

APLIKASI ALGORITMA GREEDY DALAM PERMAINAN JAWBREAKER

Albert (13506016)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl Ganesa 10, Bandung

e-mail: if16016@students.if.itb.ac.id
darkstalker_ag@yahoo.com

ABSTRAK

Pada masa sekarang ini, teknologi komputer telah berkembang dengan sangat pesat dan telah diterapkan dalam berbagai bidang. Salah satu bidang yang memanfaatkan teknologi komputer adalah industri *game* atau permainan. Pada saat ini, semakin banyak permainan komputer yang dirilis ke pasaran, dan permainan-permainan tersebut sangat diminati oleh masyarakat sebagai sarana untuk melepaskan kepenatan akibat pekerjaan atau kesibukan sehari-hari. Algoritma yang digunakan dalam permainan-permainan komputer ini pun berbeda-beda, berkisar dari algoritma yang sangat sederhana sampai dengan algoritma yang sangat rumit.

Pada kesempatan kali ini, penulis tertarik untuk membahas algoritma yang dapat digunakan dalam permainan "Jawbreaker" yang terdapat pada sistem operasi Windows Mobile 4 yang digunakan untuk Pocket PC atau PDA (*Personal Digital Assistant*). Sebenarnya tidak ada AI (*Artificial Intelligence*) yang digunakan dalam permainan ini. Hanya saja, penulis ingin mencoba menerapkan Algoritma Greedy untuk menyelesaikan permainan ini. Jawbreaker sendiri adalah sebuah permainan memecahkan bola-bola berwarna sama (dari 5 warna yang ada) yang saling menempel dalam suatu papan permainan untuk mendapatkan nilai semaksimal mungkin. Semakin banyak bola yang saling menempel saat dipecahkan, maka akan semakin besar pula nilainya. Sebenarnya terdapat 4 mode permainan pada permainan Jawbreaker ini, yaitu *Standard*, *Continuous*, *Shifter*, dan *MegaShift*. Akan tetapi, mode permainan yang akan dibahas dalam makalah ini hanyalah mode *Standard*. Sebenarnya mungkin terdapat berbagai algoritma yang dapat digunakan untuk mencari solusi permainan ini. Akan tetapi, dalam makalah ini akan dibahas mengenai algoritma Greedy saja.

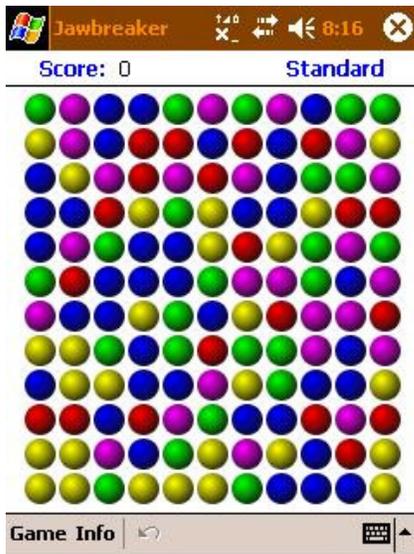
Kata kunci: Algoritma, Greedy, Aplikasi, Game, Jawbreaker.

1. PENDAHULUAN

Pada era teknologi informasi sekarang ini, teknologi komputer telah mengalami perkembangan yang sangat pesat dan menjadi kebutuhan bagi masyarakat di seluruh dunia. Hal tersebut dikarenakan banyak masalah yang sebelumnya dianggap rumit, kini dapat diselesaikan dengan lebih mudah menggunakan teknologi komputer. Salah satu aspek dalam masyarakat yang ikut terpengaruh oleh perkembangan teknologi komputer tersebut adalah dunia hiburan, dan dalam hal ini khususnya permainan komputer. Permainan komputer juga telah menjadi suatu kebutuhan bagi masyarakat sebagai sarana untuk melepas kepenatan akibat kesibukan. Salah satu permainan komputer yang cukup diminati adalah permainan Jawbreaker.

2. PERMAINAN JAWBREAKER

Permainan Jawbreaker adalah sebuah permainan standar yang terdapat dalam sistem operasi Windows Mobile 4 yang diaplikasikan dalam Pocket PC dan PDA (*Personal Digital Assistant*). Dalam permainan ini, pemain bertugas untuk memecahkan bola-bola berwarna sama (dari 5 warna yang ada) yang saling menempel satu sama lain dan membentuk suatu rangkaian dalam suatu papan permainan. Semakin banyak bola berwarna sama yang saling menempel, maka akan semakin besar pula angka atau nilai yang pemain dapatkan. Tujuan dari permainan ini adalah untuk mendapatkan nilai semaksimal mungkin untuk mengalahkan nilai tertinggi (*high score*) yang tercatat oleh program tersebut.



Gambar 1. Contoh Tampilan Game Jawbreaker

2.1 Jenis Permainan

Dalam permainan Jawbreaker yang terdapat dalam sistem operasi Windows Mobile 4, terdapat 4 jenis mode permainan yang berbeda, yaitu *Standard*, *Continuous*, *Shifter*, dan *MegaShift*. Dalam mode *Standard*, setelah bola dipecahkan, bola-bola yang berada di atas tempat yang sekarang kosong tersebut akan jatuh dan kemudian permainan dapat kembali dilanjutkan dengan memecahkan bola-bola lainnya. Mode *Continuous* pada prinsipnya sama dengan mode *Standard*. Hanya saja, jika satu kolom bola-bola telah berhasil dihilangkan, maka kolom-kolom tersebut akan dirapatkan dan pada kolom paling kiri akan ditambah dengan bola-bola baru sesuai dengan yang terlihat di bagian bawah layar permainan. Pada mode *Shifter*, setelah bola dipecahkan, apabila pada kolom sebelah kiri terdapat lebih banyak bola (lebih tinggi) dari kolom-kolom tempat bola baru saja dipecahkan, maka beberapa baris teratas dari kolom-kolom yang berada di sebelah kiri tersebut akan menggeser ke sebelah kanan. Sedangkan mode *MegaShift* adalah gabungan dari gaya permainan dalam mode *Shifter* dan *Continuous*. Akan tetapi, dalam makalah kali ini akan dibahas mengenai mode permainan *Standard* saja.

2.2 Penghitungan Poin

Setiap rangkaian bola-bola dengan warna sama yang dipecahkan dalam permainan Jawbreaker ini mempunyai nilai-nilai tertentu sesuai dengan jumlah bola sewarna yang menempel atau bersisian satu sama lain. Semakin banyak jumlah bola dalam rangkaian bola-bola sewarna tersebut, akan makin besar pula poin yang didapatkan oleh pemain.

Penulis telah mencoba untuk mencatat hasil-hasil poin yang akan didapatkan untuk setiap rangkaian bola-bola sewarna dengan jumlah tertentu. Hasilnya adalah sebagai berikut:

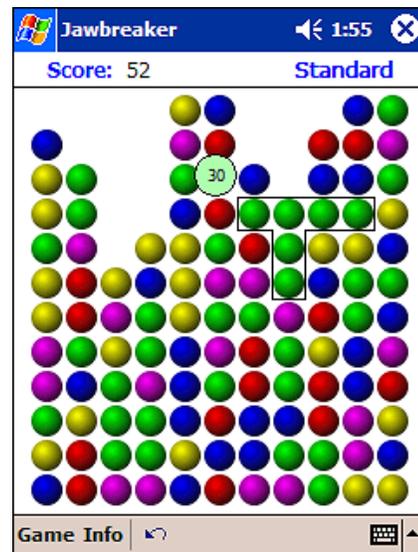
Tabel 1 Poin yang didapatkan untuk setiap rangkaian bola-bola sewarna

Jumlah bola	Poin didapat	Jumlah Bola	Poin didapat
2	2	9	72
3	6	10	90
4	12	11	110
5	20	12	132
6	30	13	156
7	42	14	182
8	56	15	210

Dengan melihat hasil pengamatan pada tabel di atas, dapat disimpulkan memang benar bahwa semakin banyak jumlah bola sewarna yang dirangkai, maka poin yang didapatkan juga akan semakin besar. Penulis kemudian mencoba mencari rumus atau formula yang digunakan untuk menentukan jumlah poin yang didapatkan tersebut. Penulis mencoba membagi jumlah poin yang didapatkan dengan jumlah bola sewarna yang terangkai. Ternyata penulis menemukan bahwa formula yang digunakan untuk menentukan jumlah poin didapat adalah sebagai berikut:

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

Dengan n adalah jumlah bola dalam rangkaian bola-bola dengan warna yang sama yang akan dipecahkan.



Gambar 2. Pemain akan memecahkan rangkaian 6 bola sewarna yang bernilai 30 poin

Rangkaian bola-bola sewarna dengan jumlah minimal yang dapat dipecahkan adalah 2 bola sewarna. Sedangkan jumlah bola sewarna pada rangkaian bola sewarna yang maksimum dapat dipecahkan adalah sebanyak semua bola yang dapat masuk dalam area papan permainan. Hanya saja tidak mungkin menjadikan semua bola dalam area permainan menjadi bola dengan warna yang sama.

3. DASAR TEORI

3.1 Definisi Algoritma Greedy

Algoritma *greedy* merupakan sebuah algoritma yang memecahkan suatu masalah secara langkah per langkah. Pendekatan yang digunakan di dalam algoritma *greedy* adalah membuat pilihan yang “tampaknya” memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal (pilihan yang terbaik untuk saat itu) pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global. Prinsip yang terdapat dalam algoritma *greedy* adalah prinsip “*take what you can get now!*”, karena algoritma tersebut mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depannya.

3.2 Skema Umum Algoritma Greedy

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 himpunan ini.
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 obyektif ini merupakan sebuah fungsi yang memaksimumkan atau meminimumkan nilai solusi.

4. METODE PENYELESAIAN

4.1 Analisis Masalah

Dalam pencarian solusi untuk penyelesaian permainan Jawbreaker ini, kita ingin mendapatkan nilai sebesar mungkin yang bisa didapatkan. Nilai yang besar baru bisa didapatkan apabila kita memilih untuk memecahkan rangkaian bola-bola sewarna 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 bola-bola 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.

4.2 Algoritma Penyelesaian Masalah

Dari analisis masalah seperti yang sudah disebutkan di atas, penulis mencoba menerapkan algoritma *greedy* sebagai penyelesaian untuk permasalahan tersebut, sebab masalah ini merupakan masalah optimisasi untuk mendapatkan hasil yang maksimal dari permainan Jawbreaker tersebut.

Langkah-langkah yang akan diambil dalam pencarian solusi ini adalah dengan pertama-tama melihat seluruh bola yang ada dalam papan atau area permainan secara baris per baris dan program akan menyimpan bola yang memiliki jumlah bola-bola sewarna paling banyak dalam rangkaiannya. Kemudian program akan memilih bola tersebut sehingga seluruh bola dalam rangkaiannya akan pecah dan kita mendapatkan poin. Kemudian program akan kembali memilih bola yang memiliki jumlah bola-bola sewarna paling banyak dalam rangkaiannya. Begitu seterusnya sampai tidak ada lagi bola-bola sewarna dengan jumlah bola pada rangkaian yang lebih dari 1.

Pada masalah pencarian solusi ini, himpunan kandidat adalah semua bola yang terdapat dalam papan permainan. Sedangkan yang menjadi himpunan solusi dalam masalah ini adalah bola-bola yang mempunyai jumlah bola terbesar dalam rangkaian.

Fungsi seleksi dalam kasus ini adalah fungsi yang akan memilih bola-bola yang mempunyai jumlah bola terbesar dalam rangkaian. Sedangkan fungsi kelayakannya adalah fungsi yang akan memastikan bahwa bola yang dipilih memiliki rangkaian yang mengandung lebih dari satu bola dengan warna yang sama. Fungsi objektif bertugas untuk memastikan bahwa semua bola yang dapat dipilih sudah dipilih, atau dengan kata lain, sudah tidak ada lagi bola dalam papan permainan yang dapat dipilih.

Dari semua analisa di atas, akhirnya didapatkan algoritma penyelesaian masalah sebagai berikut:

```
function getRangkaian (input p: papan_permainan,
posisi: Point) → integer
{ menerima masukan berupa data papan permainan
beserta seluruh isinya serta posisi bola yang
ditunjuk, kemudian mengembalikan jumlah bola
dalam rangkaian bola-bola berwarna 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 pilihBola (input p: papan_permainan,
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: papan_permainan) →
boolean
{ menerima masukan berupa data papan permainan
beserta seluruh isinya, kemudian mengembalokan
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: papan_permainan,
posisi: Point) → boolean
{ menerima masukan berupa data papan permainan
beserta seluruh isinya serta posisi bola yang
ditunjuk, kemudian mengembalokan true jika
rangkaiannya 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. }
```

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

Deklarasi

```
max: integer
ix: integer
iy: integer
posisi: Point
```

Algoritma

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

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

Deklarasi

```
posisiBola: Point
```

Algoritma

```
while (! isFinish(p))
  posisiBola = rangkaianTerbesar(p);
  if (isValid(p, posisiBola))
    pilihBola(posisiBola)
  endif
endwhile
```

4.3 Analisis Algoritma

Dengan menggunakan algoritma seperti yang dituliskan dalam *pseudo-code* di atas, kita akan dapat menyelesaikan permainan Jawbreaker dengan prinsip algoritma *greedy*. Namun apabila kita amati kembali penyelesaian dengan algoritma *greedy* tersebut dan melakukan percobaan untuk menyelesaikan permainan secara langsung, kita dapat melihat bahwa algoritma *greedy* ini memang cukup efektif untuk mendapatkan poin yang besar. Akan tetapi, terdapat cara lain yang dapat digunakan untuk mendapatkan poin yang lebih besar lagi.

Cara untuk lebih optimal mendapatkan poin yang besar adalah dengan pertama-tama membandingkan seluruh warna bola yang ada pada papan permainan. Pilih warna yang memiliki bola paling banyak. Jangan pecahkan bola dengan warna tersebut terlebih dahulu, melainkan terlebih dahulu usahakan untuk memecahkan bola-bola dengan warna lainnya agar bola-bola dengan warna yang memiliki bola terbanyak tersebut saling menempel bersisian. Setelah sebanyak mungkin bola dengan warna tersebut berhasil ditempelkan, barulah pecahkan rangkaian tersebut sehingga kita mendapatkan poin yang besar. Setelah itu barulah menghabiskan rangkaian bola-bola yang tersisa.

Cara yang kedua tersebut memang seringkali dapat menghasilkan solusi dan poin yang lebih maksimal. Akan tetapi, tidak ada jaminan pula bahwa cara tersebut akan selalu berhasil dengan lebih baik.

V. KESIMPULAN

Algoritma *greedy* dapat digunakan untuk mencari solusi dalam permainan Jawbreaker. 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 Jawbreaker dengan poin yang lebih besar. Meskipun demikian, algoritma *greedy* sudah cukup baik untuk digunakan dalam game semacam Jawbreaker ini.

REFERENSI

- [1] Munir, Rinaldi, “Diktat Kuliah IF2251 Strategi Algoritmik”, Program Studi Informatika Sekolah Teknik Elektro dan Informatika ITB, 2006.
- [2] Jawbreaker Pocket PC Guide Standard Mode
<http://www.pdagameguide.com/jawbreaker-pocket-pc.html>
tanggal akses: 15 Mei 2008