

PENGGUNAAN ALGORITMA GREEDY PADA MESIN PENCARI WWW.WEGO.COM

Gagarin Adhitama – 13508089

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
gagarin_adhitama@students.itb.ac.id

Abstraksi – Dalam makalah ini dijelaskan bahwa algoritma Greedy juga dapat digunakan dalam mesin pencari. Cara seperti ini sebenarnya sering digunakan, hanya saja sering terlewat dari pemikiran kita. Kebanyakan orang berpendapat bahwa Greedy tidak dapat dihubungkan dengan mesin pencari, padahal dalam mesin pencari tersebut terdapat proses penyeleksian, proses penyeleksian inilah yang menggunakan algoritma Greedy. Algoritma Greedy ini cukup efektif pada seleksi yang dilakukan di mesin pencari ini. Pada intinya, di mesin pencari www.wego.com, terdapat seleksi yaitu pada pencarian hotel, penerbangan, maupun paket liburan. Pada penyeleksian tersebut ada penyeleksian berdasarkan harga termurah (pada pencarian penerbangan misalnya) atau penyeleksian berdasarkan budget, atau berdasarkan tipe klasifikasi hotel. Mesin pencari ini telah meminta semua data dari tiap-tiap website, kemudian data tersebut dibandingkan dengan yang lain dan diseleksi sesuai keinginan dari penggunanya. Proses penyeleksian dari data ini digunakan Greedy, dapat menggunakan Greedy by price, atau Greedy by category, atau Greedy by day. Ada banyak proses penyeleksian di sini, hasil yang ditampilkan nantinya adalah hasil berdasarkan hasil kombinasi dari penyeleksian tersebut (semua kondisi harus terpenuhi), jika tidak terpenuhi semuanya, berarti tidak ditemukan apa yang diinginkan oleh pengguna.

Kata kunci: Greedy, mesin pencari, wego.com, seleksi, Greedy by.

I. PENDAHULUAN

Banyak orang beranggapan bahwa mesin pencari identik dengan algoritma *pattern matcher*, padahal ada mesin pencari dengan berbagai seleksi dapat juga menggunakan algoritma lain yaitu algoritma *Greedy*. Penulis mengambil objek penelitian pada situs mesin pencari untuk perjalanan wisata www.wego.com. Pada situs tersebut, kita dapat mencari info apa saja yang berhubungan dengan info harga di seluruh dunia, dari penerbangan, hotel, paket liburan, bahkan rental mobil. Cara kerja dari wego.com ini yaitu dengan mengambil informasi dari situs-situs yang berhubungan, misal kita ambil contoh di bagian penerbangan. Situs-situs maskapai di seluruh dunia ini diambil informasi mengenai harganya, dan bagaimana rutenya. Pengguna situs tersebut diminta untuk memasukkan asal keberangkatan dan tujuan. Setelah itu, pengguna dapat memilih penerbangan yang *direct flight*, atau *1-stop flight*, atau *2-stops flight*. Nah, di sinilah proses seleksi tersebut dibutuhkan. *Greedy* berguna untuk mencari berdasarkan pilihan pengguna tersebut, dan nantinya hasil yang ditampilkan adalah harga penerbangan termurah di hari itu dari tiap-tiap maskapai. Harga termurah juga merupakan proses seleksi yang merupakan *Greedy by price*. Proses penyeleksiannya yaitu dengan membandingkan harga di tiap situs maskapai penerbangan, kemudian memilih harga paling rendah. Untuk penyeleksian lainnya, misalkan untuk tipe penerbangan (apakah penerbangan langsung atau tidak), algoritma *Greedy* digunakan untuk mencari penerbangan yang diinginkan oleh pengguna. Ambil contoh, pengguna menginginkan penerbangan langsung, maka algoritma *Greedy* bekerja dengan mengambil data hanya dari maskapai yang melakukan penerbangan

langsung, untuk penerbangan tidak langsung, *Greedy* ini tidak mengambil datanya sehingga tidak menampilkannya. Pada dasarnya, proses seleksi ini menarik untuk dilihat lebih lanjut dan bagaimana *pseudo-codenya*. Di makalah ini akan dibahas tentang hubungannya algoritma *Greedy* pada proses seleksi di situs *wego.com*.

II. LANDASAN TEORI

A. Algoritma pada Proses Seleksi

Pada proses penyeleksian, banyak algoritma-algoritma yang dapat digunakan. Algoritma yang dapat digunakan untuk mencari nilai minimum misalnya, dapat digunakan algoritma pembandingan satu per satu (*brute force*) dimana semua data dibandingkan dan nantinya mendapatkan nilai terkecil. Kemudian untuk pencarian dengan karakteristik tertentu seperti klasifikasi hotel (bintang 3, bintang 4, atau lainnya) atau seperti yang telah diutarakan sebelumnya yaitu penerbangan langsung atau penerbangan dengan pemberhentian algoritma lain seperti DFS atau BFS juga dapat digunakan. Namun, muncul pertanyaan apakah algoritma tersebut cukup efektif pada permasalahannya atau belum. Karena tergantung bagaimana kebutuhan pada aplikasi tempat dimana algoritma tersebut digunakan. Proses seleksi yang lain yang mungkin digunakan dalam situs mesin pencari *wego.com* ini yaitu *selection sort*, dimana ada pengurutan berdasarkan prioritas harga.

B. Teori Algoritma Greedy

1. Elemen Algoritma Greedy

Secara umum, algoritma *Greedy* digunakan tidak hanya pada proses seleksi saja, tetapi *Greedy* juga dapat digunakan untuk mencari solusi (seperti pada penukaran koin atau menyelesaikan permainan *klotski*). Elemen-elemen yang terkait dengan algoritma ini yaitu :

- Himpunan kandidat. Himpunan ini berisi elemen-elemen pembentuk solusi. Maksudnya, himunan kandidat ini berisi elemen yang menuju solusi permasalahan yang dicari.
- Himpunan solusi. Kalau himpunan solusi ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan, atau dengan kata lain dia merupakan himpunan bagian dari himpunan kandidat.

- Fungsi seleksi. Fungsi seleksi ini yaitu fungsi yang pada setiap langkah memilih kandidat yang paling tidak memungkinkan mencari solusi optimal. Fungsi ini yang akan lebih kita bahas dalam makalah ini, yaitu seleksi yang dilakukan pada situs *wego.com*.
- Fungsi kelayakan. Fungsi ini dinyatakan dengan predikat LAYAK yang memeriksa apakah kandidat yang dipilih dapat memberikan solusi optimum atau tidak
- Fungsi objektif, yaitu fungsi memaksimalkan dan meminimumkan nilai solusi. Fungsi ini juga digunakan dalam proses penyeleksian dalam mencari harga termurah.

2. Sketsa Umum Algoritma Greedy

Secara garis besar skema umum algoritma *Greedy* ini bekerja dengan tiga metode, yaitu seleksi sehingga muncul himpunan kandidat, kemudian memilih dari kandidat tersebut menjadi himpunan solusi yaitu dengan fungsi layak, dan yang terakhir yaitu memilih mana solusi paling tepat. Itulah tiga skema utama yang dilakukan ketika algoritma *Greedy* ini bekerja. Secara detail, skema dari algoritma *Greedy* ini seperti ini

- Inisialisasi S (himpunan solusi) dengan kosong
- Pilih sebuah kandidat (dengan seleksi) dari C
- Kurangi C dengan kandidat yang sudah dipilih dari langkah (ii) di atas
- Periksa apakah kandidat yang dipilih membentuk solusi yang layak atau tidak. Jika layak masukkan kandidat dalam himpunan solusi, jika tidak maka dibuang.
- Periksa apakah himpunan solusi telah memberikan solusi yang lengkap, jika ya maka proses selesai, jika tidak maka ulangi lagi dari langkah awal.

3. Pseudo Code Algoritma Greedy

Dari skema yang telah dibuat di atas, untuk memperjelas dari segi kodenya, berikut penulis sertakan *pseudo-code* agar lebih jelas bagi para pengkode membuat kode pada program. Sehingga, kita akan mengaplikasikan algoritma ini, pembaca lebih mudah untuk memahaminya.

```
Function greedy (input C:
    himpunan_kandidat) -> himpunan
    kandidat
{Mengembalikan solusi dari optimasi
```

dengan algoritma greedy.
 Masukan: himpunan kandidat C
 Keluaran: himpunan solusi bertipe himpunan_kandidat}

Deklarasi

X: kandidat
 S: himpunan_kandidat

Algoritma

```
S <- {} {inisialisasi S dengan kosong}
while (not SOLUSI (S)) and (C ≠ {}) do
  X <- SELEKSI (C) {pilih kandidat}
  C <- C - {X} {kandidat dikurangi 1}
  if LAYAK (S U {X}) then
    S <- S U {X}
  endif
endwhile
{SOLUSI (S) or C = {}}

if SOLUSI (S) then
  Return S
else
  Write ('tidak ada solusi')
endif
```

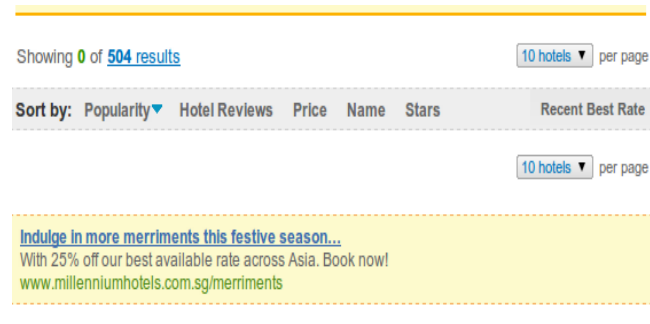
III. PENERAPAN ALGORITMA GREEDY

Pada mesin pencari wego.com, terdapat beberapa proses seleksi, dimana seleksi tersebut kadang menjadi kombinasi tertentu. Contoh pada pemilihan hotel, hotel dipilih berdasarkan lokasi terlebih dahulu, lalu pengguna dapat memilih hotel berdasarkan berdasarkan *budget* dan berdasarkan klasifikasi kelas (bintang 3, bintang 4, atau lainnya).



Gambar 1 Contoh Antarmuka Seleksi Bidang Hotel

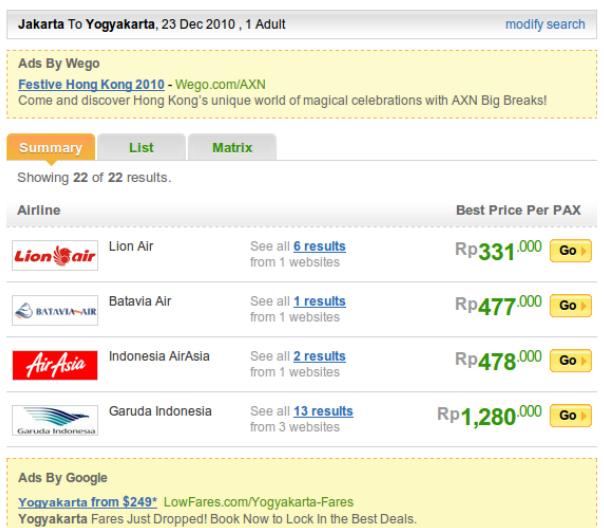
Pada contoh ini, dapat kita gunakan skema umum algoritma *Greedy* yang telah dijelaskan sebelumnya, yaitu pertama dengan membuat himpunan solusi (yang awalnya kosong), kemudian proses seleksi dimulai dengan mengambil semua data yang berhubungan dengan lokasi yang dipilih oleh pengguna. Data tersebut masuk ke dalam himpunan kandidat C pada awalnya, kemudian ketika pengguna melakukan pemilihan berdasarkan klasifikasi misalnya, maka himpunan kandidat C akan dikurangi oleh hasil seleksi X. Selanjutnya, jika pengguna menambahkan batas harga atas, maka proses seleksi dilakukan lagi dan hasil batas atas dimasukkan ke dalam variabel X, dan himpunan kandidat dikurangi dengan X itu lagi. Himpunan solusi yang tadinya kosong, digabung dengan hasil seleksi akhir (sesuai keinginan pengguna), kalau dari data tersebut ditemukan data yang sesuai, maka data tersebut ditampilkan, tetapi kalau tidak ditemukan, maka akan muncul pada layar *'showing 0 result'*.



Gambar 2 Jika pencarian tidak ditemukan

1. Penggunaan Greedy pada wego.com

Greedy by yang digunakan pada pencarian penerbangan juga pada situs ini. Ketika kita mencari penerbangan yang tersedia dengan rute tertentu pada hari tertentu, maka yang ditampilkan oleh wego.com adalah harga termurah dari tiap-tiap situs maskapai penerbangan. Kemudian seleksi lain yang dilakukan yaitu apakah penerbangan tersebut penerbangan langsung atau tidak. Untuk pencarian harga termurah ini digunakan fungsi minimum dari algoritma *Greedy* ini. Data-data dari tiap maskapai yang didapat oleh wego.com, dibandingkan harganya pada jadwal yang diinginkan oleh pengguna. Proses seleksi juga dapat dilakukan sebaliknya, maksudnya mencari berdasarkan jenis penerbangan dahulu, baru dicari harga termurah pada jenis penerbangan tersebut.



Gambar 3 Hasil pencarian penerbangan

Dari gambar di atas tampak bahwa harga tersebut adalah harga termurah pada maskapai Lion Air diambil dari situsnya. Untuk membuktikannya, ketika kita lihat harga langsung pada situs maskapai Lion Air di jadwal yang sama, harga yang termurah pada saat itu adalah harga yang tertera pada wego.com. Hal ini membuktikan bahwa algoritma ini benar dan tepat dalam mencari harga tiket termurah pada jadwal tertentu dan tujuan tertentu.

2. Ilustrasi Penggunaan Greedy

Pada bagian ini, diilustrasikan bagaimana Greedy tersebut bekerja dalam pencarian maskapai penerbangan. Contoh yang tertera di bawah ini adalah ilustrasi, sehingga dari segi nama maskapai dan harga menggunakan permasalahan. Ilustrasi ini dibuat untuk mempermudah melihat bagaimana algoritma Greedy ini bekerja.

Tabel 1 tabel ilustrasi harga dan jenis penerbangan

Maskapai	Jenis penerbangan (pemberhentian)	Harga
A	Langsung	500
B	1 x	400
C	1 x	300
D	2 x	290
E	Langsung	460
F	2 x	240
G	2 x	230
H	Langsung	450

Pada hal ini terdapat dua jenis proses seleksi, yaitu berdasarkan harga (*Greedy by price*) atau berdasarkan jenis penerbangan (*Greedy by stop*). Misalkan pengguna menginginkan penerbangan yang murah dengan syarat tidak ada pemberhentian lebih dari satu kali, maka hasil yang akan dikeluarkan

Tabel 2 hasil Greedy by stop

Maskapai	Jenis penerbangan (pemberhentian)	Harga
A	Langsung	500
B	1 x	400
C	1 x	300
E	Langsung	460
H	Langsung	450

Kemudian setelah *Greedy by stop* dilakukan, dilanjutkan oleh *Greedy by price* untuk mencari harga termurah. Dari hasil *Greedy by price* dihasilkan bahwa maskapai C dengan 1x pemberhentian dan harganya 300 adalah penerbangan yang sesuai dengan keinginan pengguna. Penggunaan algoritma Greedy ini dianggap cukup tepat karena untuk permasalahan ini terlalu berlebihan jika menggunakan algoritma *Exhaustive Search*, karena algoritma tersebut terlalu rumit untuk permasalahan seperti ini.

Untuk permasalahan pada pencarian hotel, hampir sama dengan pencarian penerbangan, hanya saja terdapat perbedaan pada pencarian hotel terdapat rentang harga. Hal ini karena terdapat banyak hotel pada data yang didapat oleh mesin pencari ini, sehingga rentang tersebut hanya menyeleksi data hotel pada rentang tersebut. Kalau pada bagian penerbangan, data yang dihasilkan atau ditampilkan adalah harga termurah (nilai harga minimum) berbeda dengan bagian hotel dimana situs ini menampilkan daftar hotel dalam rentang tertentu yang diatur oleh pengguna. Proses penyeleksian hotel di sini lebih kompleks ketimbang pencarian penerbangan karena lebih banyak hal yang bisa dipilih oleh pengguna seperti rentang harga, lokasi hotel, klasifikasi bintang hotel, hasil rekomendasi dari pengunjung situs, dan berdasarkan fasilitas yang diberikan oleh hotel tersebut. Pengguna dapat mencari hotel yang memiliki fasilitas lebih seperti taman bermain, kolam renang, atau fasilitas olahraga. Namun, terkadang hasil yang muncul tidak sesuai dengan keinginan pengguna. Hal ini bukan disebabkan oleh algoritma Greedy yang tidak efektif, tetapi karena

informasi yang didapat dari situs tersebut tidak sesuai dengan keinginan pengguna. Misal, pengguna ingin mencari hotel di daerah Bandung yang memiliki kolam renang, tetapi mesin pencari tersebut tidak dapat menampilkannya. Hal ini terjadi karena informasi yang didapat mesin pencari ini tidak menyebutkan fasilitas dari hotel-hotel di Bandung, sehingga dianggap hotel di Bandung yang memiliki kolam renang itu tidak ada.

IV. ANALISIS HASIL PENERAPAN

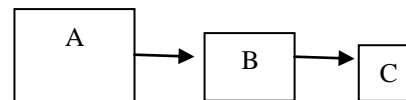
Algoritma *Greedy* ini cukup baik diaplikasikan pada situs *wego.com* ini, proses yang dilakukannya tidak membutuhkan waktu yang lama meskipun data yang diseleksi bersumber dari banyak situs di seluruh dunia ini. Pada proses seleksi mencari *Greedy by price* atau *Greedy by stops*, atau *Greedy by range* pada dasarnya prosesnya sama. Dari segi keefektifan program, *Greedy* cukup baik digunakan pada mesin pencari ini. Untuk data yang besar, yang notabene mencakup data situs-situs seluruh dunia waktu eksekusi dari algoritma ini bisa dikatakan cukup cepat.

Algoritma *Greedy* lebih dipilih untuk digunakan pada aplikasi ini karena untuk penggunaan algoritma-algoritma lain seperti pencarian harga minimum menggunakan *selection sort* sangat tidak efektif dan waktu prosesnya menjadi lebih lama. Algoritma lain seperti *Exhaustive Search* juga dianggap lebih rumit sehingga menjadi tidak efektif karena sebenarnya permasalahan yang diselesaikan tidak begitu rumit. *Greedy* ini sudah cukup tepat dalam melakukan penyeleksian yang terdapat pada mesin pencari perjalanan *wego.com* ini, kasus-kasus yang diselesaikan juga sesuai dengan skema umum daripada algoritma *Greedy* itu sendiri. Skema umum algoritma *Greedy* yaitu dengan mengurangi hasil yang ditemukan dengan hasil seleksinya. Seperti yang telah dibahas pada bagian sebelumnya, *wego.com* adalah mesin pencari dengan menggunakan pendekatan beberapa kombinasi seleksi, baik seleksi harga, seleksi tipe, klasifikasi atau yang lainnya. kombinasi seleksi di sini yang membuat algoritma *Greedy* menjadi alternatif yang tepat untuk digunakan.

V. KESIMPULAN

Setelah kita membahas banyak hal tentang algoritma *Greedy* dan penggunaannya pada situs mesin pencari *www.wego.com*, kita dapat menyimpulkan beberapa hal yang cukup penting, yaitu

1. Bahwa algoritma *Greedy* juga dapat digunakan dalam aplikasi mesin pencari. Meskipun kegunaannya bukan sebagai *pattern matcher*-nya, tetapi *Greedy* dapat digunakan untuk membantu menemukan hasil pencarian seperti yang diinginkan oleh penggunanya.
2. Untuk menyeleksi suatu hal, dimana proses penyeleksian tersebut saling berhubungan atau dengan kata lain terdapat variasi atau kombinasi seleksi, maka algoritma *Greedy* dapat digunakan dan dapat bekerja cukup efektif pada aplikasi tersebut.
3. Skema umum cara kerja algoritma *Greedy* ini yaitu dengan mengeliminasi himpunan kandidat, dimana himpunan kandidat adalah hasil seleksi dari proses sebelumnya. Sehingga himpunan kandidat tersebut nilainya makin kecil dan semakin kecil hingga menemukan solusi terakhirnya pada pencarian ini. Kalau digambarkan dalam bentuk grafis, dapat diibaratkan seperti ini



Gambar 4 Ilustrasi skema Algoritma *Greedy*

Analogikan bahwa A merupakan himpunan kandidat awal pada proses penyeleksian, kemudian setelah dieliminasi oleh hasil seleksi, menjadi himpunan B, dan diseleksi lagi maka menjadi himpunan yang lebih kecil lagi yaitu himpunan C yang menjadi solusi atau akhir dari pencarian.

4. Algoritma *Greedy* perlu dipertimbangkan dalam penggunaannya, karena masih banyak kajian tentang algoritma ini dan ternyata penggunaan algoritma ini cukup luas dan banyak aplikasi yang dapat menggunakannya.

REFERENSI

- [1] Munir, Rinaldi. 2005. *Diktat Kuliah Strategi Algoritmik IF2251 Strategi Algoritmik*. Departemen Teknik Informatika ITB
- [2] Id.wikipedia.org
- [3] Pengamatan langsung pada www.wego.com

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, 5 Desember 2010

 ttd

Gagarin Adhitama
13508089