

Penerapan Algoritma Greedy pada Permainan Bubble Breaker

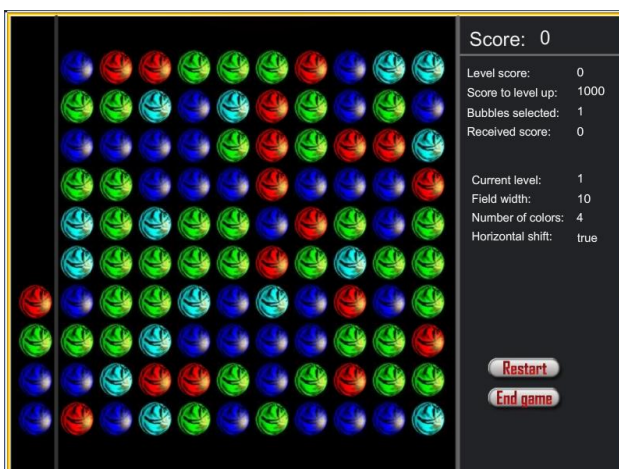
Roy Indra Haryanto - 13508026
 Program Studi Teknik Informatika
 Sekolah Teknik Elektro dan Informatika
 Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
 royindra@students.itb.ac.id

Abstract—Bubble Breaker adalah salah satu puzzle game yang menampilkan berbagai bola dengan beberapa warna yang berbeda. Bola – bola tersebut akan membentuk suatu rangkaian pada papan permainan dengan beberapa bola dengan warna yang sama saling menempel. Semakin banyak bola dengan warna yang sama yang dipilih, maka akan semakin besar poin yang didapat. Objektif dari permainan ini adalah meraih poin sebesar – besarnya dengan memilih bola dengan warna sama yang saling menempel. Untuk permasalahan ini, dapat digunakan algoritma greedy untuk meraih hasil yang optimal. Pada makalah ini akan dibahas bagaimana permasalahan dalam permainan Bubble Breaker, bagaimana pendekatan untuk memecahkan masalah dengan algoritma greedy, dan bagaimana menerapkan algoritma greedy tersebut untuk menyelesaikan permasalahan dalam permainan ini.

Index Terms—Bubble breaker, algoritma, greedy

I. INTRODUCTION

Permainan puzzle adalah salah satu pilihan permainan yang cukup sering dilirik oleh pecinta game karena selain dapat menghibur, permainan puzzle juga dapat melatih kita dalam berpikir. Permainan puzzle biasanya menggunakan logika – logika sederhana dalam memainkan permainan tersebut.



Gambar 1 Contoh tampilan game Bubble Breaker

Bubble breaker adalah salah satu permainan puzzle

dengan menampilkan berbagai bola yang memiliki kesamaan motif atau warna ataupun gambar. Tujuan dari permainan ini adalah meraih poin sebesar – besarnya dengan cara memilih bola yang berwarna sama yang saling menempel. Bola – bola tersebut biasanya terdiri dari 5 warna yang berbeda dan membentuk suatu rangkaian dalam suatu papan permainan. Semakin banyak bola berwarna sama yang saling menempel, maka akan semakin besar angka atau nilai yang pemain dapatkan.

Setiap rangkaian bola dengan warna yang sama yang dipilih atau dipecahkan dalam permainan ini mempunyai nilai – nilai tertentu sesuai dengan jumlah bola dengan warna yang sama yang menempel.

Poin yang didapat untuk setiap rangkaian bola yang berwarna sama adalah sebagai berikut :

Bola yang dipilih	Poin yang didapat
2	2
3	6
4	12
5	20
6	30
7	42
8	56
9	72
10	90
.	.
.	.
.	.
n	$n \times (n - 1)$

Tabel 1 Tabel Raihan Poin

Dapat dilihat dari tabel di atas bahwa semakin banyak jumlah bola yang memiliki warna yang sama dirangkaikan, maka akan semakin besar poin yang di dapat. Lebih jelasnya untuk penghitungan poin ini dapat dihitung dengan formula sebagai berikut :

$$\text{Poin} = n \times (n - 1)$$

Dengan n adalah jumlah bola dengan warna yang sama yang dirangkakan.

Rangkaian bola yang dapat dirangkakan dalam permainan ini adalah minimal 2 bola yang memiliki warna yang sama.

II. DASAR TEORI

Algoritma Greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Terdapat dua jenis persoalan optimasi, yaitu maksimasi dan minimasi.

Algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah dengan pada setiap langkah dilakukan hal berikut :

1. Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “*take what you can get now!*”)
2. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Persoalan optimasi untuk suatu masalah dalam konteks algoritma *greedy* disusun oleh elemen-elemen sebagai berikut:

1. Himpunan kandidat, C

Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.

2. Himpunan solusi, S

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

3. Fungsi Seleksi

Fungsi ini dinyatakan dengan predikat seleksi. Merupakan fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

4. Fungsi kelayakan

Fungsi ini dinyatakan dengan predikat layak. Fungsi kelayakan ini merupakan fungsi 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 (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi,

sedangkan yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.

5. Fungsi obyektif

Fungsi objektif ini merupakan sebuah fungsi yang memaksimalkan atau meminimumkan nilai solusi.

Berikut adalah skema umum 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 ∪ {x}) then
S ← S ∪ {x}
endif
endwhile
{SOLUSI(S) or C = {} }
if SOLUSI(S) then
return S
else
write('tidak ada solusi')
endif
```

Gambar 2 Skema umum algoritma greedy

Namun solusi optimum global yang diperoleh pada algoritma Greedy ini belum tentu merupakan solusi optimal, tetapi merupakan *sub-optimum* atau *pseudo-optimum*. Hal ini dikarenakan algoritma greedy tidak melakukan operasi secara menyeluruh pada semua alternatif solusi. Selain itu juga ada beberapa kasus yang mengakibatkan fungsi seleksi tidak bekerja dengan optimal.

III. DESKRIPSI MASALAH

Dalam pencarian solusi untuk penyelesaian permainan Bubble Breaker ini, kita ingin mendapatkan nilai sebesar mungkin yang bisa didapatkan. Nilai yang besar akan kita dapatkan apabila kita memilih untuk memecahkan rangkaian bola - bola dengan warna yang sama dengan jumlah yang besar pula. Oleh karena itu, kita harus selalu berusaha untuk memilih rangkaian dengan jumlah bola yang paling besar.

Jadi pada setiap kesempatan untuk memecahkan bola - bola tersebut, kita harus menemukan rangkaian yang paling besar dari seluruh bola yang ada pada papan permainan. Program atau pencarian solusi untuk memecahkan bolabola sewarna ini berakhir apabila tidak ada lagi bola-bola dengan warna yang sama yang memiliki posisi yang bersisian, atau dengan kata lain tidak ada lagi bola-bola sewarna dengan jumlah bola

pada rangkaian yang lebih dari 1. Kemudian permainan juga akan selesai dan nilai yang didapatkan dari memecahkan bola-bola sebelumnya tersebut akan diakumulasikan menjadi nilai akhir.

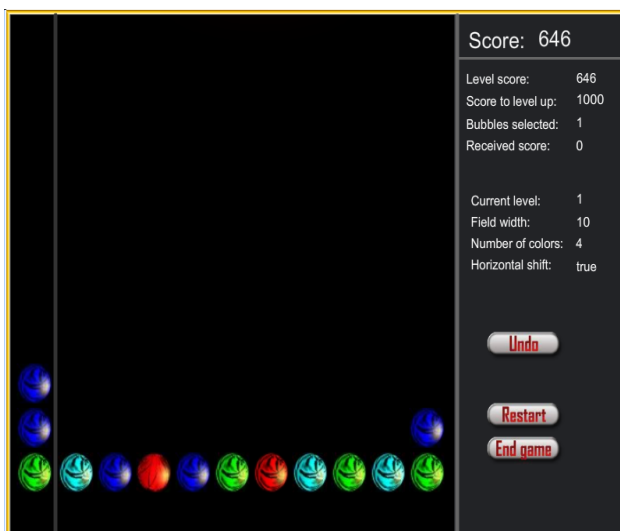
Perlu diingat bahwa apabila kita memilih salah satu bola yang membentuk rangkaian bola-bola sewarna, berarti kita akan memilih rangkaian tersebut untuk dipecahkan. Apabila terdapat 4 bola sewarna yang terangkai dan misalnya diberi nama bola 1, 2, 3, dan 4, hasilnya akan sama saja apabila kita memilih bola nomor 1, 2, 3, atau 4.

Bola manapun yang kita pilih dari keempat bola tersebut akan menyebabkan kita memecahkan keempat bola dalam rangkaian tersebut.

IV. IMPLEMENTASI GREEDY

Seperti yang telah disebutkan sebelumnya, untuk memperoleh hasil yang optimal (walaupun untuk beberapa kasus tidak akan memperoleh hasil yang maksimal) dapat digunakan algoritma greedy untuk melakukan pemilihan rangkaian bola mana yang akan dipilih.

Cara kerja algoritma greedy untuk dapat menyelesaikan permasalahan dalam permainan Bubble Breaker adalah dengan pertama – tama melihat seluruh bola yang ada dalam papan permainan. Kemudian program akan menyimpan bola yang memiliki warna yang sama yang membentuk rangkaian paling besar. Kemudian program akan memilih bola tersebut sehingga seluruh bola dalam rangkaian akan pecah dan mendapatkan poin yang optimal. Kemudian, program akan kembali melakukan hal yang sama yaitu memilih bola yang memiliki warna yang sama dengan rangkaian terbesar di dalam papan permainan yang tersisa. Program akan terus berjalan sampai jumlah bola pada papan permainan tidak memiliki rangkaian dengan jumlah bola lebih dari 1.



Gambar 3 Kondisi Berhenti

Pada masalah pencarian solusi ini, himpunan kandidat adalah semua bola yang terdapat di dalam papan permainan. Implementasi dari algoritma greedy ini adalah sebagai berikut :

- Himpunan solusi dalam permainan ini adalah rangkaian bola yang memiliki warna yang sama yang paling banyak jumlahnya.
- Fungsi seleksi dalam permasalahan ini adalah fungsi yang akan memilih bola – bola yang membuat rangkaian dengan jumlah bola terbanyak di dalam papan permainan.
- Fungsi kelayakan dalam permasalahan ini adalah fungsi yang memastikan bola yang dipilih memiliki rangkaian dengan bola yang berwarna sama dengan jumlah lebih dari 1.
- Fungsi objektif dalam permasalahan ini bertugas untuk memastikan semua bola yang dapat dipilih sudah dipilih, atau sudah tidak ada lagi bola dalam papan permainan yang dipilih.

Dari hasil analisis di atas, maka dapat dirumuskan *pseudocode*-nya sebagai berikut :

```
function getSameColor (input p: board,
posisi: Point) → integer
{ menerima masukan berupa isi papan
permainan dan mengembalikan integer jumlah
bola dalam rangkaian bola-bola dengan
warna yang sama di mana bola tersebut
tergabung. Apabila posisi yang ditunjuk
bukan merupakan bola (tempat kosong yang
tidak terisi bola), maka fungsi ini akan
mengembalikan nilai 0 }
```

```
procedure chooseBall (input p: board,
posisi: Point)
{ menerima masukan berupa data papan
permainan beserta seluruh isinya serta
posisi bola yang ditunjuk, kemudian
prosedur tersebut akan memilih bola
tersebut sehingga seluruh bola dalam
rangkaiannya akan pecah dan poin didapatkan.
}
```

```
function isFinish (input p: board) →
boolean
{ menerima masukan berupa data papan
permainan beserta seluruh isinya, kemudian
mengembalikan true jika tidak ada bola
yang dapat dipilih lagi dalam area
permainan (tidak ada yang memiliki
rangkaiannya lebih dari 1 bola) dan
mengembalikan false jika masih ada bola
yang dapat dipilih. }
```

```
function isValid (input p: board,
posisi: Point) → boolean
{ menerima masukan berupa data papan
permainan beserta seluruh isinya serta
posisi bola yang ditunjuk, kemudian
mengembalikan true jika rangkaian dari
bola yang dipilih tersebut mempunyai 2
atau lebih bola berwarna dalam
rangkaiannya, dan akan mengembalikan false
jika hanya ada 1 bola dalam rangkaian bola
yang dipilih tersebut. }
```

Fungsi seleksi :

```
function mostBall (input p:
board) → Point
{ menerima masukan berupa data papan
permainan beserta seluruh isinya dan akan
mengembalikan posisi bola yang memiliki
jumlah bola berwarna paling banyak dalam
rangkaiannya }
```

Deklarasi

```
max: integer {jumlah max rangkaian bola}
x: integer
y: integer
posisi: Point
```

Algoritma

```
for (x = 1 to 11)
  for (y = 1 to 12)
    posisi.x = x
    posisi.y = y
    if (max < getSameColor(p, posisi))
      max = getSameColor(p, posisi)
    endif
  endfor
endfor
```

```
procedure greedySolve (input p: board)
{ menerima masukan berupa data papan
permainan beserta seluruh isinya, kemudian
prosedur ini akan memilih semua bola yang
memiliki jumlah bola paling banyak dalam
rangkaiannya sampai seluruh bola yang
terdapat dalam papan permainan sudah tidak
dapat dipilih lagi }
```

Deklarasi

```
posisi: Point
```

Algoritma

```
while (! isFinish(p))
  posisi = mostBall(p)
  if (isValid(p, posisi))
    chooseBall(posisi)
  endif
endwhile
```

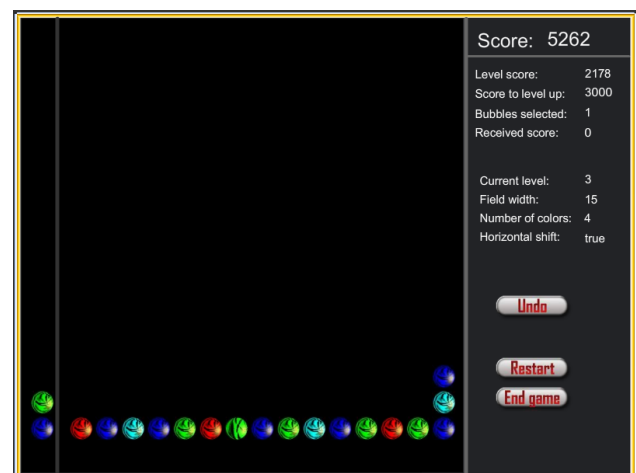
V. ANALISIS SOLUSI

Dalam algoritma greedy yang digunakan untuk penyelesaian masalah, terdapat dua fungsi utama, yaitu fungsi mostBall dan prosedur greedySolve.

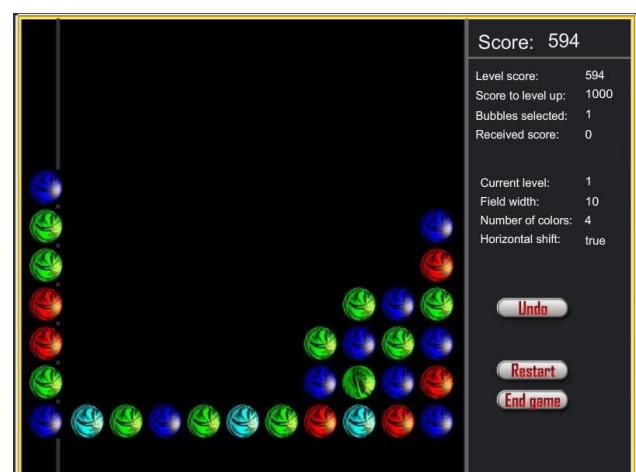
Fungsi mostBall digunakan untuk melakukan pencarian rangkaian bola dengan warna sama dengan jumlah paling banyak. Fungsi ini membantu fungsi greedySolve untuk memudahkan pemecahan masalah.

Fungsi greedySolve berguna untuk menyelesaikan permasalahan secara keseluruhan, yaitu memanggil fungsi mostBall untuk mendapatkan rangkaian bola dengan warna sama yang paling banyak, dan kemudian “memecahkan” rangkaian bola tersebut. Tujuan dari fungsi ini adalah tujuan algoritma greedy secara keseluruhan, yaitu mendapatkan poin yang optimal.

Salah satu tes uji untuk pengimplementasian greedy dalam penyelesaian masalah game Bubble Breaker adalah sebagai berikut :



Gambar 4 Tes Uji Algoritma Greedy



Gambar 5 Tes Uji Pemecahan Normal

Pada gambar 4 di atas dapat terlihat jelas perbedaan poin yang didapat jika dibandingkan dengan gambar 5. Pada gambar 4, dalam memainkan game digunakan penyelesaian dengan algoritma greedy. Sedangkan gambar 5 digunakan penyelesaian dengan cara memilih rangkaian bola pada posisi paling atas. Untuk kasus ini, algoritma greedy terbukti menghasilkan poin yang lebih optimal. Terlihat dari poin yang dihasilkan dengan algoritma greedy adalah 5262, sedangkan dengan cara biasa hanya didapatkan poin 594.

Hasil yang optimal di atas tetapi bukanlah hasil yang benar – benar optimal. Sebenarnya akan dihasilkan hasil yang lebih optimal jika pertama – tama membandingkan seluruh warna bola yang ada pada papan permainan. Pilih warna yang memiliki bola yang paling banyak. Jangan pecahkan bola dengan warna tersebut terlebih dahulu, melainkan terlebih dahulu usahakan untuk memecahkan bola – bola dengan warna lain agar bola – bola dengan warna yang memiliki bola terbanyak tersebut dapat saling menempel. Setelah sebanyak mungkin bola dengan warna tersebut berhasil ditempelkan, barulah pecahkan rangkaian tersebut sehingga kita mendapatkan poin yang besar. Setelah itu baru menghabiskan rangkaian bola – bola yang tersisa.

VI. KESIMPULAN

Algoritma *greedy* dapat digunakan untuk mencari solusi dalam permainan Bubble Brekaer. Solusi yang diberikan melalui algoritma tersebut dapat dikatakan cukup optimal dan dapat menghasilkan poin yang cukup besar. Akan tetapi, masih terdapat cara lain yang dapat digunakan untuk menyelesaikan permainan Bubble Breaker dengan poin yang lebih besar. Meskipun demikian, algoritma *greedy* sudah cukup baik untuk digunakan dalam game Bubble Breaker ini.

VII. REFERENSI

- [1] Munir, Rinaldi, “Diktat Kuliah IF2251 Strategi Algoritmik”, Program Studi Informatika Sekolah Teknik Elektro dan Informatika ITB, 2006.
- [2] <http://stackoverflow.com/questions/1541101/bubble-breaker-game-solver-better-than-greedy>
Tanggal akses : 8 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, 29 April 2010



Roy Indra Haryanto