

ANALISIS PENERAPAN ALGORITMA *BACKTRACKING* DALAM PENCARIAN SOLUSI GAME “MUMMY MAZE DELUXE”

Faradina Ardiyana¹, Kania Adityarani²

Laboratorium Ilmu dan Rekayasa Komputasi
Sekolah Teknik Elektro Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if14067@students.if.itb.ac.id¹, if14089@students.if.itb.ac.id²,

Abstrak

Algoritma Runut-balik (*backtracking*) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus. *Backtracking*, yang merupakan perbaikan dari algoritma *brute-force*, secara sistematis mencari solusi persoalan di antara semua kemungkina solusi yang ada. Hanya saja algoritma ini merupakan pencarian yang mengarah ke solusi yang dipertimbangkan saja.

Algoritma runut-balik (*backtracking*) banyak digunakan pada program permainan(*games*) dan masalah pada bidang kecerdasan buatan. Salah satu permainannya adalah mencari jalan keluar di dalam labirin (*maze problem*). Dan salah satu persoalan yang kami bahas disini adalah pencarian solusi dalam permainan “mummy maze deluxe” dengan menerapkan algoritma runut-balik(*backtracking*). Disini kami menganalisis bagaimana cara terbaik untuk keluar dari labirin disertai dengan penghindaran terhadap *mummy* yang mengejar. Jika tidak ketemu jalan keluar atau buntu atau jika posisi *player* dengan *mummy* sama, maka akan dilakukan *backtracking*.

Kata kunci: *backtracking*, *game*, *mummy maze deluxe*

1. Pendahuluan

Melakukan permainan atau bermain *game* merupakan hal yang menarik bagi hampir seluruh masyarakat di dunia ini. Sebagian masyarakat tersebut bermain *game* hanya untuk melepas lelah, dan ada juga yang ingin bersenang-senang, dan ada juga yang hanya mengisi waktu kosong, serta ada yang menganggapnya sebagai lahan untuk mengolah otak. Apalagi dengan berkembangnya teknologi di masyarakat ini, semakin berkembang juga jenis permainan yang ada pada saat ini. Dan permainan yang sering dilakukan masyarakat pada jaman ini adalah permainan yang dilakukan di komputer.

Salah satu permainan tersebut adalah permainan yang menggunakan labirin dan kita sebagai *player* harus menemukan jalan keluarnya. Dan penulis akan membahas permainan “*mummy maze deluxe*” dalam makalah ini.

Game “Mummy Maze Deluxe” pada dasarnya merupakan aplikasi *game* dari *maze problem*, disini kita sebagai pemain mencoba untuk mencari jalan keluar didalam labirin, tetapi pada permainan ini, kita sebagai pemain akan dikejar oleh *mummy*. Analisis yang dilakukan disini adalah mencari jalan keluar terbaik dan jangan sampai tertangkap oleh *mummy*. Dan dalam hal ini, algoritma yang digunakan adalah algoritma *backtracking*,

algoritma tersebut mengatasi jika jalan keluar belum ditemukan.

2. Konsep dasar

2.1 Deskripsi Umum *Game Mummy Maze Deluxe*

Mummy Maze Deluxe merupakan salah satu *game* cerdas dari *Popcap*, suatu permainan teka-teki (*puzzle*). Pemain harus melewati ruangan-ruangan di dalam suatu piramida, masing-masing merepresentasikan satu *puzzle* yang harus ditemukan solusinya(jalan keluarnya). Tiap ruangan berupa ubin-ubin persegi dengan rintangan dan objek lainnya(*mummy*). Pemain mulai di satu ubin, dan *mummy* di ubin lainnya. Untuk setiap pergerakan satu ubin dari pemain, maka *mummy* bisa bergerak dua ubin. Hal ini dapat pemain siasati dengan membuat *mummy* terperangkap di rintangan-rintangan yang ada sehingga pemain bebas bergerak sampai pintu keluar (*exit*).

Walaupun dalam *game* ini sudah tersedia solusi untuk setiap *puzzle*, penulis menganalisis bagaimana mencari solusi terbaik yang dapat diperoleh.

2.2 Algoritma *Backtracking*

Istilah runut-balik pertama kali diperkenalkan oleh D.H.Lehmer pada tahun 1950. Selanjutnya, R.J. Walker, Golomb, dan Baumert menyajikan uraian

umum tentang runut balik dan penerapannya pada berbagai persoalan. Saat ini algoritma runut-balik (*backtracking*) banyak diterapkan untuk program *games* seperti permainan *tic-tac-toe*, menemukan jalan keluar dalam sebuah labirin, catur, dan lain-lain. Serta masalah pada bidang kecerdasan buatan (*artificial intelligence*).

Algoritma runut-balik (*backtracking*) merupakan perbaikan dari algoritma *brute-force*. Algoritma ini mencari solusi dari sekian banyak kemungkinan solusi yang ada. Semua solusi dibuat dibentuk dalam pohon solusi (pohon ini berbentuk abstrak) dan algoritma akan menelusuri secara DFS (Depth Field Search) sampai ditemukan solusi yang layak.

Properti umum Metode Runut-balik :

1. Solusi persoalan

Solusi dinyatakan sebagai vector dengan n-tuple :

$$X = (x_1, x_2, \dots, x_n), x_i \in \text{himpunan berhingga } S_i$$

Mungkin saja $S_1=S_2=\dots=S_n$

Contoh: $S_i = \{0,1\}$,
 $X_i = 0$ atau 1

2. Fungsi Pembangkit nilai x_k

Dinyatakan sebagai:

$$T(k)$$

$T(k)$ membangkitkan nilai untuk x_k , yang merupakan komponen vector solusi.

3. Fungsi Pembatas

(pada beberapa persoalan fungsi ini dinamakan fungsi criteria)

Dinyatakan sebagai:

$$B(x_1, x_2, \dots, x_k)$$

Fungsi pembatas menentukan apakah (x_1, x_2, \dots, x_k) mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk x_{k+1} dilanjutkan, jika tidak, maka (x_1, x_2, \dots, x_k) dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

Fungsi pembatas tidak selalu dinyatakan sebagai fungsi matematis. Ia dapat dinyatakan sebagai predikat atau dalam bentuk lain yang ekuivalen.

**3. Pencarian Solusi Game :
 “Mummy Maze Deluxe”**

Penggunaan algoritma *backtracking* ini akan terlihat pada pencarian solusi terbaik, jika posisi pemain dengan *mummy* sama, jika pemain menghadapi jalan buntu atau menabrak dinding, jika posisi pemain hampir mendekati *mummy*, jika pemain

dengan *mummy* berada pada posisi absis atau ordinat yang sama dan dibatasi oleh tembok. Jika posisi antara pemain dengan *mummy* hampir sama ataupun sama maka *backtracking* hingga ke posisi awal. Pada *mummy* mengalami *backtracking* jika posisi absis atau ordinatnya sama dengan pemain, karena *mummy* selalu berjalan mengikuti dan mendekati pemain.

Algoritma Runut-balik (*backtracking*) yang dilakukan secara rekursi dengan parameter fungsi adalah labirin M yang menyimpan status labirin (dapat diimplementasikan sebagai matriks 0/1, rangkaian 0 untuk jalan, rangkaian 1 untuk tembok), dan terdapat sederetan langkah yang mungkin terjadi yaitu ke etes (up), ke bawah (down), ke kiri (left), ke kanan (right). Algoritma Runut-balik (*backtracking*) untuk persoalan mencari jalan keluar yang terbaik dalam permainan “Mummy Maze Deluxe” :

```
Function SolverMummyMaze(input M :
labirin)->Boolean
{true jika solusi ditemukan, false jika
tidak}
```

Deklarasi

```
arah : integer {up = 1, down =
2, left = 3, right = 4}
```

Algoritma

```
if solusi sudah ditemukan then
return true
else
for tiap arah gerakan (up,
down, left, right) do
movePlayer(M, arah)
{pindah satu langkah sesuai arah}
moveMummy(M, arah)
{pindah dua langkah sesuai arah}
if
SolveMummyMaze(M) then
return true
else if (posisi
sama) then
unmovePlayer(M, arah)
{backtracking}
unmoveMummy(M, arah)
{backtracking}
else if (player
nabrak atau buntu) then
unmovePlayer(M, arah)
{backtracking}
if (mummy
tidak nabrak atau buntu) then
unmovePlayer(M,
arah){backtracking}
moveMummy(M, arah)
else if (mummy
nabrak atau buntu) then
movePlayer(M, arah)
```

```

                                mummy
dapat bergerak mendekati player
                                jika
tidak terhalangi tembok
                                endif
                                endfor
                                endif

```

Algoritma runut-balik diatas merupakan pencarian jalan terbaik menuju jalan keluar labirin dalam permainan *mummy maze deluxe*. Pada algoritma ini akan selalu menghasilkan solusi. Jika solusi belum ditemukan maka akan dilakukan *backtracking* terus-menerus hingga solusi didapat. Untuk menghasilkan lintasan yang dilalui, perpindahan langkah akan disimpan didalam *stack* dan seluruh langkah akan dicetak setelah pemanggilan `SolverMummyMaze`.

Algoritma runut balik pada persoalan ini dipandang sebagai pembentukan pohon ruang status. Akar dari pohon adalah labirin awal, dan anak-anaknya adalah labirin yang dihasilkan dari pergerakan satu langkah dari labirin semula. Simpul-simpul pada pohon dibentuk dengan skema traversal DFS. Simpul daun merepresentasikan apakah status pergerakan langkah sampai pada titik buntu atau solusi. Jika pergerakan langkah mencapai titik buntu, maka simpul tersebut dimatikan dan lintasan dirunut-balik ke simpul-simpul diatasnya. Demikian seterusnya hingga solusi ditemukan.

Keterangan beberapa fungsi:

- Pada fungsi `movePlayer`, pemain hanya dapat berjalan satu langkah saja, tetapi arah bebas tergantung arah mana yang lebih diprioritaskan.
- Pada fungsi `moveMummy`, *mummy* dapat berjalan dua langkah sekaligus dalam arah mana saja, tergantung arah mana yang lebih diprioritaskan. Tetapi terdapat beberapa pengecualian, yaitu *mummy* dapat berjalan satu langkah jika bergerak mengikuti pemain dalam salah satu posisi yang sama serta dibatasi tembok antara pemain dengan *mummy*.
- Terdapat fungsi `unmovePlayer`, yaitu algoritma *backtracking* pada pemain. Pemain mengalami *backtrack* jika pemain dihadapkan jalan buntu, jika hampir mendekati *mummy*, dan jika dalam posisi yang sama dengan *mummy*.
- Terdapat fungsi `unmoveMummy`, yaitu algoritma *backtracking* pada *mummy*. Tetapi algoritma ini dikhususkan jika posisi *mummy* dengan pemain sama. Sisanya *mummy* akan bergerak mundur mengikuti pemain dan jika hendak mendekati pemain

4. Kesimpulan

Algoritma Runut-balik (*backtracking*) merupakan algoritma yang cukup mangkus untuk

menyelesaikan berbagai persoalan. Karena pada prinsipnya, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Pencarian hanya mengarah pada solusi yang dipertimbangkan saja. Oleh karena algoritma ini sangat efektif, maka banyak diterapkan dalam berbagai persoalan game dan dalam bidang kecerdasan buatan.

Hasil analisis kemampuan algoritma *backtracking* dalam mencari jalan keluar terbaik pada permainan "Mummy Maze Deluxe" menunjukkan bahwa algoritma ini cukup efektif untuk mendapat solusi persoalan tersebut. Sistem kerja algoritma *backtracking* yang sistematis dan yang hanya memeriksa kemungkinan solusi yang dapat dipertimbangkan sebagai solusi akhir, maka cara ini dapat diperkirakan sebagai pencarian solusi yang efektif dan efisien.

Sebagai sesuatu yang baru saja dilakukan, pencarian jalan keluar terbaik dalam permainan "mummy maze deluxe" ini belum dapat dianalisis untuk beberapa algoritma yang lain. Namun dari analisis yang kami lakukan terhadap permainan ini, kami berpendapat bahwa algoritma runut-balik (*backtracking*) cukup efektif untuk menyelesaikan persoalan ini.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi. 2005. Diktat Kuliah Strategi Algoritmik. Program Studi Teknik Informatika ITB : Bandung