

Pencarian Rute Pesawat Termurah dengan Algoritma *Best-First Search* dengan Heuristik *Airport Hub*

Mathias Novianto / 13516021

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha No. 10 Bandung 40132, Indonesia

mathiasnn98@gmail.com

Abstract—Penggunaan pesawat terbang sebagai moda transportasi saat ini sangatlah besar. Untuk mengurangi jumlah rute penerbangan yang diperlukan, maskapai penerbangan menerapkan sistem hub-and-spoke. Berdasarkan pengamatan, biaya tiket penerbangan antarbandara besar memiliki harga tiket yang lebih murah dibandingkan perjalanan ke bandara kecil. Dengan metode *best-first search*, dapat digunakan heuristik *airport hub* sebagai sebuah metode yang sistematis untuk menemukan rute pesawat termurah.

Keywords—*best-first search*, *breadth-first search*, *airport hub*, *termurah*

I. PENDAHULUAN

Seiring dengan berkembangnya ekonomi dan teknologi penerbangan, saat ini pesawat terbang sangat umum digunakan oleh masyarakat sebagai moda transportasi. Pesawat terbang sangat sering digunakan untuk perjalanan jarak menengah dan jauh karena kecepatannya. Selain itu, pesawat terbang juga tidak dibatasi oleh infrastruktur jalan, hanya membutuhkan bandara sebagai tempat mendarat dan lepas landas.

Saat ini, berkembang paradigma *hub-and-spoke* bagi maskapai penerbangan untuk membuat rute penerbangannya. Beberapa bandara dijadikan sebagai bandara penghubung bagi bandara di sekitarnya oleh maskapai penerbangan. Dengan metode ini, jumlah rute penerbangan yang dibuat dapat jauh lebih sedikit dibandingkan setiap bandara harus memiliki rute penerbangan ke bandara lainnya.

Pengguna jasa transportasi penerbangan tentunya menginginkan harga termurah untuk perjalanannya. Saat ini, penerbangan antarkota besar cenderung lebih murah dibandingkan langsung terbang ke kota kecil akibat dari *hub-and-spoke* yang diterapkan oleh maskapai penerbangan. Hal ini tentunya menjadi pertimbangan bagi konsumen untuk mendapat harga tiket yang lebih murah dengan konsekuensi harus singgah ke bandara penghubung.

Penyedia layanan pemesanan tiket pesawat juga tentunya membutuhkan metode yang mangkus untuk dapat menentukan rute pesawat termurah seiring dengan bertambahnya jumlah konsumen. Selain itu, harga tiket yang dapat berubah dalam hitungan menit membuat penyedia layanan pemesanan tiket harus lebih sering untuk mengecek harga dari maskapai penerbangan. Kemudian, pengguna jasa transportasi

penerbangan untuk perjalanan wisata sangatlah sensitif terhadap harga, sehingga pencarian tiket pesawat dilakukan terhadap beberapa destinasi dan beberapa kemungkinan waktu perjalanan.

II. DASAR TEORI

A. Graf

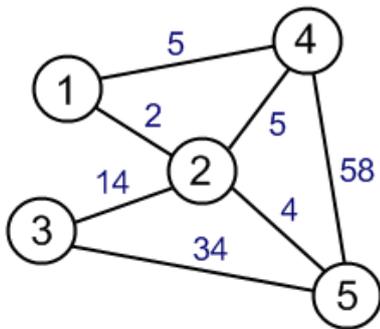
Graf adalah kumpulan simpul dan sisi yang menghubungkan simpul-simpul tersebut. Secara formal, graf adalah pasangan himpunan (V, E) , di mana V adalah himpunan dari simpul dan E adalah himpunan dari sisi yang menghubungkan simpul-simpul tersebut. E adalah *multiset*, jadi bisa saja ada elemen yang muncul lebih dari sekali. ^[1]

Dalam graf, sebuah jalan adalah rangkaian simpul dan sisi sebuah graf yang saling berselang-seling antara simpul dan sisi, yang diawali dan diakhir oleh sebuah simpul. Sebuah jalan dikatakan sebagai jejak jika setiap sisi maksimal dikunjungi sebanyak sekali. Sebuah jejak dikatakan lintasan jika setiap simpul maksimal dikunjungi sebanyak sekali, kecuali simpul awal dan simpul akhir, jika keduanya sama. Kasus tersebut dinamakan sebagai lintasan tertutup atau sirkuit.

Dua buah simpul u dan v dikatakan terhubung jika ada jalan yang berawal dari u dan berakhir di v . Jika u dan v terhubung serta v dan w terhubung, maka u dan w pasti terhubung. Sebuah graf dikatakan terhubung jika semua kemungkinan pasangan simpulnya terhubung.

Graf dapat dibagi menjadi graf berarah dan graf tidak berarah. Graf berarah berarti setiap sisi memiliki arah dan hanya menghubungkan dua simpul dalam satu arah. Graf tak berarah berarti setiap sisi dapat terhubung dalam dua arah.

Graf juga bisa dibagi menjadi graf berbobot dan graf tidak berbobot. Dalam graf berbobot, setiap simpul atau setiap sisi memiliki bobot yaitu *cost* untuk menempuh dari suatu simpul ke simpul lain yang terhubung.



Gambar 1. Sebuah graf berbobot

(Sumber:

<https://www.safaribooksonline.com/library/view/php-7-data/9781786463890/e0f9e479-fb3d-4201-98f0-aa663415c7d8.xhtml>, diakses 6 Mei 2018 pukul 20.30)

B. Algoritma Breadth-First Search

Breadth-first search (BFS) adalah sebuah algoritma untuk melintasi atau melakukan pencarian pada struktur data pohon ataupun struktur data graf. Algoritma BFS pada graf sama dengan algoritma BFS pada struktur data pohon. sama seperti algoritma Proses diawali dari akar pohon atau sebuah simpul pada graf. Untuk dapat mencegah proses dari sebuah simpul lebih dari sekali, digunakan sebuah *array* yang berisi boolean apakah sebuah simpul telah dikunjungi. Algoritma BFS dibagi menjadi dua kasus pada algoritma iteratif:^[2]

- Kasus akar: antrian pada *queue* kosong, sehingga akar harus dimasukkan sebelum kasus yang lebih umum.
- Kasus umum: Proses data yang terdapat pada *queue*, dan perluas simpul yang telah dikunjungi menjadi anaknya. Proses ini akan berakhir ketika simpul daun tidak memiliki anak.

C. Algoritma Best-First Search

Algoritma *best-first search* merupakan algoritma pencarian yang mengeksplorasi sebuah graf dengan mengunjungi simpul yang paling menjanjikan terlebih dahulu. Algoritma ini merupakan variasi dari algoritma *breadth-first search* (BFS) yang tidak memiliki informasi kemungkinan terbaik. Perbedaan algoritma *best-first search* dengan algoritma BFS biasa adalah *best-first search* memiliki fungsi untuk menentukan tetangga mana yang paling menjanjikan untuk menghasilkan hasil yang optimal, kemudian mengunjungi simpul tersebut. Algoritma *best-first search* juga merupakan kategori dari *heuristic search* dan *informed search*. Untuk dapat. Karena *best-first search* hanyalah variasi dari BFS, perbedaannya dengan BFS hanyalah menggantikan penggunaan *queue* menjadi *priority queue*.^[3]

D. Hub-and-Spoke

Hub-and-spoke adalah sebuah paradigma untuk optimisasi pada transportasi yang arus lalu lintasnya diorganisasikan menjadi sekumpulan *spoke* yang terhubung melalui *hub* pusat.

Paradigma *hub-and-spoke* sangat berbeda dengan metode *point-to-point* yang menghubungkan semua titik asal dengan semua titik tujuan.^[4]

Paradigma *hub-and-spoke* mulai digunakan oleh Delta Airlines pada tahun 1955. Seiring perkembangan teknologi, paradigma ini juga digunakan oleh bidang selain penerbangan. Bidang telekomunikasi dan bidang teknologi informasi menjadi contoh bidang yang menerapkan paradigma *hub-and-spoke* pada bisnisnya.

Keuntungan dari model *hub-and-spoke* dibandingkan *point-to-point* adalah rute yang lebih sedikit. Untuk menghubungkan n buah simpul, terdapat $n-1$ rute yang perlu untuk dihubungkan, sehingga batas atas jumlah rute yang diperlukan adalah $n(n-1)/2$. Sebagai contoh, sebuah sistem dengan 10 tujuan, dengan metode *spoke-and-hub*, dibutuhkan 9 rute untuk menghubungkan seluruh destinasi dengan sebuah *hub*. Sedangkan, dengan metode *point-to-point*, dibutuhkan 45 rute.

Karena modelnya terpusat, kegiatan operasi menjadi kurang fleksibel, perubahan pada sebuah rute dapat mengakibatkan konsekuensi terhadap seluruh jaringan. Selain itu, karena diperlukan transit di hub, perjalanan yang ditempuh dapat menjadi lebih jauh dibandingkan perjalanan langsung dari tempat asal ke tujuan. Waktu yang dibutuhkan di *hub* menambah durasi total perjalanan.



Gambar 2. Contoh *hub-and-spoke*

(Sumber:

<https://science.howstuffworks.com/transport/flight/modern/airline3.htm>, diakses 7 Mei 2018 pukul 20.00)

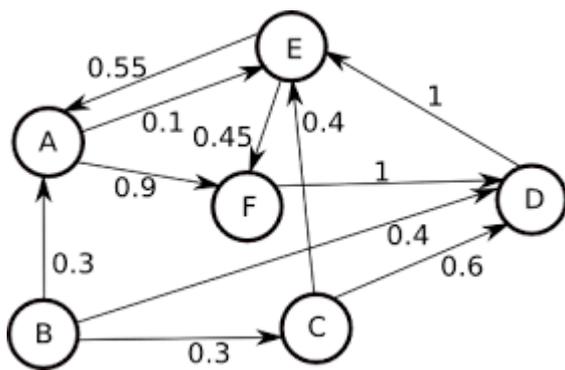
III. IMPLEMENTASI PENCARIAN TIKET PESAWAT TERMURAH DENGAN ALGORITMA BRANCH AND BOUND DAN HEURISTIK AIRPORT HUB

Pada bagian ini, penulis akan membahas mengenai implementasi algoritma branch and bound untuk mencari tiket pesawat termurah dimulai dari mendefinisikan persoalan, mendefinisikan ide pemecahan masalah, dan membuat implementasi program, dan menentukan solusi dari persoalan yang sudah dibuat:

A. Definisi Persoalan

Seseorang ingin pergi dari sebuah titik A ke B. Kita harus mencari rute dengan *cost* terkecil dari titik A ke B. Pada persoalan ini, kita mendefinisikan sebuah graf berarah berbobot yang mendefinisikan rute yang tersedia dan *cost* untuk berpindah dari suatu titik ke titik lainnya. Selain itu, kita juga mendefinisikan seberapa banyak penumpang yang diangkut oleh bandara tersebut pada suatu tahun.

Pada kasus ini, diasumsikan pengguna tidak berkeberatan apabila harus membutuhkan waktu yang lebih lama untuk sampai ke tempat tujuan dan harus berganti maskapai penerbangan. Selain itu, diasumsikan penerbangan dilakukan dalam satu hari, yaitu penerbangan kedua selalu dapat dilakukan tanpa memedulikan waktu selesai penerbangan pertama. Banyaknya asumsi yang diambil oleh penulis dikarenakan penentuan rute pesawat sangatlah kompleks, untuk memfokuskan tujuan penulisan, diambil beberapa asumsi seperti yang telah disebutkan.



Gambar 3. Graf Berarah Berbobot

(Sumber :

<http://ce.bonabu.ac.ir/uploads/30/CMS/user/file/115/EBook/Introduction.to.Algorithms.3rd.Edition.Sep.2010.pdf>, diakses 7 Mei 2018 pukul 20.20)

B. Ide Pemecahan Masalah

Saat ini, banyak maskapai yang menggunakan sistem *hub-and-spoke* dalam membuat rute penerbangan. Sebagai contoh, maskapai Lion Air, sebagai pemegang pangsa pasar penerbangan domestik terbesar di Indonesia, memiliki tiga buah *hub*, yaitu Bandara Internasional Soekarno-Hatta (Jakarta), Bandara Internasional Kuala Namu (Medan), dan Bandara Internasional Juanda (Surabaya). Maskapai-maskapai besar lainnya pun memiliki *hub*-nya masing-masing bergantung dari rute dan kota yang menjadi fokus pelayanan mereka.

Penerbangan antar-*hub* biasanya memiliki biaya yang lebih murah dibandingkan penerbangan antarkota kecil. Dengan memanfaatkan informasi ini, kita dapat menentukan prioritas pencarian solusi, sehingga pencarian dilakukan terhadap kandidat solusi yang memiliki kemungkinan terbesar untuk menjadi solusi optimum. Prioritas ditentukan berdasarkan berapa banyak penumpang yang dilayani pada tahun terakhir.

Semakin banyak penumpang yang dilayani, berarti prioritas untuk mengunjungi bandara tersebut semakin tinggi.

Untuk menyelesaikan masalah, digunakan algoritma *best-first search*. Cara kerja algoritma ini seperti algoritma *breadth-first search*, yaitu membangkitkan seluruh tetangga dari sebuah simpul. Namun, simpul yang dikunjungi oleh algoritma *best-first search* adalah simpul yang paling menjanjikan terlebih dahulu, dalam kasus ini adalah simpul yang memiliki prioritas tertinggi seperti yang telah dijelaskan sebelumnya. Struktur data *priority queue* akan digunakan untuk mengatur urutan pengunjungan simpul.

C. Pseudocode

Berikut adalah pseudocode untuk menghitung prioritas suatu rute dan algoritma *best-first search*.

```
function getPriority(int banyakPenumpang)
    if(banyakPenumpang > bandaraSangatBesar)
        return high_priority
    else if(banyakPenumpang > bandaraBesar)
        return medium_priority
    else {bandara menengah dan kecil}
        return low_priority
```

```
procedure findCheapestFlight(Airport from,
Airport to)
    //initialize the queue with possible route
    from the starting point
    for each(neighbor in from)
        addToQueue(neighbor)
    //iterate until all of the possibilities
    evaluated
    while (queue not empty)
        if(queue[0] == end and queue[0].moneySpent
        < minCost)
            minCost = queue[0].moneySpent
        for each(path in queue)
            if(path.cost > minCost)
                queue.remove(path)
        for each(neighbor in queue[0])
            if(not visited neighbor)
                queue.add(neighbor)
```

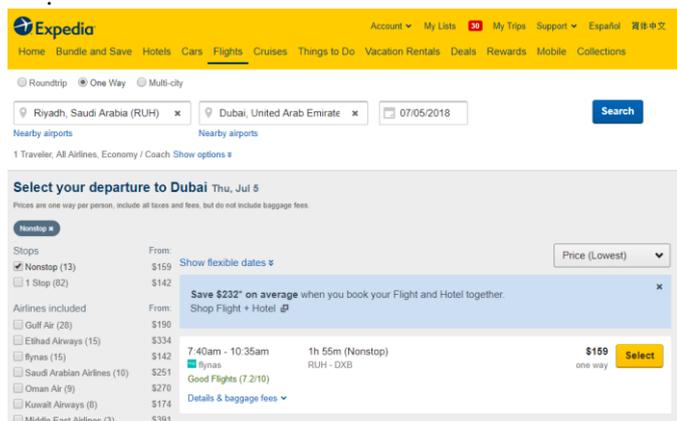
IV. HASIL EKSPERIMEN DAN ANALISIS

Pada bagian ini, akan dijelaskan contoh kasus yang diambil oleh penulis, dan penyelesaiannya menggunakan algoritma yang telah dibuat penulis pada bab III.

A. Contoh Kasus

Data yang diambil oleh penulis adalah data harga tiket penerbangan nonstop termurah dari laman situs pemesanan tiket daring, Expedia yang diakses pada 6 Mei 2018. Pencarian tiket pesawat satu arah untuk penerbangan pada 5 Juli 2018. Untuk menyederhanakan kasus, penulis hanya mengambil tujuh bandara sebagai titik yang mungkin untuk dijadikan tempat keberangkatan, tujuan ataupun transit.

Bandara yang diambil pada contoh kasus ini telah merepresentasikan ketiga jenis bandara, yaitu sangat besar, besar, dan menengah dan kecil. Selain itu, bandara-bandara yang diambil dari beberapa negara. Bandara yang diambil juga telah merepresentasikan beberapa bandara tersibuk di dunia (Dubai International Airport, Soekarno-Hatta International Airport, Franz Josef Stratuss International).



Gambar 4. Situs Pemesanan Tiket Daring

Kode IATA	Nama Bandara	Kota	Negara
CGK	Soekarno-Hatta International	Tangerang	Indonesia
KNO	Kuala Namu International	Medan	Indonesia
CMB	Bandaranaike International	Katunayake	Sri Lanka
MUC	Franz Josef Strauss International	Oberding	Jerman
DXB	Dubai International	Dubai	Uni Emirat Arab
RUH	King Khalid International	Riyadh	Arab Saudi
TXL	Berlin Tegel	Berlin	Jerman

Tabel 1. Daftar Kode IATA Bandara

	CGK	KNO	CMB	MUC	DXB	RUH	TXL
--	-----	-----	-----	-----	-----	-----	-----

CGK	0	114	291	9999	960	652	9999
KNO	119	0	9999	9999	9999	9999	9999
CMB	307	9999	0	9999	247	341	9999
MUC	9999	9999	9999	0	598	9999	45
DXB	740	9999	285	882	0	137	9999
RUH	401	9999	312	9999	159	0	9999
TXL	9999	9999	9999	59	9999	9999	0

Tabel 2. Harga Tiket Penerbangan

Keterangan Tabel :

- Nama-nama bandara direpresentasikan dengan kode bandara tersebut dari IATA
- Isi sel pada tabel merupakan harga dari bandara di kolom pertama ke bandara yang ada di baris pertama
- Harga tiket dalam satuan US Dollar

Kode Bandara	Jumlah Pengguna	Kategori
CGK	63015620	1
KNO	12245116	3
CMB	9805045	3
MUC	44577241	1
DXB	88242099	1
RUH	23400000	2
TXL	20500000	2

Tabel 3. Kapasitas Bandara yang Diuji

Keterangan Tabel :

- Definisi pembagian kategori
 - Jumlah pengguna >40000000 kategori 1
 - Jumlah pengguna >20000000 kategori 2
 - Jumlah pengguna <20000000 kategori 3
- Data jumlah pengguna diambil dari data tahun 2017 dari laman masing-masing bandara

B. Hasil Eksperimen

Untuk dapat melakukan pengujian, penulis mengimplementasikan algoritma yang telah dibuat dalam bahasa C#. Untuk algoritma *breadth-first search* (BFS), hanya akan dilakukan pengubahan prioritas suatu rute pada antrean (*queue*) dari program *best-first search*. Nilai prioritas akan selalu diatur sama pada algoritma BFS. Keluaran dari program adalah banyak operasi yang dilakukan, rute yang ditempuh, cost yang diperlukan. Berikut adalah data jumlah simpul yang dikunjungi oleh algoritma *best-first search* dan *breadth-first search*.

Asal	Tujuan	<i>Best-first search</i>		<i>Breadth-first search</i>		Cost Optimal
		Operasi	Cost	Operasi	Cost	
CGK	MUC	20	1420	17	1420	1420
KNO	CMB	6	410	6	410	410
CGK	TXL	22	1465	22	1465	1465
DXB	KNO	12	518	12	518	518
TXL	KNO	14	1175	13	1175	1175
MUC	RUH	9	735	8	735	735

Tabel 4. Hasil Pengujian dari Contoh Kasus

lebih mahal untuk perjalanan. Jika dilihat dari tabel, biaya perjalanan dari dan ke CMB memiliki harga yang lebih murah dibanding bandara lainnya. Sebagai contoh, perjalanan dari DXB menuju CGK akan lebih murah jika melalui CMB dibandingkan melakukan penerbangan nonstop. Jika dilihat dari jumlah penumpang yang diangkut, DXB dan CGK memiliki lebih banyak penumpang yang diangkut, namun heuristik yang dipakai tidak cukup membantu.

Selain itu, harga tiket pesawat sangatlah bergantung kepada berbagai macam penyebab. Penyebab-penyebab tersebut antara lain musim liburan yang ada, maskapai yang beroperasi di bandara tersebut, dan sedang berlakunya masa promo, dan lainnya. Penggunaan heuristik *airport hub* sebagai prioritas pengunjungan simpul tidak dapat bekerja sendiri, melainkan membutuhkan berbagai informasi tambahan sebagai penentu kemungkinan bandara yang memiliki harga termurah.

Algoritma *best-first search* juga tidak membantu karena tidak banyak perjalanan yang melalui bandara penghubung lebih dari dua kali. Akibatnya, pencarian rute lain tetap melakukan pencarian hingga sampai ke tujuan, karena jarang nilai pembatas dari *best-first search* dapat langsung menghentikan pencarian pada perjalanan sebelum mencapai tujuan akhir.

V. KESIMPULAN DAN SARAN

Penggunaan algoritma *best-first search* dengan heuristik *airport hub* saja tidak dapat membuat pencarian tiket pesawat termurah. Bahkan, penggunaan algoritma *best-first search* dengan heuristik *airport hub* dapat memiliki performa yang lebih buruk dibandingkan algoritma *breadth-first search*. Oleh karena itu, diperlukan gabungan berbagai heuristik untuk pencarian yang lebih cepat pada algoritma *best-first search*. Selain itu dapat juga digunakan algoritma lain untuk mempercepat pencarian.

VI. UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih pada Tuhan Yang Maha Esa, karena atas berkat-Nya saya dapat menyelesaikan makalah ini. Saya juga mengucapkan terima kasih kepada orang tua saya atas dukungan yang mereka berikan sampai saat ini. Tak ketinggalan, saya juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen IF2211 Strategi Algoritma yang telah membagikan ilmunya dengan penuh semangat dan sabar kepada saya dan K03 Teknik Informatika ITB selama satu semester.

REFERENSI

- [1] <https://mathigon.org/course/graphs-and-networks> diakses pada 6 Mei 2018 pukul 20.00
- [2] <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/> diakses 6 Mei 2018 pukul 20.10
- [3] <https://www.geeksforgeeks.org/best-first-search-informed-search/> diakses 6 Mei 2018 pukul 20.20
- [4] <https://science.howstuffworks.com/transport/flight/modern/airline3.htm> diakses pada 6 Mei 2018 pukul 20.40

```
C:\Program Files (x86)\dotnet\dotnet.exe
Tempat yang tersedia :
CGK KNO CMB MUC DXB RUH TXL
Mau berangkat dari mana ? CGK
Mau pergi kemana ? MUC
Path Traveled :
Uang yang dihabiskan : 1420
Banyak Proses : 20
CGK ==> CMB ==> DXB ==> MUC
```

Gambar 5. Hasil Pengujian Implementasi Algoritma *Best-First Search*

```
C:\Program Files (x86)\dotnet\dotnet.exe
Tempat yang tersedia :
CGK KNO CMB MUC DXB RUH TXL
Mau berangkat dari mana ? CGK
Mau pergi kemana ? MUC
Path Traveled :
Uang yang dihabiskan : 1420
Banyak Proses : 17
CGK ==> CMB ==> DXB ==> MUC
```

Gambar 6. Hasil Pengujian Implementasi Algoritma *Breadth-First Search*

C. Analisis

Dari tabel 3, diperoleh data bahwa algoritma *best-first search* dan *breadth-first search* (BFS) dapat memberikan nilai cost yang optimal. Namun, untuk jumlah simpul yang dikunjungi, penggunaan heuristik berdasarkan kapasitas bandara saja tidak cukup untuk mengurangi jumlah simpul yang dikunjungi. Bahkan, penggunaan algoritma *best-first search* menambah simpul yang dikunjungi untuk diproses.

Penyebab algoritma *best-first search* yang menambah simpul yang dikunjungi adalah bandara yang jumlah penggunaannya sedikit belum tentu memiliki harga tiket yang

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, 10 Mei 2018



Mathias Novianto
13516021