

Pemanfaatan Algoritma Greedy Untuk Penyelesaian Permainan Candy Crush

Ricky Kennedy - 13516105

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

Rickykennedy25@gmail.com

Abstract—Permainan Candy Crush merupakan permainan teka-teki yang berhasil menarik minat sebagian besar orang saat ini. Algoritma *greedy* dapat digunakan untuk membantu menyelesaikan permasalahan-permasalahan yang terdapat pada permainan Candy Crush. Oleh karena itu dalam makalah ini akan dibahas algoritma untuk menyelesaikan objektif - objektif pada setiap tingkatnya. Himpunan kandidat berisi pilihan-pilihan langkah yang dapat menyelesaikan objektif. Himpunan solusi merupakan langkah yang terpilih dari himpunan kandidat yang memberikan keuntungan terbesar.

Keywords—*permainan; algoritma; langkah; objektif;*

I. PENDAHULUAN

Pada zaman era globalisasi ini, ilmu pengetahuan dan teknologi berkembang sangat pesat sehingga menawarkan kemudahan bagi manusia untuk menjalani aktivitas sehari-hari. Dimulai dari pekerjaan, olahraga, belajar hingga mencari hiburan (bermain) yang selalu dimanjakan oleh teknologi yang berkembang saat ini.

Pada era modern ini sebagian besar permainan-permainan telah dienkapsulasi kedalam dunia *virtual*. Sehingga dapat dimainkan di handphone atau pun komputer. Akibatnya banyak developer *game* internasional dan nasional berlomba-lomba untuk menciptakan *game* yang menarik, menyenangkan, tidak membosankan bagi semua orang.

Teka-teki merupakan sebuah masalah yang diciptakan untuk menguji pengetahuan seseorang. Teka-Teki biasanya dipecah menjadi bagian-bagian kecil secara logis sehingga untuk dapat menyelesaikan sebuah teka-teki diperlukan gabungan dari solusi-solusi tiap bagian. teka-teki dapat di implementasi menjadi beberapa bentuk seperti teka-teki angka, teka-teki kata, teka-teki logika. Sehingga permainan teka-teki cocok untuk berbagai kalangan terutama pada kalangan anak-anak karena permainan teka-teki dapat dijadikan sebagai sarana belajar sambil bermain. Oleh sebab itu permainan teka teki merupakan salah satu ide yang sering diterapkan oleh para developer *game* diseluruh dunia.

Permainan teka-teki merupakan permainan yang umumnya diciptakan oleh para developer *game* di seluruh dunia untuk

menarik perhatian dari para pecinta permainan *virtual*. Permainan-permainan teka-teki yang dapat dimainkan secara *virtual* dan fenomenal saat ini adalah Tetris, PokoPang, Catur, Candy Crush dan lain lain.



Gambar 1.1 Permainan Candy Crush

Source : <http://images.pocketgamer.co.uk/artwork/na-tyoo/candy-crush-saga-ios-3.jpg>

Permainan Candy Crush ini mampu menghipnotis jutaan jiwa manusia saat ini untuk berlomba-lomba memainkan permainan Candy Crush dengan tujuan mencapai tingkat yang setinggi-tingginya dengan tingkat kesulitan yang meningkat.

II. DASAR TEORI

A. Permainan Candy Crush

Permainan Candy Crush merupakan salah satu permainan yang cukup fenomenal saat ini. Permainan ini bertujuan untuk mencapai target nilai tertentu dan/atau menghilangkan seluruh *jelly* yang terdapat pada kotak dengan menukarkan permen yang tepat besisian sehingga mencocokkan 3 atau lebih permen berturut-turut.

Mencocokkan Permen dalam petualangan teka-teki ini untuk maju ke tingkat berikutnya dengan harapan mencapai perasaan manis itu. Pemikiran cepat dan gerakan cerdas

dihadiah dengan berwarna pelangi dan permen combo yang lezat.

B. Konsep Algoritma Greedy

1. Definisi Algoritma Greedy

Dalam kehidupan sehari-hari, terdapat banyak persoalan yang menuntut pencarian solusi optimum. Persoalan tersebut dinamakan persoalan optimasi (*optimization problems*). Persoalan optimasi adalah persoalan yang tidak hanya mencari sekedar solusi, tetapi mencari solusi terbaik.

Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Algoritma *greedy* memiliki prinsip "*take what you can get now!*". Algoritma *greedy* adalah algoritma yang memecahkan masalah dengan langkah perlangkah. Pada setiap langkahnya :

- Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan.
- Berharap bahwa dengan memilih langkah optimum lokal pada setiap langkah akan berakhir dengan optimum global.

2. Definisi Algoritma Greedy

a. Himpunan kandidat, C.

Himpunan ini berisi dengan elemen-elemen pembentuk solusi. Contohnya himpunan koin, himpunan job yang akan dikerjakan, himpunan simpul di dalam graf, dan lain-lain. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.

b. Himpunan solusi, S.

Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.

- Fungsi seleksi — dinyatakan dengan predikat SELEKSI — yaitu fungsi yang pada setiap langkah-langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Biasanya setiap kandidat, x , di-assign sebuah nilai numerik, dan fungsi seleksi memilih x yang mempunyai nilai terbesar atau memilih x yang mempunyai nilai terkecil.

- Fungsi kelayakan (*feasible*) - dinyatakan dengan predikat LAYAK - yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan

solusi layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada. Kandidat yang layak dimasukkan kedalam himpunan solusi. sedangkan kandidat yang tidak layak dibuang dan tidak pernah di pertimbangkan lagi.

- fungsi objektif yaitu yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

3. Pseudocode algoritma greedy

- Predikat Fungsi SELEKSI, LAYAK, dan SOLUSI.

```
function SELEKSI (C : himpunan_kandidat) →  
kandidat  
{ me-return sebuah kandidat yang dipilih dari C  
berdasarkan kriteria yang di pilih }
```

```
function SOLUSI (S : himpunan_kandidat) →  
boolean  
{ true jika S adalah Solusi dari persoalan;  
sebaliknya false jika S belum menjadi solusi }
```

```
function LAYAK (S : himpunan_kandidat) →  
boolean  
{ bernilai true jika S merupakan solusi yang tidak  
melanggar kendala; sebaliknya bernilai false jika S  
melanggar kendala }
```

- Algoritma *greedy*.

```
function greedy (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)
```

```

    c ← c - {x}
    if LAYAK ( S ∪ {x} ) then
        S ← S ∪ {x}
    endif
endwhile
{ SOLUSI(S) or c = {} }

if SOLUSI(S) then
    return S
else
    write('tidak ada solusi')
endif

```

Catatan : Optimum global belum tentu merupakan solusi optimum (terbaik), tetapi sub-optimum atau pseudooptimum.

Alasan :

- i. Algoritma greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada (sebagaimana pada metode exhaustive search)
- ii. Terdapat beberapa fungsi SELEKSI yang berbeda, sehingga kita harus memilih fungsi yang tepat jika kita ingin algoritma menghasilkan solusi optimal

Jadi, pada sebagian masalah algoritma *greedy* tidak selalu berhasil memberikan solusi yang optimal.

4. Pemecahan Masalah Dengan *greedy*

a. Masalah Penekaran uang

Contoh 1:

misal : A = 32, koin yang tersedia: 1, 5, 10, dan 25
 Langkah 1: pilih 1 buah koin 25 (Total = 25)
 Langkah 2: pilih 1 buah koin 5 (Total = 25 + 5 = 30)
 Langkah 3: pilih 2 buah koin 1 (Total = 25+5+1+1= 32)
 Solusi: Jumlah koin minimum = 4 (solusi optimal!)

Contoh 2:

Koin: 5, 4, 3, dan 1 Uang yang ditukar = 7. Solusi greedy: 7 = 5 + 1 + 1 (3 koin) tidak optimal Solusi optimal: 7 = 4 + 3 (2 koin)

Contoh 3:

Koin: 10, 7, 1 Uang yang ditukar: 15
 Solusi greedy: 15 = 10 + 1 + 1 + 1 + 1 + 1 (6 koin)
 Solusi optimal: 15 = 7 + 7 + 1 (hanya 3 koin)

Contoh 4:

Koin: 15, 10, dan 1 Uang yang ditukar: 20
 Solusi greedy: 20 = 15 + 1 + 1 + 1 + 1 + 1 (6 koin)
 Solusi optimal: 20 = 10 + 10 (2 koin)

b. Minimisasi Waktu di dalam Sistem

Persoalan: Sebuah server (dapat berupa processor, pompa, kasir di bank, dll) mempunyai n pelanggan (customer, client) yang harus dilayani. Waktu pelayanan untuk setiap pelanggan i adalah t_i . Minimumkan total waktu di dalam sistem:

T = total waktu di dalam sistem

Ekivalen dengan meminimumkan waktu rata-rata pelanggan di dalam sistem.

Contoh :

Tiga pelanggan dengan $t_1 = 5$, $t_2 = 10$, $t_3 = 3$;

Enam urutan pelayanan yang mungkin:

Urutan	T
1, 2, 3:	$5 + (5 + 10) + (5 + 10 + 3) = 38$
1, 3, 2:	$5 + (5 + 3) + (5 + 3 + 10) = 31$
2, 1, 3:	$10 + (10 + 5) + (10 + 5 + 3) = 43$
2, 3, 1:	$10 + (10 + 3) + (10 + 3 + 5) = 41$
3, 1, 2:	$3 + (3 + 5) + (3 + 5 + 10) = 29$
3, 2, 1:	$3 + (3 + 10) + (3 + 10 + 5) = 34$

cetak hitam adalah solusi optimum yang didapatkan.

III. STRATEGI GREEDY DALAM PERMAIAN CANDY CRUSH

Perhitungan permainan Candy Crush memberikan nilai yang besar adalah berdasarkan jumlah permen yang bersisian pada satu kondisi. Semakin banyak jumlah permen yang dapat bersisian pada satu waktu maka akan mendapat nilai yang besar dan juga memberikan efek khusus terhadap permen yang berkaitan.

Oleh karena itu untuk memperoleh nilai yang maksimum pada setiap langkahnya bisa saja menggabungkan/menyusun bersisian permen yang memiliki efek khusus atau menukar permen sehingga permen dengan jumlah sebanyak mungkin (minimum 3) yang dapat bersisian dalam satu keadaan agar memperoleh nilai maksimum.

Berangkat dari aturan permainan diatas, kita dapat mendeklarasikan dua buah algoritma greedy untuk menyelesaikan permainan Candy Crush pada setiap tahapnya.

Apabila tujuan dari sebuah tingkatan permainan candy crush adalah menghancurkan jelly dan/atau mencapai nilai tertentu, maka ide greedy yang dapat diterapkan adalah sebagai berikut :

1. Melakukan Cek apakah masih terdapat jelly pada kotak
 - a. jika ya maka cek apakah terdapat permen yang dapat bersisian pada kotak yang terdapat *jelly*.
 - i. jika ya tukar kan permen agar dapat bersisian pada kotak yang terdapat jelly yang dapat memecahkan kotak jelly dengan jumlah tebanyak.
 - ii. jika tidak maka tukar permen yang dapat bersisian memberikan nilai terbesar.
 - b. jika ya maka cek apakah terdapat permen yang dapat bersisian pada kotak yang terdapat *jelly*.
2. Lakukan kembali langkah 1 sampai kesempatan melangkah habis.

jika tujuan dari sebuah tingkatan permainan candy crush hanya mencapai nilai tertentu, maka ide greedy yang dapat diterapkan adalah sebagai berikut :

1. Mencari semua kemungkinan permen yang bersisian yang dapat memberikan nilai yang maksimum pada setiap langkah tersebut.
2. Lakukan kembali langkah 1 sampai kesempatan melangkah habis.

IV. PENERAPAN ALGORITMA GREEDY PADA PERMAINAN CANDY CRUSH

Penyelesaian permainan candy crush yang memiliki tujuan hanya memecahkan sejumlah yang terdapat pada kotak. Seperti pada gambar berikut :

Gambar 4.1 permainan candy crush (1).

Contoh pada tingkat ini adalah memecahkan atau menghilangkan 12 kotak jelly yang terletak pada layar. kita asumsikan kotak paling kiri bawah memiliki koordinat (1,1). Dengan menerapkan strategi *greedy* yang telah di jelaskan pada bab sebelumnya maka langkah pertama yang kita ambil

adalah dengan menemukan permen yang dapat bersisian pada kotak jelly dan juga menghasilkan nilai maksimum. pada langkah ini kita memiliki dua pilihan yaitu :

1. menukarkan permen hijau (3, 8) dengan merah (3, 9) yang terletak pada kotak atau permen merah (4, 9) dengan permen biru (4, 8) ini akan menghasilkan hal yang sama yaitu permen merah (3 sampai 5) akan bersisian pada baris 9 sehingga melakukan penukaran permen merah terhadap permen hijau ataupun biru akan menghasilkan nilai yang sama besar.
2. menukarkan permen hijau (3, 5) dengan biru (2, 5) juga menghasilkan 3 permen yang bersisian sekaligus. Hal ini memiliki nilai yang sama dengan pilihan no 1.

misalkan kita pilih menukarkan permen hijau dan permen merah yang terdapat pada pilihan 1 maka akan menghasilkan hal sebagai berikut :

Gambar 4.2 permainan candy crush (2).

Setelah melakukan langkah sebelumnya maka tersisa 9 kotak jelly lagi yang harus dihancurkan. pada langkah ini kita punya dua pilihan untuk menghancurkan jelly pada kotak :

1. menukarkan permen hijau (2, 6) dengan permen ungu (3, 6).
2. menukarkan permen hijau (3, 5) dengan permen biru (2, 5).

Kedua langkah ini akan menghasilkan nilai yang sama besar karena 3 buah permen akan bersisian. Namun yang harus diperhatikan bahwa pilihan pertama akan menghasilkan kotak jelly yang hancur lebih banyak dibandingkan dengan pilihan kedua. maka sesuai dengan prinsip *greedy* kita memiliki pilihan pertama. sehingga menghasilkan hal sebagai berikut :

permen biru (2, 5) maka kita akan mendapat hal sebagai berikut:

Gambar 4.3 permainan candy crush (3).

Setelah melakukan langkah sebelumnya maka tersisa 5 kotak jelly lagi yang harus dihancurkan. pada langkah ini kita masih memiliki dua pilihan untuk menghancurkan jelly pada kotak :

1. menukarkan permen oranye (4, 6) dengan permen ungu (5, 6).
2. menukarkan permen ungu (2, 6) dengan permen biru (2, 5).

Kedua pilihan memiliki keuntungan yang persis sama keduanya sama-sama menyusun 3 buah permen bersisian dan menghancurkan 1 kotak jelly. Dengan mengambil pilihan permata akan menghasilkan hal sebagai berikut :

Gambar 4.4 permainan candy crush (4).

Setelah melakukan langkah sebelumnya maka tersisa 4 kotak jelly lagi yang harus dihancurkan. Pada langkah ini kita hanya memiliki satu buah pilihan untuk menghancurkan jelly yang tersisa. Dengan menukarkan permen ungu (2, 6) dengan

Gambar 4.5 permainan candy crush (5).

Setelah melakukan langkah sebelumnya maka tersisa 3 kotak jelly lagi yang harus dihancurkan. Pada langkah ini kita punya dua pilihan untuk menghancurkan jelly pada kotak :

1. menukarkan permen ungu (3, 8) dengan permen merah (2, 8) atau permen biru (3, 7).
2. menukarkan permen hijau (4, 7) dengan permen merah (4, 6).

Kedua pilihan menghasilkan nilai yang sama besar dan hanya menghancurkan 1 kotak jelly. misalkan kita menukar permen merah dan biru maka kita akan mendapatkan hal sebagai berikut :

Gambar 4.6 permainan candy crush (6).

Setelah melakukan langkah sebelumnya maka tersisa 3 kotak jelly lagi yang harus dihancurkan. Pada langkah ini kita punya tiga pilihan untuk menghancurkan jelly pada kotak :

1. menukarkan permen merah (3, 8) dengan permen ungu (2, 8).

2. menukarkan permen hijau (4, 6) dengan permen merah (4, 5).
3. menukarkan permen merah (4, 7) dengan permen hijau (4, 6) atau permen merah (4, 2) dengan permen ungu (4, 3).

Pada ketiga menghancurkan kotak jelly dengan jumlah yang sama namun harus diperhatikan bahwa pilihan kedua dan ketiga hanya menyebabkan 3 buah permen bersisian sedangkan pada pilihan pertama dapat menyebabkan 4 buah permen bersisian sehingga ini dapat menghasilkan nilai yang lebih besar dibandingkan dengan pilihan kedua atau ketiga. maka dengan mengambil langkah pertama akan menghasilkan hal sebagai berikut :

Gambar 4.7 permainan candy crush (7).

Setelah melakukan langkah sebelumnya maka tersisa 1 kotak jelly lagi yang harus dihancurkan. Pada langkah ini kita punya dua pilihan untuk menghancurkan jelly pada kotak :

1. menukarkan permen hijau (4, 6) dengan permen merah (4, 5).
2. menukarkan permen merah (4, 7) dengan permen hijau (4, 6) atau permen merah (4, 2) dengan permen ungu (4, 3).

Kedua pilihan dapat menghancurkan jelly yang tersisa dan menyelesaikan permainan pada tingkat tersebut.

Gambar 4.8 permainan candy crush (8).

V. KESIMPULAN

Salah satu cara penyelesaian permainan Candy Crush adalah dengan menggunakan algoritma greedy namun algoritma greedy tidak selalu berhasil menyelesaikan setiap tingkat pada permainan candy crush. Karena dapat terjadi keadaan dimana langkah telah habis namun nilai masih kurang dari target atau jelly masih tersisa pada kotak sehingga menyebabkan kekelahan.

VI. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur dan terima kasih kepada Tuhan Yang Maha Esa atas selesainya pembuatan makalah ini. Penulis juga mengucapkan terima kasih kepada dosen mata kuliah IF2211 Strategi Algoritma, yakni Dr. Ir. Rinaldi Munir, MT., Dr. Nur Ulfa Maulidevi, ST., MSc., dan Dr. Mesayu Lelya Khodra, ST., MT. yang telah memberikan materi dan bimbingan, baik di dalam yang telah memberikan materi dan bimbingan yang bermanfaat dalam penulisan makalah ini.

REFERENCES

[1] Munir, Rinaldi. 2009. Diktat Kuliah IF2211 Strategi Algoritma. Bandung.
 [2] <https://play.google.com/store/apps/details?id=com.king.candycrushsaga> diakses pada tanggal 12 mei 2018
 [3] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Greedy-\(2018\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Greedy-(2018).pdf) diakses pada 12 mei 2018

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, 12 Mei 2018

Ricky Kennedy - 13516105