

# Aplikasi Algoritma Greedy untuk Optimasi Sistem Booking Hotel Online

Selly Yuvita – 13508035

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
if18035@students.if.itb.ac.id

## ABSTRAK

Algoritma *Greedy* adalah salah satu algoritma yang biasa digunakan sebagai salah satu penyelesaian masalah optimasi. Pada algoritma *Greedy*, diterapkan prinsip “*take what you can get now!*”, dengan kata lain algoritma ini akan mengambil keputusan terbaik yang ada saat itu tanpa memikirkan konsekuensi ke depannya nanti. Makalah ini akan membahas aplikasi algoritma *Greedy* untuk optimasi Sistem *Booking* Hotel secara *online*. Sistem *Booking* Hotel secara *online* merupakan situs-situs web yang memfasilitasi pengunjungnya untuk memesan hotel di berbagai kota dan negara secara *online*. Saat pengunjung akan memesan hotel melalui situs tersebut, ia akan diminta untuk memasukkan tanggal *check in* tanggal *check out*. Sistem kemudian akan memperlihatkan hotel mana saja yang mempunyai kamar kosong pada rentang tanggal tersebut. Sayangnya, apabila pengunjung ingin memesan hotel untuk jangka waktu yang cukup lama di musim liburan, banyak hotel yang sudah dipesan kamar hotelnya. Akan tetapi, sistem hanya akan memperlihatkan kamar-kamar yang benar-benar kosong pada semua tanggal yang diinput pengguna, padahal ada alternatif lain, yaitu pengguna bisa berpindah kamar jika memang ada beberapa kamar yang bisa dipakai di tanggal tertentu, tetapi pada hari selanjutnya pengunjung harus berpindah kamar.

**Kata kunci :** algoritma *Greedy*, sistem *booking* hotel *online*.

## 1. PENDAHULUAN

Saat ini, teknologi berkembang dengan sangat pesat. Internet pun sudah menjadi hal yang biasa digunakan sehari-hari, kegunaannya pun sangat banyak, baik untuk sarana pembelajaran, bersosialisasi lewat jejaring social, maupun berbisnis.

Mungkin belum banyak orang yang berani berbisnis lewat internet, karena memang pembeli dan penjual tidak bertemu secara langsung sehingga banyak orang yang merasa cara ini kurang aman untuk dilakukan. Akan tetapi, tidak sedikit orang sudah mulai memanfaatkan internet sebagai sarana jual beli. Bagi sebagian orang,

asalkan situs yang dipakai sebagai sarana jual beli sudah banyak digunakan orang dan berbekal sedikit pengalaman dalam bertransaksi melalui internet, membeli atau memesan barang melalui internet sangat membantu. Alasannya, tidak semua orang punya cukup waktu untuk mendatangi tempat-tempat tertentu untuk membeli sesuatu karena kesibukan masing-masing. Selain itu, tak jarang ada diskon atau harga promosi apabila seseorang melakukan pembelian barang melalui website tertentu.

Hal ini juga dilakukan pada hotel-hotel di banyak kota dan negara. Sudah banyak hotel yang mempermudah calon tamunya dengan mengadakan fasilitas *booking online*. Ada yang menyediakannya di situs hotel itu sendiri, tapi ada juga yang mengikutsertakan hotelnya sebagai hotel yang bisa di-*booking* melalui situs pemesanan hotel global. Situs ini menyediakan *membooking* hotel-hotel di banyak kota dan negara secara *online*.

Contoh-contoh situs seperti ini antara lain [www.booking.com](http://www.booking.com), [www.agoda.com](http://www.agoda.com), [www.hotelbooking.com](http://www.hotelbooking.com), [www.hotelbook.com](http://www.hotelbook.com), dan lain-lain. Peminat situs-situs seperti ini sangat banyak terutama para calon pelanggan yang akan bepergian ke luar negeri dan tidak ingin menggunakan jasa travel agent untuk menghemat biaya.

Saat pengguna situs ini ingin *membooking* hotel, pengguna akan diminta memasukkan tanggal *check in*, tanggal *check out*, banyak kamar yang ingin di-*booking*, serta lokasi atau nama hotel yang ingin dipilih. Sistem lalu akan menampilkan hotel mana saja yang mempunyai kamar kosong pada setiap tanggal yang dimasukkan pengguna tadi. Dengan detail harga dan fasilitas yang juga akan ditampilkan, pengguna bisa melihat-lihat hotel seperti apa yang cocok untuknya dan bisa langsung *membookings*nya secara *online*.

Akan tetapi, saat pengguna ingin *membooking* kamar hotel terlalu dekat dengan tanggal *check in*-nya, atau saat pengguna ingin *membooking* kamar dalam jangka waktu yang cukup lama di musim liburan, seringkali kamar yang tersedia sangat terbatas. Hal ini dikarenakan kamar-kamar

yang ditampilkan adalah kamar-kamar yang belum di-*booking* di semua tanggal yang dimasukkan pelanggan.

Padahal, ada alternatif lainnya, yaitu pelanggan bisa menempati kamar yang kosong pada tanggal check-in-nya selama beberapa hari sampai kamar itu harus ditempati pelanggan lain yang sudah lebih dulu mem-*booking*nya, lalu pindah ke kamar lain di hotel yang sama. Hal ini tentunya juga akan membantu manajemen hotel agar semua kamar di hotelnya bisa digunakan secara efektif.

Maka dari itu, melalui makalah ini penulis ingin mengaplikasikan algoritma *Greedy* untuk optimasi sistem *booking* hotel *online*. Diharapkan dengan makalah ini rasa ingin tahu penulis terhadap aplikasi algoritma *Greedy* untuk optimasi sistem *booking* hotel *online* bisa terjawab, dan akan lebih baik lagi jika memang aplikasi ini bisa diimplementasikan lebih lanjut.

## 2. ALGORITMA GREEDY

Dalam bahasa Inggris, *Greedy* berarti rakus, tamak, atau loba. Definisi ini sangat sesuai dengan prinsip algoritma *Greedy*, yaitu “take what you can get now!”. Prinsip ini menggambarkan algoritma *Greedy* yang akan mengambil keputusan yang terbaik pada setiap langkah penyelesaian, tanpa memperhitungkan akibatnya pada keseluruhan permasalahan. Akan tetapi, agar algoritma *Greedy* benar-benar dapat menyelesaikan masalah optimasi, pada setiap langkah harus dibuat keputusan terbaik.

Algoritma *Greedy* merupakan salah satu algoritma yang sering digunakan untuk memecahkan permasalahan optimasi. Pada setiap langkah, kita harus membuat pilihan optimum lokal (local optimum), yang dapat diartikan sebagai keputusan terbaik yang diambil pada langkah tersebut. Karena algoritma *Greedy* biasanya dilakukan dalam beberapa langkah, maka dalam satu permasalahan optimasi kita akan membuat beberapa pilihan optimum lokal, sesuai dengan banyaknya langkah yang harus dilakukan. Diharapkan bahwa setiap kali diambil pilihan optimum lokal, langkah-langkah setelahnya akan mengarah ke solusi optimum global. Perlu diingat bahwa algoritma *Greedy* hanya memakai dua macam persoalan optimasi, yaitu maksimasi (maximization) dan minimisasi (minimization).

Pada algoritma *Greedy*, ada beberapa elemen yang harus diperhatikan, yaitu :

1. Himpunan kandidat (C)  
Himpunan ini berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi (S)  
Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi dari permasalahan optimasi yang akan diselesaikan

3. Fungsi seleksi  
Fungsi ini akan memilih kandidat yang paling memungkinkan untuk mencapai solusi optimal. Sesuai dengan prinsip algoritma *Greedy*, kandidat yang sudah dipilih pada suatu langkah tidak bisa diubah di langkah selanjutnya.
4. Fungsi kelayakan  
Fungsi ini akan memeriksa kelayakan suatu kandidat yang telah dipilih. Dalam arti, kandidat tersebut dan himpunan solusi yang terbentuk tidak melanggar constraints yang ada. Bila kandidat layak, maka kandidat tersebut akan dimasukkan ke dalam himpunan solusi, dan jika kandidat tersebut tidak layak, maka kandidat akan dibuang dan tidak akan dipertimbangkan lagi dalam pencarian solusi optimum.
5. Fungsi obyektif  
Fungsi ini akan membuat nilai solusi maksimum atau minimum, sesuai dengan jenis optimasi apa yang dibutuhkan.

Dengan kata lain, algoritma *Greedy* akan melakukan pencarian sebuah himpunan bagian S dari himpunan kandidat C. Himpunan bagian S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimisasi oleh fungsi obyektif.

Algoritma *Greedy* mempunyai beberapa variasi, antara lain :

1. Algoritma *Greedy* Murni
2. Algoritma *Greedy* Orthogonal
3. *Relaxed Greedy Algorithm*

Perlu diketahui bahwa tidak semua pemakaian algoritma *Greedy* akan menghasilkan solusi optimum. Contoh-contoh permasalahan yang tidak bisa menggunakan algoritma *Greedy* untuk menghasilkan solusi optimum adalah Knapsack Problem dan Travelling Salesman Problem.

## 3. IMPLEMENTASI

Pada sistem *booking* hotel *online*, algoritma *Greedy* dapat diterapkan untuk mencapai penggunaan kamar-kamar hotel secara efektif. Dalam arti, walaupun pengunjung harus berpindah kamar dalam beberapa waktu, penyelesaian ini lebih baik daripada pengunjung tidak mendapatkan kamar sama sekali karena tidak ada kamar yang benar-benar kosong di seluruh tanggal yang diinput pelanggan.

Dalam hal sistem *booking* hotel *online*, elemen-elemen yang dapat direpresentasikan dalam algoritma *Greedy* adalah sebagai berikut.

1. Himpunan kandidat (C)

- Dalam permasalahan ini, himpunan kandidat adalah himpunan kamar-kamar hotel yang mungkin untuk di-booking oleh pelanggan
- Himpunan solusi (S)  
Himpunan kombinasi kamar-kamar yang bisa di-booking oleh pelanggan dengan jumlah kamar terkecil (dalam arti pelanggan tidak perlu terlalu banyak berpindah kamar)
  - Fungsi seleksi  
Fungsi untuk membuat list kamar yang mungkin untuk di-booking oleh pelanggan sesuai dengan tanggal check in dan check out yang diinput oleh pelanggan
  - Fungsi kelayakan  
Fungsi untuk memastikan kamar belum di-booking oleh pelanggan lain di antara tanggal check in dan tanggal check out yang diinput oleh pelanggan. Dalam hal ini, fungsi ini termasuk di dalam fungsi seleksi.
  - Fungsi obyektif  
Fungsi menghitung banyaknya hari dimana kamar tersedia untuk di-booking pelanggan. Semakin banyak hasilnya, maka semakin sedikit pengunjung harus berpindah kamar, dan sebaliknya, semakin kecil hasilnya semakin banyak pelanggan harus berpindah kamar. Hasil yang optimal adalah bila pengunjung semakin sedikit berpindah kamar.

Dalam aplikasi algoritma *Greedy* untuk optimasi sistem booking hotel online ini, kamar-kamar dalam suatu hotel dan tanggal-tanggal direpresentasikan dalam tabel. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.

	6 Des	7 Des	8 Des	9 Des	10 Des
Kamar 1					
Kamar 2					
Kamar 3					
Kamar 4					
Kamar 5					

Tabel 1. Representasi Kamar Hotel dan Tanggal

Langkah selanjutnya, tabel diisi dengan ketersediaan kamar untuk di-booking. Data ini bisa direpresentasikan dengan angka 0 dan angka 1, dengan angka 0 merepresentasikan kamar yang sudah di-booking pada tanggal-tanggal tersebut dan angka 1 merepresentasikan kamar yang masih tersedia di tanggal-tanggal tersebut.

	6 Des	7 Des	8 Des	9 Des	10 Des
Kamar 1	1	0	1	0	1
Kamar 2	1	1	1	0	0

2					
Kamar 3	1	1	0	1	1
Kamar 4	0	0	1	0	0
Kamar 5	0	1	1	0	0

Tabel 2. Tabel diisi dengan 0 atau 1

Dengan menggunakan fungsi RoomAvailable dan fungsi CheckDays, hitung berapa hari setiap kamar akan tersedia di setiap tanggal dengan menjumlahkan setiap sel yang bernilai 1 yang bersebelahan. Setelah fungsi dieksekusi, tabel akan berubah menjadi seperti di bawah ini.

	6 Des	7 Des	8 Des	9 Des	10 Des
Kamar 1	1	0	1	0	1
Kamar 2	3	2	1	0	0
Kamar 3	2	1	0	2	1
Kamar 4	0	0	1	0	0
Kamar 5	0	2	1	0	0

Tabel 3. Tabel diisi dengan jumlah hari dimana kamar tersebut tersedia

Untuk mendapatkan solusi dari permasalahan ini, lakukan sort pada setiap kolom, lalu pilih kolom dengan nilai terbesar. Berpindahlah pada sel sebelahnya sampai bertemu dengan angka 0 atau saat sudah mencapai tanggal check out. Lalu cari lagi nilai terbesar yang ada pada kolom tersebut, bergerak lagi sampai bertemu dengan angka 0 atau sudah mencapai tanggal check out dan seterusnya sampai benar-benar mencapai tanggal check out.

Dengan cara ini akan didapatkan himpunan kamar yang bisa di-booking pelanggan dari tanggal check in hingga tanggal check out yang pelanggan tersebut inginkan, meski ada kemungkinan pelanggan harus berpindah-pindah kamar.

	6 Des	7 Des	8 Des	9 Des	10 Des
Kamar 1	1	0	1	0	1
Kamar 2	3	2	1	0	0
Kamar 3	2	1	0	2	1
Kamar 4	0	0	1	0	0
Kamar 5	0	2	1	0	0

**Tabel 4. Pemilihan kamar yang bisa di-booking oleh pelanggan**

Meskipun cara ini berhasil untuk mendapatkan solusi optimum pada contoh yang diberikan, tetapi cara ini tidak selalu berhasil untuk mendapatkan solusi optimum. Perhatikan contoh solusi berikut ini.

	6 Des	7 Des	8 Des	9 Des	10 Des	11 Des	12 Des
Kamar 1	1	0	1	2	1	0	0
Kamar 2	3	2	1	0	3	2	1
Kamar 3	2	1	0	1	0	0	0
Kamar 4	0	0	1	0	0	2	1
Kamar 5	0	2	1	0	0	1	0

**Tabel 5. Contoh penyelesaian pada kasus lain dengan cara yang sama**

Padahal, solusi yang lebih baik adalah sebagai berikut.

	6 Des	7 Des	8 Des	9 Des	10 Des	11 Des	12 Des
Kamar 1	1	0	1	2	1	0	0
Kamar 2	3	2	1	0	3	2	1
Kamar 3	2	1	0	1	0	0	0
Kamar 4	0	0	1	0	0	2	1
Kamar 5	0	2	1	0	0	1	0

**Tabel 6. Solusi yang relatif lebih baik**

Solusi kedua dirasa lebih optimum karena pelanggan akan berada lebih lama di suatu kamar, dengan asumsi pelanggan akan lebih nyaman bila sudah berada di suatu kamar lebih lama.

Fungsi-fungsi yang digunakan pada aplikasi algoritma *Greedy* untuk optimasi sistem *booking* hotel *online* dapat didefinisikan oleh pseudocode di berikut ini.

```
function RoomAvailable (input
CheckInDate      : Date,
CheckOutDate     : Date -> array
of integer)
{Fungsi ini akan
mengembalikan daftar kamar
yang masih tersedia diantara
tanggal Check In dan tanggal
Check Out}
```

```
function CheckDays (input
CheckInDate      : Date,
CheckOutDate     : Date, RoomID :
integer -> integer)
{Fungsi ini mengecek setiap
sel yang berisi angka 1, dan
mengembalikan untuk berapa
hari kamar tersebut tersedia
dengan cara menghitung ada
berapa angka 1 yang ada pada
sel yang berurutan}
```

```
function ChooseRoom (input
CheckInDate      : Date,
CheckOutDate     : Date -> array
of integer)
{Fungsi ini menghasilkan
pilihan perpindahan kamar
yang bisa dipertimbangkan
oleh pelanggan untuk
membooking kamar. Di dalamnya
juga dilakukan penghitungan
berapa kali pelanggan harus
berpindah kamar mulai dari
tanggal Check In sampai
tanggal Check Out}
```

#### 4. HASIL PENGUJIAN

Di makalah ini digunakan dua contoh pengujian seperti yang sudah dijabarkan pada bab sebelumnya. Hasil yang didapat adalah :

1. Pada contoh pengujian pertama, solusi yang didapatkan adalah solusi optimal
2. Pada contoh pengujian kedua, solusi yang didapatkan belum menjadi solusi optimal, meskipun jumlah perpindahan kamar antara solusi yang didapat dari aplikasi algoritma *Greedy* dengan solusi optimal adalah sama, yaitu pelanggan harus tiga kali berpindah kamar.

## 5. KESIMPULAN

Dari pemaparan yang telah dijelaskan sebelumnya, kesimpulan yang didapatkan adalah algoritma *Greedy* tidak selalu bisa memberikan solusi optimum pada permasalahan sistem *booking online* hotel *online*. Hal ini biasanya akan semakin sering terjadi seiring dengan bertambahnya waktu diantara tanggal Check In dan tanggal Check Out yang diinput oleh pelanggan.

Akan tetapi, setidaknya aplikasi algoritma *Greedy* yang telah dijelaskan di atas bisa memberikan lebih banyak pilihan kepada pelanggan, karena algoritma ini akan memasukkan daftar kamar yang tersedia, meskipun pelanggan harus berpindah-pindah kamar. Pada sistem yang ada sekarang, kamar-kamar yang akan dimasukkan ke dalam daftar pilihan hanyalah kamar-kamar yang benar-benar tersedia mulai dari waktu Check In yang diinput pelanggan sampai dengan waktu Check Out.

Solusi ini tentunya lebih baik daripada pelanggan sama sekali tidak mendapatkan kamar. Selain itu, pihak manajemen hotel juga lebih bisa lebih banyak menyewakan kamarnya.

## REFERENSI

- [1] Munir, Rinaldi. *Diktat Kuliah IF3051 Strategi Algoritma*. Bandung : Program Studi Teknik Informatika, Institut Teknologi Bandung, 2009.
- [2] <http://www.scribd.com/doc/30550680/Algoritma-Greedy> . Diakses pada tanggal 6 Desember 2010
- [3] [http://en.wikipedia.org/wiki/Greedy\\_algorithm](http://en.wikipedia.org/wiki/Greedy_algorithm) . Diakses pada tanggal 6 Desember 2010
- [4] <http://webcache.googleusercontent.com/search?q=cache:d5XPGcE5wvEJ:www.cis.upenn.edu/~matuszek/cit594-2005/Lectures/36-Greedy.ppt+Greedy+algorithm+in&cd=4&hl=id&ct=clnk&gl=id&clnt=firefox-a> . Diakses pada tanggal 6 Desember 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 2010



Selly Yuvita - 13508035