

Penerapan Algoritma Branch and Bound dalam Menyelesaikan Travelling Salesman Problem (TSP) untuk Menentukan Rute Perjalanan 11 Kantor Camat di Kota Padang

Fachry Dennis Herald - 13520139
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520139@std.stei.itb.ac.id

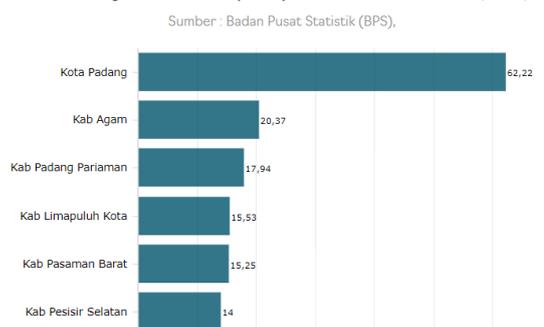
Abstrak—Kota Padang merupakan Ibu Kota Provinsi Sumatra Barat yang secara administratif dibagi menjadi 11 kecamatan. Kota ini menjadi pusat perekonomian dan administrasi untuk wilayah Sumatra Barat. Untuk mengoptimisasi kegiatan perekonomian dan administrasi, diperlukan rute yang dapat mengunjungi semua kecamatan yang diwakilkan oleh kantor camat tepat sekali dan kembali pada kecamatan yang menjadi awal dari perjalanan. Permasalahan optimisasi rute ini dikenal dengan nama Travelling Salesman Problem (TSP). Salah satu metode untuk menyelesaikan persoalan tersebut adalah dengan menggunakan algoritma Branch and Bound dengan metode heuristik matriks ongkos-tereduksi.

Kata kunci—kota padang, kecamatan, travelling salesman problem, branch and bound

I. PENDAHULUAN

Kota Padang merupakan kota yang berperan besar dalam menopang perekonomian di Provinsi Sumatra Barat. Perekonomian terbesar di Kota Padang ditopang oleh sektor perdagangan besar dan eceran serta transportasi dan pergudangan. Menurut Badan Pusat Statistik (BPS), produk domestik regional bruto (PDRB) atas dasar harga berlaku (ADHB) bernilai sebesar Rp62,22 triliun pada 2020. PDRB sektor perdagangan besar dan eceran berkontribusi sebesar Rp10,21 triliun atau 16,41% dari total PDRB tahun 2020. Sementara itu untuk sektor transportasi dan pergudangan berkontribusi sebesar Rp9,74 triliun atau 15,66% dari total PDRB 2020 [1]. Dengan angka tersebut, Kota Padang menempati posisi PDRB pertama yang kemudian diikuti oleh Kabupaten Agam dengan PDRB sebesar 20,37 triliun dan Kabupaten Padang Pariaman dengan sebesar PDRB 17,94 triliun. Dapat terlihat bahwa besar PDRB Kota Padang 3-3,5 kali lipat dibandingkan dengan Kabupaten Agam yang menempati posisi kedua dan Kabupaten Padang Pariaman yang menempati posisi ketiga. Perbedaan yang cukup besar ini menjadikan Kota Padang berpengaruh besar dalam PDRB untuk Provinsi jika dirata-ratakan berdasarkan kota dan kabupaten.

PDRB Harga Berlaku Kabupaten/Kota di Sumatera Barat (2020)



Gambar 1. Histogram PDRB Harga Berlaku Kabupaten/Kota di Sumatera Barat (2020) [1].

Berdasarkan data yang telah dipaparkan, kegiatan perekonomian di Kota Padang perlu dioptimisasi karena berperan cukup besar untuk wilayah Provinsi Sumatra Barat. Untuk itu, perlu dioptimisasi karena dengan teroptimisasinya bagian perekonomian ini, sektor lain juga akan berdampak baik. Salah satu komponen perekonomian yang dapat dioptimisasi adalah dalam kegiatan pendistribusian barang. Pendistribusian harus dilakukan secara optimal yaitu ketika seluruh wilayah mendapat pasokan barang sesuai dengan permintaan dengan biaya operasi yang minimum. Untuk meminimisasi biaya operasi, dapat dioptimalkan pada bagian transportasi karena sebagian besar kegiatan distribusi berkaitan dengan transportasi. Optimisasi yang dapat dilakukan adalah dengan mengoptimisasi jarak yang diperlukan untuk mengunjungi setiap bagian wilayah di Kota Padang yang diwakilkan oleh kantor camat.

Mencari rute yang memiliki jarak optimal (terpendek) dalam mengunjungi tiap kecamatan tepat sekali dan kembali pada kecamatan yang menjadi awal dari perjalanan analog dengan persoalan yang dikenal dengan nama Travelling Salesman Problem (TSP). Persoalan ini cukup populer dan dapat diselesaikan dengan berbagai pendekatan. Salah satu pendekatan yang dapat digunakan adalah dengan metode

Branch and Bound. Dengan menerapkan pendekatan dengan metode Branch and Bound, diharapkan dapat memberikan rute perjalanan yang optimal dalam mengunjungi 11 kantor camatan di Kota Padang. Dengan rute yang optimal ini, tentu dapat mengurangi ongkos dan juga menghemat waktu transportasi dalam kegiatan distribusi yang teroptimisasi. Dengan kegiatan distribusi yang teroptimisasi pula, sektor perekonomian mendapat dampak yang positif pula.

II. LANDASAN TEORI

A. Algoritma Branch and Bound (B&B)

Algoritma Branch and Bound (B&B) adalah salah satu algoritma yang umum digunakan untuk persoalan-persoalan optimasi, yaitu persoalan yang meminimumkan atau memaksimumkan suatu fungsi objektif selama tidak melanggar batasan (*constraints*) persoalan [2]. Sesuai dengan namanya, terdapat dua proses dasar pada algoritma ini, yaitu *branching* dan *bounding*. *Branching* adalah proses melakukan pencabangan dengan membangkitkan simpul anak pada pohon ruang status dari simpul yang diekspansi, setiap simpul anak adalah variasi kemungkinan yang mungkin dari simpul induknya dan memiliki bobot yang merepresentasikan biaya yang dibutuhkan untuk mencari simpul tersebut dari simpul asalnya. Sementara itu, *Bounding* adalah proses untuk melakukan pemangkasan jalur dengan mengeliminasi simpul yang tidak mengarah pada solusi. Pemangkasan jalur ini menggunakan fungsi pembatas, yaitu fungsi yang mempertimbangkan keberadaan simpul dengan membandingkan nilai fungsi obyektif dengan solusi terbaik saat ini, yang terbaik yang diambil.

Branch and Bound dapat dipandang sebagai Breadth First Search (BFS) dengan pendekatan *least cost search*. Perbedaan antara Branch and Bound dengan Breadth First Search murni adalah pada Breadth First Search murni, simpul berikutnya yang akan diekspansi berdasarkan urutan pembangkitannya, singkatnya First In First Out (FIFO). Sedangkan pada Branch and Bound, simpul-simpul tetap dibangkitkan sesuai dengan aturan Breadth First Search, namun setiap simpul diberi sebuah nilai *cost* yang disimbolkan dengan $\hat{c}(i)$, yaitu nilai taksiran lintasan termurah ke simpul status tujuan (*goal*) yang melalui simpul status i . Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan urutan pembangkitannya, tetapi simpul yang memiliki *cost* yang paling kecil (*least cost search*) untuk kasus minimasi [2]. Sederhananya, prinsip Branch and Bound serupa dengan antrian dengan prioritas (*priority queue*).

Algoritma Branch and Bound juga memiliki perbedaan dan persamaan dengan Algoritma Runut-balik (Backtracking). Perbedaannya dari Branch and Bound dengan Backtracking terletak pada cara pembangkitan simpul dan jenis persoalan yang dapat diselesaikan. Untuk cara pembangkitan simpul, pada Backtracking, umumnya simpul dibangkitkan dengan metode yang serupa dengan Depth First Search (DFS), sedangkan pada Branch and Bound memiliki beberapa aturan pembangkitan simpul tertentu, yang paling umum adalah *best-first rule*. Untuk jenis persoalan yang dapat diselesaikan, pada Backtracking umumnya digunakan untuk persoalan non-optimasi sehingga tidak ada batasan, sedangkan pada Branch and Bound umumnya digunakan untuk persoalan optimasi

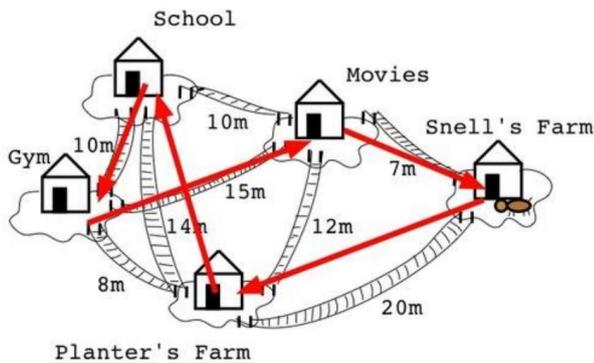
karena terdapat nilai dari solusi terbaik sejauh ini dengan setiap simpul pada pohon ruang-status, diperlukan suatu cara penentuan batas nilai terbaik fungsi objektif pada setiap solusi yang mungkin dengan menambahkan komponen pada solusi sementara yang direpresentasikan oleh simpul. Sementara itu, terdapat persamaan antara Branch and Bound dengan Backtracking yaitu pencarian solusi dilakukan dengan membentuk pohon ruang status, kemudian mengeliminasi simpul yang tidak mengarah kepada solusi yang diharapkan [2].

Adapun algoritma Branch and Bound secara umum adalah sebagai berikut.

1. Simpul akar dimasukkan kedalam antrian Q. Jika simpul akar adalah simpul tujuan (*goal*), maka proses berhenti.
2. Jika antrian Q kosong, maka proses berhenti.
3. Jika Q tidak kosong, maka pilih dari antrian Q simpul i yang memiliki *cost* $\hat{c}(i)$ paling kecil. Jika terdapat beberapa simpul i yang memenuhi, pilih salah satu secara sembarang.
4. Jika simpul i adalah simpul solusi, berarti solusi telah ditemukan. Jika hanya menginginkan satu solusi, maka proses berhenti. Untuk persoalan optimasi dengan pendekatan *least cost search*, periksa *cost* semua simpul hidup. Jika *cost* nya lebih besar dari *cost* simpul solusi, maka eliminasi simpul tersebut.
5. Jika simpul i bukan simpul solusi, maka ekspansi simpul tersebut dengan membangkitkan semua anak-anaknya. Jika simpul i tidak mempunyai anak, maka kembali ke langkah 2.
6. Untuk setiap anak j dari simpul i , hitung *cost* $\hat{c}(j)$, masukkan semua anak-anak tersebut ke dalam antrian Q.
7. Kembali ke langkah 2 [2].

B. Travelling Salesman Problem (TSP)

Travelling Salesman Problem (TSP) merupakan salah satu permasalahan populer yang berkaitan dengan aplikasi dari teori graf. Bunyi persoalannya adalah sebagai berikut. "Diberikan n buah kota yang direpresentasikan sebagai simpul graf serta diketahui jarak (bobot) antara setiap kota satu sama lain. Akan dicari perjalanan berupa sirkuit dengan jarak terpendek yang dilakukan oleh seorang pedagang sehingga ia melalui setiap kota tepat hanya sekali dan kembali lagi ke kota asal keberangkatan [2]." Berdasarkan bunyi dari persoalan tersebut dapat disimpulkan bahwa Travelling Salesman Problem merupakan permasalahan mencari sirkuit Hamilton dengan jumlah bobot yang paling kecil. Pada graf lengkap terdapat $(n-1)!$ tur di dalam graf lengkap dengan n simpul dan terdapat sejumlah $((n-1)/2)!$ sirkuit Hamilton. Persoalan ini tidak terbatas hanya untuk graf lengkap saja, namun juga berlaku juga untuk graf yang tidak lengkap selama memiliki sirkuit Hamilton. Berikut ini adalah ilustrasi dari TSP.



Gambar 2. Contoh Ilustrasi Travelling Salesman Problem (TSP) [2].

Hingga saat ini, belum ada algoritma yang mangkus untuk menyelesaikan TSP untuk sembarang n buah kota. TSP pada dasarnya merupakan sebuah persoalan dengan kompleksitas yang besar sehingga digolongkan kedalam persoalan NP-hard dalam optimisasi kombinatorial dan cukup penting dalam ilmu komputer. Oleh karena itu, banyak pendekatan yang bisa digunakan untuk memecahkan TSP. Secara naif memang dapat diselesaikan dengan algoritma Brute Force dengan konsekuensi kompleksitas yang besar. Alih-alih menggunakan Brute Force, lebih baik menggunakan metode heuristik agar didapatkan algoritma yang lebih baik untuk memecahkan masalah ini, yaitu dapat menggunakan algoritma Branch and Bound.

C. Penerapan Algoritma Branch and Bound pada Travelling Salesman Problem (TSP) Menggunakan Matriks Ongkos-Tereduksi (Reduced Cost Matrix)

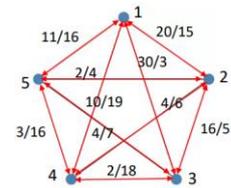
Pada penyelesaian TSP dengan Algoritma Branch and Bound, *cost* untuk setiap simpul di dalam pohon ruang status menyatakan nilai batas bawah (*lower bound*). *Cost* setiap simpul dapat dihitung secara heuristik menggunakan Matriks ongkos-tereduksi (*reduced cost matrix*) dari graf. Matriks ongkos-tereduksi adalah matriks ketetanggaan dari graf yang sudah dilakukan proses reduksi sehingga setiap kolom dan barisnya mengandung paling sedikit satu buah nol dan semua elemen lainnya nonnegatif. Pada kasus ini, matriks ketetanggaan graf berisi elemen yang menyatakan bobot dari tiap sisi. Jika tidak terdapat sisi yang menghubungkan dua node, set elemen matriks dengan nilai tak hingga. Sederhananya, sebuah matriks dikatakan tereduksi jika setiap kolom dan barisnya mengandung paling sedikit satu buah nol, dan semua elemen lainnya non-negatif. Perhatikan contoh berikut ini.



Gambar 3. Reduksi matriks M menjadi matriks tereduksi M' [2].

Tinjau sebuah TSP dengan $n = 5$, graf dinyatakan dalam matriks ketetanggaan.

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



Gambar 4. Matriks Ketetanggaan dari Graf yang Merepresentasikan TSP dengan $n = 5$ [2].

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

$$\begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix} \begin{matrix} C_1 - 1 \\ C_2 - 3 \end{matrix} \begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix} = A$$

Gambar 5. Proses Reduksi Matriks [2].

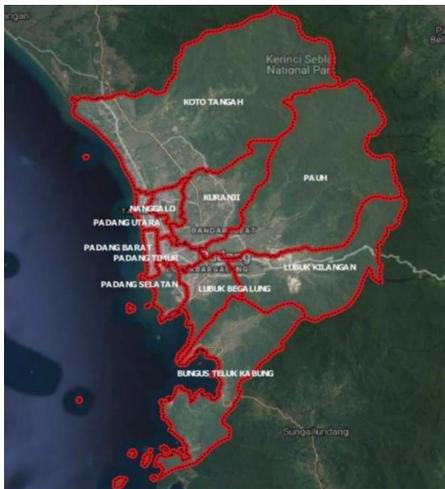
Untuk membuat setiap baris dan kolom memiliki paling sedikit satu 0, kurangi setiap baris pada matriks ketetanggaan dengan elemen terkecil pada baris tersebut. Selanjutnya, lakukan hal yang sama pada setiap kolom matriks. Setiap nilai terkecil dari baris dan kolom dijumlahkan dan dicatat sebagai *cost* dari simpul akar. Secara umum, 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.
6. Proses dilakukan pada semua simpul anak yang memungkinkan. Setelah selesai, lanjutkan dengan pemilihan simpul aktif dengan *cost* minimum.

III. IMPLEMENTASI DAN EKSPERIMEN

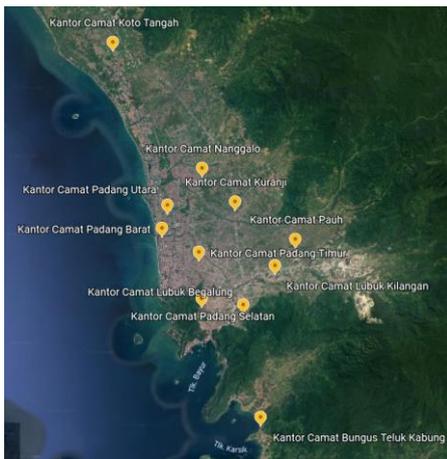
A. Pengumpulan Data dan Pemodelan Masalah

Dalam Rencana Tata Ruang Wilayah Nasional telah ditetapkan Kota Padang sebagai Pusat Kegiatan Nasional (PKN) disamping itu Kota Padang juga sebagai ibukota dan pusat pendidikan tinggi di Provinsi Sumatera Barat. Berdasarkan PP Nomor 17 tahun 1980, secara administratif luas wilayah Kota Padang adalah 165,35 km dan Kota Padang memiliki 11 Kecamatan [3].



Gambar 6. Peta Wilayah Kota Padang [4].

Selanjutnya, menggunakan bantuan perangkat lunak web Google Earth, dapat ditentukan titik-titik yang membangun rute perjalanan. Titik-titik yang ditentukan adalah 11 kantor camat di Kota Padang sebagai berikut.



Gambar 7. Titik-titik Lokasi 11 Kantor Kecamatan di Kota Padang.

Berdasarkan lokasi titik tersebut, dapat ditentukan jarak antar kantor camat. Informasi jarak antar kantor camat dicari dengan bantuan Google Maps. Jarak yang diambil adalah jarak minimum dari rute jalur-jalur yang mungkin. Untuk

memudahkan dalam pengolahan data, nama dari kecamatan diganti dengan indeks sebagai berikut.

TABEL I. INDEKS KECAMATAN KOTA PADANG

Kecamatan	Indeks
Koto Tangah	1
Pauh	2
Kuranji	3
Nanggalo	4
Padang Utara	5
Padang Barat	6
Padang Timur	7
Padang Selatan	8
Lubuk Begalung	9
Lubuk Kilangan	10
Bungus Teluk Kabung	11

Berikut ini adalah matriks ketetangaan jarak antar kantor camat. Angka direpresentasikan dan dibulatkan dalam satuan kilometer (km).

TABEL II. MATRIKS KETETANGGAN JARAK ANTAR KANTOR CAMAT KOTA PADANG

	1	2	3	4	5	6	7	8	9	10	11
1	∞	22.5	16.3	13.8	13.7	14.6	18.5	21.9	24.2	23.1	37.8
2	22.5	∞	7.0	10.6	11.3	11.6	7.8	9.3	7.6	3.8	24.1
3	16.3	7.0	∞	4.0	6.5	6.9	5.2	8.5	8.0	6.9	24.4
4	13.8	10.6	4.0	∞	5.7	6.7	9.0	12.4	10.2	10.8	28.3
5	13.7	11.3	6.5	5.7	∞	2.3	6.5	8.8	12.9	11.9	25.7
6	14.6	11.6	6.9	6.7	2.3	∞	4.5	6.9	11.1	10.0	23.8
7	18.5	7.8	5.2	9.0	6.5	4.5	∞	3.4	7.0	6.0	20.9
8	21.9	9.3	8.5	12.4	8.8	6.9	3.4	∞	5.4	7.4	16.9
9	24.2	7.6	8.0	10.2	12.9	11.1	7.0	5.4	∞	6.1	17.9
10	23.1	3.8	6.9	10.8	11.9	10.0	6.0	7.4	6.1	∞	22.2
11	37.8	24.1	24.4	28.3	25.7	23.8	20.9	16.9	17.9	22.2	∞

B. Penyelesaian Travelling Salesman Problem (TSP) dengan Algoritma Branch and Bound Menggunakan Cara Matriks Ongkos-Tereduksi (Reduced Cost Matrix)

Data matriks ketetangaan yang telah didapatkan akan diolah menggunakan program sederhana untuk menyelesaikan TSP yang ditulis dalam bahasa Python. Berikut ini adalah tangkapan layar masukan dan keluaran dari program setelah dijalankan.

```

Program Penyelesaian Travelling Salesman Problem dengan Metode Branch and Bound
Masukan nama file (.txt) : input.txt
Masukan simpul awal keberangkatan : 1

Matriks ketetangaan yang terbaca:
inf 22.5 16.3 13.8 13.7 14.6 18.5 21.9 24.2 23.1 37.8
22.5 inf 7.0 10.6 11.3 11.6 7.8 9.3 7.6 3.8 24.1
16.3 7.0 inf 4.0 6.5 6.9 5.2 8.5 8.0 6.9 24.4
13.8 10.6 4.0 inf 5.7 6.7 9.0 12.4 10.2 10.8 28.3
13.7 11.3 6.5 5.7 inf 2.3 6.5 8.8 12.9 11.9 25.7
14.6 11.6 6.9 6.7 2.3 inf 4.5 6.9 11.1 10.0 23.8
18.5 7.8 5.2 9.0 6.5 4.5 inf 3.4 7.0 6.0 20.9
21.9 9.3 8.5 12.4 8.8 6.9 3.4 inf 5.4 7.4 16.9
24.2 7.6 8.0 10.2 12.9 11.1 7.0 5.4 inf 6.1 17.9
23.1 3.8 6.9 10.8 11.9 10.0 6.0 7.4 6.1 inf 22.2
37.8 24.1 24.4 28.3 25.7 23.8 20.9 16.9 17.9 22.2 inf
    
```

Gambar 8. Tangkapan Layar Masukan Matriks Ketetangaan.

Gambar 8 menunjukkan matriks ketetangaan yang akan direduksi. Simpul awal keberangkatan yang dipilih adalah simpul 1 yang merepresentasikan Kantor Camat Koto Tangah.

Urutan baris dan kolom merepresentasikan simpul kantor camat sesuai dengan indeks yang telah ditentukan pada Tabel I. Elemen pada baris ke- i pada kolom ke- j mengindikasikan jarak tempuh perjalanan dari simpul i ke simpul j . Elemen yang bernilai inf berguna untuk mencegah penggunaan elemen tersebut untuk dihitung ketika mereduksi matriks.

Mula-mula, proses awal reduksi matriks pada simpul keberangkatan mendapatkan hasil sebagai berikut.

```
Bobot tereduksi: 86.3 | Simpul yang ditempuh: [1]
inf 8.8 2.6 0.1 0.0 0.9 4.8 8.2 9.5 9.4 11.6
8.9 inf 3.2 6.8 7.5 7.8 4.0 5.5 2.8 0.0 7.8
2.5 3.0 inf 0.0 2.5 2.9 1.2 4.5 3.0 2.9 7.9
0.0 6.6 0.0 inf 1.7 2.7 5.0 8.4 5.2 6.8 11.8
1.6 9.0 4.2 3.4 inf 0.0 4.2 6.5 9.6 10.9
2.5 9.3 4.6 4.4 0.0 inf 2.2 4.6 7.8 7.7 9.0
5.3 4.4 1.8 5.6 3.1 1.1 inf 0.0 2.6 2.6 5.0
8.7 5.9 5.1 9.0 5.4 3.5 0.0 inf 1.0 4.0 1.0
9.0 2.2 2.6 4.8 7.5 5.7 1.6 0.0 inf 0.7 0.0
9.5 0.0 3.1 7.0 8.1 6.2 2.2 3.6 1.3 inf 5.9
11.1 7.2 7.5 11.4 8.8 6.9 4.0 0.0 0.0 5.3 inf
```

Gambar 9. Matriks Tereduksi untuk Simpul Awal Keberangkatan (1)

Didapatkan nilai bobot tereduksi dari simpul awal keberangkatan (simpul akar) sebesar 86.3 yang akan menjadi batas bawah (*lower bound*) dari persoalan ini. Selanjutnya matriks tersebut akan direduksi lagi sesuai dengan langkah pembangkitan simpul anak. Batas bawah dari persoalan ini juga menjadi perkiraan nilai bobot yang diperlukan untuk menyelesaikan persoalan ini. Berikut ini adalah cuplikan proses reduksi matriks, tidak semua proses yang ditampilkan, hanya proses yang mengarah kepada solusi persoalan.

```
Bobot tereduksi: 88.9 | Simpul yang ditempuh: [1, 4, 3]
inf inf inf inf inf inf inf inf inf inf
7.6 inf inf inf 7.5 7.8 4.0 5.5 2.8 0.0 7.8
0.0 1.8 inf inf 1.3 1.7 0.0 3.3 1.8 1.7 6.7
inf inf inf inf inf inf inf inf inf inf
0.3 9.0 inf inf inf 0.0 4.2 6.5 9.6 9.6 10.9
1.2 9.3 inf inf 0.0 inf 2.2 4.6 7.8 7.7 9.0
4.0 4.4 inf inf 3.1 1.1 inf 0.0 2.6 2.6 5.0
7.4 5.9 inf inf 5.4 3.5 0.0 inf 1.0 4.0 1.0
7.7 2.2 inf inf 7.5 5.7 1.6 0.0 inf 0.7 0.0
8.2 0.0 inf inf 8.1 6.2 2.2 3.6 1.3 inf 5.9
9.8 7.2 inf inf 8.8 6.9 4.0 0.0 0.0 5.3 inf
```

Gambar 10. Matriks Tereduksi Pembangkitan Simpul 1, 4, dan 3

```
Bobot tereduksi: 92.3 | Simpul yang ditempuh: [1, 4, 3, 2, 10, 9, 11]
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
0.0 inf inf inf inf 0.0 4.2 6.5 inf inf inf
0.9 inf inf inf 0.0 inf 2.2 4.6 inf inf inf
3.7 inf inf inf 3.1 1.1 inf 0.0 inf inf inf
7.1 inf inf inf 5.4 3.5 0.0 inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
9.5 inf inf inf 8.8 6.9 4.0 0.0 inf inf inf
```

Gambar 11. Matriks Tereduksi Pembangkitan Simpul 1, 4, 3, 2, 10, 9, dan 11

```
Bobot tereduksi: 93.4 | Simpul yang ditempuh: [1, 4, 3, 2, 10, 9, 11, 8, 7, 6, 5]
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
0.0 inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
inf inf inf inf inf inf inf inf inf inf
```

Gambar 12. Matriks Tereduksi untuk Pembangkitan Simpul 1, 4, 3, 2, 10, 9, 11, 8, 7, 6, dan 5.

Proses penyelesaian persoalan ini memanfaatkan struktur data Priority Queue (Antrian dengan prioritas). Setiap pembangkitan simpul anak yang bukan merupakan simpul daun akan dimasukkan ke dalam antrian. Urutan antrian didasarkan dengan nilai bobotnya, semakin kecil nilai bobotnya maka akan tinggi prioritasnya dan akan didahulukan untuk diekspansi. Proses ini juga dilakukan secara rekursif dan akan berhenti ketika semua elemen matriks bernilai inf yang artinya pembangkitan terakhir menghasilkan simpul anak yang merupakan simpul akar.

Pada Gambar 12, Nilai didapatkan nilai bobot tereduksi terakhir adalah 93.4 dan bobot ini menjadi bobot paling minimum dan semua simpul telah dibangkitkan. Setelah mencapai simpul 5, maka simpul yang terakhir dibangkitkan adalah simpul 1 yang artinya kembali ke simpul awal keberangkatan.

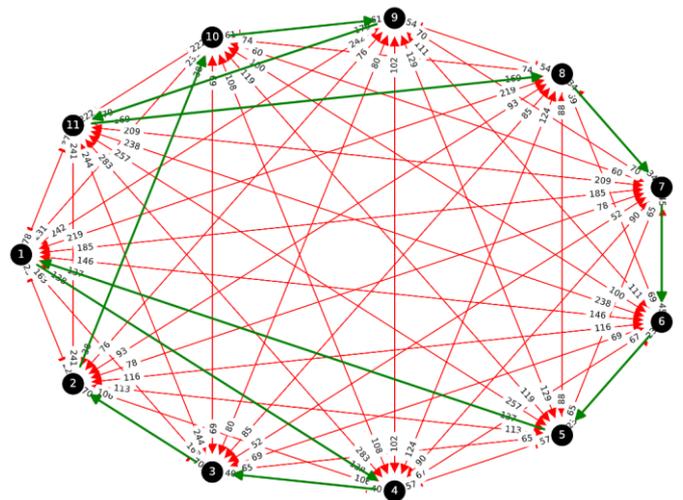
C. Hasil

Setelah proses reduksi matriks selesai dan seluruh simpul berhasil dibangkitkan, program menampilkan hasil sebagai berikut.

```
Total bobot perjalanan : 93.4
Simpul yang ditempuh : [1, 4, 3, 2, 10, 9, 11, 8, 7, 6, 5, 1]
Simpul yang dibangkitkan : 404
Waktu eksekusi : 0.4792 detik
```

Gambar 13. Hasil Akhir Perjalanan Program

Didapatkan total bobot perjalanan sebesar 93.4, simpul yang dibangkitkan sebanyak 404, dan waktu eksekusi program selama 0.4792 detik. Program juga memberikan visualisasi graf yang terbentuk dari simpul yang ditempuh sebagai berikut.



Gambar 14. Visualisasi Rute Simpul Yang Ditempuh

Simpul yang ditempuh kemudian ditranslasi ke nama kantor camat berdasarkan indeksnya sesuai dengan Tabel I. Maka didapatkan rute perjalanan sebagai berikut.

Koto Tengah → Nanggalo → Kuranji → Pauh → Lubuk Kilangan → Lubuk Begalung → Bungus Teluk Kabung → Padang Selatan → Padang Timur → Padang Barat → Padang Utara → Koto Tengah

Rute perjalan diatas memiliki jarak tempuh total sebesar 93,4 km. Peta visualisasi perjalanan jika dilihat melalui Google Earth adalah sebagai berikut.



Gambar 15. Peta Rute Perjalanan 11 Kantor Camat di Kota Padang

Selanjutnya dicari informasi jalan yang dilalui oleh rute tersebut menggunakan Google Maps dan dirincikan ke dalam tabel sebagai berikut.

TABEL III. INFORMASI JALAN YANG DILALUI RUTE

Rute	Jalan yang Dilalui
Koto Tengah – Nanggalo	Jl. Adinegoro
Nanggalo – Kuranji	Jl. Lolong Karan dan Jl. Bagindo Aziz Chan
Kuranji – Pauh	Jl. Korong Gadang
Pauh – Lubuk Kilangan	Jl. Raya Indarung
Lubuk Kilangan – Bungus Teluk Kabung	Jl. Lintas Barat Sumatera
Bungus Teluk Kabung – Padang Selatan	Jl. Lintas Barat Sumatera
Padang Selatan – Padang Timur	Jl. Sutan Syahrir
Padang Timur – Padang Barat	Jl. Jend. Sudirman
Padang Barat – Padang Utara	Jl. Ir. H. Juanda dan Jl. Beringin Raya
Padang Utara – Koto Tengah	Jl. Prof. Dr. Hamka dan Jl. Adinegoro

IV. KESIMPULAN

Dengan menerapkan Algoritma Branch and Bound dapat diselesaikan Travelling Salesman Problem untuk menentukan rute perjalanan mengunjungi 11 kecamatan yang ada di Kota Padang dengan metode matriks ongkos-tereduksi (reduced cost matrix). Mula-mula, dikumpulkan informasi mengenai jarak

antar kantor camat lalu ditranslasi ke dalam bentuk matriks ketetanggaan. Matriks ketetanggaan tersebut kemudian dapat diolah menggunakan program sederhana untuk menyelesaikan TSP. Program akan melakukan operasi perhitungan untuk mereduksi matriks ketetanggaan yang dimasukkan, kemudian menjalankan Algoritma Branch and Bound yang bekerja secara rekursif untuk menghitung bobot minimum ketika mengekspansi simpul. Program kemudian memberikan hasil berupa total bobot perjalanan sebesar 93,4, rute yang terbentuk, simpul yang dibangkitkan sebanyak 404, waktu eksekusi program selama 0.4792 detik, serta visualisasi graf urutan perjalanan pengunjungan simpul. Adapun rute yang terbentuk adalah sebagai berikut.

Koto Tengah → Nanggalo → Kuranji → Pauh → Lubuk Kilangan → Lubuk Begalung → Bungus Teluk Kabung → Padang Selatan → Padang Timur → Padang Barat → Padang Utara → Koto Tengah

dengan rincian jalan yang dilalui sebagai berikut.

Jl. Adinegoro – Jl. Lolong dan Jl. Bagindo Aziz Chan – Jl. Korong Gadang – Jl. Lintas Barat Sumatera – Jl. Sutan Syahrir – Jl. Jend. Sudirman – Jl. Ir. H. Juanda dan Jl. Beringin Raya – Jl. Prof. Dr. Hamka dan Jl. Adinegoro.

Dengan adanya rancangan rute ini, diharapkan dapat membantu pihak yang membutuhkan alternatif untuk mengurangi ongkos yang diperlukan dalam mengunjungi setiap kantor camat di Kota Padang. Dengan mengurangi ongkos tersebut, dapat ditunjang pula kegiatan ekonomi di Kota Padang.

TAUTAN VIDEO DI YOUTUBE

Sebagai pendukung dari pembuatan makalah ini dan untuk menambah pemahaman pembaca, penulis telah membuat video youtube yang dapat dilihat dalam tautan berikut:

<https://youtu.be/usJMLM9oyKs>

UCAPAN TERIMA KASIH

Segala puji dan syukur penulis panjatkan ke hadirat Allah Swt. Atas berkat rahmat dan karunia-Nya proyek makalah ini dapat selesai sesuai pada waktunya.

Ucapan terima kasih penulis sampaikan kepada Ibu Dr. Masayu Leylia Khodra, S.T., M.T. selaku dosen pengajar Mata Kuliah IF2211 Strategi Algoritma Kelas 01 dan tak terbatas pula untuk seluruh tim pengajar termasuk asisten. Atas bantuan beliau, penulis dapat banyak belajar dan mudah-mudahan ilmu yang penulis dapatkan dapat bermanfaat pula bagi orang lain melalui makalah ini.

Terakhir, penulis tak lupa pula mengucapkan terima kasih kepada orang tua, saudara, dan kolega atas dorongan, dukungan, serta saran untuk penulis dalam pengerjaan makalah ini.

REFERENSI

- [1] K. Viva Budy, "Perekonomian Kota Padang Terbesar di Sumatera Barat pada 2020", databooks, Katadata, 2021.
- [2] R. Munir, N.U. Maulidevi, dan M.L. Khodra, "Bahan Kuliah IF2211 Strategi Algoritma", Program Studi Teknik Informatika, 2021.
- [3] <https://padang.go.id/gambaran-umum-kota-padang>, diakses pada 14 Mei 2022 pukul 2.49.

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, 20 Mei 2022



Fachry Dennis Heraldi (13520139)