# Aplikasi *Travelling Salesman Problem* (TSP) dalam Menentukan Rute Perjalanan ke Pusat Oleh-Oleh di Kota Bandung

Anna Elvira Hartoyo - 13518045[1]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*[1]13518045@std.stei.itb.ac.id*

***Abstrak*—Pada abad ke-21, mayoritas orang menginginkan segala sesuatu dapat diselesaikan cepat dan efisien. Hal ini juga berlaku ketika berlibur ke suatu tempat, misalnya Kota Bandung. Bandung sebagai kota yang terkenal akan objek wisatanya memiliki banyak pusat oleh-oleh yang beragam. Wisatawan gemar untuk berbelanja buah tangan untuk dibawa pulang ke tempat asalnya. Permasalahannya adalah jarak antartempat pusat oleh-oleh tersebut dan keterbatasan waktu mengakibatkan wisatawan tidak dapat mengunjungi seluruh pusat oleh-oleh. Permasalahan ini dapat diselesaikan bila urutan dan rute jalan yang dipilih paling efisien dan tepat. Makalah ini akan membahas mengenai pengaplikasian Travelling Salesman Problem (TSP) untuk melakukan optimasi rute perjalanan kunjungan ke pusat oleh-oleh di Bandung. Penyelesaian masalah dilakukan dengan pendekatan heuristik dengan algoritma branch and bound. Pusat oleh-oleh akan direpresentasikan sebagai simpul dalam graf dan jalan yang menghubungkan antartempat sebagai sisi dalam graf, sehingga akan didapatkan sebuah graf berbobot yang akan digunakan dalam TSP. Pada akhirnya, diperoleh suatu rute yang paling optimal dan wisatawan dapat mengunjungi seluruh pusat oleh-oleh.***

***Kata Kunci*—branch and bound, graf, rute perjalanan, Travelling Salesman Problem (TSP)***

## I. PENDAHULUAN

Kota Bandung atau yang akrab disebut dengan istilah Paris Van Java merupakan kota yang populer sebagai destinasi wisata. Berbagai wisatawan domestik hingga mancanegara kerap berkunjung ke Bandung untuk menghabiskan waktu liburannya. Wisatawan umumnya berkunjung untuk menikmati berbagai objek wisata, menyantap kuliner khas, hingga berbelanja oleh-oleh. Tak heran bila di Kota Bandung ada banyak pusat oleh-oleh yang menawarkan beragam buah tangan khas kota kembang ini. Mulai dari pusat oleh-oleh yang menjual makanan khas, seperti Kartika Sari, Prima Rasa, outlet yang menjual baju di daerah Dago, hingga tempat yang menjual aksesoris lainnya seperti tas dan sepatu di Cibaduyut. Tempat-tempat tersebut tersebar di berbagai lokasi di Kota Bandung. Wisatawan biasanya berkeliling dengan menggunakan bus pariwisata atau kendaraan pribadi. Namun keterbatasan waktu seringkali menjadi kendala untuk mengunjungi seluruh pusat oleh-oleh tersebut. Permasalahan semacam ini dapat diselesaikan dengan memilih rute perjalanan yang paling efisien.

Jika melihat kenyataan, ada banyak faktor yang perlu dipertimbangkan untuk memilih rute perjalanan. Faktor tersebut meliputi jarak antartempat pusat oleh-oleh, kepadatan jalan, lebar jalan, kualitas jalan (rusak atau tidak), dan waktu yang diperlukan untuk singgah dan berbelanja di suatu tempat. Namun dalam makalah ini penulis akan memfokuskan pada satu faktor yang paling utama, yaitu jarak antartempat pusat oleh-oleh.

Pada makalah ini penulis akan menerapkan *Travelling Salesman Problem (TSP)* untuk mencari rute perjalanan paling mangkus. TSP menyelesaikan masalah optimasi rute dan menjamin seluruh tempat pusat oleh-oleh dapat dikunjungi dengan rute yang paling singkat. Wisatawan memiliki waktu yang cukup untuk singgah dan berbelanja di suatu tempat kemudian melanjutkan perjalanannya hingga seluruh tempat telah dikunjungi.

## II. LANDASAN TEORI

A. Teori Graf

1. Definisi Graf

Graf dalam sktuktur diskrit digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek tersebut. Secara visual, objek dalam graf dinyatakan sebagai titik (*node*). Sedangkan hubungan antar objek dinyatakan sebagai garis atau sisi (*edge*).

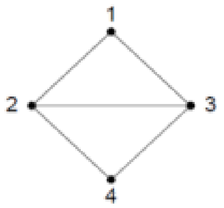
Graf G didefinsikan dengan notasi G = (V, E). V adalah himpunan tidak kosong dari simpul-simpul, biasanya disebut *ventricles* atau *node*. V = $\{v_1, v_2, v_3, \ldots, v_n\}$. E adalah himpunansisi yang menghubungkan sepadang simpul, biasa disebut juga sebagai *edge*. E = $\{e_1, e_2, e_3, \ldots, e_n\}$.

Simpul pada graf dapat diberi nama dengan huruf atau angka, sedangkan sisi pada graf biasanya diberi nama $e_1$, $e_2$, …, $e_n$. Sisi dinyatakan dalam pasangan ($v_i$, $v_j$) yang berarti sisi tersebut menghubungkan simpul $v_i$ dan $v_j$. Gelang atau kalang (*loop*) adalah sisi pada graf yang berawal dan berakhir pada simpul yang sama. Sedangkan sisi ganda (*multiple edge*) adalah dua buah sisi yang menghubungkan simpul yang sama.

2. Jenis-Jenis Graf

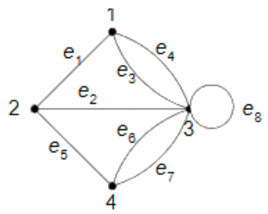
Berikut pengelompokan graf berdasarkan ada tidaknya gelang dan sisi ganda pada suatu graf :

a) Graf Sederhana (*Simple Graph*)
Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi ganda.




*Gambar 1. Graf sederhana*
*(Sumber: [2])*

b) Graf Tak Sederhana (*Unsimple Graph*)
Graf tak sederhana dapat dibagi lagi menjadi dua macam, yaitu graf ganda (*multigraph*) dan graf semu (*pseudograph)*. Graf ganda adalah graf yang mengandung sisi ganda. Sementara graf semu adalah graf yang mengandung gelang (termasuk bila memiliki sisi ganda juga), contohnya seperti graf pada gambar 2.

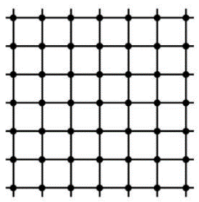



*Gambar 2. Graf tak sederhana*
*(Sumber: [2])*

Berikut pengelompokkan graf berdasaran jumlah simpul pada suatu graf:

a) Graf Berhingga (*Limited Graph*)
Graf berhingga adalah graf yang jumlah simpulnya n atau berhingga.

b) Graf Tak Berhingga (*Unlimited Graph*)
Graf tak berhingga adalah graf yang jumlah simpulnya tidak berhingga banyaknya.

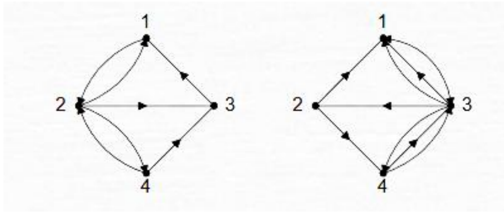


*Gambar 3. Graf tak berhingga*
*(Sumber: [2])*

Berikut pengelompokan graf berdasarkan orientasi arah pada sisi:

a) Graf Tak-Berarah (*Undirected Graph*)
Graf tak berarah adalah graf yang tidak mempunyai orientasi arah. Pada graf jenis ini urutan pasangan simpul yang menghubungkan sisi tidak diperhatikan. Dengan kata lain $(v_i, v_j) = (v_j, v_i)$.

b) Graf Berarah (*Directed Graph*)
Graf berarah adalah graf yang tiap sisinya diberikan orientasi arah. Sisi berarah ini sering disebut sebagai busur (*arc)*. Urutan pasangan simpul yang menghubungkan sisi menjadi hal penting dalam graf berarah. Dengan kata lain $(v_i, v_j) \neq (v_j, v_i)$. Untuk busur $(v_i, v_j)$, $v_i$ disebut sebagai simpul asal (*initial vertex*) dan $v_j$ disebut simpul terminal (*terminal vertex*).




*Gambar 4. Dua buah graf berarah*
*(Sumber: [2])*

3. Terminologi Graf

Berikut beberapa terminology atau istilah yang berkaitan dengan graf:

a) Bertetangga (*Adjacent*)
Dua buah simpul dikatakan bertetangga jika keduanya terhubung langsung melalui sebuah sisi. $v_i$ dikatakan bertetangga dengan $v_j$ bila terdapat sisi $(v_i,v_j)$ yang menghubungkan $v_i$ dan $v_j$.

b) Bersisian (*Incidency*)
Jika terdapat sembarang sisi e = $(v_i,v_j)$, sisi e dapat dikatakan bersisian dengan simpul $v_i$ dan $v_j$.

c) Simpul terpencil (*Isolated Vertex*)
Simpul terpencil adalah simpul yang tidak memiliki sisi yang bersisian dengannya atau simpul yang tidak memiliki simpul yang bertetangga dengannya.

d) Graf Kosong (*Null Graph*)
Graf kosong adalah graf dengan himpunan sisinya merupakan himpunan kosong atau hanya terdiri dari simpul saja tanpa sisi sama sekali. Graf kosong dapat ditulis dengan notasi $N_n$ dan n menyatakan jumlah simpul.

e) Derajat (*Degree*)
Derajat suatu simpul pada graf tak berarah didefinisikan sebagai jumlah sisi yang bersisian dengannya. Derajat ditulis dengan notasil d(v). Sedangkan pada graf berarah, derajat suatu simpul dinyatakan dengan $d_{in}(v)$ dan $d_{out}(v)$. $d_{in}(v)$ adalah derajat masuk, yaitu jumlah busur yang masuk ke simpul v. $d_{out}(v)$ adalah derajat keluar, yaitu jumlah busur yang keluar dari simpul v. Jumlah derajat pada simpul graf berarah tersebut adalah hasil pernjumlahan dari $d_{in}(v)$ dan $d_{out}(v)$.

f) Lintasan (*Path*)
Lintasan dengan panjang n dari simpul awal ke simpul tujuan di dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0$, $e_1$, $v_1$, $e_2$, $v_2$, ..., $e_n$, $v_n$ sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ..., $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf. Untuk graf sederhana, lintasan cukup dituliskan sebagai barisan dari simpul-simpul, tetapi untuk graf berarah perlu dituliskan pula sisinya untuk menghindari kerancuan. Panjang lintasan adalah jumlah sisi dalam lintasan tersebut.

g) Siklus (*Cycle)* dan Sirkuit (*Circuit*)
Siklus atau sirkuit adalah lintasan yang berawal dan

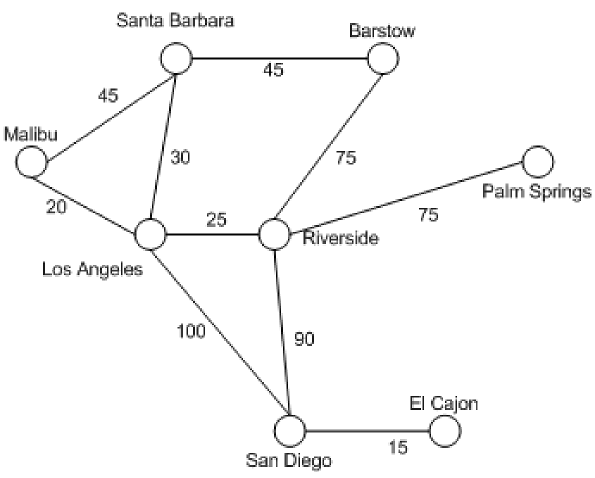
berakhir di simpul yang sama.

h) Terhubung (*Connected*)
Dua buah simpul $v_1$ dan $v_2$ dikatakan terhubung bila ada lintasan dari $v_1$ ke $v_2$. Suatu graf dikatakan graf terhubung bila untuk setiap pasangan simpul $v_i$ dan $v_j$ dalam himpunan V terdapat lintasan dari $v_i$ ke $v_j$. Jika tidak, graf disebut graf tak terhubung.

i) Upagraf (*Subgraph*) dan Komplemen Upagraf
Misalkan terdapat sebuah graf G = (V, E) dan graf $G_1$ = ($V_1$, $E_1$). $G_1$ adalah upagraf (*subgraph*) dari G jika $V_1 \subseteq V$ dan $E1 \subseteq E$. Sedangkan komplemen dari G adalahh graf $G_2$ sedemikian sehingga $E_2 = E - E_1$ dan $V_2$ merupakan himpunan simpul yang anggota-anggota $E_2$ bersisian dengannya.

j) Upagraf Rentang (*Spanning Subgraph*)
Upagraf $G_1 = (V_1, E_1)$ adalah upagraf rentang dari suatu graf G = (V, E) jika $V_1$ = V, dengan kata lain $G_1$ mengandung semua simpul dari G.

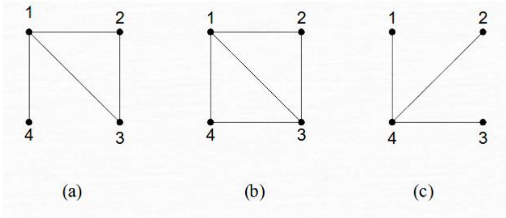
k) Graf Berbobot (*Weighted Graph*)
Graf berbobot adalah graf yang setiap sisinya diberi bobot atau harga. Bobot tersebut dapat menyatakan jarak antar dua kota, biaya tempuh perjalanan antar dua tempat, ongkos produksi, dan lain-lain. Graf berarah juga dapat direpresentasikan dengan bobot pada setiap sisinya. Panjang suatu lintasan dalam graf berbobot adalah jumlah bobot seluruh sisi yang dilalui lintasan tersebut.




*Gambar 5. Contoh graf berbobot untuk menyatakan jarak antarkota di suatu negara bagian USA (Sumber: https://simpledevcode.files.wordpress.com/2015/07/ic76956.gif?w=640)*

B. Lintasan dan Sirkuit Hamilton

Lintasan Hamilton adalah lintasan yang melewati setiap simpul pada graf tepat satu kali. Sedangkan sirkuit Hamilton adalah sirkuit yang melalui seluruh simpul pada graf tepat satu kali, kecuali simpul asal dan simpul akhir yang dilalui dua kali. Sirkuit Hamilton juga dapat didefinisikan sebagai lintasan Hamilton yang berawal dan berakhir di simpul yang sama. Graf yang memiliki lintasan Hamilton disebut graf semi-Hamilton dan graf yang memiliki sirkuit Hamilton disebut graf Hamilton.




*Gambar 6. (a) graf yang memiliki lintasan Hamilton, (b) graf yang memiliki sirkuit Hamilton, (c) graf yang tidak memiliki lintasan maupun sirkuit Hamilton (Sumber: [2])*

Terdapat beberapa teorema terkait graf Hamilton, yaitu Syarat cukup supaya graf sederhana G dengan $n \geq 3$ buah simpul adalah graf Hamilton ialah bila derajat tiap simpul paling sedikit $\frac{n}{2}$ (d(v) $\geq \frac{n}{2}$ untuk setiap simpul v di G). Teorema berikutnya adalah setiap graf lengkap adalah graf Hamilton. Di dalam graf lengkap G dengan n simpul dengan $n \geq 3$ terdapat $\frac{(n-1)!}{2}$ sirkuit Hamilton. Di dalam graf lengkap G dengan n buah simpul ($n \geq 3$) dan n ganjil, terdapat $\frac{(n-1)}{2}$ buah sirkuit Hamilton yang saling lepas. Jika n genap dan $n \geq 4$, di dalam G terdapat $\frac{(n-2)}{2}$ buah sirkuit Hamilton yang saling lepas. Akan tetapi terdapat graf di luar teorema ini tetap dapat memiliki sirkuit Hamilton dan tergolong sebagai graf Hamilton juga.

C. *Travelling Salesman Problem* (TSP)

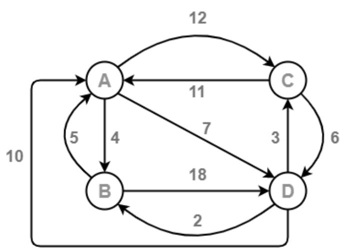
*Travelling Salesman Problem* (TSP) adalah sebuah persoalan dalam teori graf. Persoalan ini didapat dari masalah seorang pedagang keliling yang harus mengujungi sejumlah kota. Diberikan sejumlah kota dan jarak antarkota lalu harus dicari sirkuit terpendek yang akan dilalui pedagang tersebut jika pedagang itu berangkat dari kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal. Kota dapat direpresentasikan sebagai simpul dalam graf dan jalan yang menghubungkan antarkota dinyatakan sebagai sisi dari graf. Bobot pada sisi menyatakan jarak antara dua buah kota. Persoalan ini dapat diselesaikan dengan menentukan sirkuit Hamilton yang memiliki bobot paling kecil.

Untuk graf lengkap, graf dengan semua simpul memiliki sisi yang menghubungkan simpul tersebut dengan seluruh simpul lain pada graf, dapat dicari jumlah sirkuit Hamiltonnya dengan persamaan $\frac{(n-1)!}{2}$. Tetapi, persoalan TSP ini tidak hanya berlaku untuk graf lengkap saja, untuk graf tidak lengkap juga memiliki sirkuit Hamilton. Hingga saat ini belum ada algoritma mangkus untuk menyelesaikan permasalahan TSP dengan n sembarang, namun terdapat metode heuristik untuk memecahkan masalah ini. Salah satu metode heuristik yang umum digunakan adalah algoritma *branch and bound* yang akan dijelaskan pada bagian selanjutnya.

*D.* Algoritma *Branch and Bound*

Algortima *Branch and Bound* atau biasa disebut dengan B&B adalah algoritma yang digunakan untuk perncarian jalur. Untuk menyelesaikan permasalahan TSP, algoritma ini diaplikasikan untuk mencari jalur yang melalui semua titik dengan bobot paling kecil. Dalam algoritma ini ada beberapa langkah yang harus ditempuh:

1) Graf direpresetasikan ke dalam bentuk matriks ketetanggaan, contohnya sebagai berikut:




$$
\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix} \infty & 4 & 12 & 7 \\ 5 & \infty & \infty & 18 \\ 11 & \infty & \infty & 6 \\ 10 & 2 & 3 & \infty \end{bmatrix}
\end{array}
$$

*Gambar 7. Graf direpresentasikan dalam bentuk matriks (Sumber: [1])*

Jika kedua simpul tidak terhubung, diberi symbol ∞ pada matriks. Lalu dilakukan reduksi untuk setiap baris dan kolom pada matriks tersebut secara terpisah. Sebuah kolom atau baris dikatakan sudah tereduksi bila terdapat satu 0 pada baris atau kolom tersebut.

- Reduksi elemen pada baris ke-1 dengan 4
- Reduksi elemen pada baris ke-2 dengan 5
- Reduksi elemen pada baris ke-3 dengan 6
- Reduksi elemen pada baris ke-4 dengan 2
- Kolom ke-1 sudah tereduksi
- Kolom ke-2 sudah tereduksi
- Reduksi kolom ke-3 dengan 1
- Kolom ke-4 sudah tereduksi

$$
\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix} \infty & 0 & 8 & 3 \\ 0 & \infty & \infty & 13 \\ 5 & \infty & \infty & 0 \\ 8 & 0 & 1 & \infty \end{bmatrix}
\end{array}
\quad
\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix} \infty & 0 & 7 & 3 \\ 0 & \infty & \infty & 13 \\ 5 & \infty & \infty & 0 \\ 8 & 0 & 0 & \infty \end{bmatrix}
\end{array}
$$

(a) (b)

*Gambar 8. (a) Matriks yang sudah direduksi barisnya (b) matriks yang sudah direduksi baris dan kolomnya (Sumber: [1])*

Setelah itu jumlahkan seluruh elemen pereduksi matriks dan diperoleh nilai untuk *node* 1 atau cost(1). Dalam contoh ini nilai cost(1) = 4 + 5 + 6 + 2 + 1 = 18

2) Lakukan pemeriksaan terhadap simpul lainnya satu per satu hingga ditemukan simpul dengan nilai atau cost minimum. Misalnya pertama akan diperiksa untuk simpul B dari A (A→B) sebagai node 2. Dari *reduced* matriks sebelumnya, ambil nilai M[A, B] = 0. Ubah semua nilai pada baris A dan kolom B dan M[B, A] menjadi ∞. Selanjutnya lakukan lagi reduksi terhadap setiap baris dan kolom matriks lalu hitung nilai node 2 atau cost(2).

$$
\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & 0 \\ 3 & \infty & 0 & \infty \end{bmatrix}
\end{array}
$$

*Gambar 9 . Matriks setelah melalui tahap 2 (Sumber: [1])*

Cost(2) = Cost(1) + hasil penjumlahan elemen pereduksi + M[A,B]

Cost(2) = 18 + 0 + 7 = 25

Ulangi langkah ini untuk semua kemungkinan jalan dari suatu *node*. Lalu pilih *node* dengan nilai cost terkecil.

3) Lakukan langkah 2 untuk *node* pada level sebelumnya dengan cost terkecil yang sudah dipilih hingga diperoleh suatu sirkuit Hamilton dengan bobot yang minimum.

E. Pusat Oleh-Oleh di Kota Bandung

Berdasarkan beberapa rekomendasi dari *travel blog*, berikut ini daftar tempat pusat oleh-oleh di Bandung yang paling sering dikunjungi wisatawan untuk berbelanja:

1. Kartika Sari
   Pusat oleh-oleh Kartika Sari sangat terkenal dengan produk pisang bollen aneka rasa. Selain itu, pusat oleh-oleh ini juga menjual kue lainnya seperti *cheese stick*, *banana roll*, dan lain-lain. Lokasi di Jl. H. Akbar No.4, Pasir Kaliki.
2. Prima Rasa
   Prima Rasa terkenal dengan produk oleh-oleh *brownies* panggang khas Bandung. Lokasi di Jalan Kamuning No. 20, Merdeka, Sumur Bandung.
3. Kawasan Cibaduyut
   Kawasan Cibaduyut menawarkan berbagai produk kulit dengan harga terjangkau dan kualitas bagus. Produk kulit yang dijual meliputi tas, sepatu, dompet, ikat pinggang, hingga jaket. Lokasi di Jalan Cibaduyut Raya, Bojongloa Kidul.
4. Gepuk Ny. Ong
   Gerai gepuk yang paling terkenal di Bandung adalah Gepuk Ny. Ong. Gerai ini menjual gepuk ayam dan sapi. Lokasi di Jl. Dr. Djundjunan 155E.
5. Pasar Kosambi
   Pasar Kosambi menjual berbagai camilan khas Bandung seperti wajit Cililin, keripik tempe, basreng, sale pisang dan lain-lain. Lokasi di Jl. R.E. Martadinata No.85, Citarum
6. Pasar Baru Trade Centre
   Pasar Baru Trade Centre terkenal sebagai pusat grosir pakaian dengan harga terjangkau. Selain grosir, dapat juga membeli secara eceran. Lokasi di Jl. Otto Iskandar Dinata No.70.
7. Batagor Riri
   Batagor merupakan makanan khas Kota Bandung. Dari sekian banyak gerai yang menjual batagor, batagor Riri merupakan gerai yang paling laris dari masa ke masa. Lokasi di Jl. Burangrang No. 41.

8. Kawasan Cihampelas
   Di kawasan Cihampelas ada banyak toko yang menjual oleh-oleh khas Bandung, mulai dari kaus bergambar gedung sate hingga pernak-pernik seperti gantungan kunci. Lokasi di Jl. Cihampelas, Cipaganti.

Penulis mengambil delapan tempat karena pada umumnya wisatawan hanya dapat berkunjung paling banyak ke delapan tempat pusat oleh-oleh dalam sehari. Berikut lokasi delapan pusat oleh-oleh tersebut pada peta Kota Bandung:


*Gambar 10. Lokasi 8 pusat oleh-oleh di Kota Bandung (Sumber: maps.google.com)*

Keterangan:

| | |
|---|---|
| A = Gepuk Ny. Ong | E = Pasar Kosambi |
| B = Kawasan Cihampelas | F = Prima Rasa |
| C = Kartika Sari | G = Batagor Riri |
| D = Pasar Baru Trade Centre | H = Kawasan Cibaduyut |

# III. PEMBAHASAN

## A. Batasan Penelitian

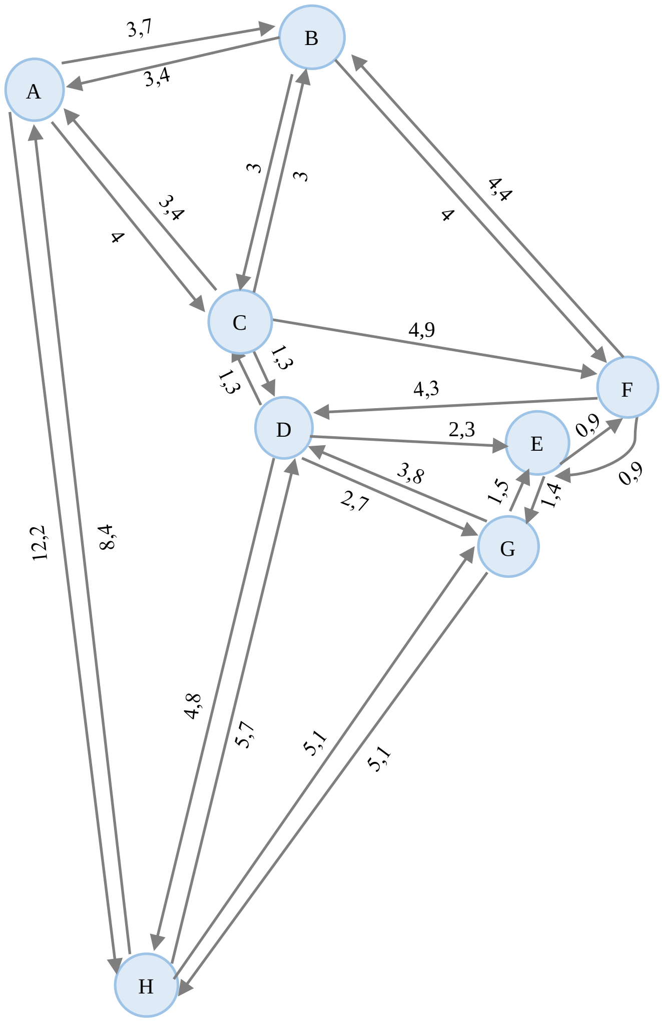
Beberapa batasan yang penulis tetapkan dalam penentuan rute perjalanan paling efisien untuk mengunjungi pusat oleh-oleh di Kota Bandung dengan penerapan teori graf dan TSP adalah:

1. Variabel faktor kepadatan jalan atau kemacetan diabaikan.
2. Variabel faktor lebar jalan diabaikan.
3. Kualitas setiap jalan dianggap sama.
4. Jika terdapat beberapa jalan dari suatu tempat ke tempat lain, diambil jalan yang paling pendek.
5. Untuk pusat oleh-oleh yang mempunyai beberapa cabang, penulis hanya mengambil satu cabang yang paling besar.

## B. Konversi Denah Lokasi Pusat Oleh-Oleh di Kota Bandung Menjadi Graf Berarah yang Berbobot

Setelah mengetahui letak setiap tempat pusat oleh-oleh akan dicari rute paling mangkus untuk mengunjungi semua tempat. Namun, diperlukan graf berarah berbobot untuk mengaplikasikan algoritma *branch and bound*. Sebelum mengonversi denah menjadi graf, perlu didefinisikan arti simpul, busur, dan bobot pada graf yang akan dibuat. Simpul merepresentasikan pusat oleh-oleh, busur merepresentasikan jalan antar pusat oleh-oleh dengan arah tertentu, dan bobot merupakan jarak antar tempat pusat oleh-oleh dalam satuan km yang telah dibulatkan menjadi satu angka di belakang koma. Graf yang terbentuk merupakan graf berarah berbobot dan bukan merupakan graf lengkap. Tetapi graf ini tetap memiliki sirkuit Hamilton.

Berikut merupakan graf berarah berbobot dari denah pada gambar 10 yang sudah disederhanakan:




*Gambar 11. Denah dalam representasi graf berarah berbobot (bobot dalam km).*

## C. Aplikasi *Travelling Salesman Problem* (TSP) dengan Algoritma *Branch and Bound* dalam Menentukan Rute Perjalanan Paling Mangkus

Langkah pertama yang harus dilakukan adalah melakukan konversi graf ke dalam bentuk matriks ketetanggaan untuk mempermudah perhitungan. Berikut matriks ketetanggaan dari graf pada gambar 11.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | 3,7 | 4 | ∞ | ∞ | ∞ | ∞ | 12,2 |
| **B** | 3,4 | ∞ | 3 | ∞ | ∞ | 4 | ∞ | ∞ |
| **C** | 3,4 | 3 | ∞ | 1,3 | ∞ | 4,9 | ∞ | ∞ |
| **D** | ∞ | ∞ | 1,3 | ∞ | 2,3 | ∞ | 2,7 | 4,8 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | 0,9 | 1,4 | ∞ |
| **F** | ∞ | 4,4 | ∞ | 4,3 | 0,9 | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 3,8 | 1,5 | ∞ | ∞ | 5,1 |
| **H** | 8,4 | ∞ | ∞ | 5,7 | ∞ | ∞ | 5,1 | ∞ |

*Gambar 12. Matriks ketetanggaan dari graf pusat oleh-oleh di Kota Bandung*

Selanjutnya, dilakukan reduksi elemen baris dan kolom pada matriks tersebut.

| | Baris | Kolom |
|---|---|---|
| A | 3,7 | 0,4 |
| B | 3 | 0 |
| C | 1,3 | 0 |
| D | 1,3 | 0 |
| E | 0,9 | 0 |
| F | 0,9 | 0 |
| G | 1,5 | 0 |
| H | 5,1 | 3,5 |

*Tabel 1. Elemen pereduksi untuk setiap baris dan kolom dari matriks mada gambar 12*

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | 0 | 0,3 | ∞ | ∞ | ∞ | ∞ | 5 |
| **B** | 0 | ∞ | 0 | ∞ | ∞ | 1 | ∞ | ∞ |
| **C** | 1,7 | 1,7 | ∞ | 0 | ∞ | 3,6 | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | 1 | ∞ | 1,4 | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 0,5 | ∞ |
| **F** | ∞ | 3,5 | ∞ | 3,4 | 0 | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,3 | 0 | ∞ | ∞ | 0,1 |
| **H** | 2,9 | ∞ | ∞ | 0,6 | ∞ | ∞ | 0 | ∞ |

*Gambar 13. Matriks setelah direduksi baris dan kolomnya.*

Jumlahkan seluruh nilai elemen pereduksi untuk memperoleh nilai cost (1)

Cost(1) = 3,7 + 3 + 1,3 + 1,3 + 0,9 + 0,9 + 1,5 + 5,1 + 0,4 + 3,5 = 21,6

Selanjutnya misalkan perjalanan akan dimulai dari titik A, maka ada 3 buah pilihan jalan dari titik A, yaitu A→B, A→C, dan A→H. Akan dicari rute mana yang memberi bobot paling minimum. Langkahnya untuk rute X→Y adalah ubah seluruh elemen pada baris X dan kolom Y dari matriks pada gambar 13 menjadi ∞. Ubah juga nilai M[Y,X] menjadi ∞. Lalu reduksi kembali matriks tersebut dan hitung nilai cost nya sesuai dengan rumus pada dasar teori.

- Rute A→B

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | 0 | ∞ | ∞ | 1,8 | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | 0 | ∞ | 3,6 | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | 1 | ∞ | 1,4 | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 0,5 | ∞ |
| **F** | ∞ | ∞ | ∞ | 3,4 | 0 | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,3 | 0 | ∞ | ∞ | 0,1 |
| **H** | 1,2 | ∞ | ∞ | 0,6 | ∞ | ∞ | 0 | ∞ |

*Gambar 14. Matriks untuk rute A→B yang sudah direduksi*

Cost(2) = 21,6 + 1,7 + 0 = 23,3

- Rute A→C

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | 0 | ∞ | ∞ | ∞ | ∞ | 1 | ∞ | ∞ |
| **C** | ∞ | 0 | ∞ | 0 | ∞ | 3,6 | ∞ | ∞ |
| **D** | ∞ | ∞ | ∞ | ∞ | 1 | ∞ | 1,4 | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 0,5 | ∞ |
| **F** | ∞ | 1,8 | ∞ | 3,4 | 0 | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,3 | 0 | ∞ | ∞ | 0,1 |
| **H** | 2,9 | ∞ | ∞ | 0,6 | ∞ | ∞ | 0 | ∞ |

*Gambar 15. Matriks untuk rute A→C yang sudah direduksi*

Cost(3) = 21,6 + (1,7) + 0,3 = 23,6

- Rute A→H

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | 0 | ∞ | 0 | ∞ | ∞ | 1 | ∞ | ∞ |
| **C** | 1,7 | 0 | ∞ | 0 | ∞ | 3,6 | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | 1 | ∞ | 1,4 | ∞ |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 0,5 | ∞ |
| **F** | ∞ | 1,8 | ∞ | 3,4 | 0 | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,3 | 0 | ∞ | ∞ | ∞ |
| **H** | ∞ | ∞ | ∞ | 0,6 | ∞ | ∞ | 0 | ∞ |

*Gambar 16. Matriks untuk rute A→H yang sudah direduksi*

Cost(4) = 21,6 + 1,7 + 5 = 28,3

Karena *cost* yang paling kecil adalah untuk rute A→B, maka dipilih *node* B untuk simpul selanjutnya. Langkah sebelumnya akan diulangi lagi untuk mencari rute paling pendek dari B. Dari matriks pada gambar 14, terdapat 2 buah pilihan jalan, yaitu B→C dan B→F.

- Rute A→B→C

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | 0 | ∞ | 3,6 | ∞ | ∞ |
| **D** | ∞ | ∞ | ∞ | ∞ | 1 | ∞ | 1,4 | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | 0,5 | ∞ |
| **F** | ∞ | ∞ | ∞ | 3,4 | 0 | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,3 | 0 | ∞ | ∞ | 0,1 |
| **H** | 1,2 | ∞ | ∞ | 0,6 | ∞ | ∞ | 0 | ∞ |

*Gambar 17. Matriks untuk rute A→B→C yang sudah direduksi*

Cost(5) = 23,3 + 1,8 + 0 = 25,1

- Rute A→B→F

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | 1 | ∞ | 1,4 | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | ∞ |
| **F** | ∞ | ∞ | ∞ | 3,4 | 0 | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,3 | 0 | ∞ | ∞ | 0,1 |
| **H** | 1,2 | ∞ | ∞ | 0,6 | ∞ | ∞ | 0 | ∞ |

*Gambar 18. Matriks untuk rute A→B→F yang sudah direduksi*

Cost(6) = 23,3 + 0,5 + 1 = 24,8

Karena *cost* yang paling kecil adalah untuk rute A→B→F, maka dipilih *node* F untuk simpul selanjutnya. Langkah akan diulangi untuk mencari rute paling pendek dari F. Dari matriks pada gambar 18, terdapat 2 buah pilihan jalan, yaitu F→D dan F→E.

- Rute A →B→F→D

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | 1 | ∞ | 1,4 | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | ∞ |
| **F** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | ∞ | 0 | ∞ | ∞ | 0,1 |
| **H** | 1,2 | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | ∞ |

*Gambar 19. Matriks untuk rute A→B→F→D yang sudah direduksi*

Cost(7) = 24,8 + 0 + 3,4 = 28,2

- Rute A →B→F→E

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | 1,4 | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | 0 | ∞ |
| **F** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,2 | ∞ | ∞ | ∞ | 0 |
| **H** | 1,2 | ∞ | ∞ | 0,6 | ∞ | ∞ | 0 | ∞ |

*Gambar 20. Matriks untuk rute A→B→F→E yang sudah direduksi*

Cost(8) = 24,8 + 0,1 + 0 = 24,9

Karena *cost* yang paling kecil adalah untuk rute A→B→F→E, maka dipilih *node* E untuk simpul selanjutnya. Langkah akan diulangi untuk mencari rute paling pendek dari F. Dari matriks pada gambar 20, terdapat 1 buah pilihan jalan, yaitu E→G

- Rute A→B→F→E→G

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **F** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | 2,2 | ∞ | ∞ | ∞ | 0 |
| **H** | 0,6 | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |

*Gambar 21. Matriks untuk rute A→B→F→E→G yang sudah direduksi*

Cost(9) = 24,9 + 0,6 + 0 = 25,5

Dipilih *node* G untuk simpul selanjutnya. Langkah akan diulangi untuk mencari rute paling pendek dari G. Dari matriks pada gambar 21, terdapat 1 buah pilihan jalan, yaitu G→D atau G→H.

- Rute A→B→F→E→G→D

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ | 0 |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **F** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **H** | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |

*Gambar 22. Matriks untuk rute A→B→F→E→G→D yang sudah direduksi*

Cost(10) = 25,5 + 0,6 + 2,2 = 28,3

- Rute A→B→F→E→G→H.

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **F** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **H** | 0,6 | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ |

*Gambar 23. Matriks untuk rute A→B→F→E→G→H yang sudah direduksi*

Cost(11) = 25,5 + 0 + 0 = 25,5

Karena *cost* yang paling kecil adalah untuk rute A→B→F→E→G→H, maka dipilih *node* H untuk simpul selanjutnya. Langkah akan diulangi untuk mencari rute paling pendek dari H. Dari matriks pada gambar 23, terdapat 2 buah pilihan jalan, yaitu H→D dan H→A. Tetapi simpul A sudah pernah dilewati sebelumnya, sehingga rute H→A ditolak.

- Rute A→B→F→E→G→H→D

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | 0 | ∞ | ∞ | ∞ | ∞ | ∞ |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **F** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **H** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |

*Gambar 24. Matriks untuk rute A→B→F→E→G→H→D yang sudah direduksi*

Cost(12) = 25,5 + 0 + 0 = 25,5

Dipilih *node* D untuk simpul selanjutnya. Langkah akan diulangi untuk mencari rute paling pendek dari D. Dari matriks pada gambar 24, hanya terdapat 1 buah pilihan jalan, yaitu D→C

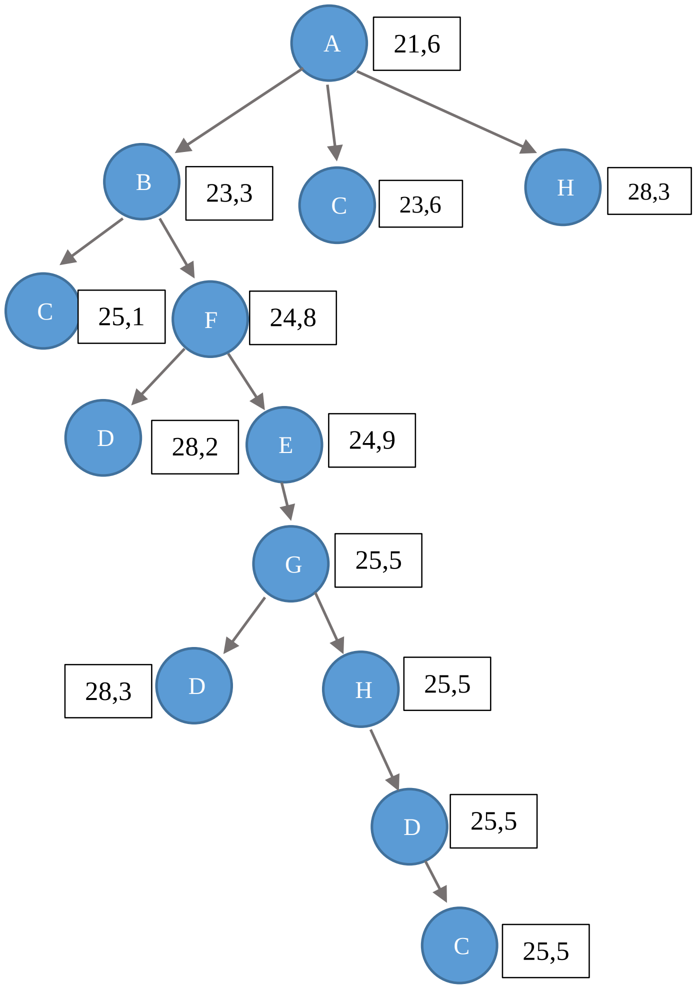
- Rute A→B→F→E→G→H→D→C

| | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** |
|---|---|---|---|---|---|---|---|---|
| **A** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **B** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **C** | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **D** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **E** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **F** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **G** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| **H** | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |

*Gambar 25. Matriks untuk rute A→B→F→E→G→H→D→C yang sudah direduksi*

Cost(14) = 25,5 + 0 + 0 = 25,5

Dari hasil perhitungan menggunakan algoritma *branch and bound*, diperoleh rute paling mangkus untuk persoalan ini adalah rute A→B→F→E→G→H→D→C→A, yaitu Gepuk Ny. Ong → Kawasan Cihampelas →Prima Rasa → Pasar Kosambi → Batagor Riri → Kawasan Cibaduyut → Pasar Baru Trade Centre → Kartika Sari → Gepuk Ny. Ong dengan bobot 25,5 km. Ilustrasi penentuan rute dalam bentuk diagram sebagai berikut:




*Gambar 26. Ilustrasi penentuan rute dengan branch and bound*

## IV. KESIMPULAN

Dalam menentukan rute perjalanan paling mangkus untuk beberapa tempat yang harus dikunjungi semua, dapat diaplikasikan konsep graf, khususnya terkait *Travelling Salesman Problem* (TSP). Persoalan TSP ini dapat diselesaikan salah satunya dengan pendekatan heuristik, yaitu dengan algoritma *branch and bound*. Hal ini dapat mengoptimasi rute yang dipilih sehingga diperoleh rute dengan jarak paling pendek tetapi tetap dapat mengunjungi seluruh tempat yang sudah ditentukan.

## V. UCAPAN TERIMA KASIH

Pertama-tama penulis ingin manyampaikan ucapan syukur kepada Tuhan Yang Maha Esa atas berkat-Nya penulis diberi kesehatan dan kekuatan sehingga dapat menyelesaikan makalah ini dengan baik. Penulis berterima kasih kepada orang tua penulis yang telah mendukung secara moral, doa, dan pembiayaan. Penulis juga mengucapkan terima kasih kepada dosen mata kuliah matematika diskrit, yaitu Bapak Dr. Ir. Rinaldi Munir, M.T. serta Ibu Fariska Zakhralativa Ruskanda, S.T., M.T. yang telah membimbing penulis dalam belajar matematika diskrit selama satu semester ini. Terakhir, penulis juga mengucapkan terima kasih kepada teman-teman Teknik Informatika 2018 yang saling mendukung dan menyemangati hingga proses pembuatan makalah ini dapat selesai tepat waktu.

## REFERENSI

[1] Anonim.2018. *Travelling Salesman Problem | Branch & Bound*. https://www.gatevidyalay.com/travelling-salesman-problem-using-branch-and-bound-approach/ diakses tanggal 2 Desember 2019 pukul 15.05

[2] Munir, Rinaldi. 2006.Diktat Kuliah IF2120: Matematika Diskrit.Bandung: Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

[3] Traveloka Blog.2018. *12 Pusat Oleh-oleh khas Bandung Paling Laris Sepanjang Masa*. https://blog.traveloka.com/bandung/12-pusat-oleh-oleh-khas-bandung-paling-laris-sepanjang-masa/ diakses tanggal 2 Desember 2019 pukul 14.51

[4] Yanuar, Muhammad.2018. *15 Tempat Belanja Oleh-Oleh Khas Bandung Yang Tidak Boleh Dilewatkan*. https://www.tripzilla.id/belanja-oleh-oleh-khas-bandung/7716 diakses tanggal 2 Desember 2019 pukul 14.35

## 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, 5 Desember 2019

Anna Elvira Hartoyo, 13518045