

Algoritma Greedy untuk Penjadwalan Penerbangan di Gerbang - Gerbang Bandara

Robertus Theodore-13509008
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
robertus.t@students.itb.ac.id

Abstrak—Dewasa ini, perkembangan industri penerbangan amatlah pesat. Operasional harian dari bandara termasuk maskapainya, dituntut untuk ekstra efektif dan efisien. Pada makalah ini, kita fokus pada salah satu aspek bagian dari operasional harian bandara yaitu, penjadwalan penerbangan di gerbang – gerbang bandara. Tujuan dari penjadwalan ini adalah untuk mengurangi penerbangan yang tidak mendapat jatah gerbang, dan untuk meminimalkan jarak tempuh penumpang di bandara. Rancangan penjadwalan ini menggunakan algoritma *greedy*. Algoritma *greedy* terbukti dapat merancang penjadwalan yang efektif dan efisien.

Kata Kunci—bandara, pesawat, penjadwalan gerbang, algoritma *greedy*.

I. PENDAHULUAN

Operasional pesawat terbang mencakup berbagai jenis layanan, seperti pemeliharaan, perbaikan, dan pemberangkatan penumpang, yang semuanya itu harus dilakukan dalam rentang waktu yang sangat cepat agar operasional dapat berjalan lancar. Perkembangan tren mobilitas manusia untuk menggunakan pesawat, juga menuntut operasional bandara agar menjalankan sistem yang paling efektif dan efisien.

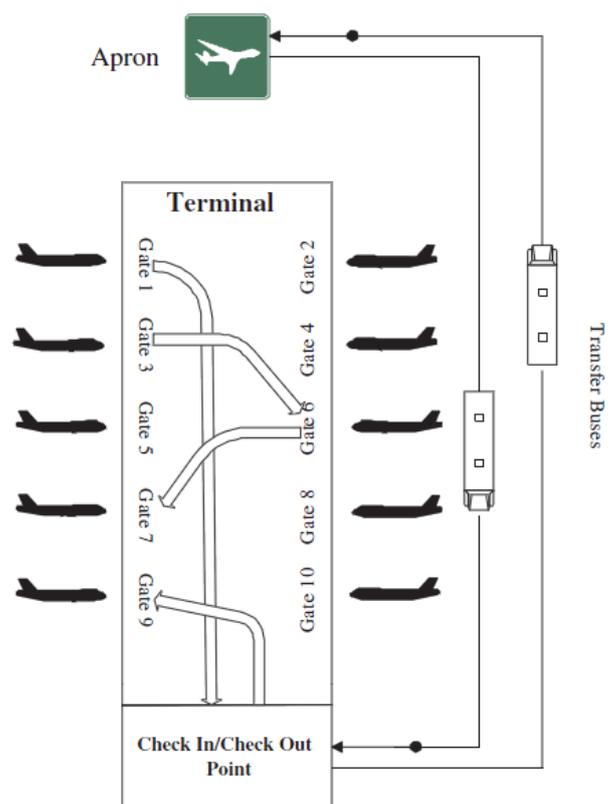
Jarak tempuh penumpang yang berjalan di bandara untuk mencapai berbagai area – area utama, termasuk pintu keberangkatan dan *baggage belts*, menjadi tolak ukur efisiensi suatu bandara. Secara khusus, jarak yang dilalui oleh penumpang dari check-in counter ke gerbang penerbangan dan dari gerbang ke gerbang (pada kasus *transfer* penumpang), sangat bergantung pada bagaimana jadwal penerbangan yang ditugaskan untuk masing – masing gerbang.

Permasalahan penjadwalan penerbangan di gerbang – gerbang bandara menjadi sesuatu yang perlu diperhatikan, karena penjadwalan ini menyangkut seberapa banyak gerbang yang diperlukan untuk sebuah maskapai, dan bagaimana alokasi penempatannya.

Fokus utama dalam penjadwalan penerbangan pada gerbang – gerbang bandara adalah kemampuan maskapai penerbangan untuk meminimalisasi biaya operasional. Operasional yang efisien sangat bergantung pada cara penjadwalan waktu kedatangan dan keberangkatan

pesawat.

Berikut adalah tata letak tempat di bandara dan alur pengunjung.



Gambar 1 *Design* Bandara dan Aliran Penumpang

Gambar di atas menunjukkan bagaimana desain umum sebuah bandara lengkap dengan gerbang – gerbang tempat pesawat. Dalam kenyataannya, ketika jumlah pesawat yang ada lebih banyak dari jumlah gerbang yang tersedia, pesawat akan ditempatkan di bagian *apron*.

Berbagai jenis analisis model tentang penjadwalan penerbangan pada gerbang – gerbang bandara sudah pernah dikembangkan sebelumnya, sebagai contoh Mangoubi dan Mathaisel 1985, Vanderstraetan 1988, Cheng 1997, Haghani 1985, Vanderstraetan 1988, Cheng 1997, Haghani dan Chen 1998. Pada waktu yang sama, berbagai teknik sudah diaplikasikan untuk mengatasi

masalah ini. Sebagai contoh, linear binary programming (Babic 1984), 0 – 1 linear programming (Bihl 1990), genetic algorithm (Gu & Chung 1999), mixed 0 – 1 quadratic integer programming dan tabu search (Xu & Bailey 2001), multi-objective programming (Yan & Huo 2001), simulated annealing (Ding 2002), stochastic programming (Lim & Wang 2005) [1]. Meskipun sudah banyak penelitian tentang minimisasi biaya untuk jarak di bandara, masih sedikit yang membahas tentang meminimalkan jumlah pesawat yang tidak mendapatkan gerbang.

II. KAJIAN TEORI

Algoritma greedy.

Algoritma greedy merupakan metode yang paling populer untuk memecahkan masalah optimasi. Secara harfiah, greedy berarti tamak atau rakus. Prinsip dari algoritma greedy adalah mengambil setiap kesempatan yang ada saat itu juga, tanpa memperhatikan konsekuensi kedepannya. Algoritma greedy membentuk solusi dari langkah demi langkah, dan pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Di setiap langkahnya algoritma greedy mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan. Setiap keputusan yang diambil diharapkan merupakan langkah optimum pada langkah tersebut, dikenali sebagai solusi optimum lokal, kemudian dengan setiap langkah yang ditempuh diharapkan dapat memperoleh solusi optimum di akhir proses, yaitu solusi optimum global.

Algoritma *greedy* tidak selalu memperoleh solusi optimum untuk keseluruhan permasalahan yang ditangani, dikarenakan algoritma *greedy* tidak melakukan operasi secara *exhaustive* kepada semua data, dan seringkali tidak mementingkan solusi optimum global. Akan tetapi, algoritma *greedy* merupakan solusi yang baik digunakan dikarenakan algoritma *greedy* bekerja dengan cepat dan sering memberikan perkiraan nilai optimum yang baik di setiap langkahnya. Dan tidak jarang dapat menghasilkan solusi optimum global pada suatu permasalahan dari pengambilan solusi optimum lokal di setiap langkahnya. Elemen-elemen algoritma *greedy* dalam persoalan optimasi adalah sebagai berikut :

1. Himpunan kandidat: Himpunan ini berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi: Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi merupakan himpunan bagian dari himpunan kandidat.
3. Fungsi seleksi: Fungsi seleksi dinyatakan sebagai predikat seleksi, merupakan fungsi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal.
4. Fungsi kelayakan: Fungsi kelayakan dinyatakan sebagai predikat LAYAK merupakan fungsi yang

memeriksa apakah suatu kandidat yang dipilih dapat memberikan solusi yang layak, yaitu kandidat tersebut tidak melanggar aturan yang ada.

5. Fungsi objektif: Fungsi objektif merupakan fungsi yang memaksimalkan atau meminimumkan nilai solusi.

Kelebihan utama dari algoritma greedy adalah :

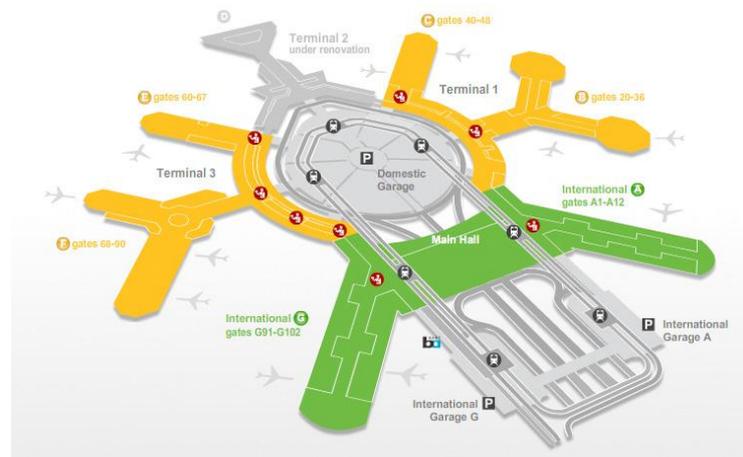
- a. Mudah diimplementasikan, algoritma greedy sangat mudah untuk diimplementasikan pada persoalan-persoalan yang ada.
- b. Efisien, algoritma greedy merupakan algoritma yang sangat efisien karena algoritma ini selalu berusaha memilih hasil optimal.

Kedua hal ini menjadikan algoritma greedy terkadang mampu memberikan hasil optimal, seperti untuk permasalahan *activity-selection*, *fractional knapsack* dan *minimum spanning tree* [2].

III. RUMUSAN MASALAH

Kendala operasi terdiri dari:

1. Setiap pesawat harus ditugaskan ke tepat hanya satu gerbang. Dengan kata lain, untuk setiap gerbang dapat ditempati oleh satu dan hanya penerbangan pada saat yang sama. Juga disebut gerbang bebas, jika tidak sedang melanyani penerbangan pada waktu tersebut.
2. Untuk pertimbangan keamanan, dua pesawat dilarang ditempatkan pada gerbang yang sama secara simultan. Dengan kata lain, jika gerbang telah ditempati oleh satu pesawat, tidak dapat diisi dengan pesawat yang lain sampai gerbang telah dikosongkan.



Gambar 2 Kondisi real dari Denah Sebuah Bandara di San Fransisco

Dapat dilihat pada gambar bahwa jarak dari *main hall* ke gerbang dan jarak antara gerbang relatif jauh, sehingga pada kenyataannya, penjadwalan gerbang di bandara adalah proses yang rumit. Demi

menyederhanakan masalah, hal utama yang dibahas terdiri dari tiga faktor berikut:

- Jumlah penerbangan yang tiba dan keberangkatan.
- Jumlah gerbang yang tersedia untuk penerbangan datang.
- Penerbangan dan waktu *landing* berdasarkan jadwal penerbangan.

Jarak yang perlu kita perhatikan dalam perhitungan mulai dari

- Check-in ke gerbang (kasus keberangkatan penumpang),
- Gerbang ke *baggage claim areas* (check-out / kepulangan penumpang),
- Gerbang ke gerbang (kasus transfer atau *connecting passengers*)

Selain itu, ada kasus khusus ketika jumlah pesawat terbang yang ada melebihi jumlah gerbang yang *available*, untuk itu, kita perlu menambahkan perhitungan jarak antara apron area ke terminal untuk pengisian penumpang pesawat di area tersebut.

Dalam model ini, kita tidak akan menetapkan kendala sekunder yang menghitung lama *boarding*, dan waktu antri lain antara keberangkatan dan kedatangan pesawat. Hal – hal sekunder tersebut dapat ditanggulangi dengan menambahkan durasi lama keberangkatan dan kedatangan pesawat. H. Ding dalam makalahnya tentang “Aircraft and Gate Scheduling Optimization at Airports” mendefinisikan beberapa notasi terkait model penjadwalan yang digunakan:

- N: Sekumpulan waktu kedatangan pesawat dan/atau keberangkatan pesawat di bandara.
M: Sekumpulan gerbang yang tersedia di bandara.
n: Jumlah total penerbangan, misal, |N|, dimana |N| menandakan kardinalitas dari N.
m: Jumlah total gerbang, misal, |M|.
a_i: Waktu kedatangan untuk penerbangan i.
d_i: Waktu keberangkatan untuk penerbangan i.
w_{k,l}: Jarak tempuh penumpang dari gerbang k ke gerbang l.
f_{i,j}: Jumlah penumpang yang berpindah dari penerbangan i ke penerbangan j.

Sebagai tambahan, penggunaan dua gerbang contoh yang merepresentasikan jalur masuk atau keluar dari airport, dan gerbang m+1 yang merepresentasikan *apron* (tempat yang digunakan ketika ada pesawat datang dan tidak ada gerbang yang tersedia). Jadi, w_{k,0} merepresentasikan jarak tempuh antara gerbang k dan jalur masuk / jalur keluar airport, dan f_{0,i} merepresentasikan nomor dari penerbangan i; f_{i,0} merepresentasikan nomor kedatangan pesawat dari penerbangan i; w_{m+1,k} merepresentasikan jarak tempuh antara apron dan gerbang k (biasanya lebih jauh dari jarak

tempuh antara gerbang).

Variable biner y_{i,k} = 1 menandakan bahwa penerbangan i diisi ke gerbang k (0 < k ≤ m+ 1), y_{i,k} = 0 juga sebaliknya. Lalu, batasan yang harus dipenuhi adalah $\forall(i, j) y_{i,k} = y_{j,k} = 1 (k = m + 1) \text{ berarti } a_i > d_j \text{ atau } a_j > d_i$.

Seperti pada batasan masalah yang telah didefinisikan sebelumnya, kondisi ini tidak mengijinkan dua penerbangan dijadwalkan pada gerbang yang sama secara simultan (kecuali jika penerbangan tersebut dijadwalkan berada di apron).

Formulasi untuk penjadwalan penerbangan pada gerbang – gerbang di bandara ini dapat dirumuskan sebagai:

$$\text{Minimize } \sum_{i=1}^n y_{i,m+1}$$

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{m+1} \sum_{l=1}^{m+1} f_{i,j} w_{k,l} y_{i,k} y_{j,l} + \sum_{i=1}^n f_{0,i} w_{0,i} + \sum_{i=1}^n f_{i,0} w_{i,0}$$

subject to:

$$\sum_{k=1}^{m+1} y_{i,k} = 1 (\forall i, 1 \leq i \leq n) \quad (1)$$

$$a_i < d_i (\forall i, 1 \leq i \leq n) \quad (2)$$

$$y_{i,k} y_{j,k} (d_j - a_i) (d_i - a_j) \leq 0 (\forall i, j, 1 \leq i, j \leq n, k \neq m+1) \quad (3)$$

$$y_{i,k} \in \{0, 1\} (\forall i, 1 \leq i \leq n, \forall k, 1 \leq k \leq m+1) \quad (4)$$

Persamaan (1) menjamin bahwa setiap penerbangan harus diisi ke satu gerbang atau ke apron. Persamaan (2) menspesifikasikan bahwa setiap waktu kedatangan lebih dahulu daripada waktu keberangkatan. Persamaan (3) menyatakan bahwa dua jadwal tidak dapat saling tindih, jika mereka diisikan ke gerbang yang sama.

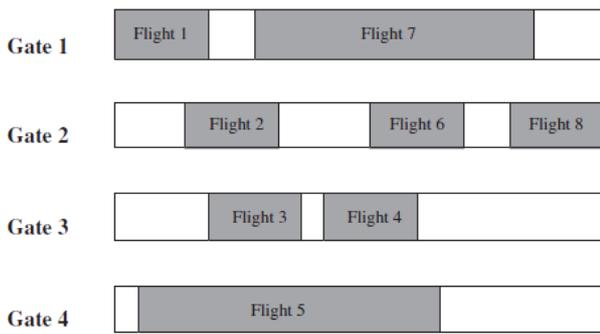
III. ANALISIS DAN PENERAPAN

A. Pemodelan Algoritma

Sebagai bahan perbandingan penggunaan algoritma: C. Li dalam makalahnya yang berjudul “Airport Gate Assignment—A Hybrid Model and Implementation” memodelkan penjadwalan ini, dan memperoleh kompleksitas waktu algoritmanya sebesar T(n) = 3n + a. Dengan menggunakan algoritma greedy, kita dapat memperoleh kompleksitas waktu sebesar T(n) = n. Meskipun tidak terlalu jauh perbedaan kompleksitasnya, alasan pemilihan algoritma greedy untuk memodelkan permasalahan penjadwalan ini adalah alasan kemudahan dalam implementasi dan terbukti lebih efisien.

Cost tambahan yang paling utama, yang dikeluarkan baik dari pihak bandara, maskapai, maupun dari pihak penumpang adalah biaya ketika gerbang di bandara tidak

tersedia untuk pesawat yang datang. Dengan kata lain, kita perlu menghindari penggunaan apron sebagai gerbang untuk pesawat.



Gambar 3 Algoritma Greedy untuk Mengoptimalkan Penjadwalan Penerbangan pada Gerbang – Gerbang di Bandara

Ide dasar dari algoritma greedy adalah: Setelah mengurutkan semua penerbangan menurut waktu keberangkatannya, penerbangan diisi ke gerbang yang ada satu per satu. Penerbangan yang selanjutnya diisikan ke gerbang yang tersedia dengan waktu penerbangan yang paling terakhir. Jika tidak ada lagi gerbang yang tersedia, penerbangan akan diisikan ke apron.

Greedy menurut selisih waktu tunggu yang paling minimum.

- Himpunan kandidat: himpunan waktu kedatangan dan/atau keberangkatan pesawat, dan himpunan gerbang yang tersedia di bandara.
- Himpunan solusi: himpunan pasangan gerbang dan jadwal pesawat yang diisi ke dalamnya.
- Fungsi seleksi: isi penjadwalan pesawat pada suatu gerbang dengan penerbangan ke- i yang memiliki waktu kedatangan lebih besar daripada waktu keberangkatan terakhir pada gerbang tersebut.
- Fungsi layak: memeriksa apakah penerbangan yang diisi ke suatu gerbang memiliki waktu kedatangan lebih besar daripada waktu keberangkatan penerbangan sebelumnya.
- Fungsi obyektif: selisih waktu antara jadwal kedatangan pesawat dengan jadwal keberangkatan pesawat sebelumnya (pada suatu gerbang), yang paling minimum.

B. Algoritma dan Pembuktian

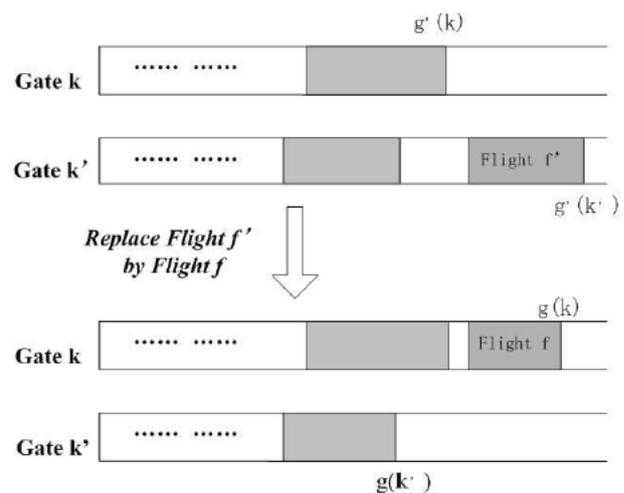
Detil dari algoritma greedy untuk penjadwalan penerbangan pada gerbang – gerbang pesawat di bandara, adalah sebagai berikut:

1. Urutkan semua penerbangan menurut waktu keberangkatannya d_i ($1 \leq i \leq n$). Diberikan g_k ($1 \leq k \leq m$) yang merepresentasikan waktu terdekat yang tersedia (sesungguhnya waktu keberangkatan dari

penerbangan terakhir) dari gerbang k . Ubah nilai $g_k = -1$ untuk semua k .

2. Untuk semua penerbangan i
 - a. Cari gerbang k yang memiliki $g_k < a_i$ dan g_k dimaksimumkan.
 - b. Jika k ditemukan, isikan penerbangan i ke gerbang k , perbaharui isi $g_k = d_i$;
 - c. Jika k tidak ditemukan, isi penerbangan i ke apron.
3. Keluarkan hasil.

Pembuktian kebenaran untuk algoritma ini salah satunya dengan menggunakan induksi [3]. Asumsikan kita menemukan solusi optimal setelah menjadwalkan penerbangan i dengan algoritma greedy. Lalu kita isi penerbangan f ke gerbang k . Tetapi solusi optimal adalah menghilangkan penerbangan f dan mengisinya f' ($f' > f$) ke gerbang k' . Jadi kita selalu mengganti f' oleh f untuk membuat solusi greedy kita tidak lebih buruk daripada solusi optimal.



Gambar 4 Pembuktian Penjadwalan dengan Menggunakan Algoritma Greedy

Ada dua kasus yang perlu diperhatikan.

1. Jika $k = k'$, karena kita mengurutkan penerbangan menurut waktu keberangkatan, $d_f \leq d_{f'}$. Kita memiliki $g_k \leq g_{k'}$. Seperti yang telah kita bahas, waktu yang tersedia untuk gerbang, kita menemukan bahwa algoritma greedy menghasilkan solusi yang lebih baik, atau setidaknya sama.
2. Jika $k \neq k'$ kita menemukan bahwa $g_k \leq g_{k'}$ dan $g_{k'} \leq g_k$ karena kita memilih maksimum g_k pada solusi greedy.

C. Rekomendasi Metode

Kita dapat menambahkan beberapa parameter lain, untuk membuat algoritma penjadwalan yang lebih mangkus. Sebagai contoh, untuk kasus tertentu kita dapat menambahkan parameter banyaknya penerbangan di suatu gerbang dengan jarak dari gerbang tersebut ke *Main Hall*

Airport. Hal ini bertujuan untuk meminimalkan waktu tempuh penumpang.

Gerbang yang memiliki penjadwalan penerbangan yang paling banyak, ditempatkan di lokasi yang paling dekat dengan *Main Hall Airport*, karena banyaknya jadwal penerbangan mengakibatkan banyaknya lalu – lintas penumpang ke gerbang tersebut. Ketika kita membuat jarak gerbang tersebut paling dekat dengan *Main Hall Airport*, kita dapat lebih meminimalkan waktu tempuh penumpang ke gerbang.

V. KESIMPULAN

Jadi, terbukti algoritma greedy memberikan kita solusi penjadwalan termasuk jumlah optimal dari penerbangan yang dapat dijadwalkan dalam gerbang – gerbang di bandara.

VI. UCAPAN TERIMA KASIH

Saya ingin mengucapkan terimakasih yang sebesar – besarnya kepada Ibu Dr. Eng. Ayu Purwarianti, S.T., M.T. dan Bapak Dr. Ir. Rinaldi Munir, M.T., yang telah mengajar mata kuliah IF3051, Strategi Algoritma. Atas pelajaran yang Bapak dan Ibu berikan, membimbing saya dalam menyelesaikan makalah ini.

REFERENSI

- [1] Chendong, Li. “Airport Gate Assignment” Texas: Texas Tech University, 2008.
- [2] Munir, Rinaldi. “Diktat Kuliah IF3051 Strategi Algoritma”. Institut Teknologi Bandung, 2009..
- [3] H. Ding, A. Lim, B. Rodrigues, Y. Zhu. “Aircraft and Gate Scheduling Optimization at Airports” Singapore: School of Business, Singapore Management University, 2004.
- [4] Dobson Gregory, Lederer Philip. “Airline Scheduling and Routing in a Hub-and-Spoke System” New York: University of Rochester.
- [5] Li Xiaofeng, Zhao Hai. “Greedy Algorithm Solution of Flexible Flow Shop Scheduling Problem” Changchun: College of Information Science and Engineering, 2010.

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, 8 Desember 2011

ttd



Robertus Theodore 13509008