

# Implementasi Algoritma Greedy Pada Permainan Linyca

Anselmus Krisma Adi Kurniawan - 13508012

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

If18012@students.if.itb.ac.id

Pada makalah ini akan dibahas mengenai pengimplementasian algoritma greedy pada permainan linyca. Pada bagian pertama dijelaskan latar belakang pembuatan makalah ini kemudian selanjutnya pada bagian kedua dijelaskan mengenai algoritma greedy itu sendiri. Algoritma greedy yang akan kita gunakan adalah algoritma greedy yang umum, bukan algoritma greedy yang sudah divariasikan seperti *Relaxed Greedy Algorithm*. Setelah kita mengetahui mengenai algoritma greedy kita juga perlu tau mengenai permainan yang akan menjadi objek implementasi kita yaitu linyca yang dijelaskan pada bagian ketiga makalah ini. Linyca merupakan permainan freeware yang dengan mudah dapat ditemukan di internet. Tujuan dari permainan ini adalah untuk mendapatkan nilai tertinggi dengan menghancurkan pasangan balok yang satu warna.

Setelah kita mengetahui mengenai algoritma greedy dan permainan linyca, pada bagian keempat kita mulai melakukan implementasinya. Implementasi yang kita lakukan pertama kita harus mengubah permainan linyca ini menjadi kumpulan *list of integer*, nilai integer disini merepresentasikan warna yang ada dalam permainan ini. Pada bagian kelima terdapat analisis mengenai permasalahan yang kita bahas. Kemudian setelah itu terdapat kesimpulan serta daftar referensi yang digunakan dalam makalah ini.

**Keyword :** Greedy, Linyca, List, puzzle game

## I. PENDAHULUAN

Sering kali kita dapatkan di internet berbagai *puzzle game* yang menarik bagi kita. Tanpa sadar sebagian besar permainan tersebut sebenarnya dapat kita selesaikan dengan banyak sekali algoritma pemecahan masalah yang ada. Kita ambil contoh algoritma greedy, pada algoritma greedy sendiri kita hanya perlu menentukan kriteria yang akan kita gunakan sebagai patokan misalnya berdasarkan jumlah koin terbanyak. Dengan algoritma ini kita dapat menyelesaikan banyak sekali *puzzle game* yang menggunakan kriteria tertentu pula sebagai tujuan akhir game tersebut.

Linyca merupakan salah satu *puzzle game* yang ada di internet dan tersedia dengan gratis, kita dapat langsung memainkannya secara *online*. Permainan ini dapat kita gunakan sebagai contoh implementasi algoritma greedy yang akan kita bahas lebih jelas pada makalah ini.

## II. ALGORITMA GREEDY

*Greedy* berarti rakus, tamak, dan lainnya. Prinsip utama algoritma greedy adalah mengambil apa yang bisa didapatkan sekarang. Selain itu juga merupakan sebuah algoritma optimasi dalam menyelesaikan masalah dengan membuat hasil optimal lokal pada langkah yang dilakukan dengan harapan dapat menghasilkan hasil yang optimal global di akhirnya nanti.

Pada umumnya algoritma *greedy* memiliki 5 elemen utama yaitu

1. Himpunan kandidat, C
2. Himpunan solusi, S
3. Fungsi seleksi
4. Fungsi objektif
5. Fungsi kelayakan

dari kelima itu dapat kita tarik kesimpulan bahwa algoritma *greedy* melibatkan pencarian terhadap sebuah himpunan bagian, S, dalam himpunan kandidat, C. Yang dimana S harus memenuhi kriteria tertentu dan S dioptimisasi oleh fungsi objektif. Kita ambil contoh masalah penukaran uang :

- Himpunan kandidat : himpunan koin yang merepresentasikan nilai 1, 5, 10, 25 paling sedikit mengandung satu koin untuk setiap nilai.
- Himpunan solusi : total nilai koin yang dipilih tepat sama dengan nilai yang ditukarkan.
- Fungsi seleksi : pilih koin yang bernilai tertinggi dari koin yang tersisa.
- Fungsi layak : memeriksa apakah nilai total dari himpunan koin yang dipilih tidak melebihi uang yang harus dibayar.
- Fungsi objektif : jumlah koin yang digunakan minimum.

Skema umum algoritma *greedy* dapat dilihat di bawah ini :

```
function greedy(input C:
himpunan_kandidat) →
himpunan_kandidat
Deklarasi
  x : kandidat
  S : himpunan_kandidat
Algoritma :
  S ← {}
```

```

while (not SOLUSI(S) and (C ≠ {}))
do
    x ← SELEKSI(S)
    C ← C - {x}
    if LAYAK(S U {x}) then
        S ← S U {x}
    endif
endwhile

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

```

Meskipun algoritma *greedy* dapat dikatakan sebagai algoritma yang cukup mudah diimplementasikan pada permasalahan optimasi tapi tidak selalu algoritma ini dapat memberikan hasil yang optimum, hal ini dikarenakan :

1. Algoritma *greedy* tidak melakukan operasi secara menyeluruh terhadap semua kemungkinan yang ada.
2. Banyak sekali fungsi seleksi yang dapat diaplikasikan sehingga kita harus menentukan fungsi seleksi mana yang paling optimal.

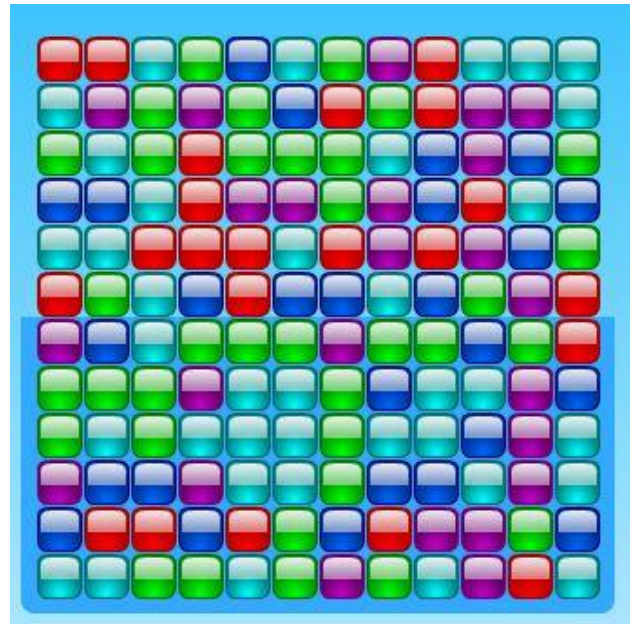
Algoritma *greedy* juga memiliki beberapa variasi yang dapat diaplikasikan juga yaitu :

1. Pure Greedy Algorithm
2. Orthogonal Greedy Algorithm
3. Relaxed Greedy Algorithm

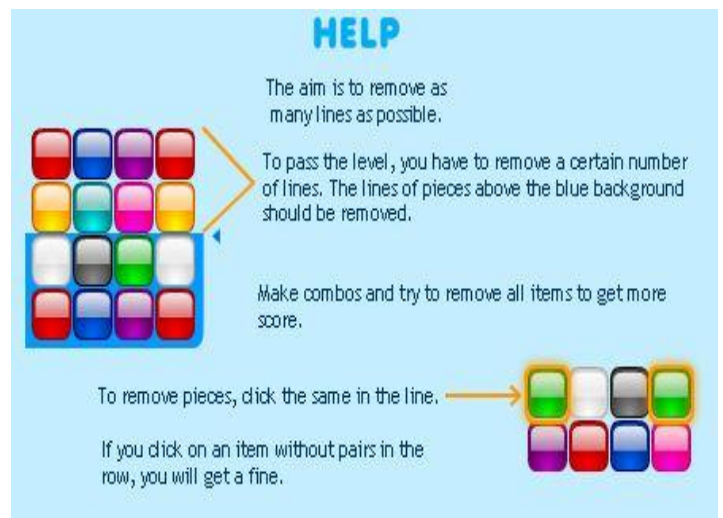
masing-masing dari algoritma diatas memiliki karakteristik yang berbeda-beda sehingga dapat menghasilkan hasil yang optimal untuk masing-masing kasus yang ditangani.

### III. LINYCA

Merupakan *puzzle game* sederhana yang ada di internet dan kita dapat memainkannya tanpa harus melakukan pembayaran. Tujuan dari game ini adalah untuk mendapatkan nilai tertinggi dengan cara menghancurkan balok-balok yang memiliki warna yang sama. Semakin banyak balok yang kita hancurkan setiap waktu, semakin besar pula nilai yang akan kita dapatkan. Salah satu langkah yang dapat diterapkan adalah menghitung semua kemungkinan balok yang memiliki pasangan warna yang paling banyak.



Gambar 3.1 Permainan Linyca



Gambar 3.2 Menu Bantuan untuk Linyca

### IV. IMPLEMETASI

Dengan melihat sekilas tentang permainan linyca, kita dapat menerapkan algoritma *greedy* untuk membantu kita menyelesaikan permasalahan yang ada. Pertama kali, kita harus mendefinisikan tujuan kita. Tujuan kita adalah mengimplementasikan algoritma *greedy* agar nilai yang kita dapatkan menjadi optimal. Hal tersebut dapat diperoleh dengan menghancurkan balok-balok yang memiliki warna yang sama dan paling banyak diantara baris dan warna yang mungkin. Kemudian setelah itu kita perlu menspesifikasikan bagaimana kita akan menggunakan algoritma *greedy* kita. Dari pembahasan sebelumnya kita tau bahwa algoritma *greedy* memiliki lima elemen, yaitu

1. Himpunan kandidat, C
2. Himpunan solusi, S
3. Fungsi seleksi

4. Fungsi objektif
  5. Fungsi kelayakan
- maka melihat dari masalah yang ada kita dapat menentukan kelima elemen tersebut untuk permasalahan permainan Linyca ini, yaitu

1. Himpunan kandidat  
Semua balok yang mungkin, yaitu balok yang dalam satu baris terdapat balok lain yang memiliki warna yang sama.
2. Himpunan solusi  
Semua balok berhasil dihancurkan atau mendapatkan nilai yang optimal
3. Fungsi seleksi  
Pilih balok yang memiliki pasangan dengan jumlah paling banyak
4. Fungsi objektif  
Apakah balok yang dipilih tidak membuat balok yang lain tidak memiliki pasangan
5. Fungsi kelayakan  
Balok yang dipilih maksimal

Dengan menggunakan kelima elemen diatas kita dapat membuat sebuah program yang mampu mensimulasikan permasalahan kita. Program ini menggunakan pendekatan *list*, yaitu kolom yang ada dirubah menjadi *list*, sehingga dengan contoh pada gambar 4.1 didapatkan 12 *list*. *List* yang kita gunakan adalah *list integer* dimana warna-warna yang ada direpresentasikan ke dalam angka integer.

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

Gambar 4.1 Representasi dalam List

Dari kumpulan *list* tersebut kita perlu mendapatkan nilai angka yang sering muncul untuk setiap *index* di *list*.

1	2	3	4	5	6	7	8	3	1	2	2	2=3
3	2	1	4	1	1	7	3	9	1	2	3	1=4
1	2	3	1	5	6	7	8	9	3	1	2	1=3
1	2	3	1	5	6	2	1	3	1	2	3	.
1	2	3	4	8	1	7	3	9	1	2	3	.
3	1	7	4	2	4	4	8	9	3	1	3	.
1	2	3	4	5	0	1	3	9	1	1	2	.
2	2	3	2	0	6	7	8	9	1	3	2	.
1	2	6	8	5	2	3	8	3	1	3	2	.
3	2	3	7	3	6	7	8	2	1	2	2	.
1	3	3	4	5	6	7	4	9	1	2	3	.
1	3	3	4	5	6	7	3	9	1	2	2	3=3
List 1	List 2	List 3	List 4	List 5	List 6	List 7	List 8	List 9	List 10	List 11	List 12	

Gambar 4.2 List setelah dilakukan perhitungan

Dari *index* yang ada, cari nilai yang paling banyak memiliki pasangan, dalam kasus ini adalah *index* ke-11 untuk nilai 1 yang memiliki 4 pasangan. Setelah kita mendapatkan pasangan tersebut, kita hapus elemen *list* pada *index* ke-11 yang memiliki nilai 1, sehingga didapatkan list dibawah ini

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

Gambar 4.3 List setelah elemen dihapus

Proses ini diulang sampai tidak dapat lagi menemukan balok yang memiliki pasangan warna.

*Pseudo code* dari program dapat dilihat di bawah ini

```

procedure linyca(input/output L :
array of list)
{
Prosedur melakukan rediksi terhadap
elemen list yang dipilih
I.S : list representasi permainan
Linyca
F.S : elemen list yang memiliki
pasangan paling banyak dihapus dari
list
}
Deklarasi
  Li : list
  X : integer
Algoritma
  Li ← elemen_max_per_kolom(L)
  X ← max(Li)
  For Semualist dalam L do
    Hapus elemen list yang sama
    dengan X
  endfor

```

Dengan *pseudo code* didapatkan hasil dibawah ini

```

Hasil :
List 1 : {}
List 2 : {}
List 3 : {}
List 4 : {}
List 5 : {}
List 6 : {}
List 7 : {}
List 8 : {}
List 9 : {}
List 10 : {}
List 11 : {}
List 12 : {}
Nilai : 2200

```

Gambar 4.4 Hasil Program 1

```

Hasil :
List 1 : {}
List 2 : {}
List 3 : {}
List 4 : {}
List 5 : {}
List 6 : {}
List 7 : {}
List 8 : {}
List 9 : {}
List 10 : {}
List 11 : {}
List 12 : {}
Nilai : 2600

```

Gambar 4.5 Hasil Program 2

Dapat kita lihat bahwa semua elemen *list* dapat habis dan kita mendapatkan hasil yang optimal, tetapi dengan pola *puzzle* yang berbeda didapatkan hasil dibawah ini

```

Hasil :
List 1 : {1}
List 2 : {}
List 3 : {}
List 4 : {3, 4}
List 5 : {}
List 6 : {}
List 7 : {2, 3}
List 8 : {}
List 9 : {}
List 10 : {4, 1, 2}
List 11 : {}
List 12 : {}
Nilai : 1500

```

Gambar 4.6 Hasil Program 3

```

Hasil :
List 1 : {}
List 2 : {}
List 3 : {6}
List 4 : {}
List 5 : {}
List 6 : {}
List 7 : {5, 4}
List 8 : {}
List 9 : {}
List 10 : {}
List 11 : {}
List 12 : {4}
Nilai : 1800

```

Gambar 4.7 Hasil Program 4

Gambar diatas merupakan salah satu contoh bahwa algoritma *greedy* tidak selalu menghasilkan hasil yang optimal.

## V. ANALISIS

Algoritma *greedy* merupakan salah satu algoritma yang dapat diimplementasikan ke sebagian besar masalah yang ada di kehidupan kita. Tetapi algoritma ini tidak menjamin ditemukannya solusi yang optimal untuk setiap permasalahan.

Melihat dari permasalahan yang kita angkat pada makalah ini mengenai penyelesaian permainan Linyca dengan menggunakan algoritma *greedy* dapat kita lihat bahwa algoritma *greedy* yang diimplementasikan memiliki anomali. Terkadang mendapatkan hasil yang optimal tapi terkadang juga tidak mendapatkan hasil yang optimal, tetapi dilihat dari hasilnya dapat kita ketahui juga bahwa algoritma ini cukup membantu dalam menyelesaikan permasalahan kita.

## VI. KESIMPULAN

Algoritma *greedy* sangat mudah untuk diimplementasikan tetapi hasil yang didapatkan tidak selalu merupakan hasil yang optimal hal ini dikarenakan algoritma *greedy* tidak melakukan operasi terhadap semua kemungkinan yang ada serta banyaknya fungsi seleksi yang dapat kita pilih sehingga belum tentu fungsi yang kita pilih merupakan fungsi yang tepat.

## REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika ITB, 2008.
- [2] Linyca Puzzle Game, <http://www.freepuzzlegames.biz/game/613/Linyca.html>, Tanggal akses 7 Desember 2010, pukul 23.50.
- [3] J. Bang-Jensen, G. Gutin and A. Yeo, When the greedy algorithm fails. *Discrete Optimization 1* (2004), 121–127.
- [4] *Introduction to Algorithms* (Cormen, Leiserson, and Rivest) 1990, Chapter 16 "Greedy Algorithms" p. 329.

## 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, 7 Desember 2010



Anselmus Krisma Adi Kurniawan - 13508012