

Penerapan Algoritma *Branch and Bound* pada Permasalahan *Travelling Salesman Problem* untuk Menentukan Rute Terdekat antar Kecamatan di Jakarta Pusat

Fayza Nadia - 13520001

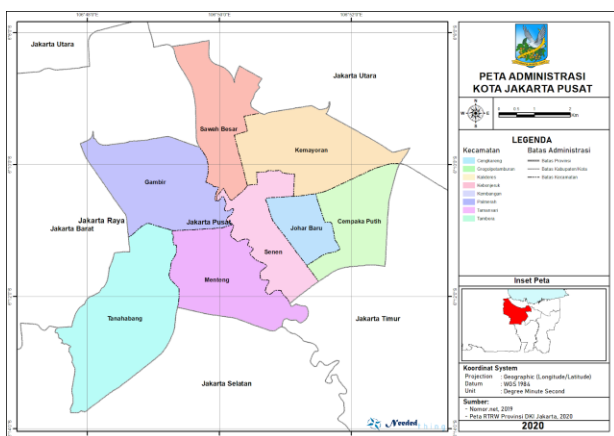
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520001@std.stei.itb.ac.id

Abstrak—Mobilisasi merupakan kegiatan yang rutin dilakukan masyarakat setiap harinya. Jakarta Pusat, sebagai jantung dari ibu kota Indonesia, memiliki kepadatan penduduk yang cukup tinggi, berakibat pada padatnya jalanan serta transportasi umum yang tersedia. Pemilihan rute yang efisien dan efektif agar dapat sampai ke tujuan menjadi hal krusial yang harus direncanakan setiap harinya. Algoritma *Branch and Bound* menjadi salah satu solusi yang dapat diterapkan untuk menyelesaikan tipe permasalahan seperti ini, disebut juga dengan *Travelling Salesman Problem*.

Keywords—*branch and bound*; mobilisasi; rute; travelling salesman problem;

I. PENDAHULUAN

Dalam kehidupan sehari-hari, masyarakat melakukan mobilisasi baik antarkota maupun di dalam kota. Seringkali masyarakat bermobilisasi dengan destinasi yang banyak dalam satu waktu. Perencanaan rute yang baik menjadi salah satu hal krusial yang harus diperhatikan bagi masyarakat, mengingat Kota Jakarta yang erat dengan kemacetan di sebagian besar ruas jalanan.



Gambar 1. Peta Wilayah Administrasi Jakarta Pusat

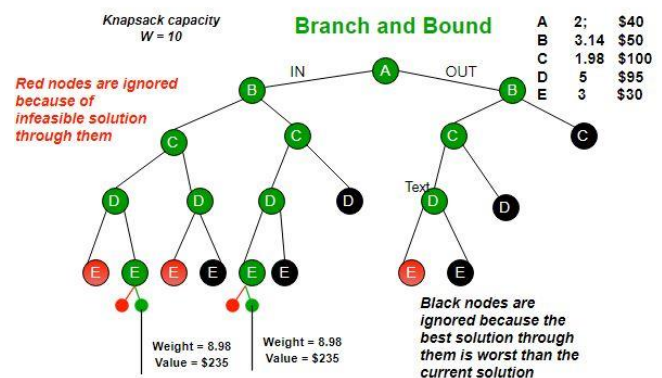
(sumber: <https://1.bp.blogspot.com/-EUhmKtVP6E8/X4mhgCv8HNI/AAAAAAAAAKWg/yP6-gYB2UyU-EV7vKM3ikrSdDK6jWEKEQCLcBGAsYHQ/s2048/Jakpus.png> diakses pada 22 Mei 2022 pukul 05.45)

Kota Jakarta sendiri terbagi menjadi lima wilayah administrasi. Salah satu dari wilayah tersebut adalah Kota Administrasi Jakarta Pusat. Sebagai pusat dari ibu kota Indonesia, Jakarta Pusat memiliki banyak destinasi yang tersebar di masing-masing kecamatannya. Hal ini tidak menutup kemungkinan adanya masyarakat yang melakukan mobilisasi ke banyak destinasi tersebar di kecamatan yang terletak di Jakarta Pusat.

Pemilihan rute dalam mengelilingi kecamatan yang terletak di Kota Administrasi Jakarta Pusat dapat dilakukan dengan menggunakan algoritma *branch and bound*. Dengan algoritma tersebut, dapat dihasilkan pilihan rute dengan tingkat efisiensi sebaik mungkin dalam melewati setiap kecamatan di Jakarta Pusat. Efisiensi tersebut dapat menghemat waktu tempuh yang dihabiskan saat mobilisasi.

II. LANDASAN TEORI

A. Algoritma *Branch and Bound*



Gambar 2. Ilustrasi Algoritma *Branch and Bound*

(sumber: <https://media.geeksforgeeks.org/wp-content/uploads/knapsack3.jpg> diakses pada 22 Mei 2022 pukul 05.51)

Algoritma *Branch and Bound* adalah algoritma yang umum digunakan dalam permasalahan optimisasi. Optimisasi dalam hal ini adalah masalah di mana terdapat suatu fungsi objektif yang dimaksimalkan ataupun diminimalkan dengan memperhatikan batasan atau *constraint* tertentu yang telah ditetapkan.

Branch and Bound mencari solusi dari suatu permasalahan dengan membentuk pohon ruang status atau *state space tree*, sama halnya dengan algoritma *graph traversal* lain seperti BFS (*Breadth First Search*), DFS (*Depth First Search*), DLS (*Depth Limited Search*), IDS (*Iterative Deepening Search*), serta *backtracking*. Pada dasarnya, algoritma *branch and bound* merupakan algoritma BFS dengan ekspansi simpul dari simpul dengan *cost* terkecil terlebih dahulu atau dikenal juga dengan istilah *least cost search* pada kasus minimasi dan berlaku sebaliknya pada kasus maksimasi.

Perhitungan *cost* dari masing-masing simpul dapat dihitung dengan persamaan berikut

$$c(i) = f(i) + g(i)$$

$c(i)$ = bobot perjalanan simpul i

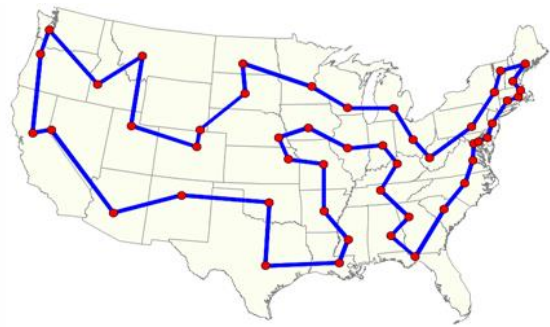
$f(i)$ = bobot dari simpul akar ke simpul i

$g(i)$ = bobot dari simpul i ke simpul tujuan

Secara tahap per tahap, algoritma ini memiliki tahapan sebagai berikut:

1. Memasukkan simpul akar ke dalam sebuah antrian atau *queue* Q . Algoritma berakhir jika simpul tersebut merupakan simpul solusi (*goal node*) dan hanya terdapat satu simpul solusi yang diinginkan.
2. Jika *queue* Q kosong, algoritma berhenti.
3. Jika *queue* Q tidak kosong, pilih simpul i pada *queue* Q yang memiliki nilai *cost* $c(i)$ terkecil. Jika terdapat lebih dari satu simpul, maka pilihlah salah satunya secara sembarang.
4. Algoritma berakhir jika simpul i merupakan simpul solusi dan hanya terdapat satu solusi yang diinginkan. Pada kasus minimasi, periksa *cost* dari seluruh simpul hidup. Jika *cost* simpul solusi lebih kecil dari *cost* simpul yang dibangkitkan, maka simpul tersebut akan dimatikan.
5. Jika simpul i tidak memiliki simpul anak, kembali ke langkah 2. Namun, jika simpul i bukan merupakan simpul solusi, maka simpul anak akan dibangkitkan.
6. Menghitung nilai *cost* $c(j)$ setiap simpul anak j dari simpul i dan memasukkan seluruh simpul anak j ke dalam *queue* Q .
7. Kembali ke langkah 2.

B. Travelling Salesman Problem (TSP)



Gambar 4. Travelling Salesman Problem (TSP)

(sumber: https://physics.aps.org/assets/a38de7c6-00ac-45fb-9bcf-3b3e14a72b41/es32_1.png diakses pada 22 Mei 2022 pukul 12.00)

Travelling Salesman Problem (TSP) adalah permasalahan matematis di mana diberikan sejumlah simpul, dalam hal ini adalah kota, serta jarak sebagai bobot antarkota. Tujuan utama dari TSP adalah untuk menemukan jarak terpendek yang dapat ditempuh jika melalui setiap simpul tepat sekali dengan tujuan simpul akhir merupakan simpul awal pula.

Pemecahan masalah TSP dapat dilakukan dengan berbagai cara, seperti algoritma *brute force*, *greedy*, *branch and bound*, dan lain-lain. Penyelesaian TSP dengan *branch and bound* diawali dengan membuat matriks bobot atau *cost matrix* dari jarak-jarak antarkota. Namun, kedua simpul atau kota yang tidak saling terhubung akan dituliskan sebagai tak hingga (∞). Setiap simpul kemudian dihitung *cost*-nya secara heuristik berdasarkan salah satu dari dua cara, yaitu matriks ongkos-tereduksi (*reduced cost matrix*) atau bobot minimum tur lengkap.

Perhitungan *cost* atau nilai batas dengan *reduced cost matrix* diawali dengan mereduksi matriks pemetaan graf G hingga setiap kolom dan barisnya memiliki minimal satu buah nol dan elemen positif atau nol pada elemen lainnya. Reduksi matriks dapat dilakukan berulang kali hingga dihasilkan matriks bobot tereduksi.

M					M'				
12	20	30	10	11	1	10	17	0	1
15	8	16	4	2	12	6	11	2	0
3	5	11	2	4	0	3	6	0	2
19	6	18	9	3	15	3	12	6	0
16	4	7	16	8	11	0	0	12	4



Gambar 3. Proses Reduksi Matriks pada Reduced Cost Matrix

(sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian2.pdf> diakses pada 22 Mei 2022 pukul 07.47)

Persamaan fungsi pembatas (*bounding function*) dari metode *reduced cost matrix* adalah

$$c(S) = c(R) + A(i, j) + r$$

$c(S)$ = bobot perjalanan minimum simpul S

$c(R)$ = bobot perjalanan minimum simpul R

$A(i, j)$ = bobot sisi (i, j) pada graf G yang berkoresponden dengan sisi (R, S)

r = jumlah semua pengurang pada proses reduksi matriks

$Cost$ yang didapatkan tersebut kemudian dijadikan acuan dalam ekspansi simpul atau kota yang menjadi simpul solusi. Dalam hal ini, $cost$ dengan bobot terkecil menjadi prioritas utama dalam ekspansi simpul.

Sedangkan, berbeda dengan metode matriks ongkos-tereduksi, metode bobot minimum tur lengkap tidak memerlukan matriks. Perhitungan dilakukan dengan persamaan bobot tur lengkap yaitu

$$M \equiv cost = \text{bobot minimum tur lengkap} \\ \geq \frac{1}{2} \sum \text{bobot sisi } i_1 + \text{bobot sisi } i_2$$

$Cost$ yang didapatkan kemudian dipetakan ke dalam bentuk pohon ruang status. Seterusnya, perhitungan kemudian diulang hingga semua simpul sudah dilewati atau dihitung.

C. Kota Administrasi Jakarta Pusat



Gambar 5. Jakarta Pusat

(sumber: <https://www.javatrans.com/wp-content/uploads/2020/10/Tempat-Wisata-Jakarta-Pusat.jpg> diakses pada 22 Mei 2022 pukul 11.58)

Berdiri sejak tahun 1527, Provinsi DKI Jakarta memiliki enam wilayah administrasi, yaitu adalah Kabupaten Administrasi Kepulauan Seribu, Kota Administrasi Jakarta Barat, Kota Administrasi Jakarta Selatan, Kota Administrasi Jakarta Timur, Kota Administrasi Jakarta Utara, dan Kota Administrasi Jakarta Pusat.

Sebagai wilayah terpusat dari Indonesia, Jakarta Pusat tergolong padat. Banyak perkantoran, mulai dari kantor pemerintahan, perwakilan negara asing, dan bisnis, yang menghuni Jakarta Pusat. Selain itu, banyak pula ditemukan tempat-tempat bersejarah yang mendukung sisi pariwisata dari Jakarta Pusat.

III. PEMBAHASAN

Agar dapat mengunjungi seluruh titik lokasi terpilih di Jakarta Pusat, kemudian dilakukan algoritma *branch and bound* untuk menyelesaikan permasalahan *Travelling Salesman Problem* (TSP) dengan metode matriks ongkos-tereduksi.

A. Pemilihan Titik Lokasi

Titik-titik lokasi sebagai simpul graf dipilih dari kecamatan yang berada di wilayah administrasi Jakarta Pusat. Berikut adalah titik lokasi yang diambil:

- Cempaka Putih (CP)
- Gambir (GB)
- Johar Baru (JB)
- Kemayoran (KM)
- Menteng (MT)
- Sawah Besar (SB)
- Senen (SN)

Gambir (GB) dipilih sebagai simpul awal dan juga simpul akhir rute karena terdapat stasiun kereta di daerah tersebut. Kemudian, kecamatan lain akan dikunjungi satu per satu sebanyak satu kali.

B. Pembentukan Matriks

Bobot antarsimpul, dalam hal ini adalah jarak antarkecamatan, diambil dengan menggunakan bantuan Google Maps. Jarak tidaklah sepenuhnya akurat, namun tetap dapat dijadikan acuan dalam perhitungan jarak. Untuk mempermudah penulisan pula, jarak pada matriks merupakan jarak dalam skala kilometer.

TABLE I. MATRIKS SEBELUM REDUKSI

	CP	GB	JB	KM	MT	SB	SN
CP	∞	7.8	1.7	4.9	4	7.9	2.3
GB	7.8	∞	6.5	6.3	4.6	2.1	5.9
JB	1.7	6.5	∞	4.2	3	7.2	2.1
KM	4.9	6.3	4.2	∞	5.8	4.1	5.2
MT	4	4.6	3	5.8	∞	5.8	1.9
SB	7.9	2.1	7.2	4.1	5.8	∞	5.7
SN	2.3	5.9	2.1	5.2	1.9	5.7	∞

Kemudian, dilakukan reduksi hingga didapatkan minimal satu buah nol pada masing-masing baris dan kolom.

TABLE II. MATRIKS SETELAH REDUKSI

	CP	GB	JB	KM	MT	SB	SN
CP	∞	6.1	0	1.2	4	7.9	2.3
GB	5.7	∞	4.4	2.2	2.5	0	3.8
JB	0	4.8	∞	0.5	1.3	5.5	0.4
KM	0.8	2.2	0.1	∞	1.7	0	1.1
MT	2.1	2.7	1.1	1.9	∞	3.9	0

SB	5.8	0	5.1	0	3.7	∞	3.6
SN	0.4	4	0.2	1.3	0	3.8	∞

Pada reduksi, didapatkan batas minimum sebagai berikut:

$$\text{Minimum baris} = 1.7 + 2.1 + 1.7 + 4.1 + 1.9 + 2.1 + 1.9 = 15.5$$

$$\text{Minimum kolom} = 0 + 0 + 0 + 2 + 0 + 0 + 0 = 2$$

$$\text{Batas minimum} = 15.5 + 2 = 17.5$$

C. Proses Iterasi Routing

Dalam mencari kecamatan terbaik yang dikunjungi pertama kali dari kecamatan Gambir (GB), diperlukan perhitungan untuk masing-masing cost dari keenam kecamatan lainnya.

1) Iterasi ke-1

TABLE III. MATRIKS COST LINTASAN GB-CP

	CP	GB	JB	KM	MT	SB	SN
CP	∞	∞	0	1.2	4	7.9	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	∞	4.4	∞	0.1	0.9	5.1	0
KM	∞	2.2	0.1	∞	1.7	0	1.1
MT	∞	2.7	1.1	1.9	∞	3.9	0
SB	∞	0	5.1	0	3.7	∞	3.6
SN	∞	4	0.2	1.3	0	3.8	∞

Didapatkan hasil perhitungan cost sebagai berikut

$$c(\text{CP}) = c(\text{GB}) + A(\text{GB}, \text{CP}) + r \\ = 17.5 + 5.7 + 0.4 = 23.6$$

TABLE IV. MATRIKS COST LINTASAN GB-JB

	CP	GB	JB	KM	MT	SB	SN
CP	∞	4.9	∞	0	2.8	6.7	1.1
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	∞	∞	0.5	1.3	5.5	0.4
KM	0.8	2.2	∞	∞	1.7	0	1.1
MT	2.1	2.7	∞	1.9	∞	3.9	0
SB	5.8	0	∞	0	3.7	∞	3.6
SN	0.4	4	∞	1.3	0	3.8	∞

Didapatkan hasil perhitungan cost sebagai berikut

$$c(\text{JB}) = c(\text{GB}) + A(\text{GB}, \text{JB}) + r \\ = 17.5 + 4.4 + 1.2 = 23.1$$

TABLE V. MATRIKS COST LINTASAN GB-KM

	CP	GB	JB	KM	MT	SB	SN
CP	∞	6.1	0	∞	4	7.9	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	4.8	∞	∞	1.3	5.5	0.4
KM	0.8	∞	0.1	∞	1.7	0	1.1

MT	2.1	2.7	1.1	∞	∞	3.9	0
SB	5.8	0	5.1	∞	3.7	∞	3.6
SN	0.4	4	0.2	∞	0	3.8	∞

Didapatkan hasil perhitungan cost sebagai berikut

$$c(\text{KM}) = c(\text{GB}) + A(\text{GB}, \text{KM}) + r \\ = 17.5 + 2.2 + 0 = 19.7$$

TABLE VI. MATRIKS COST LINTASAN GB-MT

	CP	GB	JB	KM	MT	SB	SN
CP	∞	6.1	0	1.2	∞	7.9	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	4.8	∞	0.5	∞	5.5	0.4
KM	0.8	2.2	0.1	∞	∞	0	1.1
MT	2.1	∞	1.1	1.9	∞	3.9	0
SB	5.8	0	5.1	0	∞	∞	3.6
SN	0.4	4	0.2	1.3	∞	3.8	∞

Didapatkan hasil perhitungan cost sebagai berikut

$$c(\text{MT}) = c(\text{GB}) + A(\text{GB}, \text{MT}) + r \\ = 17.5 + 2.5 + 0.2 = 20.2$$

TABLE VII. MATRIKS COST LINTASAN GB-SB

	CP	GB	JB	KM	MT	SB	SN
CP	∞	4	0	1.2	4	∞	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	2.7	∞	0.5	1.3	∞	0.4
KM	0.7	0	0	∞	1.6	∞	1
MT	2.1	0.6	1.1	1.9	∞	∞	0
SB	5.8	∞	5.1	0	3.7	∞	3.6
SN	0.4	1.9	0.2	1.3	0	∞	∞

Didapatkan hasil perhitungan cost sebagai berikut

$$c(\text{SB}) = c(\text{GB}) + A(\text{GB}, \text{SB}) + r \\ = 17.5 + 0 + 2.2 = 19.7$$

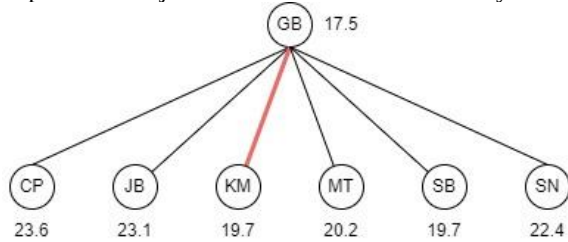
TABLE VIII. MATRIKS COST LINTASAN GB-SN

	CP	GB	JB	KM	MT	SB	SN
CP	∞	6.1	0	1.2	4	7.9	∞
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	4.8	∞	0.5	1.3	5.5	∞
KM	0.8	2.2	0.1	∞	1.7	0	∞
MT	1	1.6	0	0.8	∞	2.8	∞
SB	5.8	0	5.1	0	3.7	∞	∞
SN	0.4	∞	0.2	1.3	0	3.8	∞

Didapatkan hasil perhitungan cost sebagai berikut

$$c(\text{SN}) = c(\text{GB}) + A(\text{GB}, \text{SN}) + r \\ = 17.5 + 3.8 + 1.1 = 22.4$$

Dari keenam simpul kecamatan tersebut, dapat dilihat bahwa lintasan GB - KM atau Gambir – Kemayoran dan GB – SB atau Gambir – Sawah Besar memiliki *cost* terkecil dibandingkan kecamatan lainnya. Oleh karena itu, salah satu dari kedua simpul tersebut dipilih untuk kemudian dibangkitkan simpul anaknya. Kali ini, penulis memilih simpul Kemayoran, sehingga simpul solusi sejauh ini adalah Gambir – Kemayoran.



Gambar 6. Pohon Ruang Status Iterasi ke-1
(sumber: Dokumen Pribadi)

2) Iterasi ke-2

TABLE IX. MATRIKS COST LINTASAN GB-KM-CP

	CP	GB	JB	KM	MT	SB	SN
CP	∞	6.1	0	∞	4	4.1	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	∞	4.4	∞	∞	0.9	1.3	0
KM	∞	∞	∞	∞	∞	∞	∞
MT	∞	2.7	1.1	∞	∞	0.1	0
SB	∞	0	5.1	∞	3.7	∞	3.6
SN	∞	4	0.2	∞	0	0	∞

Didapatkan hasil perhitungan *cost* sebagai berikut
 $c(\text{CP}) = c(\text{KM}) + A(\text{KM}, \text{CP}) + r$
 $= 19.7 + 0.8 + 4.2 = 24.7$

TABLE X. MATRIKS COST LINTASAN GB-KM-JB

	CP	GB	JB	KM	MT	SB	SN
CP	∞	3.8	∞	∞	1.7	1.8	0
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	4.8	∞	∞	1.3	1.7	0.4
KM	∞	∞	∞	∞	∞	∞	∞
MT	2.1	2.7	∞	∞	∞	0.1	0
SB	5.8	0	∞	∞	3.7	∞	3.6
SN	0.4	4	∞	∞	0	0	∞

Didapatkan hasil perhitungan *cost* sebagai berikut
 $c(\text{JB}) = c(\text{KM}) + A(\text{KM}, \text{JB}) + r$
 $= 19.7 + 0.1 + 6.1 = 25.9$

TABLE XI. MATRIKS COST LINTASAN GB-KM-MT

	CP	GB	JB	KM	MT	SB	SN
--	----	----	----	----	----	----	----

CP	∞	6.1	0	∞	∞	4.3	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	4.8	∞	∞	∞	1.9	0.4
KM	∞	∞	∞	∞	∞	∞	∞
MT	2.1	2.7	1.1	∞	∞	0.3	0
SB	5.8	0	5.1	∞	∞	∞	3.6
SN	0.2	3.8	0	∞	∞	0	∞

Didapatkan hasil perhitungan *cost* sebagai berikut
 $c(\text{MT}) = c(\text{KM}) + A(\text{KM}, \text{MT}) + r$
 $= 19.7 + 1.7 + 3.8 = 25.2$

TABLE XII. MATRIKS COST LINTASAN GB-KM-SB

	CP	GB	JB	KM	MT	SB	SN
CP	∞	6.1	0	∞	4	∞	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	4.8	∞	∞	1.3	∞	0.4
KM	∞	∞	∞	∞	∞	∞	∞
MT	2.1	2.7	1.1	∞	∞	∞	0
SB	5.8	0	5.1	∞	3.7	∞	3.6
SN	0.4	4	0.2	∞	0	∞	∞

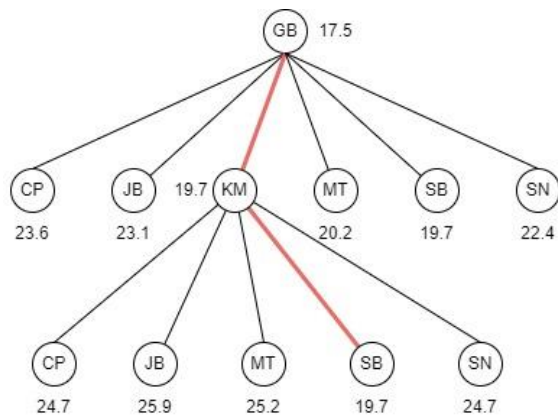
Didapatkan hasil perhitungan *cost* sebagai berikut
 $c(\text{SB}) = c(\text{KM}) + A(\text{KM}, \text{SB}) + r$
 $= 19.7 + 0 + 0 = 19.7$

TABLE XIII. MATRIKS COST LINTASAN GB-KM-SN

	CP	GB	JB	KM	MT	SB	SN
CP	∞	6.1	0	∞	4	5.1	∞
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	4.8	∞	∞	1.3	2.7	∞
KM	∞	∞	∞	∞	∞	∞	∞
MT	1	1.6	0	∞	∞	0	∞
SB	5.8	0	5.1	∞	3.7	∞	∞
SN	0.4	4	0.2	∞	0	1	∞

Didapatkan hasil perhitungan *cost* sebagai berikut
 $c(\text{SN}) = c(\text{KM}) + A(\text{KM}, \text{SN}) + r$
 $= 19.7 + 1.1 + 3.9 = 24.7$

Dari kelima simpul kecamatan tersebut, dapat dilihat bahwa lintasan KM-SB atau Kemayoran – Sawah Besar memiliki *cost* terkecil. Sehingga, simpul solusi sejauh ini adalah Gambir – Kemayoran – Sawah Besar.



Gambar 7. Pohon Ruang Status Iterasi ke-2

(sumber: Dokumen Pribadi)

3) Iterasi ke-3

TABLE XIV. MATRIKS COST LINTASAN GB-KM-SB-CP

	CP	GB	JB	KM	MT	SB	SN
CP	∞	3.4	0	∞	4	∞	2.3
GB	∞	∞	∞	∞	∞	∞	∞
JB	∞	1.7	∞	∞	0.9	∞	0
KM	∞	∞	∞	∞	∞	∞	∞
MT	∞	0	1.1	∞	∞	∞	0
SB	∞	∞	∞	∞	∞	∞	∞
SN	∞	1.3	0.2	∞	0	∞	∞

Didapatkan hasil perhitungan *cost* sebagai berikut

$$c(\text{CP}) = c(\text{SB}) + A(\text{SB}, \text{CP}) + r \\ = 19.7 + 5.8 + 3.1 = 28.6$$

TABLE XV. MATRIKS COST LINTASAN GB-KM-SB-JB

	CP	GB	JB	KM	MT	SB	SN
CP	∞	1.1	∞	∞	1.7	∞	0
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	2.1	∞	∞	1.3	∞	0.4
KM	∞	∞	∞	∞	∞	∞	∞
MT	2.1	0	∞	∞	∞	∞	0
SB	∞	∞	∞	∞	∞	∞	∞
SN	0.4	1.3	∞	∞	0	∞	∞

Didapatkan hasil perhitungan *cost* sebagai berikut

$$c(\text{JB}) = c(\text{SB}) + A(\text{SB}, \text{JB}) + r \\ = 19.7 + 5.1 + 5 = 29.8$$

TABLE XVI. MATRIKS COST LINTASAN GB-KM-SB-MT

	CP	GB	JB	KM	MT	SB	SN
CP	∞	3.4	0	∞	∞	∞	2.3
GB	∞	∞	∞	∞	∞	∞	∞

JB	0	2.1	∞	∞	∞	∞	0.4
KM	∞	∞	∞	∞	∞	∞	∞
MT	2.1	0	1.1	∞	∞	∞	0
SB	∞	∞	∞	∞	∞	∞	∞
SN	0	1.1	0	∞	∞	∞	∞

Didapatkan hasil perhitungan *cost* sebagai berikut

$$c(\text{MT}) = c(\text{SB}) + A(\text{SB}, \text{MT}) + r \\ = 19.7 + 3.7 + 2.9 = 26.3$$

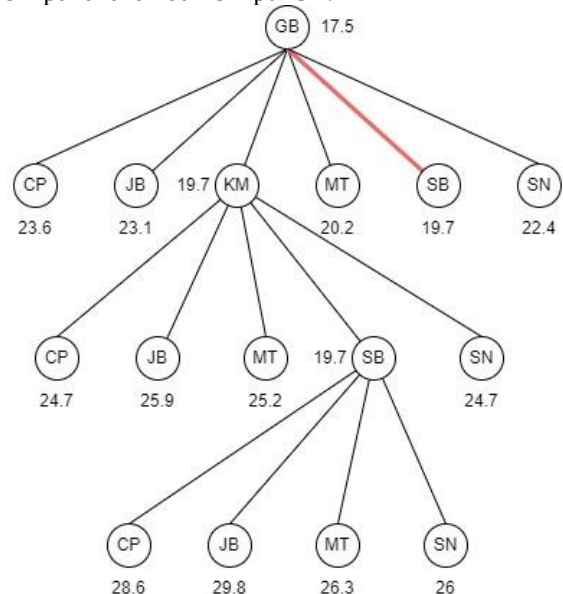
TABLE XVII. MATRIKS COST LINTASAN GB-KM-SB-SN

	CP	GB	JB	KM	MT	SB	SN
CP	∞	4.5	0	∞	4	∞	∞
GB	∞	∞	∞	∞	∞	∞	∞
JB	0	3.2	∞	∞	1.3	∞	∞
KM	∞	∞	∞	∞	∞	∞	∞
MT	1	0	0	∞	∞	∞	∞
SB	∞	∞	∞	∞	∞	∞	∞
SN	0.4	2.4	0.2	∞	0	∞	∞

Didapatkan hasil perhitungan *cost* sebagai berikut

$$c(\text{SN}) = c(\text{SB}) + A(\text{SB}, \text{SN}) + r \\ = 19.7 + 3.6 + 2.7 = 26$$

Dari keempat simpul kecamatan tersebut yang dibangkitkan, dapat dilihat bahwa simpul dengan *cost* terkecil yang dapat di-expand merupakan simpul SB dari lintasan GB – SB dengan *cost* bernilai 19.7. Oleh karena itu dilakukan *backtracking* untuk memeriksa simpul anakan dari simpul SB.



Gambar 8. Pohon Ruang Status Iterasi ke-3

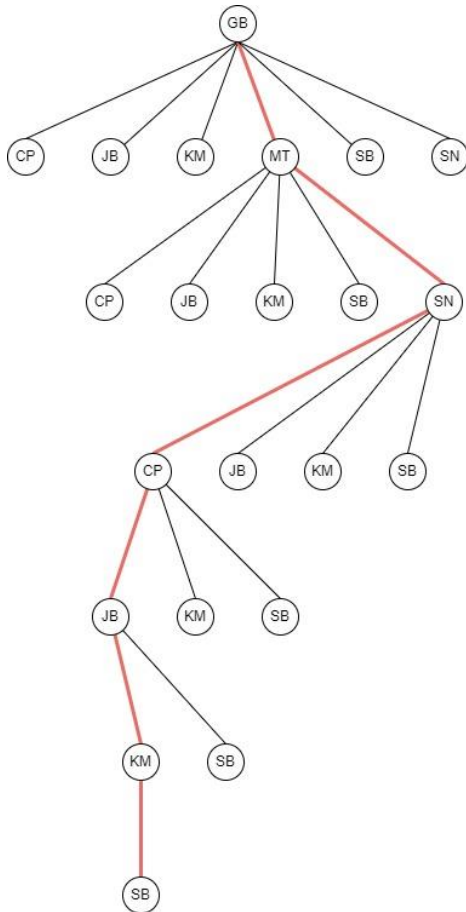
(sumber: Dokumen Pribadi)

Selanjutnya, dilakukan iterasi berkali-kali hingga ditemukan simpul solusi berupa seluruh simpul

kecamatan yang dikunjungi tepat sekali dengan *cost* seminimum mungkin.

D. Rute Akhir

Setelah dilakukan iterasi berulang-ulang kali, didapatkan hasil sebagaimana berikut:



Gambar 9. Pohon Ruang Status Iterasi Akhir

(sumber: Dokumen Pribadi)

Rute akhir dengan jarak seminimum mungkin dapat dicapai dengan rute Gambir – Menteng – Senen – Cempaka Putih – Johar Baru – Kemayoran – Sawah Besar – Gambir. Rute tersebut memiliki total jarak tempuh sebesar 20.9 km.

IV. KESIMPULAN DAN SARAN

Algoritma *branch and bound* memiliki banyak kegunaan dalam kehidupan sehari-hari. Salah satunya adalah dalam permasalahan TSP atau *Travelling Salesman Problem*. Dalam hal ini, rute dengan jarak tempuh terdekat yang dapat dicapai pada kecamatan di wilayah administrasi Jakarta Pusat adalah Gambir – Menteng – Senen – Cempaka Putih – Johar Baru – Kemayoran – Sawah Besar – Gambir dengan total jarak sebesar 20.8 km. Saran dalam pengerjaan penelitian ini adalah pembuatan program dengan *coding* untuk memudahkan pengerjaan karena iterasi yang dapat terjadi berulang-ulang kali, sesuai dengan banyaknya simpul yang ada.

LINK VIDEO YOUTUBE

Dalam rangka penyebaran makalah yang telah dibuat, penulis menyertakan tautan video Youtube yang dapat diakses pada link berikut:

<https://youtu.be/62C50cED-BQ>

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan yang Maha Esa atas rahmatnya sehingga makalah ini, yang berjudul “Penerapan Algoritma *Branch and Bound* pada Permasalahan *Travelling Salesman Problem* untuk Menentukan Rute Terdekat antar Kecamatan di Jakarta Pusat”, dapat diselesaikan dengan baik dan tepat pada waktunya. Tentunya, penulis tidaklah sendiri dalam proses penyusunan makalah ini. Penulis mengucapkan terima kasih untuk seluruh dosen mata kuliah IF2211 Strategi Algoritma 2021/2022 yang telah membimbing dan membagikan ilmunya kepada seluruh mahasiswa Teknik Informatika ITB angkatan 2020. Kemudian, terima kasih juga penulis ucapkan untuk teman-teman yang selalu menemani dan memberikan dukungan di setiap saat. Terakhir, tidak lupa penulis mengucapkan terima kasih sebesar-besarnya untuk kedua orang tua yang selalu bersama penulis.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. 2021. Algoritma branch and bound (Bagian 1) <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian1.pdf>. Diakses pada 21 Mei 2022.
- [2] Munir, Rinaldi. 2021. Algoritma branch and bound (Bagian 2) <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian2.pdf>. Diakses pada 21 Mei 2022.
- [3] Munir, Rinaldi. 2021. Algoritma branch and bound (Bagian 3) <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian3.pdf>. Diakses pada 21 Mei 2022.
- [4] Munir, Rinaldi. 2021. Algoritma branch and bound (Bagian 4) <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf>. Diakses pada 21 Mei 2022.
- [5] <https://www.geeksforgeeks.org/branch-and-bound-algorithm/>. Diakses pada 22 Mei 2022.

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, 22 Mei 2022

Fayza Nadia
13520001