

# Penggunaan Algoritma Greedy dalam Optimasi Masalah Perkebunan

Daniel Widya Suryanata / 13509083  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
flgphoenix@yahoo.com

**Abstract**—Makalah ini membahas tentang penggunaan algoritma Greedy dalam optimasi masalah perkebunan. Secara lebih spesifik, masalah perkebunan yang akan dioptimasi adalah masalah penanaman tanaman agar tanaman jual tersebut dapat memberikan hasil atau keuntungan yang optimal kepada pemilik kebun. Pada makalah ini, akan dibahas beberapa algoritma Greedy dengan operand yang berbeda, setelah itu akan dibandingkan, operand mana yang paling sering memberikan keuntungan optimal. Makalah ini juga memberikan beberapa contoh kasus serta memberikan penyelesaian optimalnya.

**Index Terms**—Algoritma Greedy, Greedy by Profit, Greedy by Time, Greedy by Space

## I. PENDAHULUAN

### 1.1 Perkebunan

Di Indonesia, perkebunan menjadi subsektor dari sektor pertanian. Subsektor perkebunan penting karena subsektor ini sangat berperan dalam penyediaan lapangan kerja, pertumbuhan ekonomi, sumber devisa, pengentasan kemiskinan, dan konservasi lingkungan.

Dengan begitu banyaknya jenis tanaman yang dapat ditanam di perkebunan di Indonesia, seringkali pemilik perkebunan menjadi bingung untuk menanam tanaman, dan tidak jarang seorang pemilik kebun hanya mengkhususkan menanam satu jenis tanaman saja. Padahal dengan menggunakan algoritma greedy untuk mengoptimasi masalah perkebunan, sangat mungkin untuk mendapatkan penghasilan tambahan.

### 1.2 Algoritma Greedy

Greedy pada bahasa Inggris dapat diartikan sebagai rakus atau tamak. Sesuai dengan namanya, algoritma Greedy adalah algoritma yang sifatnya tamak, tetapi tamak itu tidak selalu buruk. Dalam dunia informatika, algoritma Greedy adalah algoritma yang sangat populer digunakan untuk mencari nilai optimal dari banyak permasalahan.

Algoritma Greedy membentuk solusi langkah per langkah. Banyak pilihan yang dapat diambil dalam setiap langkah, dan algoritma Greedy mengambil solusi yang dapat memberikan nilai optimum lokal, dengan harapan

bahwa nilai tersebut mengarah pada solusi optimum global walaupun kadang solusi yang diberikan tidak mengarah pada solusi yang optimum global.

Skema umum dari algoritma Greedy adalah :

- Himpunan Kandidat (C)  
Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.
- Himpunan Solusi (S)  
Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan ini adalah himpunan bagian dari himpunan kandidat.
- Fungsi Seleksi  
Fungsi ini dinyatakan dengan predikat SELEKSI, yaitu fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal.
- Fungsi Kelayakan / Feasible  
Fungsi ini dinyatakan dengan predikat LAYAK, yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada.
- Fungsi Obyektif  
Fungsi ini adalah fungsi yang memaksimumkan atau meminimumkan nilai solusi.

## II. ALGORITMA GREEDY

Berikut ini adalah pseudocode umum dari algoritma greedy :

```
Function greedy (input C : himpunan kandidat)
→ himpunan kandidat
{ Mengembalikan solusi dari persoalan optimasi
dengan algoritma greedy.
Masukan : himpunan kandidat C
Keluaran : himpunan solusi yang 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 sebuah kandidat dari C}
  C ← C - {X} {elemen himpunan kandidat berkurang satu}
  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
```

Untuk masalah optimasi kebun, skema umum dari algoritma Greedy adalah :

- Himpunan Kandidat (C) : himpunan tanaman dengan atribut : nama, harga jual, space (ruang yang dibutuhkan untuk menanamnya), dan waktu panen.
- Himpunan Solusi (S) : Salah satu dari C yang memiliki hasil optimum lokal pada setiap metode Greedy yang digunakan.
- Fungsi seleksi : pada umumnya, fungsi seleksi pada algoritma greedy mengeliminasi elemen C yang sudah pernah dipilih, namun pada permasalahan ini, fungsi seleksi memungkinkan suatu elemen dari C untuk terpilih lagi. Fungsi-fungsi seleksi pada masalah ini akan dibahas pada Bab III.
- Fungsi kelayakan : Fungsi kelayakan pada permasalahan ini didefinisikan seperti pada pseudocode di bawah ini :
  - Gunakan fungsi seleksi. Fungsi seleksi akan menghasilkan CurrentLahan (akumulasi luas lahan yang sudah terpakai + luas lahan yang dibutuhkan oleh tanaman tersebut) dan CurrentWaktu (akumulasi waktu tanam yang sudah terpakai + waktu panen tanaman tersebut)
  - Jika CurrentLahan dan CurrentWaktu masih kurang dari

atau sama dengan luas lahan dan waktu tanam yang ditentukan oleh user, pemilihan tanaman berikutnya melalui fungsi seleksi dilanjutkan. Tetapi apabila melanggar, CurrentLahan dan CurrentLuas akan dikembalikan nilainya menjadi nilai sebelum dilakukan fungsi seleksi.

- Setelah CurrentLahan dan CurrentWaktu dikembalikan nilainya, akan dipanggil fungsi yang mengecek apakah masih ada tanaman yang layak untuk ditanam (tanaman yang spacenya tidak melebihi luas lahan dan waktu panennya tidak melebihi waktu tanam).
  - Tanaman-tanaman tersebut akan disimulasikan menurut batas waktu yang ditentukan (dibatasi oleh waktu tanam yang dipilih oleh pengguna).
  - Apabila ada tanaman yang sudah memenuhi waktu panen, maka keuntungan dari tanaman tersebut akan diakumulasikan dan akan ditanam tanaman baru yang tidak melebihi luas lahan dan waktu tanam pada space tanaman yang lama.
  - Setelah simulasi selesai, keuntungan yang didapat dari tiga jenis fungsi seleksi akan dibandingkan dan akan dipilih satu jenis fungsi seleksi yang menghasilkan keuntungan maksimal. Report dari tanaman yang harus ditanam juga akan ditampilkan. Report tersebut berisi tanaman apa yang harus ditanam pada hari ke-sekian.
- Fungsi obyektif : Fungsi obyektif pada permasalahan ini ada 3 tergantung dari jenis algoritma Greedy yang ingin digunakan. Fungsi obyektif ini juga akan dibahas pada Bab III.

### III. PEMBAHASAN MASALAH

Pengguna akan diminta input luas lahan serta waktu tanam. Keduanya akan menjadi pembatas dalam algoritma Greedy yang dipakai. Pengguna juga dapat mengubah

kamus tanaman yang akan dipakai sebagai referensi data tanaman. Kamus tanaman berada pada file yang berbeda. Dalam kamus tanaman tersebut, pengguna diminta untuk memasukkan nama tanaman, keuntungan yang diperoleh dari penjualan tanaman, waktu panen tanaman, serta space yang dibutuhkan oleh tanaman.

Ada tiga jenis algoritma Greedy yang akan digunakan dalam mencari nilai optimum global, yaitu Greedy by Profit, Greedy by Space, dan Greedy by Time.

Secara umum, ketiga algoritma Greedy tersebut sama. Yang berbeda hanyalah pengaturan kamus tanaman. Setelah diload ke dalam file, kamus tanaman akan diurutkan berdasarkan algoritma Greedy yang dipakai.

Apabila hendak menggunakan Greedy by Profit, maka kamus tanaman akan diurutkan dari tanaman yang memiliki keuntungan paling besar sampai tanaman yang memiliki keuntungan paling kecil.

Apabila hendak menggunakan Greedy by Time, maka kamus tanaman akan diurutkan dari tanaman yang memiliki waktu panen paling kecil sampai tanaman yang memiliki waktu panen paling besar.

Apabila hendak menggunakan Greedy by Space, maka kamus tanaman akan diurutkan dari tanaman yang memiliki space paling kecil sampai tanaman yang memiliki space paling besar.

Pseudocode dari algoritma Greedy tersebut :

Procedure Greedy (input/output T : tanaman,  
temp : tanaman, LuasLahan : integer,  
WaktuTanam : integer, pos : integer)

KAMUS LOKAL

i : integer  
ptr : integer  
CurrentLahan : integer

ALGORITMA

```

i ← 1
ptr ← 1
CurrentLahan ← 0

while (T.namai != "") and (CurrentLahan <
LuasLahan) do
    if (T.waktupaneni ≤ WaktuTanam) and
(CurrentLahan + T.spacei ≤
LuasLahan) then
        temp.namaptr ← T.namai
        temp.hargaptr ← T.hargai
        temp.waktupanenptr ←
T.waktupaneni
        temp.spaceptr ← T.spacei
        CurrentLahan ← CurrentLahan +
T.spacei
        ptr ← ptr + 1
        pos ← pos + 1
    else
        i ← i + 1

```

```

if (CurrentLahan > LuasLahan) then
    temp.namaptr ← T.namai
    temp.hargaptr ← T.hargai
    temp.waktupanenptr ← T.waktupaneni
    temp.spaceptr ← T.spacei
    pos ← pos - 1

```

Tanaman pada pseudocode diatas adalah suatu tipe bentukan dengan elemen: nama, profit, space, serta waktupanen.

Algoritma Greedy ini memiliki kompleksitas algoritma sebesar O(n) karena hanya ada satu kalang yang dipakai untuk membandingkan tanaman-tanaman.

### A. Greedy by Profit

Greedy by Profit adalah cara optimasi yang didasarkan pada besarnya keuntungan yang didapatkan dari penjualan tanaman. Ide dasar dari metode ini adalah mencari tanaman dengan keuntungan yang paling besar untuk ditanam, tetapi masih tetap dibatasi oleh fungsi kelayakan.

Pada kasus dimana terdapat kesamaan harga antara tanaman yang sedang dibandingkan dengan tanaman yang sedang dipilih oleh Greedy by Profit, maka faktor-faktor lain dari kedua tanaman tersebut akan dibandingkan. Yang diprioritaskan adalah waktu panen, apabila waktu panen keduanya juga sama, yang dibandingkan kemudian adalah space tanaman tersebut.

### B. Greedy by Time

Greedy by Time adalah cara optimasi yang didasarkan pada besarnya minimalnya waktu yang dipakai untuk menanam suatu tanaman. Ide dasar dari metode ini adalah mencari tanaman dengan waktu panen yang terkecil, tetapi masih tetap dibatasi oleh fungsi kelayakan.

Pada kasus dimana terdapat kesamaan waktu panen antara tanaman yang sedang dibandingkan dengan tanaman yang sedang dipilih oleh Greedy by Time, maka faktor-faktor lain dari kedua tanaman tersebut akan dibandingkan. Yang diprioritaskan adalah keuntungan, apabila keuntungan keduanya juga sama, yang dibandingkan kemudian adalah space tanaman tersebut.

### C. Greedy by Space

Greedy by Space adalah cara optimasi yang didasarkan pada besarnya minimalnya space (lahan) yang dipakai untuk menanam suatu tanaman. Ide dasar dari metode ini adalah mencari tanaman dengan space yang terkecil, tetapi masih tetap dibatasi oleh fungsi kelayakan.

Pada kasus dimana terdapat kesamaan space antara tanaman yang sedang dibandingkan dengan tanaman yang sedang dipilih oleh Greedy by Space, maka faktor-faktor lain dari kedua tanaman tersebut akan dibandingkan. Yang diprioritaskan adalah keuntungan, apabila keuntungan keduanya juga sama, yang dibandingkan kemudian adalah waktu panen tanaman tersebut.

#### IV. CONTOH KASUS

Pada bab ini akan dibahas beberapa contoh kasus dalam masalah optimasi perkebunan.

Tabel 1. Contoh Kamus Tanaman menunjukkan contoh-contoh tanaman yang dipakai beserta data-datanya.

Tabel 1. Contoh Kamus Tanaman

No	Nama	Profit (Rp)	Waktupanen (hari)	Space (m <sup>2</sup> )
1	Jagung	4.000	90	0.5
2	Jambu	10.000	150	1
3	Apel	30.000	280	3
4	Cabai	400	110	0.04

Luas Lahan : 20 m<sup>2</sup>.

Waktu Tanam : 380 hari.

- Penyelesaian dengan Greedy by Profit :  
Apabila diselesaikan dengan metode Greedy by Profit, maka solusi yang akan didapatkan adalah :

Apel : 6 x 1 penanaman (280 hari, 18 m<sup>2</sup>).

Jambu : 2 x 2 penanaman (300 hari, 2 m<sup>2</sup>)

Jagung : 36 x 1 penanaman (90 hari, 18m<sup>2</sup>)

Yaitu dengan cara, tanam 6 pohon apel pada hari pertama. Penanaman ini akan memakai 18 m<sup>2</sup> dari total lahan. Karena masih ada sisa 2 m<sup>2</sup>, tanam 2 pohon jambu. Setelah 150 hari, jambu akan berbuah dan siap dipanen, kemudian pada hari itu juga, tanam kembali 2 pohon jambu di tempat yang sama. Pada hari ke-280, seluruh aple akan berbuah dan lahan seluas 18 m<sup>2</sup> siap untuk ditanami. Karena waktu tanam yang tersisa hanya 100 hari, maka hanya memungkinkan untuk menanam jagung. Jadi tanamlah 36 tanaman jagung pada lahan yang dulunya ditanami apel. Pada hari ke-300, kedua pohon jambu akan berbuah lagi, namun karena waktu yang tersisa hanya 80 hari, maka sisa lahan itu tidak ditanami.

Dari solusi di atas, keuntungan yang didapat :

Apel : 6 x Rp 30.000,00 = Rp 180.000,00

Jambu : 4 x Rp 10.000,00 = Rp 40.000,00

Jagung : 36 x Rp 4.000,00 = Rp 144.000,00

Maka total keuntungan yang didapatkan dengan cara Greedy by Profit adalah Rp 364.000,00.

- Penyelesaian dengan Greedy by Time :  
Apabila diselesaikan dengan metode Greedy by Time, maka solusi yang akan didapatkan adalah :

Jagung : 40 x 4 penanaman (360 hari, 20 m<sup>2</sup>)

Untuk merealisasikan perhitungan ini, tanam 40 tanaman jagung yang akan memenuhi semua lahan. Hal ini dapat dilakukan sebanyak 4 kali, yaitu setiap 90 hari.

Dari solusi ini, keuntungan yang akan didapatkan adalah sebesar 160 x Rp 4.000,00, yaitu Rp 640.000,00.

- Penyelesaian dengan Greedy by Space :  
Apabila diselesaikan dengan metode Greedy by Space, maka solusi yang akan didapatkan adalah :

Cabai : 500 x 3 penanaman (330 hari, 20 m<sup>2</sup>)

Merealisasikan metode ini sama dengan merealisasikan metode Greedy by Time, yaitu dengan menanam sebanyak 500 tanaman cabai yang akan memenuhi semua lahan. Hal tersebut dilakukan sebanyak 3 kali, yaitu setiap 110 hari.

Dari solusi ini, keuntungan yang akan didapatkan adalah sebesar 1.500 x Rp 400,00, yaitu Rp 600.000,00.

Dari ketiga hasil diatas, dapat dilihat bahwa yang memberikan keuntungan optimal pada contoh kasus ini adalah metode Greedy by Time.

Apabila variabel masukan pengguna diubah, misalnya:

Luas lahan : 20 m<sup>2</sup>.

Waktu Tanam : 150 hari.

- Penyelesaian dengan Greedy by Profit  
Penyelesaian yang diberikan oleh metode Greedy by Profit adalah :

Jambu : 20 x 1 penanaman (150 hari, 20 m<sup>2</sup>)

Dengan metode ini, keuntungan yang didapat adalah : 20 x Rp 10.000,00, yaitu Rp 200.000,00.

- Penyelesaian dengan Greedy by Time  
Penyelesaian yang diberikan oleh metode Greedy by Time adalah :

Jagung : 40 x 1 penanaman (90 hari, 20 m<sup>2</sup>)

Dengan metode ini, keuntungan yang didapat adalah : 40 x Rp 4.000,00, yaitu Rp 160.000,00.

- Penyelesaian dengan Greedy by Space  
Penyelesaian yang diberikan oleh metode Greedy by Space adalah :

Cabai : 500 x 1 penanaman (110 hari, 20 m<sup>2</sup>)

Dengan metode ini, keuntungan yang didapat adalah : 500 x Rp 400,00, yaitu Rp 200.000,00.

Dari ketiga hasil diatas, yang memberikan keuntungan optimal adalah metode Greedy by Profit dan Greedy by

Space. Walaupun secara logika, metode Greedy by Space lebih menguntungkan pada contoh soal ini (karena dengan metode Greedy by Space, pengguna mendapatkan Rp 200.000,00 dalam waktu 110 hari, sedangkan dengan metode Greedy by Profit, pengguna mendapatkan Rp 200.000,00 dalam waktu 150 hari). Hasil ini berbeda dari contoh sebelumnya dimana hasil optimal dicapai oleh metode Greedy by Time.

Pada contoh kasus terakhir ini, pengguna memasukan data baru pada kamus tanaman, tambahan tersebut dapat dilihat pada Tabel 2. Tambahan Data Tanaman

Tabel 2. Tambahan Data Tanaman

No	Nama	Profit (Rp)	Waktupanen (hari)	Space (m2)
5	BlueBerry	9.000	140	0.8

Luas Lahan : 12 m<sup>2</sup>

Waktu Tanam : 150 hari

Apabila data ini diperiksa dengan tiga metode, maka :

- Penyelesaian dengan Greedy by Profit  
Penyelesaian yang diberikan oleh metode ini adalah :

Jambu : 12 x 1 penanaman (150 hari, 12m<sup>2</sup>)

Keuntungan bila memakai metode ini : 12 x Rp 10.000,00, yaitu sebesar Rp 120.000,00.

- Penyelesaian dengan Greedy by Time  
Penyelesaian yang diberikan oleh metode ini adalah :

Jagung : 24 x 1 penanaman (90 hari, 12m<sup>2</sup>)

Keuntungan bila memakai metode ini : 24 x Rp 9.000,00, yaitu sebesar Rp 96.000,00.

- Penyelesaian dengan Greedy by Space  
Penyelesaian yang diberikan oleh metode ini adalah :

Cabai : 240 x 1 penanaman (110 hari, 12m<sup>2</sup>)

Keuntungan bila memakai metode ini : 240 x Rp 400,00, yaitu sebesar Rp 96.000,00.

Ketiga metode diatas memberikan hasil maksimal sebesar Rp 120.000,00, yaitu dari hasil Greedy by Profit.

Padahal untuk contoh kasus ini, ada cara realisasi lain yang akan memberikan hasil optimal, yaitu :

BlueBerry : 15 x 1 penanaman (140 hari, 12m<sup>2</sup>)

Keuntungan dari hasil ini adalah : 15 x Rp 9.000,00,

yaitu sebesar Rp 135.000,00.

## V. KESIMPULAN

Ketiga pengembangan dari algoritma Greedy ini (Greedy by Profit, Greedy by Time, dan Greedy by Space) hanya merupakan pendekatan untuk mencari hasil optimal dari permasalahan optimasi perkebunan. Terkadang ketiga metode ini tidak dapat memberikan hasil yang optimal, seperti pada contoh kasus terakhir pada Bab IV.

Setiap metode memiliki kelebihan dan kekurangannya masing-masing. Kelebihan dan kekurangan tersebut ditentukan oleh faktor-faktor seperti luas lahan, waktu tanam, serta data-data tanaman, yaitu profit, waktu panen, serta space.

Algoritma ini cocok digunakan untuk pengguna yang membutuhkan komputasi yang cepat. Hal ini disebabkan oleh kecilnya kompleksitas algoritma ini, yaitu hanya sebesar  $O(n)$  saja.

## REFERENSI

- [1] Munir, Rinaldi. 2009. "Diktat Kuliah IF3051, Strategi Algoritma," Bandung : Program Studi Teknik Informatika ITB.
- [2] [http://www.ipard.com/art\\_perkebun/des14-04\\_wrs-Lasp](http://www.ipard.com/art_perkebun/des14-04_wrs-Lasp). Tanggal akses : 20 Agustus 2011, pukul 13.00.

## 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



Daniel Widya Suryanata  
13509083