

# Pengaturan Sistem Lampu Lalu Lintas dengan Algoritma Branch and Bound dengan Waktu Tunggu Menggunakan Algoritma Greedy

Aldo Suwandi / 13509025  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13509025@std.stei.itb.ac.id

**Abstrak**—Lampu lalu lintas merupakan sebuah alat pengatur kendaraan beroda agar dapat dengan teratur dan tertib dalam berlalu lintas. Pada zaman modern sekarang ini, lampu lalu lintas merupakan sebuah sistem yang hampir digunakan oleh seluruh dunia untuk mengatur sistem lalu lintas. Masalah yang harus diatasi oleh lampu lalu lintas ini adalah bagaimana cara mengatur jalur mana yang harus berjalan dan jalur mana yang harus berhenti dengan tidak adanya tabrakan antar jalur. Selain itu, dibutuhkan juga sebuah pengaturan untuk menentukan lamanya waktu tunggu untuk kendaraan yang berhenti yang sebanding dengan banyaknya antrian pada jalur tersebut. Semakin lama, penggunaan lampu lalu lintas ini semakin banyak, maka diperlukan sebuah metode untuk mengefektifkan kinerja dari lampu lalu lintas ini. Algoritma Branch and Bound memiliki karakteristik dapat melihat ruang solusi, sehingga dapat digunakan untuk menentukan lampu-lampu mana saja yang harus menyala dan mati. Lalu, algoritma greedy digunakan untuk menentukan waktu tunggu dari masing-masing jalur, karena algoritma ini memiliki karakteristik untuk mengambil keputusan di mana hasilnya akan paling menguntungkan.

**Kata kunci**—Lampu Lalu Lintas, Waktu Tunggu, Branch and Bound, Greedy.

## I. PENDAHULUAN

Lampu lalu lintas merupakan sebuah sistem pengaturan lalu lintas. Lampu ini terdiri dari 3 kondisi, yaitu penanda jalan, penanda waspada, dan penanda berhenti. Penanda itu wajib dipatuhi oleh setiap kendaraan beroda yang sedang melintasi daerah dengan lampu lalu lintas tersebut. Lampu berwarna merah merupakan penanda berhenti, lampu berwarna kuning merupakan penanda waspada, dan lampu berwarna hijau penanda jalan.

Umumnya seperti yang kita ketahui saat ini, bahwa lampu lalu lintas biasanya berada pada pertigaan, perempatan, perlimaan jalan, atau lebih. Hal ini berguna untuk memudahkan pekerjaan dari pada polisi atau petugas pengatur lalu lintas, agar tidak harus secara terus menerus dalam mengatur laju dari kendaraan beroda yang sedang melintas.

Pada zaman sekarang ini, hampir seluruh jalan pada kota besar maupun kecil menggunakan sistem dari lampu lalu lintas ini, dan pengaturan dari pada sistem lampu lalu

lintas ini sudah terkomputasi, sehingga tidak lagi memerlukan tenaga manusia untuk mengoperasikan sekuens dari nyala masing-masing lampu dalam lampu lalu lintas.

Akibat dari pada terkomputasinya sebuah sistem lampu lalu lintas, maka diperlukan sebuah algoritma untuk menentukan pengaturan agar tidak adanya lampu yang nyala dalam setiap jalur sehingga lampu tersebut dapat menyebabkan jalur menjadi berpotongan dengan jalur lainnya, karena hal ini dapat menyebabkan kemacetan dan kecelakaan. Maka, dengan pendekatan melalui algoritma Branch and Bound, diharapkan dapat menemukan seluruh kombinasi yang cocok dalam nyalanya setiap lampu lalu lintas, agar tidak ada jalur yang saling berpotongan.

Selain dari hal di atas, diperlukan juga sebuah pengaturan dalam menentukan jalur mana yang memiliki waktu tunggu terlama dan jalur mana yang memiliki waktu tunggu tercepat, sehingga hal ini dapat mencegah kemacetan dan mengefektifkan pekerjaan dari pada lampu lalu lintas ini. Untuk ini, pendekatan yang paling tepat adalah dengan menggunakan algoritma greedy, karena algoritma merupakan algoritma yang dapat mengambil keputusan berdasarkan kondisi yang paling menguntungkan, dan dalam kasus ini adalah akan mengambil jalur yang lebih pendek antriannya sehingga memiliki waktu tunggu lebih pendek.

## II. DASAR TEORI

### *Algoritma Branch and Bound*

Algoritma ini merupakan metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi diorganisasikan ke dalam pohon ruang status. Pembentukan pohon ruang status pada algoritma B&B dibangun dalam skema BFS.

Metode BFS berjalan dengan membangkitkan akar pertama kali, kemudian semua simpul anak-anaknya, kemudian successor dari setiap simpul anak, dan seterusnya. Secara umum, semua simpul pada akar dibangkitkan terlebih dahulu sebelum simpul-simpul pada kar  $n+1$ . Dengan metode BFS, jika terdapat sebuah solusi, maka BFS menjamin dapat menemukannya, dan jika

terdapat lebih dari satu solusi, BFS selalu menemukan solusi pertama pada akar pohon yang lebih rendah. Meskipun BFS kelihatannya metode yang bagus untuk mencari solusi, namun kompleksitas algoritmanya eksponensial.

Perbedaan antara metode BFS dan B&B adalah dimana B&B melakukan optimasi pencarian ke simpul solusi dengan memberi sebuah nilai ongkos (cost) pada setiap simpulnya. Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan urutan pembangkitannya, di mana pada BFS murni sesuai dengan urutan pembangkitan.

*Algoritma Greedy*

Algoritma ini membentuk solusi langkah per langkah per langkah. Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan, Keputusan yang telah diambil satu langkah tidak dapat dibuat lagi pada langkah selanjutnya.

Skema umum algoritma greedy,

1. Himpunan kandidat, C

Himpunan ini berisi elemen-elemen pembentuk solusi.

2. Himpunan solusi, S

Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan.

3. Fungsi seleksi

Fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal

4. Fungsi kelayakan

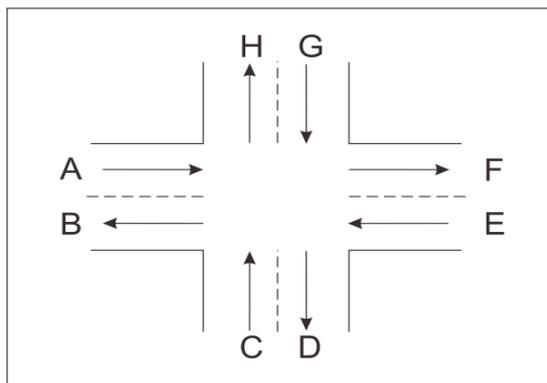
Fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada.

5. Fungsi obyektif

Fungsi yang memaksimalkan atau meminimumkan nilai solusi.

III. ANALISIS MASALAH

Lampu lalu lintas pada umumnya berada pada pertigaan, perempatan, perlimaanan jalan atau lebih. Maka, dibutuhkan sebuah mekanisme agar sekuens dari masing-masing lampu tidak bertabrakan, sebagai contoh, kita dapat melihat perempatan berikut :



Gambar Persimpangan Jalan

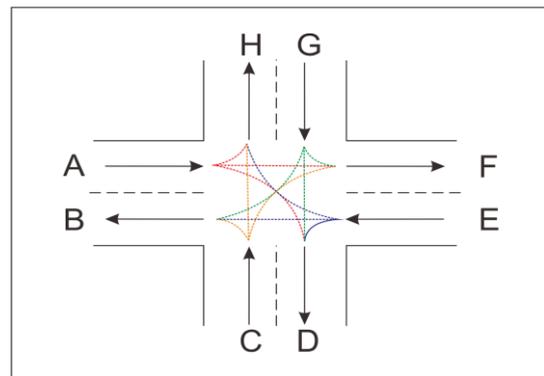
Pada gambar dapat dilihat, bahwa pada perempatan jalan terdapat sejumlah n jalur yang dapat dilewati oleh kendaraan di jalur A, B, C, D dan seterusnya. Banyaknya jalur dapat ditentukan dengan rumus :

$$n = p \times (p-1)$$

n = banyaknya jalur

p = jumlah persimpangan jalan.

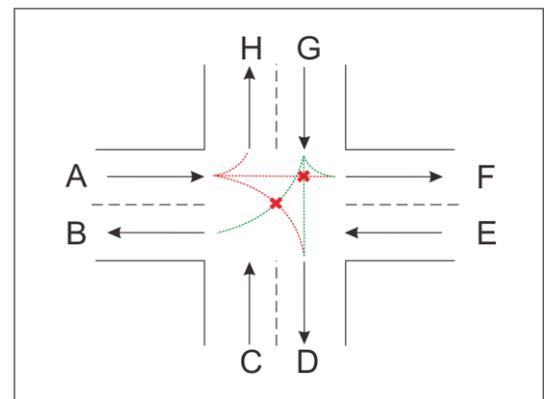
Dari hasil perhitungan di atas, dapat dilihat bahwa jalur yang terbentuk dari perempatan pada gambar adalah 12 jalur (4 x (4-1)). Bentuk jalur yang diperoleh adalah sebagai berikut :



Gambar Jalur dari Setiap Mobil

Dapat dilihat pada gambar, bahwa setiap mobil memiliki 3 jalur yang berbeda (setiap mobil memiliki jalur dengan warna yang sama, contohnya mobil pada jalur A memiliki jalur berwarna merah, mobil di jalur E memiliki warna biru, dan seterusnya). Kita akan mengambil contoh mobil di jalur A, maka mobil tersebut dapat menuju ke H, F, dan D. Mobil di jalur E, dapat menuju ke B, H, dan D.

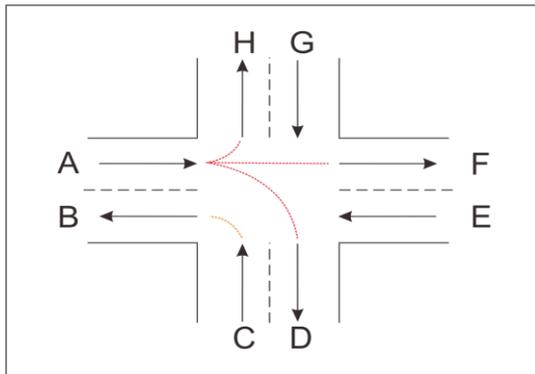
Fungsi dari pada lampu lalu lintas ini adalah dengan melakukan sebuah pengaturan pada mobil dari setiap jalur agar dapat berjalan dengan teratur dan tidak bertabrakan pada mobil pada jalur lain. Sebagai contoh :



Gambar Tabrakan antar Jalur

Maka, diperlukan sebuah pengaturan agar lampu-lampu pengatur lalu lintas dapat menyala dengan

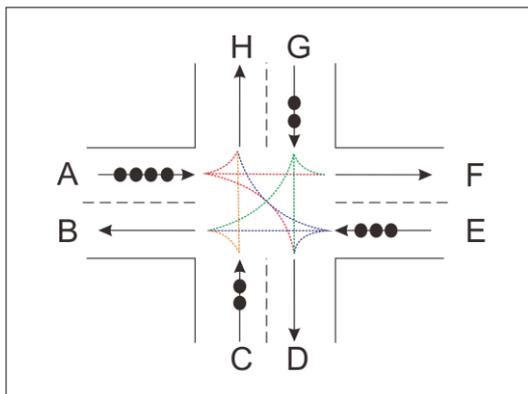
mempertimbangkan jalur mana yang tidak boleh menyala secara bersamaan, dan mana yang boleh secara bersamaan. Sebagai contoh sebagai berikut :



Gambar Sekuens Lampu Lalu Lintas yang Benar

Gambar di atas menunjukkan bagaimana pengaturan lampu lalu lintas secara baik dan benar, karena jalur di atas tidak ada yang bertabrakan, maka diperlukan sebuah algoritma pengaturan yang dapat menemukan sekuens - sekuens apa saja yang dapat mengatur lampu lalu lintas yang benar.

Selain itu, diperlukan juga sebuah sistem pengaturan dalam menentukan waktu tunggu untuk setiap jalur. Hal ini dikarenakan pada setiap jalur, jumlah mobil yang menunggu berbeda-beda jumlahnya. Maka, bobot tunggu untuk setiap jalurpun berbeda-beda. Sebagai contoh,



Gambar Banyaknya Mobil Setiap Jalur

Dapat dilihat pada gambar di atas, jalur A memiliki mobil yang menunggu paling banyak. Kita asumsikan bahwa setiap lampu lalu lintas memiliki sebuah sensor untuk mengetahui panjangnya antrian dalam setiap jalurnya. Maka diperlukan sebuah pengaturan untuk menentukan lampu mana yang harus memberi penanda berhenti paling lama (lampu merah) dan lampu mana yang harus memberi penanda jalan (lampu hijau) paling lama, begitu juga dengan kebalikannya.

#### IV. PENERAPAN ALGORITMA BRANCH AND BOUND DAN ALGORITMA GREEDY UNTUK PEMECAHAN MASALAH

Pertama, algoritma B&B digunakan untuk memecahkan masalah dalam penentuan himpunan kandidat untuk algoritma greedy. Himpunan kandidat yang dimaksud adalah bentuk dari mana sekuens - sekuens lampu lalu lintas tanpa ada lampu yang saling bertabrakan. Algoritma B&B dipilih sebagai pemecah masalah ini karena algoritma ini memiliki karakteristik untuk menemukan solusi secara sistematis atas masalah yang ada.

Sebagai contoh, kita akan mengambil sample sebuah perempatan jalan (gambar seperti pada analisis masalah) yang akan dibuat solusinya menggunakan algoritma B&B. Bentuk dari solusi dengan pencarian algoritma B&B adalah sebagai berikut.

$\{x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}\}$

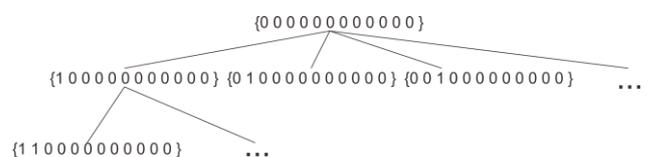
Setiap elemen memiliki sebuah nilai berupa,  
 0 = lampu merah  
 1 = lampu hijau

Urutan dalam solusinya adalah sebagai berikut,

- x1 = lampu pengatur A-H
- x2 = lampu pengatur A-F
- x3 = lampu pengatur A-D
- x4 = lampu pengatur C-B
- x5 = lampu pengatur C-H
- x6 = lampu pengatur C-F
- x7 = lampu pengatur E-D
- x8 = lampu pengatur E-B
- x9 = lampu pengatur E-H
- x10 = lampu pengatur G-F
- x11 = lampu pengatur G-D
- x12 = lampu pengatur G-B

Dalam pencarian menggunakan B&B ini terdapat sebuah fungsi yang berguna sebagai penentu cost dalam setiap cabangnya. Untuk kasus kali ini, fungsi cost tersebut merupakan fungsi penyeleksi apakah dalam setiap elemen x ada x lain yang menyebabkan jalur bertabrakan. Sebagai contoh, apabila x1, x2, x3 dalam kondisi menyala (1), maka yang boleh menyala hanyalah x4 (1) dengan kondisi x lainnya adalah merah (0).

Bentuk pohon yang akan dibangun adalah sebagai berikut :



Gambar Pohon BFS dari Perempatan Jalan

Dapat dilihat pada gambar di atas, bahwa pencarian solusi pada pohon tersebut akan dimulai dengan kondisi lampu merah seluruhnya. Lalu, dipastikan bahwa hasil solusi dari pada pohon tersebut akan tercapai pada seluruh cabang yang paling bawah.

Algoritma Pencariannya adalah sebagai berikut :

Sebelumnya, diketahui bahwa persimpangan merupakan tipe bentukan yang berisikan elemen-elemen sesuai dengan banyaknya jumlah jalur yang ada, sebagai contoh {0 0 0 0 0 0 0 0 0 0} (perempatan jalan).

```

Queue T;
procedure TrafficSolution (input s :
    persimpangan)
{
    if (!CekChild(s)) then
        -> return
    else
        if (CekJalur(s)) then
            T.Push(MakeChild(s))
            TrafficSolution (T.Pop())
}

```

Fungsi CekChild() adalah untuk memeriksa apakah cabang tersebut dapat memiliki anak lagi atau tidak.

Fungsi Cek Jalur() merupakan fungsi pembentuk cost yang gunanya untuk melihat apakah cabang tersebut memenuhi kriteria dalam pengaturan jalur. (Apakah ada jalur yang bertabrakan atau tidak).

Fungsi MakeChild() merupakan fungsi untuk membuat anak secara banyak, yang nantinya akan dimasukkan ke dalam T

Dengan algoritma ini, maka solusi yang di dapat adalah 4 buah yang nantinya akan terdapat di dalam Queue T. Solusinya adalah,

{1 1 1 1 0 0 0 0 0 0}, {0 0 0 1 1 1 1 0 0 0},  
 {0 0 0 0 0 1 1 1 1 0}, {1 0 0 0 0 0 0 0 1 1},

Maka telah didapatkan himpunan solusi dari algoritma B&B untuk menentukan pola atau sekuens dari pada pengaturan jalur dalam lampu lalu lintas ini.

Tahap selanjutnya adalah untuk menentukan lamanya waktu menunggu dalam setiap jalur. Maka, himpunan solusi dari algoritma B&B ini kita tempatkan pada himpunan kandidat dari algoritma greedy yang nantinya akan kita gunakan dalam penentuan waktu tunggu untuk setiap jalur.

Mari kita asumsikan bahwa untuk setiap waktu tunggu untuk 1 mobil dalam setiap jalurnya adalah 1 detik. Maka apabila kembali pada persoalan di atas, kita dapat melihat pada perempatan jalan terdapat 4 buah antrian mobil. Fungsi seleksi dalam algoritma greedy merupakan fungsi yang memilih himpunan kandidat yang ada. Setelah itu digunakan fungsi kelayakan untuk melihat kandidat mana yang memperoleh waktu lebih lama atau lebih cepat dalam menunggu.

Fungsi kelayakan di sini harus melihat bahwa dalam setiap kandidat terdapat 2 buah jalur yang harus diperhitungkan. Contohnya, pada kandidat {1 1 1 1 0 0 0 0 0 0}. Maka harus diperhitungkan jumlah mobil pada jalur A dan C, dengan perbandingan jalur A lebih tinggi dari pada jalur C, karena jalur A menggunakan seluruh jalur miliknya, sedangkan C hanya 1 jalur.

Algoritma perhitungannya sebagai berikut :

Sebelumnya, dimisalkan bahwa setiap kandidat solusi memiliki tipe bentukan,

n = yang menyimpan banyak mobil yang ada pada setiap jalurnya.

wait\_time = waktu tunggu.

rank = ranking dari pada tingkat banyaknya mobil dibanding dengan jalur lain.

```

procedure ArrangeTime(input a : kandidat)
{
    temp : kandidat
    MostLong(T)
    if (a = temp) then
        a.wait_time = rank x time_default
}

```

Fungsi MostLong() merupakan fungsi kelayakan dan seleksi dimana mengubah rank dari setiap kandidat yang ada. Fungsi harusnya dipanggil ketika satu putaran lampu lalu lintas berakhir. (1 putaran yang dimaksud adalah seluruh pola telah dimunculkan).

Dengan algoritma ini maka, waktu tunggu dalam setiap jalur akan berbeda-beda bergantung dari banyaknya jumlah mobil (n) dari setiap kandidat yang ada.

## V. KESIMPULAN

Metode yang algoritma B&B dan algoritma greedy yang dilakukan dapat menjawab masalah agar tidak adanya jalur yang bertabrakan dalam suatu pengaturan lampu lalu lintas, dan bagaimana sistem pengaturan dari pada waktu tunggu untuk setiap jalur yang ada. Namun pemecahan masalah dengan metode ini masih memiliki beberapa kekurangan, di antaranya pada persimpangan jalan yang sifatnya ganjil, fungsi penentu cost dalam algoritma B&B tidak dapat bekerja seperti pada persimpangan yang jumlahnya genap. Lalu, metode ini belum menjelaskan dan mengimplementasikan bagaimana apabila ada jalur yang one-way. Namun, secara garis besar dasar dari pemecahan masalah untuk lampu lalu lintas sudah dapat terjawab dengan 2 solusi dari algoritma yang ada pada makalah ini.

## PUSTAKA

[1] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Bandung : Institut Teknologi Bandung, 2009, hal 41-84, 171 - 188.

## 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, 8 Desember 2011

A handwritten signature in black ink, appearing to read 'Aldo', with a long horizontal stroke extending to the right.

Aldo Suwandi / 13509025