

Penerapan Algoritma *Branch & Bound* dalam Regulasi Ruang Parkir Pesawat Terbang di Bandara

Moh Taufan Tripurnasatria

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10, Bandung
Email : if15121@students.if.itb.ac.id

ABSTRAK

Industri transportasi udara yang berkembang pesat ternyata tidak diiringi dengan peningkatan kapasitas ruang parkir pesawat terbang di bandara. Untuk itu, perlu dilakukan regulasi yang intensif guna memanfaatkan ruang yang ada untuk ditempati pesawat sebanyak-banyaknya, sekaligus meningkatkan keuntungan yang diperoleh pihak administrasi bandara. Dalam makalah ini, digunakan algoritma *branch and bound* untuk menyelesaikan persoalan optimasi yang diberikan. Algoritma ini didasarkan pada skema *Breadth First Search (BFS)*. Dengan algoritma ini, pembangkitan pohon ruang status tidak terlalu banyak seperti pada *BFS* murni karena adanya fungsi *cost* yang memberikan ongkos(*cost*) yang diperlukan untuk membangkitkan simpul solusi selanjutnya sehingga jumlah simpul dapat diminimalkan.

Kata Kunci : *Branch and Bound*, Regulasi, Ruang Parkir.

1. PENDAHULUAN

Industri transportasi, khususnya transportasi udara saat ini berkembang dengan pesatnya. Hal ini disebabkan meningkatnya populasi manusia, serta meningkatnya taraf hidup masyarakat di berbagai negara sehingga perpindahan tempat dalam waktu singkat menjadi salah satu 'kebutuhan pokok' manusia. Selain itu, meningkatnya kemampuan raksasa industri pesawat terbang seperti BOEING dan AIRBUS dalam memproduksi pesawat-pesawat yang kompetitif baik dari segi ukuran, efisiensi, maupun harga, menjadikan bidang ini semakin menggairahkan.

Namun, perkembangan industri pesawat terbang ternyata tidak serta merta diikuti oleh kemampuan bandar udara untuk menampungnya. Selain karena ukuran pesawat saat ini yang semakin 'rakus' ruang parkir, hal ini juga disebabkan perluasan bandara

seringkali menghadapi kendala yang sangat signifikan, baik karena letaknya ada yang di tengah perkotaan, maupun karena pembebasan lahan yang terkendala oleh keadaan sosial ekonomi masyarakat setempat yang menentangnya (seperti di Indonesia).

Untuk itu, regulasi (pengaturan) ruang parkir pesawat di bandara menjadi salah satu alternatif solusi, jika bukan satu-satunya yang dapat diandalkan untuk memanfaatkan sebanyak mungkin ruang yang tersedia untuk memarkir pesawat yang ada. Namun, tentu saja masalah ini tergolong masalah yang sangat rumit. Selain karena banyaknya kendala yang dihadapi, juga karena keinginan pihak administrasi bandara untuk mendapatkan keuntungan semaksimal mungkin. Salah satu pemecahannya adalah dengan memanfaatkan algoritma *branch and bound (b&b)*. Namun, algoritma ini bukanlah satu-satunya metode pemecahan masalah yang dapat digunakan. Selain *b&b*, dapat pula digunakan algoritma *greedy*, atau *dynamic programming*.

2. DESKRIPSI PERSOALAN

Diberikan sebuah bandara, yang memiliki ruang parkir untuk n buah pesawat. Tujuan kita adalah bagaimana memanfaatkan ruang parkir tersebut agar diperoleh keuntungan yang maksimum bagi pihak administrasi bandara. Dalam keadaan tertentu, sebuah pesawat yang berukuran besar dapat melakukan *overlapping* terhadap ruang parkir di sebelahnya.

2.1 Kendala (*constraints*)

1. Setiap pesawat, dengan nomor penerbangan tertentu memiliki waktu kedatangan dan waktu keberangkatan tertentu yang kadang tidak sesuai dengan jadwal yang telah ditentukan.
2. Keadaan fisik pesawat yang berbeda-beda antara satu pesawat dengan pesawat lainnya. Perbedaan ini dapat

berupa berat pesawat, dan ukuran (dimensi) pesawat. Hal ini akan sangat mempengaruhi keberadaan pesawat dalam penentuan ruang parkir.

2.2 Kelayakan (*feasibility*)

Solusi dikatakan layak apabila :

1. Setiap pesawat terbang ditempatkan pada tempat yang sesuai dengan ukurannya dan kondisinya.
2. Tidak ada interval waktu parkir pesawat yang saling beririsan, artinya jadwal keberangkatan suatu pesawat di suatu posisi harus lebih kecil dari waktu kedatangan pesawat lain yang telah dijadwalkan akan memakai tempat tersebut.

3. RUANG LINGKUP

Dalam menyelesaikan permasalahan ini, penulis membuat asumsi-asumsi sebagai berikut:

1. Semua *airline* (penyedia jasa penerbangan) dianggap memberikan biaya sewa parkir yang sama kepada pihak bandara.
2. Kondisi pesawat tidak diperhitungkan apakah perlu perawatan (sehingga diharuskan dekat dengan hanggar), atau tidak
3. Manuver pesawat tidak diperhitungkan.

4. ALGORITMA *BRANCH AND BOUND*

Algoritma *branch and bound* merupakan metode pencarian di dalam ruang solusi secara sistematis. Algoritma ini didasarkan atas skema *Breadth First Search (BFS)*. Untuk mempercepat pencarian ke simpul solusi, maka setiap simpul diberi sebuah nilai ongkos (*cost*). Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan urutan pembangkitannya (sebagaimana pada *BFS* murni), tetapi simpul yang memiliki ongkos yang paling kecil (*least cost search*).

Nilai ongkos pada setiap simpul i menyatakan taksiran ongkos termurah lintasan dari simpul ke i ke simpul solusi (*goal node*).

$C(i)$ = nilai taksiran lintasan termurah dari simpul status i ke status tujuan

Dengan kata lain, $\hat{c}(i)$ menyatakan batas bawah (*lower bound*) dari ongkos pencarian solusi dari status i .

5. METODE

Sebelum menyelesaikan permasalahan ini terlebih dahulu dihitung fungsi kendala yang ada. Hal ini dilakukan untuk menghitung *cost* yang diperlukan dalam pembentukan pohon.

```
function constraint(input : p : pesawat)->
boolean
{mengembalikan true apabila tidak sesuai
dengan kriteria yang ditetapkan}
```

Selanjutnya, dihitung *cost* yang diperlukan simpul status untuk sampai ke simpul solusi.

```
function cost(input : p : pesawat) ->
integer
{mengembalikan mengembalikan waktu kosong
yang ada dalam selang waktu keberangkatan
pesawat dengan kedatangan pesawat lain,
juga diperhitungkan dimensi pesawat(akan
menambah cost) karena dapat terjadi
overlapping dengan tetangganya, hal ini
dapat mengurangi keuntungan pihak
administrasi bandara}
```

$C(i) = cost(p);$

Selanjutnya, dilakukan proses pembangkitan pohon ruang status untuk setiap status yang ada. Dalam menggunakan algoritma *BFS*, kita memerlukan pohon untuk menyimpan status setiap simpul agar memudahkan kita dalam membangkitkan anak-anak setiap simpul.

Setiap simpul menyatakan status pesawat apakah diterima pada posisi tersebut atau tidak.

Aturan pencabangan (*branch*): simpul yang akan dibangkitkan pada suatu level pada pohon ditentukan dari apakah simpul tersebut tidak terkendala, artinya jika dikenakan fungsi *constraint* di atas harus mengembalikan *false*. Jika tidak maka anak-anaknya tidak boleh dibangkitkan.

Pada pohon ruang status, ditetapkan bahwa akar adalah keadaan di mana tidak ada pesawatpun yang dipilih pada posisi tersebut.

Selanjutnya, pembangkitan pohon dapat dilakukan secara dinamis.

Pada setiap simpul ditetapkan *bound (cost)* dari fungsi tadi. Jadi, setiap pembangkitan simpul tertentu dilakukan pemanggilan terhadap fungsi *cost*. Dalam membangkitkan simpul selanjutnya, dilihat dulu *cost* yang paling rendah. Dari situ, kita dapat memulai untuk membangkitkan simpul lain. Sementara simpul yang tidak mengarah ke solusi, harus 'dibunuh'. Setiap selesai membangkitkan simpul anak, diperiksa lagi simpul yang memiliki *cost* paling rendah.

Pada akhir solusi (pada daun) akan didapatkan himpunan pesawat yang akan mendapat tempat tertentu dalam bandara. Hal ini diperoleh dengan menelusuri *path* (jalur) dari akar sampai ke daun.

Hasil ini dapat dikembangkan lebih jauh lagi dengan mencari solusi lain. Ini dimaksudkan untuk memperoleh keuntungan yang lebih besar lagi.

6. KESIMPULAN

Algoritma *branch and bound* dapat dijadikan sebagai alternatif dalam melakukan regulasi ruang parkir pesawat terbang di bandar udara. Dengan menggunakan algoritma ini, pencarian solusi yang tidak perlu dapat 'dibunuh', sehingga diperoleh hasil yang lebih cepat. Selain itu, algoritma ini juga memungkinkan untuk dicari lagi solusi lain. Dalam hal ini digunakan untuk memaksimalkan keuntungan yang diperoleh. Cara yang digunakan tidak harus berupa *branch and bound*, tetapi dapat juga cara heuristik, greedy, ataupun program dinamis. Oleh karena itu, algoritma *branch and bound* ini dapat digunakan bersama dengan algoritma lain

7. REFERENSI

- [1] Filippi Carlo, et.al. "Real Time Optimal Assignment of Transit Aircraft to Overlapping Parking Places in an Airport". Department of Pure and Applied Mathematics, University of Padova.
- [2] Munir, Rinaldi. Strategi Algoritmik. 2007. Teknik Informatika, Institut Teknologi Bandung : Bandung.

Nama, "Judul Makalah", Na