

Aplikasi Algoritma B&B untuk Memperoleh Poin Maksimum pada Permainan “Diner Dash”

Chandra Sutikno Oemaryadi

Program Studi Teknik Informatika ITB
Alamat : Veteran 84, Bandung 40112
e-mail: chandra_oey@yahoo.com

ABSTRAK

Algoritma *Branch and Bound*, atau yang umum dikenal dengan nama B&B, dikemukakan oleh A. H. Land dan A. G. Doig pada tahun 1960 untuk menyelesaikan masalah pemrograman linear. Algoritma B&B merupakan metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi diorganisasikan ke dalam pohon ruang status. Pohon ruang status pada algoritma B&B dibangun dengan skema DFS (*Depth First Search*), sedangkan pada algoritma runut-balik dibangun dengan skema BFS (*Breadth First Search*).

Algoritma *Branch and Bound* memiliki kompleksitas waktu lebih baik daripada algoritma runut-balik, DFS murni, maupun BFS murni. Hal ini dikarenakan algoritma B&B tidak memperluas (meng-*expand*) seluruh simpul pada pohon ruang status, melainkan hanya simpul-simpul yang memenuhi suatu kriteria tertentu.

Algoritma *Branch and Bound* dapat diaplikasikan untuk menemukan solusi optimal dari berbagai masalah optimasi, terutama masalah optimasi diskrit dan kombinatorial. Masalah perolehan keuntungan maksimum dalam permainan “Diner Dash” merupakan masalah optimasi, yang dapat diselesaikan dengan menggunakan algoritma *Branch and Bound*.

Kata kunci: *Branch and Bound*, pemrograman linear, optimasi, Diner Dash.

1. PENDAHULUAN

Masalah optimasi seringkali menjadi momok bagi pengusaha yang ingin memperoleh keuntungan sebanyak-banyaknya. Dalam permainan “Diner Dash”, pemain berperan sebagai seorang pengusaha rumah makan yang harus melayani tamu-tamunya.

Tamu-tamu yang akan berkunjung ke rumah makan terdiri dari beberapa jenis, antara lain tamu normal, tamu lambat,

dan tamu cepat. Setiap jenis tamu memiliki karakteristik tersendiri, dilihat dari tingkat kesabaran, waktu yang dibutuhkan untuk menghabiskan santapan, dan nominal poin yang akan diberikan kepada pemain.

Banyaknya tamu yang berkunjung ke rumah makan tidak sebanding dengan jumlah meja yang tersedia. Pemain yang tidak memperoleh meja akan menunggu, namun akan meninggalkan rumah makan apabila kesabartannya habis. Tantangan terbesar bagi pemain adalah menentukan pemilihan urutan tamu yang akan dilayani guna mengeruk poin sebanyak-banyaknya.

Masalah pemilihan urutan tamu ini dapat dikategorikan sebagai masalah optimasi. Masalah ini akan diselesaikan dengan menggunakan algoritma *Branch and Bound*.

2. METODE

2.1 Deskripsi Umum Algoritma B&B

Algoritma *Branch and Bound* pertama kali dikemukakan oleh A. H. Land dan A. G. Doig pada tahun 1960 untuk menyelesaikan masalah pemrograman linear. Dalam matematika, pemrograman linear meliputi seluruh masalah optimasi dan fungsi-fungsi linear, yang dapat dinyatakan sebagai persamaan linear dengan batasan-batasan tertentu. Pemrograman linear umumnya digunakan untuk menemukan cara untuk memperoleh hasil optimum, misalnya keuntungan maksimal atau biaya minimal, dengan sejumlah sumber daya (misalnya waktu, uang, dan tempat) yang diberikan.

Algoritma B&B merupakan metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi diorganisasikan ke dalam pohon ruang status. Berbeda dengan algoritma runut-balik, yang membentuk pohon ruang status dengan menggunakan skema BFS (*Breadth First Search*), algoritma B&B membentuk pohon status dengan menggunakan skema DFS (*Depth First Search*). Algoritma B&B memiliki kompleksitas waktu lebih baik dibandingkan dengan algoritma runut-balik, BFS murni, maupun DFS murni, karena adanya fungsi pembatas, yang akan dijelaskan pada bagian berikutnya.

Berdasarkan penyelidikan *Optimization Technology Center*, algoritma *Branch and Bound* merupakan algoritma yang paling banyak diimplementasikan di dalam perangkat lunak. Algoritma B&B telah terbukti cukup sangkil untuk menyelesaikan masalah praktis.

Berdasarkan riset yang telah dilakukan oleh SEAS, *School of Engineering and Applied Science*, B&B merupakan algoritma yang *powerfull*, karena dapat diaplikasikan dalam berbagai masalah yang tidak dapat diselesaikan dengan menggunakan metode *greedy* dan pemrograman dinamis.

2.2 Prinsip Pencarian Solusi pada Algoritma B&B

Simpul hidup pada pohon ruang status yang dibentuk oleh algoritma B&B memiliki ongkos tertentu, yang dinotasikan dengan simbol $c(i)$. i menyatakan simbol simpul, $c(i)$ menyatakan nilai batas (*bound*) dari simpul i .

Tiga jenis simpul yang terdapat dalam pohon ruang status :

- Simpul hidup, yaitu simpul yang belum diperluas (*di-expand*)
- Simpul mati, yaitu simpul yang tidak dapat diperluas lagi
- Simpul-E, yaitu simpul hidup dengan ongkos terbaik (umumnya terbesar atau terkecil), yang akan diperluas

Algoritma B&B memiliki tiga buah prosedur yang digunakan dalam membentuk pohon ruang status, antara lain :

- *Branching*, atau perluasan
Branching adalah proses pembentukan pohon ruang solusi dengan memperluas simpul-E yang terdapat pada pohon ruang solusi.
- *Bounding*, atau pemberian nilai batas
Bounding adalah proses pemberian nilai batas pada semua simpul hidup yang terdapat pada pohon ruang solusi.
- *Pruning*, atau penghapusan
Pruning adalah penghapusan simpul hidup dari daftar simpul yang akan diperluas, membuat simpul tersebut menjadi simpul mati.

Pruning dapat dikatakan sebagai proses mematikan simpul, misalnya simpul x , dimana program menganggap simpul x tidak akan menghasilkan status akhir (solusi), sehingga simpul x tidak akan diperluas oleh program. Umumnya program akan memperluas simpul yang memiliki ongkos terbesar atau terkecil, dan melakukan *pruning* pada simpul-simpul sisanya. Proses *pruning* ini

mengakibatkan kompleksitas waktu algoritma B&B jauh lebih baik dibandingkan algoritma runut-balik, yang sama-sama membangun pohon ruang solusi untuk memperoleh penyelesaian dari permasalahan.

Ongkos dari setiap simpul hidup dihasilkan dalam proses *bounding*. Notasi umum dari ongkos :

$$c(i)=f(i)+g(i)$$

$f(i)$ = ongkos untuk mencapai simpul i dari akar
 $g(i)$ = ongkos untuk mencapai simpul tujuan dari akar
 $c(i)$ = ongkos total simpul i

Definisi dari $f(i)$ dan $g(i)$ dapat berbeda-beda, sesuai dengan persoalan yang dihadapi.

2.3 Deskripsi Permainan "Diner Dash"

Diner Dash adalah permainan yang dipopulerkan oleh GameHouse. "Diner Dash" dapat dikatakan cukup sukses di pasaran, terbukti dengan dirilisnya sekuel ketiga dari permainan ini, yaitu "Diner Dash : Hometown Hero". Dengan cara bermain yang tidak terlalu sulit, fitur-fitur yang menarik, dan tingkat kesulitan permainan yang bervariasi, permainan ini dapat membuat pemain betah berjam-jam di depan layar komputer. Berikut adalah tampilan umum dari permainan ini :



Gambar 1 Tampilan umum permainan Diner Dash : Hometown Hero

Dalam permainan "Diner Dash", pemain berperan sebagai seorang pengelola rumah makan, yang bernama Flo. Sebagai Flo, pemain harus melayani berbagai jenis tamu yang berkunjung ke rumah makan, dengan jumlah meja yang terbatas. Tantangan yang dihadapi pemain adalah memperoleh poin sebanyak-banyaknya, dengan menentukan urutan tamu yang dilayani oleh Flo.

Berdasarkan karakteristik, tamu dapat dibedakan menjadi tiga jenis, antara lain :

- Tamu cepat(fst)
- Tamu normal(nor)
- Tamu lambat(slw)

Setiap jenis tamu memiliki tingkat kesabaran(pat), lama waktu yang diperlukan untuk menghabiskan santapan(eat), dan pemberian poin(pro) untuk pemain, yang berbeda-beda. Kesabaran yang masih dimiliki oleh tamu dilambangkan dengan *icon* hati (lihat gambar 2).

	pat(s)	eat(s)	pro
nor	50	40	150
slw	60	70	200
fst	40	40	150

Tabel 1 Tabel properti dibagi per jenis tamu

Poin yang diberikan oleh tamu (pro) adalah hasil penjumlahan poin yang diberikan ketika tamu memesan makanan, makanan diantarkan, tamu selesai makan, dan ketika Flo membersihkan meja yang sudah digunakan oleh tamu. Poin diakumulasikan ketika tamu sudah menyelesaikan hidangannya.



Gambar 2 tamu cepat, tamu normal, dan tamu lambat

Tamu menyatakan **sekumpulan orang** yang akan bersantap di rumah makan. Tamu dapat terdiri dari **satu atau dua orang** dengan **jenis sama**. Bersesuaian dengan hal ini, meja yang tersedia yang hanya mampu menampung maksimal dua orang.

3. PENYELESAIAN DENGAN *BRANCH AND BOUND*

3.1 Pengaplikasian Algoritma B&B pada Permainan Diner Dash

Seperti telah disebutkan di atas, masalah perolehan keuntungan maksimum pada game “Diner Dash”, akan didekati dengan menggunakan algoritma *Branch and Bound*.

Properti :

1. Solusi permainan
Permainan terselesaikan jika tidak ada lagi tamu yang dapat dilayani
2. Fungsi pembangkit
Membangkitkan segala kemungkinan langkah pada setiap langkahnya. Contohnya, pada keadaan antrian dua tamu lambat, satu tamu cepat, dan dua buah meja, terdapat 2 kemungkinan langkah berikutnya, yaitu dua meja diisi dua tamu lambat dan dua meja diisi seorang tamu lambat dan seorang tamu cepat.
3. Fungsi pembatas
Fungsi untuk membatasi sejauh mana algoritma B&B akan menelusuri setiap kemungkinan yang ada. Misalnya ada batasan bahwa banyak tamu yang dilayani tidak boleh melebihi meja yang tersedia.

Setiap simpul yang terdapat dalam pohon ruang solusi memiliki properti nomor simpul, satu buah tabel dan ongkos dari simpul tersebut. Nomor simpul adalah urutan pembangkitan dari simpul.

Tabel berisi keadaan tamu-tamu yang mengantri dan tamu-tamu yang sedang bersantap. Setiap isi dari sel tabel menyatakan tingkat kesabaran dari tamu yang sedang mengantri(pat), dan sisa waktu yang dibutuhkan tamu yang sedang menyelesaikan santapannya(eat). Meja yang kosong dinyatakan secara implisit di dalam tabel, berupa selisih banyak meja dan banyak tamu yang sedang bersantap.

	fst	nor	slw
eat	10		
pat		30	20,30

Tabel 2 Tabel yang terdapat pada simpul hidup

Contohnya, nilai 10 pada sel[fst][eat] menyatakan terdapat satu tamu cepat yang sedang bersantap, dan akan menyelesaikan santapannya dalam 10 detik. Nilai 20, 30 pada sel[slw][pat] menyatakan terdapat dua tamu lambat yang sedang mengantri dengan kesabaran tersisa masing-masing 20 dan 30. Jika kesabaran sudah habis, tamu akan meninggalkan rumah makan. Jika banyak meja yang terdapat adalah 3, maka dengan melihat tabel 2, kita dapat mengambil kesimpulan bahwa masih terdapat dua meja kosong yang dapat diisi oleh tamu yang sedang mengantri.

Setiap simpul yang diperluas menyatakan keadaan di mana setidaknya satu tamu telah menyelesaikan hidangannya, alias terdapat setidaknya satu meja yang kosong dan dapat ditempati oleh tamu yang sedang mengantri. Hal ini berarti kondisi waktu pada tiap simpul tidaklah selalu sama.

Nilai yang terdapat pada cabang menyatakan pilihan tamu yang dilayani, diurutkan sebagai tamu cepat, tamu normal, tamu lambat. Misal, nilai 2,0,0, berarti melayani 2 tamu cepat, 0 tamu normal, dan 0 tamu lambat.

Ongkos dari setiap simpul dinyatakan sebagai :

$$c(i) = f(i) + g(i)$$

$f(i)$ adalah poin yang sudah diperoleh oleh pemain, poin diperoleh ketika tamu sudah menyelesaikan santapannya
 $g(i)$ adalah poin maksimum yang masih dapat diperoleh oleh pemain, yaitu penjumlahan poin yang (akan) diberikan oleh tamu yang sedang mengantri
 $c(i)$ adalah hasil penjumlahan $f(i)$ dan $g(i)$
 Simpul-E adalah simpul hidup yang memiliki ongkos terbesar.

Tahap-tahap pencarian solusi :

1. Memperluas simpul-E untuk setiap kemungkinan kombinasi penempatan tamu pada meja
2. Menghitung ongkos untuk setiap simpul hidup
Simpul hidup adalah kondisi di mana setidaknya satu tamu telah selesai makan.
3. Menentukan simpul-E, yaitu simpul hidup dengan ongkos terbesar
4. Mematikan semua simpul hidup dengan ongkos lebih kecil daripada simpul-E
5. Mengulangi tahap 1 s/d 4 hingga mencapai solusi permainan

3.2 Contoh Kasus

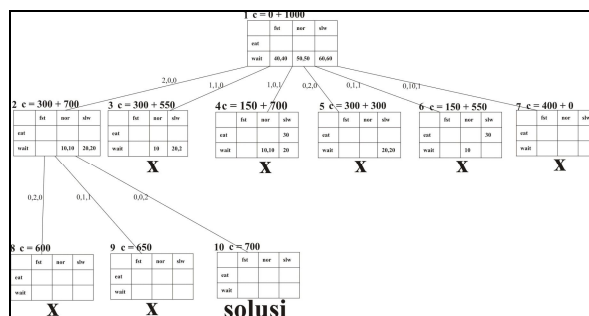
Berikut dikemukakan contoh kasus keadaan pada permainan Diner Dash, dan penyelesaian pencarian keuntungan terbesar dengan menggunakan algoritma B&B.

Properti :

1. Meja : 2 buah
2. Tamu : 6 tamu, 2 tamu cepat, 2 tamu normal, 2 tamu lambat

Simpul akar adalah simpul di mana belum satupun tamu memperoleh meja.

Berdasarkan tahap-tahap yang telah dijelaskan pada sub-bab 3.1, akan terbentuk suatu pohon ruang status sebagai berikut :



Gambar 3 Pohon ruang status untuk contoh kasus

Gambar 3 dalam ukuran yang diperbesar dilampirkan pada halaman terakhir makalah.

Berdasarkan pohon ruang status yang dibentuk dengan menggunakan algoritma *Branch and Bound*, keuntungan maksimal yang dapat diperoleh adalah 700, dengan urutan tamu yang dilayani :

1. dua tamu cepat
2. dua tamu lambat

Pada simpul (2), kita dapat melihat “ $c = 300 + 700$ ”. Nilai 300, merupakan $f(2)$, menyatakan total poin yang telah diterima (dari dua tamu cepat yang telah menyelesaikan santapan). Nilai 700, merupakan $g(2)$, menyatakan total poin yang masih dapat diperoleh pemain, dengan menjumlahkan poin yang (akan) diberikan oleh tamu yang sedang mengantri (dua tamu sedang dan dua tamu lambat).

Antara simpul (1) dan (2), kita dapat melihat nilai 2,0,0. Seperti telah dijelaskan, nilai pertama menyatakan banyak tamu cepat yang dilayani, nilai kedua menyatakan banyak tamu normal yang dilayani, dan nilai ketiga menyatakan banyak tamu lambat yang dilayani. Nilai 2,0,0 menyatakan program melayani 2 tamu cepat, 0 tamu normal, dan 2 tamu lambat. Simpul(2) merupakan kondisi di mana 2 tamu cepat telah selesai bersantap, dan pemain memperoleh poin sebesar $2 * 150$.

IV. KESIMPULAN

Berdasarkan contoh kasus, kita dapat melihat bahwa algoritma *Branch and Bound* dapat membantu pemain dalam menentukan tamu yang dilayani oleh Flo. Jika menggunakan metode DFS murni, waktu yang dibutuhkan untuk mencapai simpul solusi tidak sedikit, karena harus mencoba seluruh kemungkinan dari setiap simpul. Dengan menggunakan metode B&B, simpul yang diperluas hanya simpul dengan ongkos terbesar.

Kesimpulan secara keseluruhan :

1. *Algoritma Branch and Bound* membangun pohon ruang status dengan menggunakan skema DFS.
2. *Algoritma Branch and Bound* merupakan algoritma dengan kompleksitas waktu lebih baik daripada algoritma runut-balik, DFS murni, dan BFS murni, karena tidak memperluas seluruh simpul hidup, melainkan simpul hidup dengan ongkos optimum.
3. *Algoritma Branch and Bound* dapat diaplikasikan pada permainan “Diner Dash” untuk menentukan tamu yang dilayani, sehingga pemain akan memperoleh keuntungan maksimum.

REFERENSI

- [1] Munir, Rinaldi, “Diktat Kuliah IF2251 Strategi Algoritmik”,2006.
- [2] www-fp.mcs.anl.gov diakses 16 Mei 2008 pukul 14.32.
- [3] www.seas.gwu.edu diakses 16 Mei 2008 pukul 14.19.
- [4] www.stanford.edu diakses 16 Mei 2008 pukul 14.32.
- [5] www.wikipedia.org diakses 11 Mei 2008 pukul 14.31.

