

# Penerapan Exhaustive Search dan Algoritma Branch and Bound untuk Menentukan Rute Terbaik dari Kereta Api di Pulau Jawa

Shadiq Harwiz - 13520038

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13520038@std.stei.itb.ac.id

**Abstract**—Kereta api menjadi salah satu alat transportasi yang cukup diminati masyarakat Indonesia, terutama di Pulau Jawa. Dengan tarif yang terjangkau, masyarakat dapat sampai di kota tujuan dengan waktu yang tidak terlalu lama. Penulis hendak membantu masyarakat yang hendak menggunakan kereta api sebagai transportasi dalam berpegian untuk menentukan rute terbaik mereka dalam perjalanan mereka ke kota tujuan. Dengan menggunakan algoritma Branch and Bound dan dibantu Exhaustive Search, penerapan algoritma ini diharapkan bisa membantu masyarakat untuk menentukan tidak hanya rute tersingkat, melainkan rute termurah dalam perjalanannya hingga ke kota tujuan.

**Kata Kunci**—masyarakat, jawa, kereta api, a star, rute terbaik

## I. PENDAHULUAN

Kereta api merupakan salah satu alat transportasi darat antarkota yang diminati oleh seluruh lapisan masyarakat. Kereta api memiliki beberapa kelebihan, diantaranya berjalan pada jalur khusus, memiliki kecepatan yang tinggi, dapat menjangkau daerah yang jauh, dan hemat bahan bakar. Dengan demikian, efisiensi dan efektifitas perjalanan manusia dengan kereta api sangat besar nilainya. Bila diteliti lebih lanjut, akan ditemukan berbagai penghematan dalam segi ekonomi jika menggunakan moda transportasi ini.

Di Indonesia penggunaan moda transportasi kereta api sangat diminati oleh masyarakat. Menurut data BPS, jumlah penumpang kereta api di Indonesia mencapai 149,763 juta orang sepanjang 2021. Dilihat dari wilayahnya, penumpang kereta api di Jawa (Jabodetabek + non-Jabodetabek) menjadi yang terbanyak sebesar 147,521 juta orang. Selain karena harga yang terjangkau, ketepatan waktu kereta api di Indonesia bisa dikatakan cukup baik sehingga pengguna jasa kereta api setiap tahunnya semakin meningkat.

Pulau Jawa yang hampir seluruh wilayahnya tidak dibatasi oleh laut sangat bergantung dengan keberadaan kereta api, masyarakat di wilayah ini menggunakan jasa kereta api untuk melakukan perjalanan dari satu wilayah ke wilayah lainnya. Tidak hanya kereta api jarak jauh, tetapi juga kereta api jarak dekat memiliki banyak peminat.



**Gambar 1-1:** Peta Jalur Kereta Api Jawa

Sumber: <https://www.arcgis.com/apps/View/index.html>

Penggunaan moda transportasi kereta api yang didominasi oleh masyarakat Pulau Jawa mengakibatkan banyaknya stasiun yang ada pada Pulau Jawa. Sehingga ada kemungkinan lebih dari satu jalur yang dapat dilewati hingga mencapai kota tujuan. Tentunya, biaya yang dikeluarkan bervariasi sehingga terdapat untuk suatu rute yang memiliki biaya yang lebih murah dibandingkan rute yang lain dalam mencapai kota tujuan.

Pada makalah ini, penulis akan membahas mengenai pencarian rute terbaik kereta api di Pulau Jawa. Rute terbaik yang dimaksud adalah rute termurah dalam mencapai kota tujuan. Oleh karena itu, penulis menggunakan penggabungan algoritma Brute Force dalam Exhaustive Search dan Algoritma Branch and Bound untuk memenuhi kebutuhan tersebut.

## II. TEORI DASAR

### A. Algoritma Brute Force

Algoritma Brute Force adalah metode yang paling sederhana untuk menyelesaikan masalah. Algoritma Brute Force memiliki pengertian pendekatan yang bersifat *straightforward* untuk menyelesaikan masalah. Algoritma Brute Force menyentuh langsung ke masalah sehingga cara penyelesaiannya sederhana, langsung, dan jelas. Algoritma ini bersifat “tidak cerdas” dan tidak efektif. Meskipun demikian, seringkali algoritma Brute Force dibutuhkan untuk dibandingkan dengan algoritma-algoritma lainnya.

## B. Exhaustive Search

Exhaustive Search adalah teknik pencarian solusi secara Brute Force untuk masalah pencarian elemen. Metode yang digunakan pada Exhaustive Search adalah:

1. Enumerasi semua kemungkinan solusi
2. Evaluasi semua kemungkinan solusi dan simpan solusi terbaik yang ditemukan selama ini (*the best solution found so far*)
3. Setelah pencarian berakhir, umumkan solusi terbaik

## C. Algoritma Branch and Bound

Algoritma Branch and Bound merupakan suatu algoritma yang digunakan untuk persoalan optimasi, yaitu meminimalkan atau memaksimalkan suatu fungsi objektif, yang tidak melanggar batasan (*constraint*) persoalan.

Algoritma Branch and Bound bisa dibilang sebagai gabungan dari algoritma Breadth-First Search (BFS) dan least cost search. Pada BFS murni, simpul berikutnya akan diekspansi berdasarkan urutan pembangkitannya (*First in First Out*). Pada algoritma Branch and Bound, setiap simpul diberi sebuah nilai 'cost'  $\hat{c}(i)$  yang merupakan nilai taksiran lintasan termurah ke simpul status tujuan yang melalui simpul status  $i$ . Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan urutan pembangkitannya, tetapi simpul yang memiliki cost paling kecil (*least cost search*), pada kasus minimasi, atau yang memiliki cost terbesar, untuk kasus maksimasi.

Algoritma Branch and Bound memiliki persamaan dengan algoritma Backtracking, yaitu pencarian solusi membentuk pohon ruang status dan simpul yang tidak 'mengarah' ke solusi akan 'dibunuh'. Namun, kedua algoritma ini memiliki perbedaan, yaitu pada algoritma Backtracking, tidak ada batasan terhadap persoalan yang dapat diselesaikan, walaupun umumnya untuk kasus non-optimisasi. Sedangkan algoritma Branch and Bound hanya bisa dilakukan untuk kasus optimisasi. Untuk setiap simpul pada pohon ruang status, ditentukan suatu batas nilai terbaik fungsi objektif pada setiap solusi yang mungkin, dan akan disimpan nilai dari solusi terbaik sejauh ini. Selain itu, pada algoritma Backtracking, pembangkitan simpul umumnya dilakukan dengan Depth-First Search (DFS), sedangkan pada algoritma Branch and Bound, pembangkitan simpul dilakukan dengan beberapa aturan tertentu, umumnya *best-first rule*.

Algoritma Branch and Bound menerapkan fungsi pembatas yang memangkas jalur yang dianggap tidak lagi mengarah ke solusi. Secara umum, kriteria pemangkasannya adalah sebagai berikut.

- Nilai simpul tidak lebih baik dari nilai terbaik sejauh ini (*the best solution so far*)
- Simpul tidak merepresentasikan solusi yang 'feasible' karena ada batasan yang dilanggar
- Solusi pada simpul tersebut hanya terdiri atas satu titik, tidak ada pilihan lain. Bandingkan nilai fungsi objektif dengan solusi terbaik saat ini, yang terbaik yang diambil

Algoritma Branch and Bound dapat dibuat secara umum sebagai berikut.

1. Masukkan simpul akar ke dalam antrian Q. Jika simpul akar adalah simpul solusi (goal node), maka solusi telah ditemukan. Stop.
2. Jika Q kosong, tidak ada solusi. Stop.
3. Jika Q tidak kosong, pilih dari antrian Q simpul  $i$  yang mempunyai nilai 'cost'  $\hat{c}(i)$  paling kecil. Jika terdapat beberapa simpul  $I$  yang memenuhi, pilih satu secara sembarang.
4. Jika simpul  $I$  adalah simpul solusi, berarti solusi sudah ditemukan, stop. Jika simpul  $I$  bukan simpul solusi, maka bangkitkan semua anak-anaknya. Jika  $i$  tidak mempunyai anak, kembali ke langkah 2.
5. Untuk setiap anak  $j$  dari simpul  $i$ , hitung  $\hat{c}(j)$ , dan masukkan semua anak-anak tersebut ke dalam Q.
6. Kembali ke langkah 2.

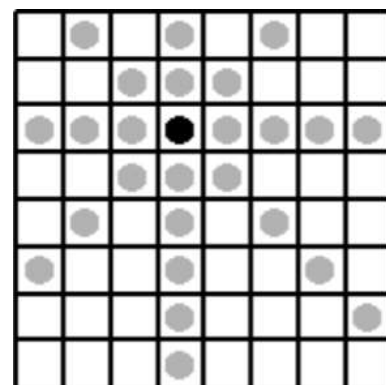
## D. Persoalan Branch and Bound

Terdapat beberapa persoalan yang dapat ditemukan solusinya dengan menggunakan algoritma branch and bound, yaitu:

### 1. Persoalan N-Ratu (The N-Queens Problem)

Merupakan persoalan sebuah papan permainan yang berukuran  $N \times N$  dan  $N$  buah ratu. Ratu-ratu tersebut harus ditempatkan pada petak papan tersebut sedemikian rupa agar tidak ada dua ratu atau lebih yang terletak pada baris, kolom, dan diagonal yang sama. Simpul yang memiliki nilai biaya terkecil adalah simpul hidup yang menjadi simpul expand, dan untuk setiap simpul  $X$ , nilai biayanya dapat berupa:

- a. Jumlah simpul yang terdapat di dalam upapohon  $X$  dan perlu dibangkitkan sebelum ditemukan simpul solusinya
- b. Panjang lintasan dari simpul  $X$  sampai ke simpul solusi terdekat



Gambar 2-1: Persoalan N-Ratu

Sumber: [2]

### 2. Permainan 15-Puzzle

Permainan 15-Puzzle merupakan persoalan yang menentukan urutan langkah dari pergeseran sebuah kotak kosong untuk dijadikan sebuah puzzle dengan posisi dari setiap

nomornya sesuai dengan posisi pada solusi akhirnya. Perhitungan biaya yang dilakukan pada permainan 15-Puzzle adalah

$$\hat{c}(P) = f(P) + \hat{g}(P)$$

dengan  $f(P)$  adalah panjang lintasan dari simpul akar ke  $P$  dan  $\hat{g}(P)$  adalah taksiran panjang lintasan terpendek dari  $P$  sampai ke simpul solusi pada upapohon yang akarnya adalah  $P$ , atau jumlah ubin tidak kosong yang posisinya tidak sesuai dengan posisi pada solusi akhirnya.

|   |   |    |    |
|---|---|----|----|
| 1 | 3 | 4  | 15 |
| 2 |   | 5  | 12 |
| 7 | 6 | 11 | 14 |
| 8 | 9 | 10 | 13 |

(a) Susunan awal

|    |    |    |    |
|----|----|----|----|
| 1  | 2  | 3  | 4  |
| 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 |
| 13 | 14 | 15 |    |

(b) Susunan akhir

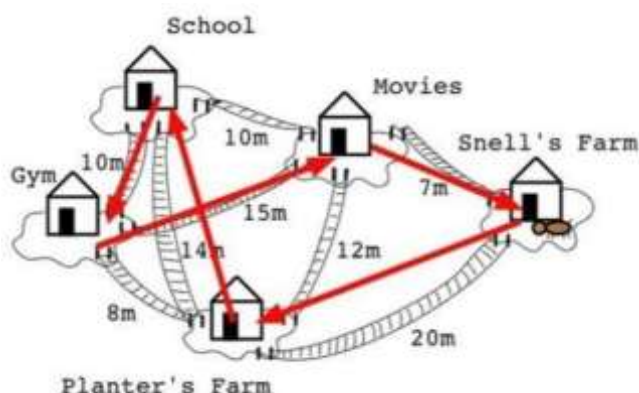
Gambar 2-2: Persoalan 15-Puzzle

Sumber: [2]

### 3. Travelling Salesperson Problem (TSP)

Persoalan dari TSP adalah suatu kondisi di mana diberikan sebuah  $n$  kota (simpul) dan diketahui jarak (bobot) antarkota tersebut. Perlu ditemukan perjalanan (tour) dengan jarak terpendek yang dapat dilalui oleh seorang pedagang, dengan syarat pedagang tersebut harus melalui setiap kota tepat hanya sekali saja dan kembali ke kota asal keberangkatannya. Cost yang diperhitungkan untuk setiap simpulnya di dalam pohon ruang status menyatakan nilai batas bawah (lower bound) ongkos untuk mencapai simpul solusi dari simpul tersebut, dan dapat dihitung secara heuristik berdasarkan salah satu dari kedua cara ini:

- Matriks ongkos-tereduksi (reduced cost matrix) dari graf
- Bobot minimum tur lengkap



Gambar 2-3: Travelling Salesperson Problem

Sumber: [2]

### 4. Assignment Problem

Assignment Problem merupakan persoalan di saat terdapat  $n$  orang dan  $n$  buah pekerjaan (job) dan masing-masing orang akan diberikan pekerjaan. Biaya untuk memerikan pekerjaan untuk setiap orang dinyatakan dalam sebuah matriks. Cara perhitungan untuk biaya (lower bound) setiap simpul hidup di dalam pohon ruang status bervariasi, seperti:

- Menggunakan matriks ongkos tereduksi
- Menjumlahkan nilai minimum untuk setiap baris pada matriks tersebut

|       | Job 1 | Job 2 | Job 3 | Job 4 |         |
|-------|-------|-------|-------|-------|---------|
| $C =$ | 9     | 2     | 7     | 8     | Orang a |
|       | 6     | 4     | 3     | 7     | Orang b |
|       | 5     | 8     | 1     | 8     | Orang c |
|       | 7     | 6     | 9     | 4     | Orang d |

Gambar 2-4: Assignment Problem

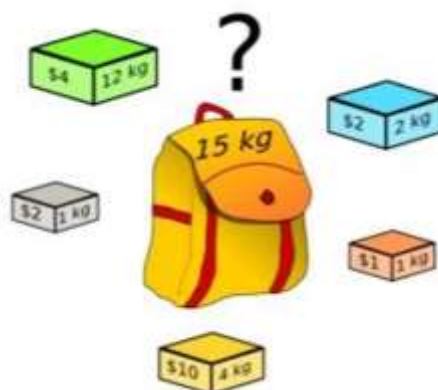
Sumber: [2]

### 5. Integer Knapsack Problem

Integer Knapsack Problem merupakan persoalan dengan situasi ketika diberikan  $n$  buah objek dan sebuah knapsack dengan kapasitas bobot  $K$ . Untuk setiap objeknya, akan memiliki bobot (weight)  $w_i$  dan keuntungan (profit)  $p_i$ , dan perlu diperhatikan cara pemilihan objek-objek untuk dimasukkan ke dalam knapsack agar dapat memperoleh keuntungan yang maksimal, dengan batasannya adalah jumlah bobotnya tidak melebihi kapasitas knapsack. Persoalan ini dapat dikatakan sebagai persoalan maksimasi, atau mencari keuntungan maksimum, oleh karena itu biaya ada setiap simpul di pohon ruang status dinyatakan sebagai batas atas (upper bound) dari solusi optimal. Simpul yang di-expand adalah simpul hidup yang memiliki biaya terbesar. Perhitungan untuk biayanya adalah

$$\hat{c}(i) = F + (K - W)p_{i+1}/w_{i+1}$$

dengan  $F$  adalah total keuntungan yang sudah dicapai,  $K$  adalah bobot knapsack,  $W$  adalah total bobot knapsack yang sudah terpakai, dan  $(p_{i+1}/w_{i+1})$  adalah rasio keuntungan per bobot objek yang tersisa untuk berikutnya.



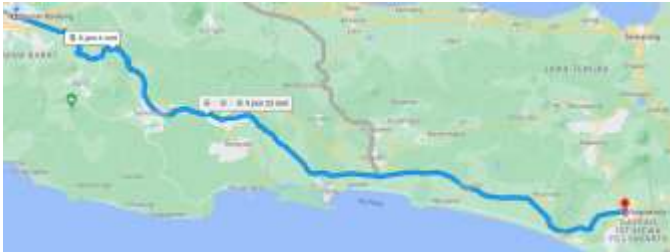
### Gambar 2-5: Integer Knapsack Problem

Sumber: [2]

### III. PENERAPAN ALGORITMA A STAR DALAM PENENTUAN RUTE TERBAIK

#### A. Inisialisasi Persoalan

Misalkan seseorang ingin berangkat dari Stasiun Bandung menuju Stasiun Yogyakarta pada tanggal 30 Mei 2022.



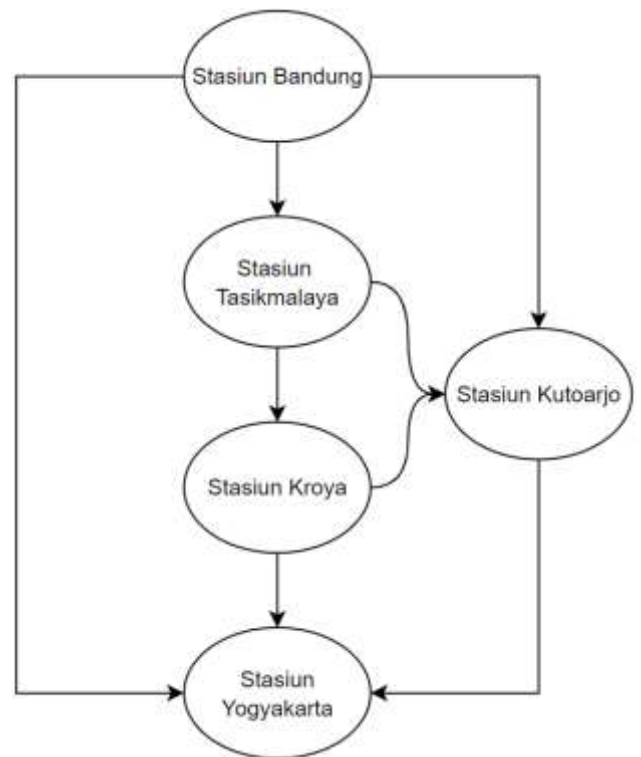
**Gambar 3-1: Salah Satu Peta Rute Kereta Api dari Stasiun Bandung Menuju Stasiun Yogyakarta**

Sumber: maps.google.com

Selain rute yang tertera pada gambar 3-1, terdapat beberapa jalur alternatif yang dapat ditempuh dari Stasiun Bandung menuju Stasiun Yogyakarta sebagai berikut.

1. Stasiun Bandung – Stasiun Kutoarjo – Stasiun Yogyakarta
2. Stasiun Bandung – Stasiun Tasikmalaya – Stasiun Kroya – Stasiun Yogyakarta
3. Stasiun Bandung – Stasiun Tasikmalaya – Stasiun Kutoarjo – Stasiun Yogyakarta
4. Stasiun Bandung – Stasiun Tasikmalaya – Stasiun Kroya – Stasiun Kutoarjo – Stasiun Yogyakarta

Kemudian, jalur-jalur tersebut dapat digambarkan graf sebagai berikut.



**Gambar 3-2: Graf Rute Kereta Api dari Stasiun Bandung Menuju Stasiun Yogyakarta**

Pada persoalan ini, cost dihitung berdasarkan jumlah biaya yang telah dikeluarkan atau jumlah waktu yang telah dihabiskan. Jika S adalah anak dari simpul R, maka

$$\hat{c}(S) = \hat{f}(R) + \hat{g}(R)$$

dengan  $\hat{f}(R)$  adalah biaya yang telah dikeluarkan atau waktu yang telah dihabiskan dari simpul akar ke simpul R dan  $\hat{g}(R)$  adalah biaya yang telah dikeluarkan atau waktu yang telah dihabiskan pada simpul ke R.

#### B. Implementasi Algoritma Branch and Bound dalam Menentukan Rute Termurah

Berikut data yang diambil pada tanggal 23 Mei 2022 dari aplikasi Traveloka pada pemesanan tiket tanggal 30 Mei 2022.

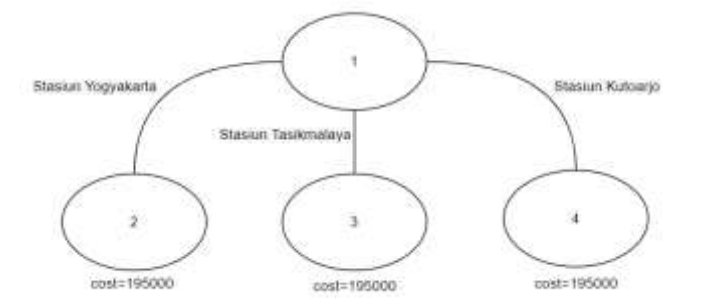
**Tabel 3-1: Daftar Biaya Rute Kereta Api**

| Stasiun Awal | Stasiun Tujuan | Biaya    |
|--------------|----------------|----------|
| Bandung      | Bandung        | Rp0      |
| Bandung      | Yogyakarta     | Rp195000 |
| Bandung      | Tasikmalaya    | Rp195000 |
| Tasikmalaya  | Kroya          | Rp58000  |
| Kroya        | Yogyakarta     | Rp75000  |
| Tasikmalaya  | Kutoarjo       | Rp58000  |
| Kroya        | Kutoarjo       | Rp58000  |

|          |            |          |
|----------|------------|----------|
| Bandung  | Kutoarjo   | Rp195000 |
| Kutoarjo | Yogyakarta | Rp50000  |

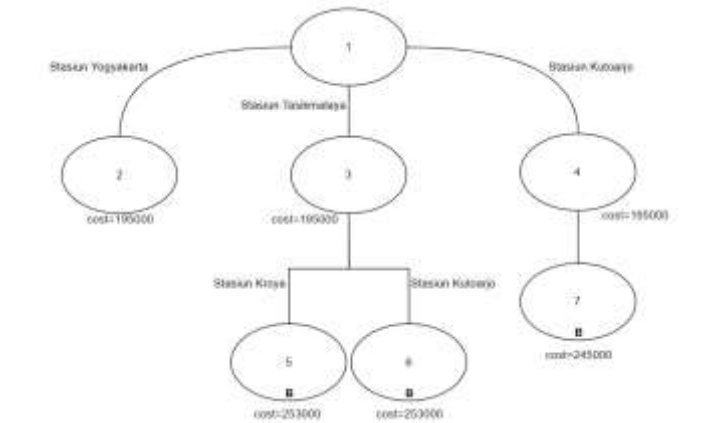
|             |             |     |
|-------------|-------------|-----|
| Bandung     | Yogyakarta  | 451 |
| Bandung     | Tasikmalaya | 183 |
| Tasikmalaya | Kroya       | 169 |
| Kroya       | Yogyakarta  | 141 |
| Tasikmalaya | Kutoarjo    | 285 |
| Kroya       | Kutoarjo    | 102 |
| Bandung     | Kutoarjo    | 399 |
| Kutoarjo    | Yogyakarta  | 57  |

Berikut merupakan pohon status pencarian awal untuk pencarian solusi menggunakan Algoritma Branch and Bound.



**Gambar 3-3: Pohon Status Pencarian Awal**

Dapat dilihat dari pohon tersebut bahwa terdapat sisi yang merupakan tempat tujuan stasiun selanjutnya dan *cost*, yaitu jumlah biaya yang telah dikeluarkan. Dari gambar 3-3, dapat dilihat bahwa jumlah biaya yang telah dikeluarkan sama setelah bergerak ke tempat tujuan stasiun selanjutnya. Dari pohon ruang status yang terbentuk, dapat diketahui bahwa telah diperoleh solusi pertama, yaitu stasiun Yogyakarta dengan bobot 195000. Dengan demikian, bunuh semua simpul dengan *cost* > 195000 yang dalam hal ini ditandai dengan B.



**Gambar 3-4: Pohon Status Akhir**

Dengan demikian, diperoleh satu solusi yaitu solusi ke-2: dari Stasiun Bandung menuju Stasiun Yogyakarta dengan biaya yang dikeluarkan sebesar Rp195.000.

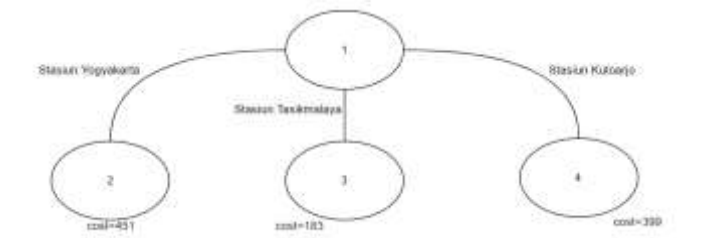
C. Implementasi Algoritma Branch and Bound dalam Menentukan Rute Tercepat

Berikut data yang diambil pada tanggal 23 Mei 2022 dari aplikasi Traveloka pada pemesanan tiket tanggal 30 Mei 2022.

**Tabel 3-2: Daftar Waktu Perjalanan Rute Kereta Api**

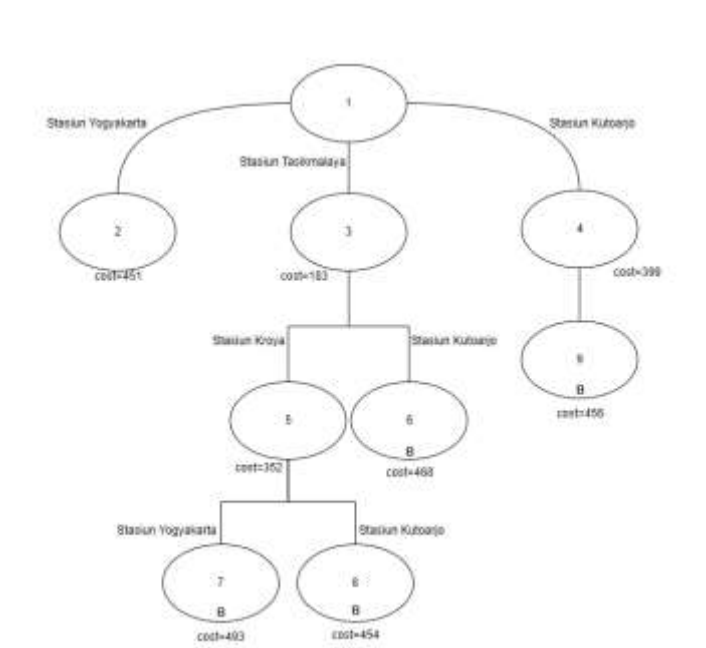
| Stasiun Awal | Stasiun Tujuan | Waktu (menit) |
|--------------|----------------|---------------|
| Bandung      | Bandung        | 0             |

Berikut merupakan pohon status pencarian awal untuk pencarian solusi menggunakan Algoritma Branch and Bound.



**Gambar 3-5: Pohon Status Pencarian Awal**

Dapat dilihat dari pohon tersebut bahwa terdapat sisi yang merupakan tempat tujuan stasiun selanjutnya dan *cost*, yaitu jumlah biaya yang telah dikeluarkan. Dari gambar 3-5, dapat dilihat bahwa jumlah waktu yang telah dihabiskan berbeda-beda setelah bergerak ke tempat tujuan stasiun selanjutnya. Dari pohon ruang status yang terbentuk, dapat diketahui bahwa telah diperoleh solusi pertama, yaitu stasiun Yogyakarta dengan bobot 451. Dengan demikian, bunuh semua simpul dengan *cost* > 451 yang dalam hal ini ditandai dengan B.



### **Gambar 3-6: Pohon Status Akhir**

Dengan demikian, diperoleh satu solusi yaitu solusi ke-2: dari Stasiun Bandung menuju Stasiun Yogyakarta dengan waktu yang dihabiskan selama 451 menit.

#### **D. Penarikan Rute Terbaik dengan Exhaustive Search**

Telah didapat satu rute terbaik, yakni dari Stasiun Bandung langsung menuju Stasiun Yogyakarta dengan total biaya yang dikeluarkan sebesar Rp195.000 dan lama waktu yang dihabiskan selama 451 menit.

Karena hanya diperoleh satu rute terbaik, maka didapatkan bahwa rute terbaik adalah dari Stasiun Bandung menuju Stasiun Yogyakarta.

### **IV. KESIMPULAN**

Kecepatan dan pengeluaran yang minim menjadi hal yang umum diterapkan oleh masyarakat membuat algoritma ini sangat berguna. Tentunya dengan tarif perjalanan kereta yang tidak selalu sama pada setiap hari nya menimbulkan suatu momen yang membuat tarif perjalanan kereta ke suatu kota tertentu menjadi lebih mahal dibanding tarif pada umumnya. Apabila algoritma ini dikembangkan dalam bentuk aplikasi, aplikasi ini bagus untuk membantu masyarakat dalam menentukan rute perjalanan terbaik mereka dari suatu kota ke kota tujuan.

### **UCAPAN TERIMA KASIH**

Saya bersyukur dan berterima kasih kepada Allah swt. atas kasih Rahmat-Nya dan karunia-Nya yang selalu berlimpah dalam hidup saya. Alhamdulillah, saya bisa menyelesaikan makalah ini dengan baik. Tentunya itu semua karena anugerah dari Allah swt. Saya mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T., Ibu Dr. Nur Ulfa Maulidevi, S.T, M.T., dan Ibu Dr. Masayu Leylia Khodra selaku dosen mata kuliah IF2211 Strategi Algoritma. Saya juga mengucapkan terima kasih kepada orang tua dan keluarga saya yang selalu mendukung saya baik secara langsung, maupun melalui dalam doa. Terima kasih juga kepada teman dan sahabat yang mendukung saya dalam menyelesaikan makalah ini

### **VIDEO LINK AT 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/1QznFLNDZag>

### **REFERENCES**

- [1] <https://www.bps.go.id/indicator/17/72/2/jumlah-penumpang-kereta-api.html>  
Waktu akses: 23 Mei 2022 pukul 14.03 WIB
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian1.pdf>  
Waktu akses: 23 Mei 2022 pukul 14.31 WIB
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian2.pdf>  
Waktu akses: 23 Mei 2022 pukul 14.35 WIB
- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian3.pdf>  
Waktu akses: 23 Mei 2022 pukul 14.37 WIB
- [5] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Branchand-Bound-2022-Bagian4.pdf>  
Waktu akses: 23 Mei 2022 pukul 14.43 WIB

### **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, 23 Mei 2022

Ttd

Nama dan NIM



Shadiq Harwiz / 13520038