

PEMANFAATAN ALGORITMA *BRANCH AND BOUND* UNTUK PENCARIAN RUTE TERBAIK *FOOD DELIVERY*

Anatariani Virniawati

Program Studi Teknik Informatika, Sekolah Tinggi Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesa no. 10, Bandung
e-mail: anatariani@hotmail.com

ABSTRAK

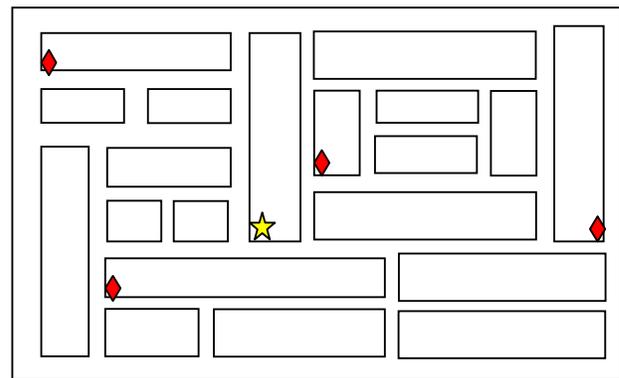
Saat ini, banyak restoran atau rumah makan yang menawarkan layanan *delivery*. Layanan ini memungkinkan pelanggan untuk memesan makanan melalui telepon kemudian pihak restoran akan mengantarkan pesanan ke alamat pelanggan. Proses *food delivery* ini membutuhkan waktu yang cepat untuk menjaga kepuasan pelanggan. Sedangkan untuk pihak restoran sendiri membutuhkan agar proses pengiriman memakan biaya seminim mungkin. Pada makalah ini akan dijelaskan suatu metode untuk mempermudah petugas pengantar untuk menentukan rute terbaik untuk pengantaran pesanan makanan dengan memanfaatkan algoritma *branch and bound* dan pembentukan ruang status dengan skema *Breadth First Search* (BFS).

Kata kunci: *branch and bound*, *breadth first search*, *food delivery*

1. PENDAHULUAN

Kualitas pelayanan merupakan salah satu faktor kunci keberhasilan suatu usaha. Pada restoran atau tempat makan, kecepatan pelayanan merupakan salah satu isu penting untuk meningkatkan kepuasan pelanggannya. Untuk meningkatkan pelayanan terhadap pelanggannya, maka saat ini banyak restoran yang menawarkan jasa *food delivery* yaitu jasa pengantaran makanan ke alamat pelanggan. Pelanggan cukup memesan makanan melalui telepon dan pihak restoran akan mengantarnya.

Pada makalah ini, penulis akan menjelaskan suatu metode penentuan rute perjalanan yang efektif untuk *food delivery*. Perjalanan yang dilakukan oleh petugas pengantar harus memakan waktu dan biaya yang seminimal mungkin. Untuk membuat rute perjalanan ini perlu mempertimbangkan jarak yang perlu ditempuh serta tingkat kepadatan jalan yang akan dilalui untuk mencapai alamat pelanggan.



★ = Restoran

◆ = Rumah pelanggan

Gambar 1. Peta Lokasi Wilayah Pengiriman

2. ALGORITMA *BRANCH AND BOUND*

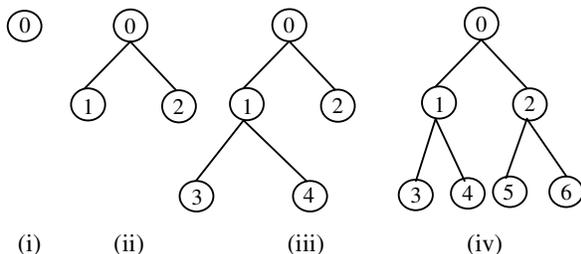
Algoritma *Branch and Bound* (B&B) merupakan suatu metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi diorganisasikan ke dalam pohon ruang status yang dibangun secara dinamis dengan skema *Breadth First Search* (BFS).

Tiap simpul yang dibangkitkan akan diberikan nilai ongkos (*cost*). Untuk mempercepat pencarian solusi maka simpul berikut yang diekspansi tidak berdasarkan urutan pembangkitannya tetapi berdasarkan *cost* yang dimiliki oleh simpul-simpul hidup. *Cost* pada setiap simpul *i* menyatakan perkiraan ongkos termurah lintasan dari simpul *i* ke simpul solusi.

2.1 ALGORITMA *BREADTH FIRST SEARCH*

Pada pembentukan pohon ruang status dengan metode BFS, simpul akar dibangkitkan pertama kali. Kemudian setiap simpul anaknya dibangkitkan, kemudian suksesor

dari setiap simpul anak, dan seterusnya. semua simpul pada aras a dibangkitkan terlebih dahulu sebelum simpul-simpul pada aras $a + 1$. Simpul-simpul yang dibangkitkan diberikan nomor secara melebar dari kiri ke kanan dengan angka 0, 1, 2, ... sesuai dengan urutan pembangkitannya.



Gambar 2. Tahapan pembentukan pohon pencarian BFS

2.2 PRINSIP PENCARIAN PADA ALGORITMA B&B

Pada algoritma B&B, pencarian ke simpul solusi dipercepat dengan memilih simpul hidup berdasarkan nilai ongkos dan tiap simpul X diasosiasikan dengan suatu ongkos yang menyatakan nilai batas (*bound*) yang dapat berupa :

- Jumlah simpul dalam upapohon X yang perlu dibangkitkan sebelum simpul solusi ditemukan.
- Panjang lintasan dari simpul X ke simpul solusi terdekat.

Fungsi heuristik untuk menghitung taksiran nilai ongkos tersebut dinyatakan secara umum sebagai berikut :

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

dimana

$\hat{c}(i)$ = ongkos untuk simpul i

$f(i)$ = ongkos mencapai simpul i dari akar

$g(i)$ = ongkos mencapai simpul tujuan dari simpul i

Nilai \hat{c} digunakan untuk mengurutkan pencarian dimana simpul dengan nilai \hat{c} terkecil akan menjadi simpul yang diekspansi.

2.3 SKEMA UMUM ALGORITMA B&B

Berikut langkah-langkah pencarian solusi dengan menggunakan algoritma B&B :

1. Masukkan simpul akar ke dalam antrian Q . Jika simpul akar adalah simpul solusi maka solusi telah ditemukan dan pencarian berhenti.

2.
 - 2.1 Jika antrian Q kosong maka solusi tidak ada dan pencarian berhenti
 - 2.2 Jika antrian Q tidak kosong maka pilih dari antrian Q simpul i yang mempunyai nilai \hat{c} paling kecil. Jika terdapat beberapa simpul i yang memenuhi maka pilih satu secara sembarang.
3.
 - 3.1 Jika simpul i adalah simpul solusi maka solusi telah ditemukan dan pencarian berhenti.
 - 3.2 Jika simpul i bukan simpul solusi maka bangkitkan semua anaknya. Jika simpul tidak memiliki anak maka kembali ke langkah 2.
4. Untuk setiap anak j dari simpul i , hitung nilai \hat{c} dan masukkan semua anak yang sudah dibangkitkan ke dalam antrian Q .
5. Kembali ke langkah 2.

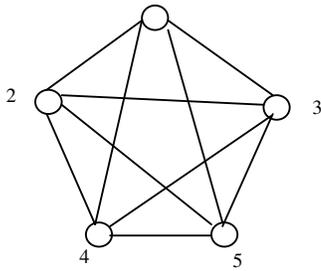
3. PENERAPAN ALGORITMA BRANCH AND BOUND DALAM PENCARIAN RUTE PERJALAN

Untuk menemukan rute perjalanan terbaik untuk *food delivery*, terdapat beberapa parameter yang perlu ditentukan sebelumnya untuk memperhitungkan nilai ongkos (*cost*). Parameter tersebut antara lain adalah jarak antara restoran ke alamat pelanggan, jarak antara alamat pelanggan satu ke pelanggan lainnya serta tingkat kepadatan jalan yang akan dilalui petugas pengantar dimana kepadatan ini dapat berbeda antara dua titik yang dilalui arah yang berbeda.

Rumah dari tiap pelanggan akan diasosiasikan sebagai simpul dan restoran sebagai akar. Peta yang menggambarkan lokasi dari restoran dan rumah pelanggan akan direpresentasikan dalam bentuk graf lengkap.

Misalkan :

- (i) $G = (V, E)$ adalah graf yang merepresentasikan peta lokasi
- (ii) $|V| = n$ = jumlah simpul dalam graf
- (iii) $\hat{c}(i, j)$ = nilai ongkos atau bobot sisi i, j
- (iv) perjalanan (tur) berawal dan berakhir di simpul 1 (restoran)
- (v) S adalah ruang solusi, yang dalam hal ini $S = \{(1, \pi, 1) \mid \pi \text{ adalah permutasi bilangan } (2, 3, \dots, n)\}$
- (vi) $|S| = (n - 1)!$ = banyaknya kemungkinan solusi



Gambar 3. Graf yang merepresentasikan peta lokasi pada Gambar 1

Penghitungan nilai ongkos $\hat{c}(i, j)$ adalah sebagai berikut:

$$\hat{c}(i, j) = d(i, j) + \rho(i, j)$$

dimana

$\hat{c}(i, j)$ = bobot sisi i, j

$d(i, j)$ = jarak pada peta antara simpul i dan simpul j

$\rho(i, j)$ = kepadatan jalan pada peta antara simpul i dan simpul j

Tabel 1. Tabel Bobot Sisi Graf

	$d(i, j)$	$\rho(i, j)$	$\hat{c}(i, j)$
(1,2)	10	3	13
(1,3)	5	5	10
(1,4)	7	3	10
(1,5)	3	8	11
(2,1)	10	9	19
(2,3)	4	5	9
(2,4)	9	8	17
(2,5)	1	1	2
(3,1)	5	4	9
(3,2)	4	8	12
(3,4)	3	1	4
(3,5)	3	9	12
(4,1)	7	3	10
(4,2)	9	7	16
(4,3)	3	4	7
(4,5)	8	2	10
(5,1)	3	7	10
(5,2)	1	3	4
(5,3)	3	6	9
(5,4)	8	8	16

Langkah pertama yang perlu dilakukan adalah hitung bobot dari tiap sisi (*cost* dari 1 titik pada peta ke titik lainnya). Untuk mempermudah pengerjaan pencarian solusi, maka representasikan bobot dari tiap sisi graf kedalam suatu matriks. Berikut sebuah matriks yang merepresentasikan bobot pada graf berarah pada Gambar2.

$$\begin{bmatrix} \infty & 13 & 10 & 10 & 11 \\ 19 & \infty & 9 & 17 & 2 \\ 9 & 12 & \infty & 4 & 12 \\ 10 & 16 & 7 & \infty & 10 \\ 10 & 4 & 9 & 16 & \infty \end{bmatrix}$$

Untuk menentukan nilai \hat{c} untuk simpul akar maka perlu dilakukan reduksi baris dan reduksi kolom. Jumlah elemen pengurang baik untuk baris maupun kolom akan menjadi batas bawah dari perjalanan dengan total bobot minimum yaitu \hat{c} untuk simpul akar.

Reduksi baris :

$$\begin{bmatrix} \infty & 13 & 10 & 10 & 11 \\ 19 & \infty & 9 & 17 & 2 \\ 9 & 12 & \infty & 4 & 12 \\ 10 & 16 & 7 & \infty & 10 \\ 10 & 4 & 9 & 16 & \infty \end{bmatrix} \begin{matrix} R_1 - 10 \\ R_2 - 2 \\ R_3 - 4 \\ R_4 - 7 \\ R_5 - 4 \end{matrix} \begin{bmatrix} \infty & 3 & 0 & 0 & 1 \\ 17 & \infty & 7 & 15 & 0 \\ 5 & 8 & \infty & 0 & 8 \\ 3 & 9 & 0 & \infty & 3 \\ 6 & 0 & 5 & 12 & \infty \end{bmatrix}$$

Kemudian reduksi kolom :

$$\begin{bmatrix} \infty & 3 & 0 & 0 & 1 \\ 17 & \infty & 7 & 15 & 0 \\ 5 & 8 & \infty & 0 & 8 \\ 3 & 9 & 0 & \infty & 3 \\ 6 & 0 & 5 & 12 & \infty \end{bmatrix} C_1 - 3 \begin{bmatrix} \infty & 3 & 0 & 0 & 1 \\ 14 & \infty & 7 & 15 & 0 \\ 2 & 8 & \infty & 0 & 8 \\ 0 & 9 & 0 & \infty & 3 \\ 3 & 0 & 5 & 12 & \infty \end{bmatrix}$$

Total jumlah pengurangan adalah = $(10 + 2 + 4 + 7 + 4) + (3) = 30$. Maka batas untuk simpul akar ($\hat{c}(\text{root})$) = 30.

Ini berarti bahwa rute yang akan dibentuk paling tidak memiliki total bobot minimum 30.

Langkah selanjutnya adalah, misalkan A adalah suatu matriks tereduksi untuk simpul X . Misalkan Y adalah anak dari simpul X sehingga sisi (X, Y) pada pohon ruang status berkoresponden dengan sisi (i, j) pada peta lokasi. Jika Y bukan simpul daun, maka perlu dilakukan :

- (i) Ubah semua nilai pada baris i dan kolom j menjadi ∞ agar tidak adalintasan yang keluar dari simpul i atau masuk pada simpul j .
- (ii) Ubah $A(j, 1)$ menjadi ∞ untuk mencegah penggunaan sisi $(j, 1)$
- (iii) Reduksi kembali semua baris dan kolom pada matriks A kecuali elemen ∞ .

Jika z adalah total semua pengurang maka nilai batas untuk simpul Y adalah :

$$\hat{c}(Y) = \hat{c}(X) + A(i, j) + z$$

dimana

$\hat{c}(Y)$ = bobot perjalanan minimum yang melalui simpul Y

$\hat{c}(X)$ = bobot perjalanan minimum yang melalui simpul X

$A(i, j)$ = bobot sisi (i, j) pada graf G yang

berkoresponden dengan sisi (X, Y) pada pohon ruang status
 z = jumlah semua pengurang.

Setelah itu, pilih simpul hidup yang memiliki nilai batas terkecil. Simpul tersebut akan menjadi simpul-E yang kemudian akan diekspansi. Ulangi langkah ini hingga ditemukannya simpul daun. Solusi yang ditemukan pada tahap ini merupakan solusi terbaik sampai saat itu.

Semua simpul hidup yang nilai batasnya lebih besar dari simpul terakhir yang dikunjungi dibunuh (*bound*). Ulangi langkah-langkah tersebut hingga tidak terdapat lagi simpul hidup di dalam pohon ruang status.

IV. KESIMPULAN

Algoritma *Branch and Bound* (B&B) untuk menyelesaikan permasalahan pencarian rute terbaik untuk *food delivery* sangat efektif jika dibandingkan dengan algoritma *Brute Force*. Hal ini dapat dilihat dari kompleksitas waktu yang dibutuhkan untuk pencarian solusi.

Proses pencarian solusi dengan menggunakan algoritma B&B dapat dipercepat karena adanya pemilihan simpul yang akan diekspansi sehingga simpul-simpul yang tidak menuju solusi tidak perlu diproses lebih lanjut. Parameter perhitungan *cost* untuk tiap simpul tidak hanya berdasarkan jarak namun juga memasukkan faktor kepadatan lalu lintas yang akan dilewati oleh petugas pengantar.

REFERENSI

- [1] Munir, Rinaldi. Diktat Kuliah IF2251 *Strategi Algoritmik*. Institut Teknologi Bandung. 2007.
- [2] Wiener, Richard. *Branch and Bound Implementations for the Travelling Sales Person Problem*. Journal of Object Technology. Volume 2. No. 6. 2003.