Aplikasi Branch and Bound Pada Pencarian Jalan Pada Software Navigasi

Rita Sarah / 13512009
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512009@std.stei.itb.ac.id

***Abstrak*— Makalah ini membahas tentang algoritma branch and bound , serta perannya dalam pencarian jalan pada software-software navigasi yang banyak digunakan sekarang ini (contoh pada Google Maps, Waze , dll.) untuk mencari beberapa jalan yang optimal menuju suatu tempat lainnya menggunakan algoritma branch and bound.**

***Kata Kunci*—Branch and Bound , navigation , rute terpendek, map**

I. PENDAHULUAN

. Pada era perkembangan teknologi informasi, banyak sekali hal yang dapat dilakukan dengan bantuan perangkat lunak pada komputer. Teknologi yang dikenal dengan teknologi informasi itu berperan dengan menganalisis suatu kasus dengan algoritma tertentu hingga memudahkan manusia di kehidupan sehari-hari.

Hal ini juga berdampak pada peta. Peta yang dulunya digunakan untuk mencari jalan secara manual oleh pengguna peta tersebut kini menjadi lebih mudah diakses dan mudah digunakan karena perkembangan teknologi . Untuk menggunakan peta dulunya adalah kesulitan menemukan posisi sekarang , namun dengan adanya bantuan teknologi hal itu pun menjadi mudah. Banyak software yang dibuat untuk memudahkan navigasi. Software tersebut dapat diakses melalui mesin GPS mobil ataupun smartphone dan perangkat elektronik lainnya.

Karena itu dibutuhkan sebuah algoritma yang cukup baik dalam pengolahan data untuk pencarian informasi yang diinginkan tersebut. Untuk itu dibuat banyak jenis pengolahan data dengan bermacam-macam algoritma. Untuk pencarian jalan seperti pada software navigasi digunakan algoritma seperti greedy (umumnya menggunakan Djikstra Algorithm) , pencarian transversal pada graf (BFS , DFS) , Branch and Bound, serta Dynamic Programming.

Software navigasi sekarang ini banyak digunakan untuk pencarian jalan menuju banyak tempat sehingga memudahkan para pengguna jalan mengetahui jalan mana yang perlu diambil menuju suatu tempat.Software navigasi tersebut mempunyai database berupa peta seluruh dunia beserta jalur-jalur yang dapat dilewati pada seluruh dunia . Selain itu juga memiliki algoritma pengolahan jalan yang akan dipakai saat kita sudah menentukan jalan mana yang akan diambil .

Software navigasi yang terkenal adalah Google Maps, Apple Maps, Waze ,dll.

Contoh pada Google Maps ketika kita sudah memilih tujuan yang akan dituju , maka kita dapat memilih rute mana yang akan dipilih. Pada rute tersebut terdapat kalkulasi waktu yang akan dilewati beserta kilometer yang akan ditempuh.

Untuk lebih mengerti cara kerja dari pencarian jalur tersebut maka kita dapat mencoba belajar dari algoritma branch and bound.

II. DASAR TEORI

A.Breadth First Search

Branch and Bound sangat dekat hubungan nya dengan algoritma Breadth First Search. Algoritma Breadth First Search (BFS) dikenal juga dengan algoritma pencarian melebar . Aplikasi dari algoritma ini adalah pencarian ruang pohon status dari sebuah graf.

Misal didalam sebuah kasus kita memiliki graf G yang mempunyai n buah simpul . Kita akan melakukan transversal dimulai dari simpul v.

Cara kerja algoritma BFS adalah dengan mengunjungi simpul v, kemudian semua simpul yang bertetangga dengan simpul v dikunjungi terlebih dahulu. Selanjutnya simpul yang belum dikunjungi dan bertetangga dengan simpul tadi dikunjungi . Jika graf berbentuk pohon berakar , maka semua simpul pada aras d dikunjungi lebih dahulu sebelum simpul-simpul pada aras d+1.

Algoritma BFS menggunakan sebuah antrian (queue) untuk mencatat simpul mana yang telah dikunjungi. Simpul-simpul yang telah dikunjungi suatu saat diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetangga lainnya. Untuk menjamin tidak adanya repetisi , tiap simpul hanya masuk ke antrian tersebut sekali. Selain itu dibutuhkan tabel boolean untuk mencatat simpul-simpul yang sudah dikunjungi. Awalnya isinya kosong (false) . Jika sudah dikunjungi maka isi nya akan diubah menjadi true .

Untuk memudahkan pengertian terhadap Breadth First Search maka dapat membaca pseudo-code BFS.

```
procedure BFS ( input v : integer)

w :integer
q:queue //processing queue
visited : array of boolean
A: matrix  // adjacency matrix for graph
n: integer //matrix size
procedure makeQueue( input/output  q: queue)
procedure  pushQueue(input/output  q:queue,input  v: integer)
procedure  popQueue(input/output  q:  queue,output  v :integer)
function isQueueEmpty ( input q:queue) -> boolean

makeQueue(q);
visited[v]<-true
pushQueue(q,v)

while not isQueueEmpty(q) do
    popQueue(q,v)
    for (w<-1 to n) do
        if (A[v,w] = 1) then
            if not visited[w] then
              pushQueue(q,w)
              visited[w] <- true
            endif
        endif
    endfor
endwhile
```

B.Branch and Bound

Branch and Bound , adalah sebuah strategi algortima yang digunakan untuk menyelesaikan suatu persoalan. Persoalan pencarian rute ini dapat diselesaikan dengan branch and bound.

Branch and bound (BB atau B&B) digunakan untuk mencari solusi optimal pada pencarian beberapa jumlah solusi. Algoritma ini merupakan metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi diorganisasikan kedalam pohon ruang status. Pembentukan pohon ruang status pada algoritma Branch & Bound berbeda dengan pembentukan pohon seperti di algoritma runut balik atau DFS. Pada algoritma Branch & Bound digunakan pembentukan pohon ruang solusi dengan skema BFS. Untuk mempercepat pencarian ke simpul solusi , maka setiap simpul diberi sebuah nilai ongkos / cost. Simpul berikutnya yang diekspansi adalah simpul yang memiliki cost terendah .

Nilai cost ini menyatakan taksiran ongkos termurah lintasan dari simpul i ke simpul solusi (goal node)

c(i)= nilai taksiran lintasan termurah dari simpul status i ke status tujuan .

c(i) menyatakan batas bawah (lower bound) dari ongkos pencarian solusi dari status i. Ongkos ini dihitung dengan suatu fungsi pembatas.

Metode ini diusulkan oleh A.H. Land dan A. G. Doig pada tahun 1960. Algoritma yang mirip dengan Branch and Bound adalah algoritma A* yang dikemukanan Hart , Nilsson, dan Raphael(1986) untuk menemukan lintasan terpendek dari suatu simpul ke simpul lain di dalam graf ruang status.

Branch and Bound dapat diaplikasikan untuk mencari rute seperti kasus berikut.

Terdapat graf lengkap berbobot sebagai berikut :

Bentuk adjacency matrix yang digunakan

x	20	30	10	11
15	x	16	4	2
3	5	x	2	4
19	6	18	x	3
16	4	7	16	x

Karena perjalanan di dalam graf melalui sisi (i,j) dengan i=1,2,3,4,5 dan dengan j = 1,2,3,4,5 , maka mengurangkan setiap elemen pada suatu baris atau pada suatu kolom dengan konstanta t akan mengurangi panjang setiap perjalanan sebesar t . Jika t dipilih dari elemen minimum pada baris i dan kolom j maka mengurangkan elemen pada baris i kolom j dengan t akan menghasilkan sebuah nol pada baris i kolom j tersebut . Dengan mengulangi proses ini menyebabkan matriks bobot tereduksi . Jumlah total elemen pengurang dari semua baris dan kolom menjadi batas bawah dari perjalanan dengan total bobot minimum . Nilai ini digunakan sebagai cost untuk simpul akar dalam pohon ruang status.

Contoh jika ingin mencari rute simpul 1 ke simpul 5 maka langkah pengerjaan awal yang harus dilakukan adalah :

Lakukan reduksi baris

x	18	28	8	9
14	x	15	3	1
3	5	x	2	4
17	4	16	x	1
15	3	6	15	x

R1- 8 , R2-1 , R3-2 , R4-1 , R5-3

Hasil reduksi baris

x	10	20	0	1
13	x	14	2	0
1	3	x	0	2
16	3	15	X	0
12	0	3	12	x

Lakukan reduksi kolom

x	10	20	0	1
13	X	14	2	0
1	3	X	0	2
16	3	15	x	0
12	0	3	12	x

C1 – 1 , C3 – 3

Hasil reduksi kolom

x	10	17	0	1
12	X	11	2	0
0	3	x	0	2
15	3	12	x	0
11	0	0	12	x

Jumlah semua pengurang = (R1+R2 + R3 + R4 +R5) + (C1+C3) = (8+1+2+1+3)+(1+3) = 19 .

Nilai yang didapat diatas adalah nilai batas untuk simpul akar yang dapat ditulis sebagai :

c(root) = 19

Artinya, persoalan tersebut paling tidak memiliki total bobot minimum 25. Pohon ruang status saat ini berisi satu buah simpul yang merupakan akar dengan nilai batas 25.

Selanjutnya misalnya A adalah matriks tereduksi untuk simpul R . Misalkan S adalah anak dari simpul R sedemikian sehingga sisi (R,S) pada pohon ruang status berkoresponden dengan sisi (i,j) . Jika S bukan simpul daun , maka matriks bobot tereduksi untuk simpul S dapat dihitung dengan cara :

- Ubah nilai pada baris i dan kolom j menjadi x . Ini mencegah agar tidak ada lintasan yang keluar dari simpul i atau masuk simpul j .
- Ubah A[j,1] menjadi x untuk mencegah penggunaan sisi (j,1).
- Reduksi kembali semua baris dan kolom pada matriks A kecuali elemen x .

Jika r adalah total pengurang pada matriks yang sudah direkduksi kembali , maka nilai batas pada simpul S adalah :

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

dimana

c(S) = bobot perjalanan minimum yang melalui simpul S (simpul di pohon ruang status)

c(R) = bobot perjalanan yang melalui simpul R , yang dalam hal ini adalah orangtua S pada pohon ruang status

A(i,j) = bobot sisi (i,j) pada graf G yang berkoresponden dengan sisi (R,S) pada pohon ruang status.

r = jumlah pengurang pada proses memperoleh matriks tereduksi simpul S

Perhitungan selanjutnya untuk simpul-simpul lain pada pohon ruang status adalah sebagai berikut

Mencari cost simpul 2 lintasan 1,2

Reduksi

x	x	x	X	x
x	x	11	2	0
0	x	x	0	2
15	x	12	X	0
11	x	0	12	x

Tidak dapat direduksi lagi maka c(2) = c(1) +A(1,2) + r = 19+10+0 = 29

Mencari cost simpul 3, lintasan 1,3

Reduksi

x	x	x	X	x
12	x	x	2	0
x	3	x	0	2
15	3	x	x	0
11	0	x	12	x

C1-11

Hasil Reduksi

x	x	x	x	x
12	x	x	2	0
x	3	x	0	2
15	3	x	x	0
11	0	x	12	x

Didapatkan r=11 . Nilai batas untuk simpul 3
c(3) = c(1) + A(1,3)+r = 19+17+11=47

Mencari cost simpul 4, lintasan 1,4
Reduksi

x	x	x	x	x
12	x	11	x	0
0	3	x	x	2
x	3	12	x	0
11	0	0	x	x

Tidak dapat direduksi lagi , maka nilai batas untuk simpul 4 c(4) = c(1) + A(1,4)+r = 19+0+0 = 19.

Mencari cost simpul 5, lintasan 1,5 .

Reduksi

x	x	x	x	x
12	x	11	2	x
0	3	x	0	x
15	3	12	x	x
x	0	0	12	x

R2-2 , R4-3

Hasil Reduksi

x	X	x	x	x
10	X	9	0	x
0	3	x	0	x
12	0	9	x	x
x	0	0	12	x

Diperoleh r=2+3 =5 . Nilai batas untuk simpul 5 pada pohon ruang status adalah c(5)=c(1)+A(1,5)+r= 19+1+5= 25.

Maka sudah tidak mungkin simpul 2,3 yang menuju ke 5 dengan rute terpendek, maka itu kita dapat mematikan simpul 2 dan 3. Kita dapat meninjau simpul 4 agar dapat membandingkan dengan simpul 5.

Simpul 6, Lintasan 1,4,2
Reduksi

x	X	x	x	x
12	X	11	x	0
0	X	x	x	2
x	X	x	x	x
11	X	0	x	x

Tidak dapat direduksi , maka nilai batas untuk simpul 6 c(6) =c(4) + B(4,2)+r = 19+3+0 = 22

Simpul 7 , Lintasan 1,4,3
Reduksi

x	X	x	x	x
12	X	x	x	0
x	3	x	x	2
x	X	x	x	x
11	0	x	x	x

R3-2 , C1-11

Hasil Reduksi

x	X	x	x	x
1	X	x	x	0
x	1	x	x	0
x	X	x	x	x
0	0	x	x	x

Diperoleh r = 2+11 = 13. Maka nilai batas untuk simpul 7
c(7)=c(4)+B(4,3)+r =19+12+13= 44

Simpul 8 , Lintasan 1,4,5

Reduksi

x	X	x	x	x
12	X	11	x	x
0	3	x	x	2
x	X	x	x	x
x	0	0	x	x

R2-11

Hasil Reduksi

x	x	x	x	x
1	x	0	x	x
0	3	x	x	2
x	x	x	x	x
x	0	0	x	x

Diperoleh r=11. Nilai batas untuk simpul 8 pada pohon ruang status adalah
c(8) = c(4) + B(4,5)+r= 19+0+11=30

Simpul yang memiliki cost terendah pada pohon adalah simpul 6 . Simpul 7 dan simpul 8 memiliki cost lebih besar dari simpul 5, itu berarti simpul 5 pasti lebih optimal daripada simpul 7 dan 8.

Maka kita harus mengekspansi simpul 6 .

Simpul 9 , Lintasan 1,4,2,3
Reduksi

x	x	x	x	x
x	x	x	x	x
x	x	x	x	2
x	x	x	x	x
11	x	0	x	x

R3-2 , R5-11

Hasil Reduksi

x	x	x	x	x
x	x	x	x	x
x	x	x	x	0
x	x	x	x	x
0	x	x	x	x

Diperoleh r=2+11=13 . Nilai batas untuk simpul 9 pada pohon ruang status :
c(9) = c(6) + C(2,3) + r =22+11+13 = 46

Simpul 10, Lintasan 1,4,2,5
Reduksi

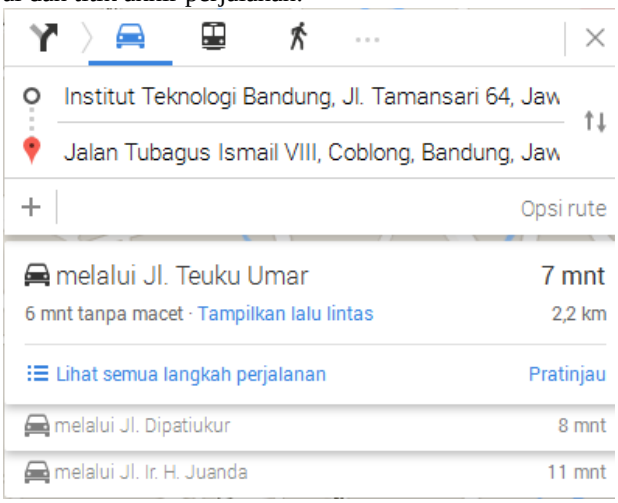
x	x	x	x	x
x	x	x	x	x
0	x	x	x	x
x	x	x	x	x
x	x	0	x	x

Tidak dapat direduksi , sehingga r=0.Nilai batas untuk simpul 10 pada pohon ruang status
c(9) = c(6)+C(2,5)+r=22+0+0 = 22.

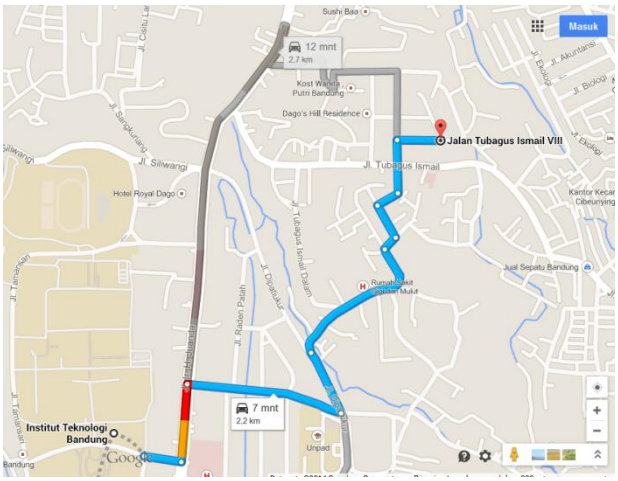
Kita dapat mematikan simpul 9 karena costnya lebih besar dari 5 . Simpul 10 mencapai tujuan dengan cost lebih rendah dari 5 , maka simpul 10 adalah rute paling optimal .

III. APLIKASI PENCARIAN RUTE TERPENDEK

Pada aplikasi google maps ,pencarian dimulai dengan tombol jalan lalu dengan mengisi atau menginput titik awal dan titik akhir perjalanan.



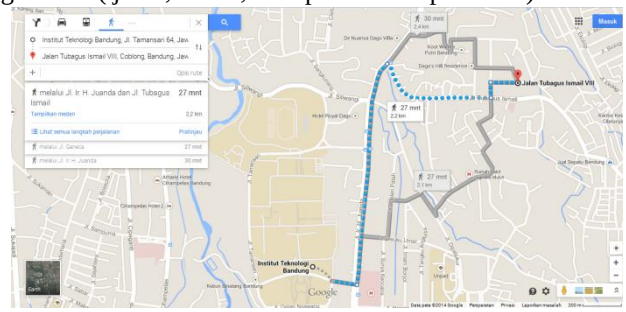
Gambar 3.1 List pilihan jalur paling optimal yang disarankan Google Maps dengan menggunakan Driving Directions



Gambar 3.2 Pilihan jalur yang paling optimal yang disarankan Google Maps dengan menggunakan Driving Directions pada peta

Akan ditunjukan sebuah rute yang dianggap paling optimal , dan beberapa rute alternatif yang mungkin ditempuh .

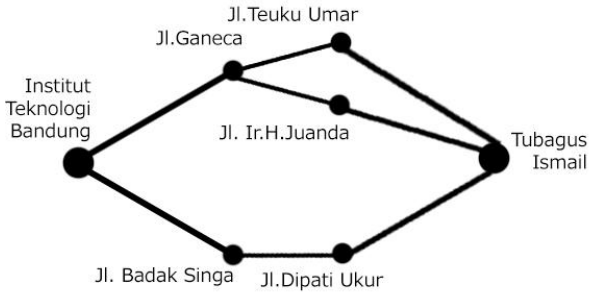
Pada pemilihan rute juga dapat memilih mode yang digunakan (jalan, mobil, atau public transportation).



Gambar 3.2 Pilihan jalur paling optimal yang disarankan Google Maps menggunakan Walking Directions

Pada kasus diatas lokasi yang dicari adalah dari Institut Teknologi Bandung ke Tubagus Ismail.

Jika dibuat graf maka kira-kira seperti ini



Gambar 3.3 Graf Hasil Pencarian Google Maps dari Institut Teknologi Bandung ke Tubagus Ismail

Untuk melewati jalan tersebut melewati jarak tempuh yang berbeda dan waktu tempuh yang berbeda sehingga hal tersebut bisa menjadikan graf diatas graf berbobot berdasarkan waktu atau jarak tempuh.

Pada pencarian rute paling optimal dapat menggunakan Dynamic Programming sehingga terdapat memorisasi penyimpanan bobot dan rute-rute . Namun pada makalah ini membahas Branch and Bound untuk pencarian rute.

IV. KESIMPULAN

Dapat disimpulkan bahwa pada software navigasi digunakan peta dan digunakan algoritma pencarian rute paling optimal. Algoritma pencarian rute sangat banyak jenis nya, seperti Djikstra Algorithm , Breadth First Search, Depth First Search, Branch and Bound, Dynamic Programming. Pada makalah ini dijelaskan Branch and Bound untuk pencarian rute.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan yang Maha Esa, karena oleh karena-Nya makalah ini dapat diselesaikan. Penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir dan Ibu Masayu Leylia Khodra selaku guru dari mata kuliah Strategi Algoritma IF 2211 yang telah mengajari mata kuliah ini sehingga lebih mudah dipahami dan memudahkan dalam pembuatan paper .

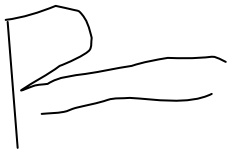
DAFTAR PUSTAKA

[1]Rinaldi, Munir. *Diktat Kuliah IF2211 Strategi Algoritma*. Bandung: Institut Teknologi Bandung. 2009.
[2] http://en.wikipedia.org/wiki/Branch_and_bound
Diakses tanggal 17 Mei 2014
[3]http://www.stanford.edu/class/ee364b/lectures/bb_slides.pdf
Diakses tanggal 17 Mei 2014
[4]https://www.google.co.id/maps/dir/Institut+Teknologi+Bandung,+Jl.+Tamansari+64,+Jawa+Barat+40132/Jalan+Tubagus+Ismail+VIII,+Coblong,+Bandung,+Jawa+Barat+40134/@-6.8873719,107.6115185,16z/data=!4m14!4m13!1m5!1m1!1s0x2e68e65767c9b183:0x2478e3dcdce37961!2m2!1d107.611009!2d-6.89293!1m5!1m1!1s0x2e68e6ffcbf04ca3:0x61dee7f7e0d2c1ed!2m2!1d107.620611!2d-6.884362!3e0
Diakses tanggal 17 Mei 2014

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, 19 Mei 2014



Rita Sarah /13512009