

Paduan Aplikasi Algoritma Branch and Bound dan Backtracking dalam Pencarian n - Jarak Tersingkat dalam Sistem GPS

Ryan Sosiawan¹, Pradita Herdiansyah², Nashirudin Anwar³

*Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganeshha 10, Bandung*

E-mail : if14065@students.if.itb.ac.id¹, if14073@students.if.itb.ac.id²,
if14086@students.if.itb.ac.id³

Abstrak

Dewasa ini, jumlah penggunaan kendaraan bermotor di Indonesia meningkat dengan pesat sehingga mengakibatkan tingkat kemacetan yang sangat tinggi di beberapa ruas jalan raya di beberapa wilayah di Indonesia. Keadaan ini membuat routing perjalanan dari satu tempat ke tempat lain menjadi sangat krusial untuk memperoleh waktu tempuh yang paling singkat. Salah satu solusi dari permasalahan di atas ialah dengan pemanfaatan peta kota yang routingnya dapat dilakukan secara manual. Namun, untuk lebih memudahkan penggunaannya, aplikasi Global Positioning System (GPS) yang dapat menunjukkan jalur terpendek dari satu tempat ke tempat lain dapat dijadikan opsi yang lebih baik. Aplikasi GPS yang ideal selanjutnya, dapat dimanfaatkan untuk menunjukkan n-jalur terpendek yang terurut yang dapat dijadikan alternatif dalam routing jalur jika terjadi hambatan pada rute utama yang merupakan rute terpendek. Pencarian n-jalur terpendek secara terurut tersebut yang dapat dicari solusinya dengan menggunakan paduan algoritma Branch and Bound dengan algoritma Backtracking.

Kata kunci: routing, global positioning system, Branch and Bound, Backtracking

1. Pendahuluan

Sistem yang menyerupai Global Positioning System atau yang sering kita dengar sebagai GPS ini sebenarnya telah muncul pertama kali di bidang militer dan pertahanan di Amerika tepatnya pada saat Perang Dunia II. Sistem tersebut bernama LORAN. Kemudian Uni Sovyet menyusul teknologi tersebut dengan meluncurkan Satelit Sputnik-nya yang di mana pemanfaatannya adalah dengan mempelajari Efek Doppler yang diberikan oleh satelit tersebut.

Seiring perkembangan zaman, pemanfaatan GPS tidak lagi terbatas pada kebutuhan militer saja melainkan telah meluas untuk memenuhi berbagai kebutuhan di berbagai bidang. Beberapa bidang itu antara lain, untuk transportasi, pemetaan (pembacaan wilayah), maupun kompetisi (Rally Dakkar).

Pemanfaatan GPS dalam bidang transportasi dapat dimanfaatkan untuk menggantikan fungsi routing jalan konvensional yang menggunakan peta biasa. Dengan GPS, pengguna yang bersangkutan akan mendapatkan solusi rute yang paling dekat dengan lebih mudah tanpa harus melakukan routing manual pada peta. Metode ini lebih familiar dengan sebutan *Automotive Navigation System*. Cara kerja yang dipakai berdasarkan dengan database jalan yang dimiliki dipadu dengan pengiriman posisi yang dikirim oleh alat GPS.

Berdasarkan kerja GPS yang telah ada, kami mencoba sedikit memodifikasi agar pengguna mendapatkan lebih dari 1 rute untuk mengantisipasi kejadian rute

terbaik tidak dapat ditempuh jika ada halangan. Untuk mendapatkan n-solusi rute terurut dimulai dari solusi yang terbaik, diperlukan suatu algoritma pencarian berdasarkan jarak tempuh rute. Untuk mengatasi masalah itu, kami mencoba menawarkan solusi dengan menggunakan algoritma branch and bound untuk mendapatkan solusi terbaik berdasarkan bobot yang dipadu dengan algoritma backtracking untuk proses pencarian solusi lainnya.

2. Automotive Navigation System

Automotive Navigation System (Sistem Navigasi Otomotif) adalah sebuah sistem navigasi yang berbasis satelit yang dikhususkan penggunaannya pada sebuah kendaraan/alat transportasi. Pada sistem navigasi ini penggunaan dari GPS lebih ditekankan pada pengiriman posisi berdasarkan pantauan satelit untuk diintegrasikan dengan database jalan yang dimiliki penerima (receiver) untuk kemudian ditampilkan ke dalam layar hasil penyatuan tersebut. Namun terkadang sistem navigasi ini masih terhalang oleh kendala cuaca dan letak geografis misalnya di dalam terowongan dan gua atau tempat-tempat yang sekiranya sinyal-sinyal dari satelit tidak dapat diterima.

Sebagai perintis sistem navigasi ini adalah Honda pada tahun 1983 yang digunakan pada tipe Acura Legend, sedangkan yang berbasis pada GPS baru diperkenalkan oleh Pioneer pada tahun 1990.

Database jalan raya yang digunakan harus dibuat berdasarkan detail-detail yang sama dengan keadaan yang sebenarnya lengkap dengan dengan kode-kode secara geografis baik itu garis lintang, garis bujur dan ketinggian dari tiap-tiap spot yang terdapat dalam database jalan raya. Selain itu dalam peta jalan raya tersebut bisa memuat informasi-informasi yang sekiranya bisa bermanfaat bagi pengendara seperti pom bensin, restoran dan tempat istirahat.

3. Algoritma Branch and Bound dan Backtracking

3.1. Algoritma Branch and Bound

Algoritma Branch and Bound merupakan metode pencarian dalam ruang solusi yang ditransformasikan dalam bentuk ruang pohon pencarian. Prinsip algoritma ini sama dengan algoritma BFS (Breadth First Search) hanya saja untuk mendapatkan solusi dengan lebih cepat dan lebih mangkus, tiap simpul pada pohon diberikan cost tertentu. Hal ini berfungsi untuk melakukan proses pembangkitan simpul selanjutnya hanya pada simpul yang memiliki cost terkecil

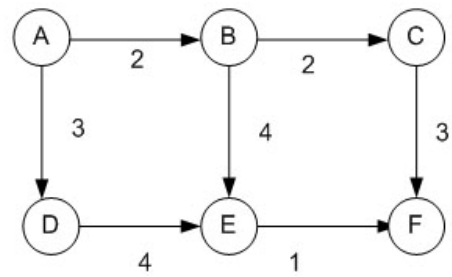
3.2. Algoritma Backtracking

Algoritma Backtracking merupakan algoritma yang berbasiskan DFS (Depth First Search). Yang dilakukan oleh algoritma ini adalah mencari kemungkinan solusi dengan menelusuri hingga node terdalam. Kemudian dilakukan perjalanan kembali dengan melalui node-node calon solusi yang telah dikunjungi untuk menemukan jalur solusi lain yang lebih sesuai. Dengan kata lain, algoritma ini akan melakukan pencarian solusi secara berurutan dari jalur solusi satu ke jalur solusi lain. Namun akan berhenti bila solusi yang sesuai telah ditemukan.

4. Aplikasi Algoritma Branch and Bound (BnB) dan Backtracking dalam Routing GPS

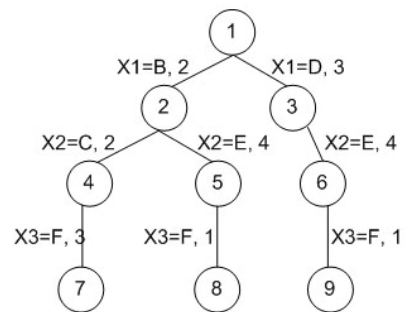
Penyelesaian masalah routing dengan GPS diawali dengan pencarian solusi terbaik dengan algoritma pencarian solusi *Branch and Bounds*. Berikut ini adalah contoh penerapan dari solusi yang kami tawarkan.

Setelah ditemukan solusi terbaik akan kami tawarkan kepada user pengguna apakah solusi tersebut akan digunakan oleh user untuk digunakan atau oleh proses selanjutnya ditemukan bahwa jalur tersebut tidak bisa dilalui dikarenakan oleh suatu hal(ex: macet, kecelakaan, blokir jalan dll) maka akan dijalankan *backtracking* yang akan kemudian dijalankan kembali algoritma BnB untuk mencari solusi terbaik kedua. Hal tersebut kita ulang terus menerus sampai akhirnya ditemukan solusi yang bisa diterima oleh user.



Gambar 4.1

Misalkan pada contoh di atas kita ingin melakukan perjalanan dari titik asal A menuju titik akhir F. Pohon ruang status dari semua solusi yang ada, diharapkan akan seperti ini :



Gambar 4.2

Solusi permasalahan di atas dengan menggunakan algoritma perpaduan antara BnB dengan backtracking adalah sebagai berikut:

1. Memecahkan masalah di atas dengan cara BnB untuk menemukan solusi terbaik. Langkah berikutnya ialah mentransformasikan graf tersebut diatas menjadi matriks ketetangaan seperti pada gambar 4.3.

	A	B	C	D	E	F
A		∞	2	∞	3	∞
B	∞		∞	2	∞	∞
C	∞	∞		∞	∞	3
D	∞	∞	∞		∞	4
E	∞	∞	∞	∞		∞
F	∞	∞	∞	∞	∞	

Gambar 4.3

Reduksi baris untuk tiap baris dari matriks di atas menjadi langkah selanjutnya. Reduksi di sini bertujuan untuk memunculkan nilai 0 pada tiap tiap baris. Maka matriks hasilnya akan menjadi seperti gambar 4.4 di bawah ini.

	A	B	C	D	E	F
A	∞	0	∞	1	∞	∞
B	∞	∞	0	∞	∞	∞
C	∞	∞	∞	∞	0	∞
D	∞	∞	∞	∞	0	∞
E	∞	∞	∞	∞	∞	0
F	∞	∞	∞	∞	∞	∞

Gambar 4.4

Reduksi kolom dilakukan sebagai langkah selanjutnya dengan metode yang sama dengan reduksi baris, hanya saja asosiasi gbaris digantikan oleh kolom. Pada akhirnya kita akan memperoleh matriks tereduksi sebagai berikut.

	A	B	C	D	E	F
A	∞	0	∞	0	∞	∞
B	∞	∞	0	∞	∞	∞
C	∞	∞	∞	∞	0	∞
D	∞	∞	∞	∞	0	∞
E	∞	∞	∞	∞	∞	0
F	∞	∞	∞	∞	∞	∞

Gambar 4.5

2. Pada tahap ini, persamaan fungsi pembatas sebagai berikut diperlukan:

$$C(S) = C(R) + A(I,j) + r$$

Yang dalam hal ini,

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

$C(R)$ = bobot perjalanan minimum yang melalui simpul R (yang dalam hal ini R adalah orangtua S)

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

r = jumlah semua pengurang pada proses memperoleh matriks tereduksi untuk simpul S

Pengerjaan dengan menggunakan rumus di atas adalah sebagai berikut :

$$C(B) = C(A) + A(A,B) + r \quad (I)$$

$$= 13 + 0 + 0 = 13$$

$$C(D) = C(A) + A(A,D) + r \quad (II)$$

$$= 13 + 1 + 0 = 14$$

$$C(C) = C(B) + A(B,C) + r \quad (III)$$

$$= 13 + 0 + 0 = 13$$

$$C(E) = C(B) + A(B,E) + r \quad (IV)$$

$$= 13 + 0 + 0 = 13$$

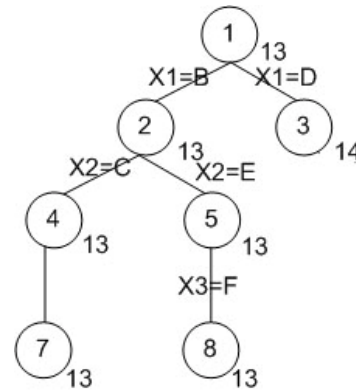
$$C(F) = C(C) + A(C,F) + r \quad (V)$$

$$= 13 + 0 + 0 = 13$$

$$C(F) = C(E) + A(E,F) + r \quad (VI)$$

$$= 13 + 0 + 0 = 13$$

3. Setelah melakukan perhitungan di atas, maka kita akan memperoleh pohon ruang status akhir sebagai berikut



Gambar 4.6

Pada gambar 4.6 di atas kita telah berhasil menemukan dua buah solusi optimum yang berbeda, yaitu A-B-C-F dan A-B-E-F. Dengan algoritma BnB ini, node yang bukan merupakan node dengan bobot terkecil akan dimatikan dan tidak dibangun menjadi pohon yang lebih dalam lagi.

4. Langkah selanjutnya adalah backtracking. Backtracking di sini tidak seperti backtrack ing pada umumnya. Pada algoritma perpaduan ini, backtracking yang terjadi adalah proses runut balik hingga sampai ke node asal (start). Setelah proses backtracking dilalui dan telah kembali ke simpul asal, dilakukan pencarian terhadap semua node yang langkahnya telah dimatikan pada tahap 2. Di antara semua node yang telah dimatikan tersebut, akan dipilih satu node yang memiliki bobot terkecil.

5. Pada gambar di atas, titik 1 adalah node asal. Sedangkan node 6 dan 7 merupakan node tujuan dan rute menuju node node tersebut adalah rute optimal sesuai cara BnB. Backtrack akan terjadi dari node 6 dan 7 berjalan balik bersama-sama menuju node 1 kembali.

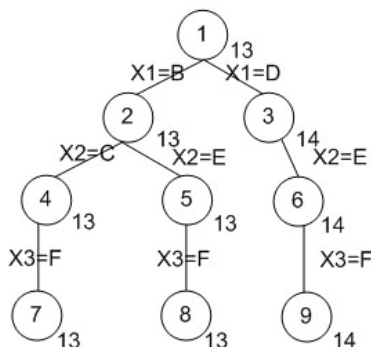
Jika kita perhatikan, node 3 pada gambar 4.6 di atas merupakan node yang dihentikan pembangunannya. Di samping itu, node 3 juga merupakan node yang memiliki cost berbeda sedikit dari node yang kita bangun sebelumnya (node 2). Oleh karena itu, node 3 kita pilih sebagai node berikutnya untuk dibangun dengan menggunakan algoritma BnB.

6. Dengan menggunakan rumusan yang sama, kita dapat melakukan perhitungan untuk pembangunan node 3 sebagai berikut

$$C(E) = C(D) + A(D,E) + r \\ = 14 + 0 + 0 = \mathbf{14}$$

$$C(F) = C(E) + A(E,F) + r \\ = 14 + 0 + 0 = \mathbf{14}$$

7. Setelah perhitungan di atas, pohon ruang status akhir pun juga akan berubah. Hal ini disebabkan adanya pembangunan pada node baru. Pohon ruang status akhir akan menjadi seperti berikut



Gambar 4.7

Solusi alternatif yang kita peroleh adalah: A-D-E-F

8. Secara keseluruhan kita telah mendapatkan 3 buah solusi, yaitu :

- Dua buah solusi optimal : **A-B-C-F** dan **A-B-E-F**
- Satu solusi alternatif tidak optimal: **A-D-E-F**

5. Kesimpulan

Paduan aplikasi algoritma branch and bound dengan backtracking dapat dimanfaatkan dalam kasus pencarian n-jalur dalam penerapannya di perangkat GPS. Secara umum, penerapan algoritma yang kami tawarkan ini dapat dikatakan kurang mangkus, karena proses backtracking yang dilakukan akan kembali lagi ke akar pohon ruang solusi dan tidak bisa langsung menuju node (yang telah dimatikan) yang memiliki bobot terkecil untuk menemukan solusi. Harapan kami, penerapan algoritma ini akan dipadukan dengan kemampuan satelit yang dapat membaca kondisi kelayakan rute. Hal ini dimplementasikan untuk

menghentikan routing jika rute yang didapat sudah dinilai layak untuk dilalui sehingga proses pencarian tidak perlu dilakukan lagi ke rute yang prioritasnya lebih rendah.

6. Daftar Pustaka

1. Munir, Rinaldi.2006. "Strategi Algoritmik". Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
2. http://en.wikipedia.org/wiki/Automotive_navigation_system
Diakses tanggal 16 Mei pukul 20.35 WIB
3. en.wikipedia.org/wiki/Global_Positioning_System
Diakses tanggal 16 Mei pukul 20.35 WIB
4. <http://mathworld.wolfram.com/BranchandBoundAlgorithm.html>
Diakses tanggal 16 Mei pukul 20.35 WIB
5. http://www.codersource.net/csharp_branch_and_bound_algorithm_implementation.aspx
Diakses tanggal 16 Mei pukul 20.35 WIB