

# Penyelesaian *Tile-Matching Game* Menggunakan *Greedy*

Fakhri

13510048

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13510048@std.stei.itb.ac.id

## ABSTRAK

*Tile Matching game* merupakan genre permainan yang sudah marak di zaman sekarang. Pola permainanannya adalah memanipulasi objek permainan sehingga objek tersebut hilang dengan mengikuti aturan – aturan tertentu. Permainan dengan genre ini biasanya merupakan permainan dengan satu pemain saja, tanpa lawan baik itu AI maupun player lain. Tantangan yang ada adalah skor tertinggi dan kesulitan dalam pemecahan tantangan setiap level permainan. Normalnya objek permainan ini terdiri atas ‘tiles’ yang bervariasi. Kabanyakan permainan dengan genre ini mengharuskan pemain untuk menemukan tiga tile yang berjenis sama sehingga genre ini sering disebut dengan nama ‘match-three games’. Namun nama ‘match-three games’ sebenarnya sudah kurang tepat sekarang karena perkembangan genre permainan ini yang kian lebih kompleks. Dalam tulisan ini akan dibahas penyelesaian dari genre permainan ini melalui contoh game terkenal di saat sekarang seperti PopCap Games: COLLAPSE!, GameHouse: BEJEWELED, dan MumboJumbo : LUXOR dengan level penyelesaian standar. Metode penyelesaian yang digunakan adalah greedy untuk sistem penilaian, dan Decrease & Conquer yaitu DFS/BFS untuk penentuan nilai. Dua algoritma ini dipilih karena dua algoritma ini tergolong sederhana dan lempeng (straightforward) sehingga mudah untuk diimplementasikan.

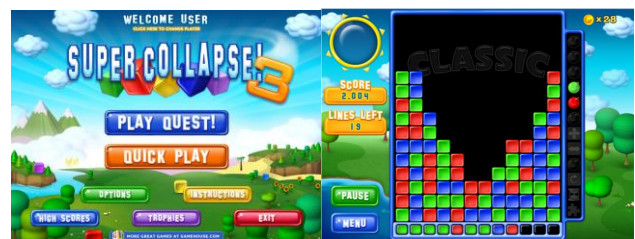
**Kata Kunci :** Tile Matching, PopCap Games: COLLAPSE!, GameHouse: BEJEWELED, MumboJumbo : LUXOR, Greedy, Decrease & Conquer, DFS/BFS.

## I. PENDAHULUAN

Tile Matching Game pada dasarnya menuntut pemain untuk fokus dalam mengidentifikasi pola – pola objek permainan sehingga menemukan aturan yang membuat objek tersebut dapat dieliminasi, memberikan point, dan mungkin dilakukan sebelum waktu atau kondisi selesai terpenuhi. Tantangan yang diberikan bisa dibedakan melalui variasi pola yang ada, jumlah tile/objek permainan, eaktu yang diberikan, jumlah langkah dan lain

– lain.

Permainan ini pada dasarnya sudah muncul sejak komputer belum ada seperti *mahjong solitaire* maupun *card solitaire*. Setelah komputer muncul maka mulailah muncul permainan digital/video game dengan aturan dasar yang sama, yaitu penyamaan pola tile. Permainan digital generasi pertama dengan genre ini adalah Alexey Pajitnov : TETRIS dan Kuniaki Moribe : CHAIN SHOT! (SAMEGAME) yang dirilis pada tahun 1980an. Kedua permainan ini merupakan landasan pertama untuk mekanisme permainan dengan *pattern matching* pertama. Walaupun muncul dengan genre yang sama, kedua permainan ini memiliki perbedaan pada nilai penarik, sistem penilaian, aturan pemanipulasian, maupun pembatasan waktu sehingga dapat menimbulkan perbedaan yang terasa diantaranya. Generasi kedua yang terkenal, memodifikasi genre yang telah ada, dan menghasilkan mekanisme baru adalah Gunpei Yokoi : DR.MARIO dan Taito : PLOTTING yang melahirkan *tile-matching* dengan mekanisme tembak/shooting lalu Famicom: MAGIC JEWELRY yang melahirkan *tile-matching* dengan mekanisme tukar/geser. Mekanisme – mekanisme tersebut digunakan dan dikembangkan hingga saat ini, contoh perkembangan untuk mekanisme *pattern matching* murni adalah COLLAPSE!, mekanisme *swapping* adalah BEJEWELED yang terinspirasi dari permainan Rusia, Shariki, dan Robot Communications : ZOO KEEPER, lalu mekanisme *shooting* adalah LUXOR dan PopCap Games : ZUMA.



Gambar 1 : Antarmuka SUPER COLLAPSE! 3

Sebelum tahun 2000, game seperti ini merupakan permainan yang bisa dikatakan berkelas tinggi dan bernilai ekonomis tinggi.



Gambar 2 : Antarmuka BEJEWELED 2

Namun di era 2000an ke atas permainan dengan genre ini sudah bisa dikategorikan ke dalam permainan kasual dan sehari – hari karena bersifat gratis untuk dimainkan di internet maupun untuk diunduh sekalipun, sebagai contoh adalah tiga game yang akan kita jadikan bahan pembahasan : seri PopCap Games : SUPER COLLAPSE! 3, seri GameHouse : BEJEWELED 2 DELUXE, dan MumboJumbo : LUXOR 2.



Gambar 3 : Antarmuka LUXOR 2

Pada perkembangannya, unsur Tile matching Game dicampurkan dengan unsur lain seperti arcade, turn based, real time strategy, maupun Role playing game. Akan tetapi, pencampuran ini pada dasarnya tidak mengubah inti permainan melainkan hanya pembungkus berupa tambahan alur cerita maupun *side-game* yang ada di dalam permainan agar menimbulkan karakteristik berbeda antar game dengan genre yang sama.

## II. ATURAN PERMAINAN

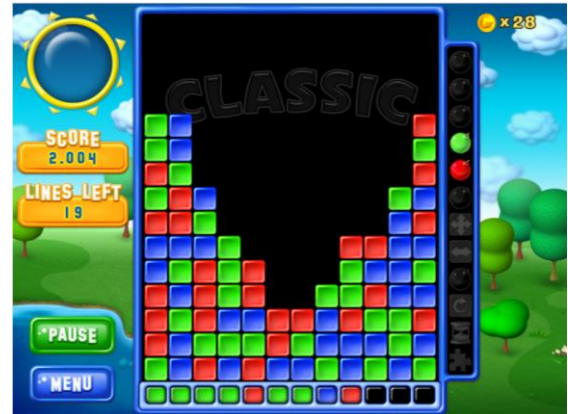
Permainan *tile-matching* memiliki konsep inti permainan yang sama, hanya saja terdapat perbedaan khusus antara tiap permainan berbeda yang menimbulkan ciri khas masing – masing. Namun pada mekanisme yang berbeda, perbedaan yang terasa membuat seolah game dengan genre yang sama memiliki genre berbeda. Hal ini akan terasa pada 3 mekanisme yang akan kita bahas berikut :

### A. Mekanisme Classic Tile-Matching

Mekanisme ini merupakan mekanisme yang paling standar dalam permainan *tile matching* dan telah digunakan sejak pertama kali permainan digital *tile-matching* ada, yaitu memilih beberapa *tile* dengan pola objek yang sama, seperti warna, gambar, dan lain-lain sebanyak 3 agar

menghilangkannya dari permainan dan pemain memperoleh nilai dari hal tersebut.

Untuk mempermudah penjelasan, kita akan menggunakan satu contoh permainan yaitu GameHouse : SUPER COLLAPSE! 3 dengan antar muka sebagai berikut



Gambar 4 : Layar permainan SUPER COLLAPSE! 3

Dari gambar di atas terlihat ada beberapa warna *tile*. User memilih salah satu *tile*, jika yang dipilih memiliki minimal tiga dempetan *tile* dengan warna yang sama di atas, kiri, kanan, atau, bawah, maka semua yang berwarna sama dan berdempetan itu akan hancur. Setelah ada yang hancur, maka *tile* yang berada di atasnya akan jatuh ke tempat yang kosong. Nilai yang diperoleh berdasarkan dengan jumlah *tile* yang berhasil dihancurkan ditambah dengan nilai – nilai kombo. Batasan permainan adalah jumlah *line* yang akan muncul dari bawah yang jika habis akan membuat pemain menang. Petunjuk sisa *line* terdapat di bawah petunjuk *score*.

### B. Mekanisme Swapping

Mekanisme ini merupakan mekanisme yang muncul pada generasi ke dua dari *tile-matching game*. Sistem permainannya mirip dengan sistem pada mekanisme *Classic Tile-Matching*, hanya saja pemain tidak memilih satu *tile*, melainkan memilih dua. Dua *tile* yang dipilih akan saling bertukar tempat dan hasil pertukarantempat akan menjadi penentuan. Jika menimbulkan adanya tiga pola *tile* yang sama akibat dua *tile* yang ditukar itu, maka *tile* berpola sama

Untuk mempermudah penjelasan, kita akan menggunakan satu contoh permainan yaitu PopCap Games : BEJEWELED 2 DELUXE dengan antar muka sebagai berikut :



**Gambar 5 : Layar Permainan BEJEWELED 2**

dari gambar di atas, kita dapat melihat ada banyak variasi *tile* yang digambarkan melalui batu permata. Terlihat pula ada dua permata yang sedang dalam proses *swapping*, yaitu permata biru yang awalnya di kiri dan permata hijau yang awalnya di kanan. Ketika dua permata tersebut selesai berpindah tempat, permata hijau, dua permata hijau berurutan di kiri akan hilang dan diubah menjadi poin berdasarkan jumlah *tile* yang hancur dan kombo. *Tile* di kiri bawah tidak ikut hancur karena yang hancur hanya jika terdapat dalam satu baris atau kolom. *Tile* yang muncul tidak mungkin berkumpul dalam 3 deret sekaligus karena sudah pasti hancur. Tiap *tile* yang hancur akan digantikan dengan *tile* yang berada di atasnya, di jika tidak ada maka akan muncul *tile* baru dari layar kosong di atasnya. Batasan permainan ini adalah penuhnya *bar* yang ada di bagian bawah layar. *Bar* ini akan terisi layaknya *loading* tiap kali ada *tile* yang hancur dan makin cepat jika jumlah *tile* yang hancur ada banyak. Batasan untuk kalah adalah jika tidak ada kemungkinan satupun untuk menghancurkan *tile* dari semua kemungkinan *swapping*. Di bagian kiri layar terdapat rombol *hint* yang membantu pemain dalam menemukan solusi yang dibayar dengan pengurangan isi *bar*.

### C. Mekanisme Shooting

Mekanisme ini sama – sama muncul pada permainan generasi kedua dari genre *tile-matching*. Sistem permainannya mirip dengan sistem permainan pada mekanisme *Classic Tile-Matching*, hanya saja pemain tidak memilih *tile* pada tempatnya melainkan menembakkan suatu *tile* ke lokasi tertentu. Jika *tile* yang ditembak berhenti di suatu titik, maka semua *tile* yang terhubung langsung maupun tidak langsung akibat *tile* lain akan hancur. Pemain harus mengarahkan *tile* tersebut dengan mekanisme yang berbeda di tiap permainan. Bahkan kebanyakan permainan

dengan mekanisme ini juga menggerakkan target *tile* yang akan ditembakkan pada jalur tertentu.

Untuk mempermudah penjelasan, kita akan menggunakan satu contoh permainan yaitu MumboJumbo : LUXOR 2 dengan antar muka sebagai berikut :



**Gambar 6 : Layar Permainan LUXOR 2**

Pada gambar di atas dapat dilihat suatu jalur yang dilalui banyak bola dengan warna dan motif tertentu. Pada permainan tersebut, pemain dapat mengarahkan sekap dengan bola bermotif yang sama dengan bola lain ke arah kanan maupun kiri, namun tetap disumbu horisontal yang sama. Tiap klik yang dilakukan pemain akan membuat bola pada sayap ditembakkan secara vertikal ke bagian atas layar, jika bola yang ditembakkan menabrak bola pada jalur, maka bola tersebut akan menyelip di antara bola di jalur lalu bergerak mengikuti jalur yang ada. Bola baru pada sayap akan muncul tiap usai penembakan dengan warna yang acak. Jika bagian yang diselipi bola yang ditembakkan menimbulkan terjadinya deretan pola dengan warna yang sama dengan bola yang ditembakkan, maka seluruh bola akan hancur dan pemain akan mendapatkan poin sesuai dengan bola yang hancur dan kombo. Batasan permainan ini adalah habis dan musnahnya bola yang disediakan di tiap level yang membuat pemain menang atau adanya bola di jalur yang masuk ke akhir jalur yang menyebabkan pemain kalah.

## III. STRATEGI PERMAINAN

Pada dasarnya tiga jenis mekanisme tersebut punya target pencapaian yang sama yaitu memperoleh nilai yang paling tinggi dari penemuan pola yang sama. Dari pernyataan kalimat di atas, metode paling *straightforward* yang dapat diterapkan adalah Algoritma Greedy.

### A. Mekanisme Classic Tile-Matching

Hal yang paling pertama dilakukan adalah menentukan seluruh kemungkinan nilai untuk tiap



*tile* yang mungkin dipilih. Untuk menentukan hal ini digunakan *decrease & conquer* bisa itu BFS ataupun DFS. Asumsikan penyelesaian menggunakan DFS, maka algoritmanya adalah :

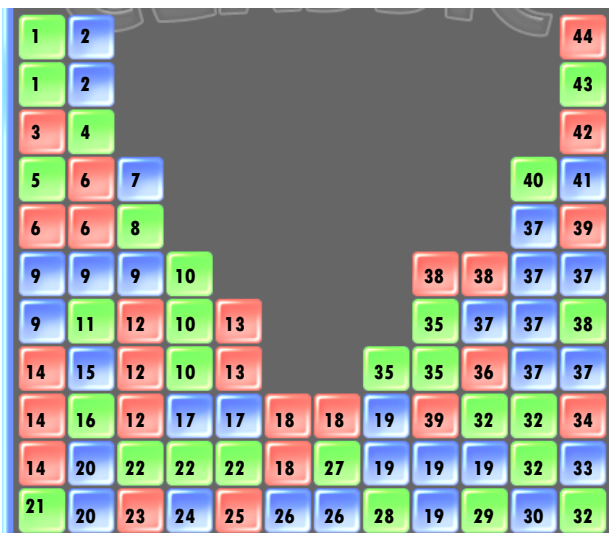
```

procedure trace (x, y, input/output counter)
begin // DFS
  if tile[x,y].nilai=false /*cek sudah
    pernah diakses atau belum*/
    tile[x,y].nilai=true
    if tile[x-1,y] = tile[x,y]
      trace(x-1,y, counter+1)
    if tile[x+1,y] = tile[x,y]
      trace(x+1,y, counter+1)
    if tile[x,y-1] = tile[x,y]
      trace(x,y-1, counter+1)
    if tile[x,y+1] = tile[x,y]
      trace(x,y+1, counter+1)
end

begin //main program
  k=0
  for i = 0 to sizeof(row)
    for j = 0 to sizeof(column)
      n = 0
      trace(0, j, n)
      k=k+1
      choosen[k]=n, i, j //suatu record
    descending_sort_by_n(choosen) // Sorting
    choose(choosen[1]) //GREEDY
  end.

```

Sehingga jika hingga dipetakan per zona :



**Gambar 7 : Layar permainan  
SUPER COLLAPSE! 3**

Zona di atas bukan urutan perulangan seperti pada algoritma melainkan pembeda semata. Lalu untuk *choosen* pada algoritma akan terisi hal berikut

| n   | x   | y   |
|-----|-----|-----|
| 7   | 4   | 10  |
| 5   | 8   | 7   |
| ... | ... | ... |
| 0   | 10  | 11  |

Dari tabel di atas maka keputusan yang akan diambil secara greedy adalah *tile* pada posisi 5,11 yaitu pada zona 37. Dengan memilih hal ini maka pemain akan memperoleh nilai maksimal untuk kondisi saat itu.

## B. Mekanisme Swapping

Hal yang paling pertama dilakukan adalah menentukan seluruh kemungkinan nilai untuk tiap pertukaran *tile* yang mungkin dipilih. Untuk menentukan hal ini bisa menggunakan *decrease & conquer* seperti Mekanisme *Classic Tile-Matching*, cukup dengan iterasi biasa sudah cukup. Algoritmanya jika memakai DFS adalah :

```

procedure traceV(x, y, input/output counter)
begin // DFS
  if tile[x-1,y] = tile[x,y]
    trace(x-1,y, counter+1)
  if tile[x+1,y] = tile[x,y]
    trace(x+1,y, counter+1)
end

procedure traceH(x, y, input/output counter)
begin // DFS
  if tile[x,y-1] = tile[x,y]
    trace(x,y-1, counter+1)
  if tile[x,y+1] = tile[x,y]
    trace(x,y+1, counter+1)
end

begin //main program
  k=0
  for i = 0 to sizeof(row)
    for j = 0 to sizeof(column)
      if n =3 then break
      swap(tile[i, j], tile[i+1, j])
      n = 0
      traceV(0, j, n)
      k=k+1
      choosen[k]=n, i, j, i+1, j //record
      swap(tile[i, j], tile[i+1, j])

      if n =3 then break
      swap(tile[i, j], tile[i-1, j])
      n = 0
      traceV(0, j, n)
      k=k+1
      choosen[k]=n, i, j, i+1, j //record
      swap(tile[i, j], tile[i-1, j])
      if n =3 then break
      swap(tile[i, j], tile[i, j+1])

```

```

n = 0
traceH(0, j, n)
k=k+1
choosen[k]=n, i, j, i+1, j //record
swap(tile[i, j], tile[i, j+1])

if n =3 then break
swap(tile[i, j], tile[i, j-1])
n = 0
traceH(0, j, n)
k=k+1
choosen[k]=n, i, j, i+1, j //record
swap(tile[i, j], tile[i, j-1])

descending_sort_by_n(choosen) // Sorting
choose(choosen[1]) //GREEDY
end.

```

Hasil dari algoritma di atas sesuai gambar 5 melalui tabel choosen akan seperti berikut:

| n   | x   | y   | Next x | Next y |
|-----|-----|-----|--------|--------|
| 3   | 1   | 1   | 2      | 1      |
| ... | ... | ... | ...    | ...    |
| 0   | 7   | 7   | 7      | 6      |

Pada kode terdapat break karena greedy di kasus ini adalah greedy terhadap waktu. Total point tidak mungkin bisa lebih besar dari 3 point untuk *pure greedy*, yaitu hanya memperhatikan kondisi nilai maksimum untuk kondisi sekarang.

### C. Mekanisme Shooting

Hal yang pertama dilakukan adalah menentukan lokasi target dengan minimal ada dua *tile* berwarna sama, sorting, lalu memilih yang terbesar. Jika dialgoritamkan :

```

procedure trace (x,input/ouput counter)
begin // DFS
  if tile[x].nilai=false /*cek sudah
    pernah diakses atau belum*/
    tile[x].nilai=true
    if tile[x-1] = tile[x,y]
      trace(x-1,counter+1)
    if tile[x+1,y] = tile[x,y]
      trace(x+1,counter+1)
end

begin //main program
k=0
for i = 0 to sizeof(row)
  n = 0
  if tile[i]==maintile then
    trace(0, j, n)
    k=k+1
    choosen[k]=n, i //suatu record
  descending_sort_by_n(choosen) // Sorting
  choose(choosen[1]) //GREEDY

```

end.

Hasil dari algoritma di atas adalah tabel choosen yang mengurutkan nilai kemungkinan terbesar dan menembak ke arah tersebut.

## V. KESIMPULAN

Algoritma Greedy dapat diterapkan untuk hasil yang bagus dalam permainan Tile-Matching khususnya yang bermekanisme Classic Tile-Matching dan Shooting. Untuk mekanisme swapping, greedy tidak terlalu efisien akibat jumlah maksimal yang diperoleh adalah tiga diluar kombo dan kondisi setelahnya.

## REFERENCES

- [1] [http://en.wikipedia.org/wiki/Match\\_three\\_game](http://en.wikipedia.org/wiki/Match_three_game)  
waktu akses : 21 Desember 2012 pukul 12.03
- [2] [http://en.wikipedia.org/wiki/Luxor\\_\(video\\_game\)](http://en.wikipedia.org/wiki/Luxor_(video_game))  
waktu akses : 21 Desember 2012 pukul 12.16
- [3] Levitin, Anany . Introduction to The Design & analysis of Algorithms : Addison Wiley

## 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, 21 Desember 2012



Fakhri  
13510048