurang baris dan kolom. Jika sudah menjadi matriks tereduksi, matriks tersebut akan menjadi simpul akar dari pencarian, dan jumlah total semua pengurang akan menjadi cost akar.
2. Membangkitkan semua anak yang mungkin dimiliki simpul saat ini. Misalkan rute perjalanan dimulai dari 1, A adalah matriks yang tereduksi untuk simpul R, S adalah anak dari simpul R sehingga sisi (R, S) pada pohon ruang status berkoresponden dengan sisi (i, j) pada perjalanan.
3. Jika S bukan simpul daun, maka matriks bobot tereduksi untuk simpul S dapat dihitung sebagai berikut:
 - a. Ubah semua nilai pada baris i dan kolom j menjadi ∞.
 - b. Ubah nilai A(j,1) menjadi ∞.
 - c. Reduksi kembali semua baris dan kolom pada matriks A.
 - d. Hitung bobot simpul S dengan persamaan berikut:

$$\hat{c}(S) = \hat{c}(R) + A(i,j) + r$$

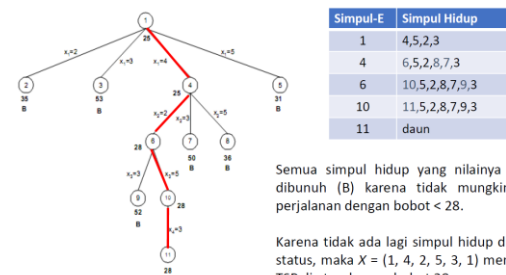
$\hat{c}(S)$: bobot perjalanan minimum yang melalui simpul S

$\hat{c}(R)$: bobot perjalanan minimum yang melalui simpul R, R adalah simpul orang tua dari S

$A(i,j)$: bobot sisi (i,j) pada graf G yang berkoresponden dengan sisi (R,S) pada pohon ruang status

r : jumlah semua pengurang pada proses reduksi matriks untuk simpul S

4. Jika belum mencapai daun, ulangi dari langkah 2 dengan simpul yang akan diekspans adalah simpul dengan bobot terkecil.
5. Jika sudah mencapai daun, bunuh semua simpul hidup yang memiliki bobot yang lebih besar daripada bobot terbaik simpul saat ini.
6. Jika masih ada simpul hidup yang bobotnya tidak lebih besar dengan bobot terbaik simpul saat ini, periksa kembali simpul tersebut sampai mencapai daun. Proses ini terus diulang hingga mendapatkan simpul dengan bobot terbaik atau tidak ada lagi simpul hidup.



Semua simpul hidup yang nilainya lebih besar dari 28 dibunuh (B) karena tidak mungkin lagi menghasilkan perjalanan dengan bobot < 28.

Karena tidak ada lagi simpul hidup di dalam pohon ruang status, maka X = (1, 4, 2, 5, 3, 1) menjadi solusi persoalan TSP di atas dengan bobot 28.

Gambar 5. Pohon ruang status untuk menyelesaikan TSP menggunakan algoritma *Branch and Bound* dengan metode matriks ongkos-tereduksi (Sumber: *Diktat Kuliah Strategi Algoritma ITB*)

III. IMPLEMENTASI DAN EKSPERIMEN

A. Pengambilan Data

Pada makalah ini, penulis melakukan percobaan dengan cara mengambil data tidak sebenarnya secara acak. Data yang digunakan berupa kumpulan titik-titik dan koordinat pada sebuah peta.

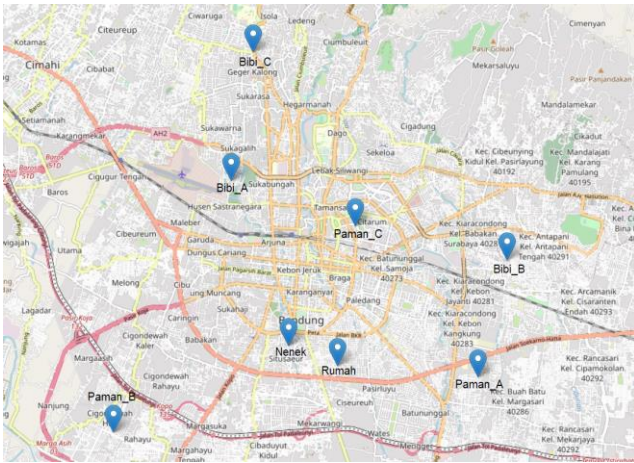
Titik-titik koordinat yang diambil adalah sebagai berikut :

1. Rumah (-6.9457, 107.6136)
2. Nenek (-6.9408, 107.6018)
3. Paman_A (-6.9484, 107.6476)
4. Paman_B (-6.9617, 107.5593)
5. Paman_C (-6.9121, 107.6180)
6. Bibi_A (-6.9011, 107.5878)
7. Bibi_B (-6.9203, 107.6546)
8. Bibi_C (-6.8703, 107.5931)

Pada makalah ini, penulis melakukan percobaan dengan cara mengambil data tidak sebenarnya secara acak. Data yang digunakan berupa kumpulan titik-titik dan koordinat pada sebuah peta.

Titik *Rumah* dipilih sebagai titik awal rute karena rute perjalanan melakukan halalbihalal mengunjungi kerabat dimulai dari rumah sendiri. Selanjutnya akan dicari rute

perjalanan sehingga setiap titik koordinat dikunjungi lalu kembali lagi ke *Rumah* dan rute tersebut adalah rute dengan bobot yang paling sedikit. Pada kasus ini, diasumsikan jika dari suatu titik bisa ke titik lainnya pada peta, dengan kata lain graf yang digunakan adalah graf lengkap, sehingga terdapat $(n - 1)!$ rute perjalanan yang dapat dipilih.



Gambar 6. Letak titik-titik koordinat rumah kerabat yang akan dikunjungi (Sumber: Dokumentasi Pribadi, dibuat menggunakan Leaflet-OpenStreetMap)

Kemudian, dihitung jarak antar titik koordinat menggunakan persamaan harvesine dan hasil dari perhitungan tersebut disimpan dalam sebuah matriks ketetangaan.

inf	1403	3768	6244	3765	5711	5334	8675
1403	inf	5121	5233	3654	4675	6251	7893
3768	5121	inf	9853	5195	8435	3219	10557
6244	5233	9853	inf	8500	7430	11474	10819
3765	3654	5195	8500	inf	3543	4141	5390
5711	4675	8435	7430	3543	inf	7668	3473
5334	6251	3219	11474	4141	7668	inf	8762
8675	7893	10557	10819	5390	3473	8762	inf

Gambar 7. Matriks ketetangaan untuk data pada Gambar. 6 (Sumber: Dokumentasi Pribadi)

Elemen pada baris ke- i dan kolom ke- j untuk matriks pada Gambar. 7 merupakan bobot yang berupa jarak untuk menempuh perjalanan dari simpul S ke simpul R . Urutan baris ke- i sama dengan urutan titik-titik koordinat. Setelah itu dibuatlah matriks ongkos-tereduksi dari matriks pada Gambar 7 untuk menghitung ongkos atau bobot minimum sebagai batas bawah.

inf	0	2365	1011	2292	4308	3931	7272
0	inf	3718	0	2181	3272	4848	6490
549	1902	inf	2804	1906	5216	0	7338
1011	0	4620	inf	3197	2197	6241	5586
222	111	1652	1127	inf	0	598	1847
2238	1202	4962	127	0	inf	4195	0
2115	3032	0	4425	852	4449	inf	5543
5202	4420	7084	3516	1847	0	5289	inf

Batas bawah simpul akar = 28866

Gambar 8. Hasil reduksi matriks (Sumber: Dokumentasi Pribadi)

Setelah berhasil mereduksi matriks, didapatkan nilai batas bawah untuk simpul akar sebesar 28866. Nilai ini merupakan perkiraan bobot paling minimum untuk menempuh rute perjalanan mengunjungi semua simpul dan kembali ke simpul awal. Kemudian, semua simpul anak dari simpul akar dibangkitkan dan dihitung kembali bobot untuk semua simpul anak tersebut menggunakan matriks ongkos-tereduksi.

No	Ekpansi Simpul																																																																
1	<p>No simpul : 2 Simpul ekspans : Rumah Simpul yang dibangkitkan : Paman_A Matriks ongkos-tereduksi :</p> <table border="1"> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>0</td><td>inf</td><td>inf</td><td>0</td><td>2181</td><td>3272</td><td>4848</td><td>6490</td></tr> <tr><td>inf</td><td>1902</td><td>inf</td><td>2804</td><td>1906</td><td>5216</td><td>0</td><td>7338</td></tr> <tr><td>1011</td><td>0</td><td>inf</td><td>inf</td><td>3197</td><td>2197</td><td>6241</td><td>5586</td></tr> <tr><td>222</td><td>111</td><td>inf</td><td>1127</td><td>inf</td><td>0</td><td>598</td><td>1847</td></tr> <tr><td>2238</td><td>1202</td><td>inf</td><td>127</td><td>0</td><td>inf</td><td>4195</td><td>0</td></tr> <tr><td>1263</td><td>2180</td><td>inf</td><td>3573</td><td>0</td><td>3597</td><td>inf</td><td>4691</td></tr> <tr><td>5202</td><td>4420</td><td>inf</td><td>3516</td><td>1847</td><td>0</td><td>5289</td><td>inf</td></tr> </table> <p>Total bobot simpul = 32083</p>	inf	inf	inf	inf	inf	inf	inf	inf	0	inf	inf	0	2181	3272	4848	6490	inf	1902	inf	2804	1906	5216	0	7338	1011	0	inf	inf	3197	2197	6241	5586	222	111	inf	1127	inf	0	598	1847	2238	1202	inf	127	0	inf	4195	0	1263	2180	inf	3573	0	3597	inf	4691	5202	4420	inf	3516	1847	0	5289	inf
inf	inf	inf	inf	inf	inf	inf	inf																																																										
0	inf	inf	0	2181	3272	4848	6490																																																										
inf	1902	inf	2804	1906	5216	0	7338																																																										
1011	0	inf	inf	3197	2197	6241	5586																																																										
222	111	inf	1127	inf	0	598	1847																																																										
2238	1202	inf	127	0	inf	4195	0																																																										
1263	2180	inf	3573	0	3597	inf	4691																																																										
5202	4420	inf	3516	1847	0	5289	inf																																																										
2	<p>No simpul : 35 Simpul ekspans : Paman_A Simpul yang dibangkitkan : Bibi_B Matriks ongkos-tereduksi :</p> <table border="1"> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>0</td><td>inf</td><td>inf</td><td>0</td><td>2181</td><td>3272</td><td>inf</td><td>6490</td></tr> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>1011</td><td>0</td><td>inf</td><td>inf</td><td>3197</td><td>2197</td><td>inf</td><td>5586</td></tr> <tr><td>222</td><td>111</td><td>inf</td><td>1127</td><td>inf</td><td>0</td><td>inf</td><td>1847</td></tr> <tr><td>2238</td><td>1202</td><td>inf</td><td>127</td><td>0</td><td>inf</td><td>inf</td><td>0</td></tr> <tr><td>inf</td><td>2180</td><td>inf</td><td>3573</td><td>0</td><td>3597</td><td>inf</td><td>4691</td></tr> <tr><td>5202</td><td>4420</td><td>inf</td><td>3516</td><td>1847</td><td>0</td><td>inf</td><td>inf</td></tr> </table> <p>Total bobot simpul = 32083</p>	inf	inf	inf	inf	inf	inf	inf	inf	0	inf	inf	0	2181	3272	inf	6490	inf	inf	inf	inf	inf	inf	inf	inf	1011	0	inf	inf	3197	2197	inf	5586	222	111	inf	1127	inf	0	inf	1847	2238	1202	inf	127	0	inf	inf	0	inf	2180	inf	3573	0	3597	inf	4691	5202	4420	inf	3516	1847	0	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf																																																										
0	inf	inf	0	2181	3272	inf	6490																																																										
inf	inf	inf	inf	inf	inf	inf	inf																																																										
1011	0	inf	inf	3197	2197	inf	5586																																																										
222	111	inf	1127	inf	0	inf	1847																																																										
2238	1202	inf	127	0	inf	inf	0																																																										
inf	2180	inf	3573	0	3597	inf	4691																																																										
5202	4420	inf	3516	1847	0	inf	inf																																																										
3	<p>No simpul : 39 Simpul ekspans : Bibi_B Simpul yang dibangkitkan : Paman_C Matriks ongkos-tereduksi :</p> <table border="1"> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>0</td><td>inf</td><td>inf</td><td>0</td><td>inf</td><td>3272</td><td>inf</td><td>6490</td></tr> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>1011</td><td>0</td><td>inf</td><td>inf</td><td>inf</td><td>2197</td><td>inf</td><td>5586</td></tr> <tr><td>inf</td><td>111</td><td>inf</td><td>1127</td><td>inf</td><td>0</td><td>inf</td><td>1847</td></tr> <tr><td>2238</td><td>1202</td><td>inf</td><td>127</td><td>inf</td><td>inf</td><td>inf</td><td>0</td></tr> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>5202</td><td>4420</td><td>inf</td><td>3516</td><td>inf</td><td>0</td><td>inf</td><td>inf</td></tr> </table> <p>Total bobot simpul = 32083</p>	inf	inf	inf	inf	inf	inf	inf	inf	0	inf	inf	0	inf	3272	inf	6490	inf	inf	inf	inf	inf	inf	inf	inf	1011	0	inf	inf	inf	2197	inf	5586	inf	111	inf	1127	inf	0	inf	1847	2238	1202	inf	127	inf	inf	inf	0	inf	inf	inf	inf	inf	inf	inf	inf	5202	4420	inf	3516	inf	0	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf																																																										
0	inf	inf	0	inf	3272	inf	6490																																																										
inf	inf	inf	inf	inf	inf	inf	inf																																																										
1011	0	inf	inf	inf	2197	inf	5586																																																										
inf	111	inf	1127	inf	0	inf	1847																																																										
2238	1202	inf	127	inf	inf	inf	0																																																										
inf	inf	inf	inf	inf	inf	inf	inf																																																										
5202	4420	inf	3516	inf	0	inf	inf																																																										
4	<p>No simpul : 45 Simpul ekspans : Paman_C Simpul yang dibangkitkan : Bibi_C Matriks ongkos-tereduksi :</p> <table border="1"> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>0</td><td>inf</td><td>inf</td><td>0</td><td>inf</td><td>3272</td><td>inf</td><td>inf</td></tr> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>1011</td><td>0</td><td>inf</td><td>inf</td><td>inf</td><td>2197</td><td>inf</td><td>inf</td></tr> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>2111</td><td>1075</td><td>inf</td><td>0</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td><td>inf</td></tr> <tr><td>inf</td><td>4420</td><td>inf</td><td>3516</td><td>inf</td><td>0</td><td>inf</td><td>inf</td></tr> </table> <p>Total bobot simpul = 34057</p>	inf	inf	inf	inf	inf	inf	inf	inf	0	inf	inf	0	inf	3272	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	1011	0	inf	inf	inf	2197	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	2111	1075	inf	0	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	inf	4420	inf	3516	inf	0	inf	inf
inf	inf	inf	inf	inf	inf	inf	inf																																																										
0	inf	inf	0	inf	3272	inf	inf																																																										
inf	inf	inf	inf	inf	inf	inf	inf																																																										
1011	0	inf	inf	inf	2197	inf	inf																																																										
inf	inf	inf	inf	inf	inf	inf	inf																																																										
2111	1075	inf	0	inf	inf	inf	inf																																																										
inf	inf	inf	inf	inf	inf	inf	inf																																																										
inf	4420	inf	3516	inf	0	inf	inf																																																										

5	<pre>No simpul : 154 Simpul ekspans : Bibi_C Simpul yang dibangkitkan : Bibi_A Matriks ongkos-tereduksi : inf inf inf inf inf inf inf inf 0 inf inf 0 inf inf inf inf inf inf inf inf inf inf inf inf 1011 0 inf inf inf inf inf inf inf inf inf inf inf inf inf inf inf 1075 inf 0 inf Total bobot simpul = 34057</pre>
6	<pre>No simpul : 156 Simpul ekspans : Bibi_A Simpul yang dibangkitkan : Paman_B Matriks ongkos-tereduksi : inf inf inf inf inf inf inf inf 0 inf Total bobot simpul = 34057</pre>
7	<pre>No simpul : 157 Simpul ekspans : Paman_B Simpul yang dibangkitkan : Nenek Matriks ongkos-tereduksi : inf Total bobot simpul = 34057</pre>

Tabel 1. Proses pembangkitan simpul akar hingga daun (Sumber: Dokumentasi Pribadi)

Semua simpul anak yang dibangkitkan disimpan dalam suatu antrian simpul hidup, kemudian dipilih simpul hidup yang memiliki total bobot paling minimum untuk kemudian diekspans seperti proses sebelumnya. Proses ini dilakukan secara terus menerus dan berulang-ulang hingga dicapai simpul daun yang merupakan akhir dari proses pencarian dengan algoritma *Branch and Bound*. Tabel 1 memperlihatkan proses pembangkitan simpul solusi dari akar hingga ke daun. Pada kasus ini diasumsikan jika rute yang diambil adalah rute yang pertama kali memenuhi kriteria hasil. Lintasan dari simpul akar ke simpul daun merepresentasikan rute perjalanan yang ditempuh dari simpul *Rumah* melalui semua simpul pada peta lalu kembali lagi ke simpul *Rumah*.

```
Hasil pencarian rute:
Rumah → Paman_A → Bibi_B → Paman_C → Bibi_C → Bibi_A
→ Paman_B → Nenek → Rumah

Panjang rute: 34057 m
Banyaknya pembangkitan simpul: 158
```

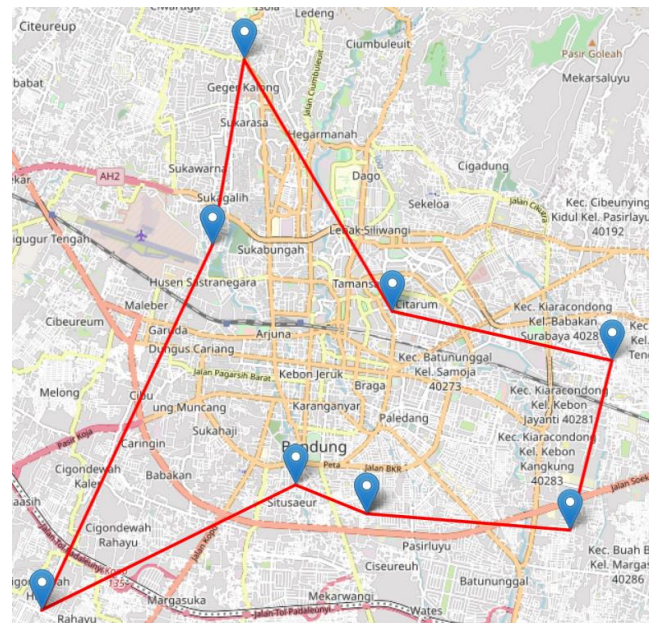
Gambar 9. Hasil keluaran program untuk menyelesaikan Travelling Salesman Problem (Sumber: Dokumentasi Pribadi)

Pada gambar diatas, didapatkan hasil keluaran program dengan rute *Rumah → Paman_A → Bibi_B → Paman_C → Bibi_C → Bibi_A → Paman_B → Nenek → Rumah*. Program juga menampilkan bahwa rute tersebut memiliki bobot berupa jarak atau Panjang rute sebesar 34.057 m atau sekitar 34 km. Diperlukan pembangkitan simpul sebanyak 158 kali untuk mendapatkan hasil tersebut.

IV. KESIMPULAN

Algoritma *Branch and Bound* dapat digunakan untuk menyelesaikan persoalan *Travelling Salesman Problem* untuk menentukan rute halalbihalal saat hari raya Idulfitri dengan menggunakan teknik matriks ongkos-tereduksi.

Pada implementasi dan eksperimen kasus dari data yang diberikan didapatkan hasil sebagai berikut:



Gambar 10. Peta hasil pencarian rute halalbihalal saat hari raya Idulfitri (Sumber: Dokumentasi Pribadi, dibuat menggunakan Leaflet-OpenStreetMap)

Rute halalbihalal yang dapat ditempuh melalui *Rumah → Paman_A → Bibi_B → Paman_C → Bibi_C → Bibi_A → Paman_B → Nenek → Rumah*. Bobot total yang berupa jarak tempuh melalui rute diatas adalah sebesar 34.057 m.

Implementasi yang digunakan tidak hanya terbatas untuk kasus yang diberikan, namun dapat digunakan untuk kasus-kasus lain selama diberikan data berupa titik-titik koordinat dan titik awal rute perjalanan.

Perhitungan bobot berupa jarak masih belum optimal karena hanya menggunakan jarak berupa garis lurus menggunakan persamaan harvesine. Jarak sebenarnya akan lebih jauh daripada jarak yang diberikan karena jalan yang ditempuh tidak selalu lurus dan juga terdapat banyak persimpangan jalan.

Walaupun perhitungan bobot masih belum sempurna, hasil dari algoritma *Branch and Bound* untuk menyelesaikan persoalan *Travelling Salesman Problem* tetap dapat dijadikan

rujukan untuk menentukan rute halalbihalal saat hari raya Idulfitri.

TAUTAN VIDEO DI YOUTUBE

<https://youtu.be/gINXvHkl4RY>

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan kepada kehadiran Allah SWT, karena berkat rahmat, hidayah dan karunia-Nya maka penulis dapat menyelesaikan makalah ini tepat pada waktunya.

Penulis juga mengucapkan terimakasih sebesar-besarnya kepada semua pihak yang telah turut membantu penulis dalam menyelesaikan makalah ini. Terimakasih banyak kepada Bapak Ir. Rila Mandala, M.Eng., Ph.D. selaku dosen pengampu mata kuliah IF2211 Strategi Algoritma kelas K1, dan juga kepada seluruh tim pengajar mata kuliah IF2211 yang telah banyak memberikan ilmunya kepada penulis hingga penulis mampu menyelesaikan makalah ini.

Terimakasih juga kepada seluruh teman-teman penulis yang selalu memberikan dorongan motivasi dan memberikan

saran-saran kepada penulis dalam mengerjakan tugas makalah ini.

REFERENSI

- [1] R. Munir, "Diktat Kuliah IF2211 Strategi Algoritma ITB", 2020, diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/stmik.htm> pada 9 Mei 2021.
- [2] L. Anany, "Introduction to the Design and Analysis of Algorithms 3th Edition", Pearson, 2011.

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, 10 Mei 2021



Muhammad Azhar Faturahman - 13519020