

Aplikasi Algoritma Runut Balik (*Backtracking*) Pada Permainan Tiga Pengelana bersama Tiga Perampok

Doni Arzinal

Program Studi Informatika
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail: if15109@students.if.itb.ac.id

ABSTRAK

Runut-balik (*Backtracking*) adalah algoritma berbasis DFS untuk mencari solusi persoalan secara lebih mangkus yang merupakan perbaikan dari algoritma brute-force dengan ciri memangkas (*prunning*) simpul-simpul yang tidak mengarah ke solusi. Algoritma Runut-balik (*Backtracking*) banyak digunakan untuk program games dan masalah-masalah pada bidang kecerdasan buatan (*Artificial Intelegence*). Salah satu jenis permainan yang dapat diselesaikan dengan algoritma Runut-balik (*Backtracking*) adalah permainan Tiga Pengelana bersama Tiga Perampok. Pada awal permainan, akan kita jumpai tiga orang pengelana dan tiga orang perampok pada suatu daratan A dimana tiga orang pengelana hendak menyebrang ke daratan B. Untuk bisa sampai ke daratan B, pengelana harus memanfaatkan perahu yang hanya bisa dinaiki oleh 2 orang. Karena perahu hanya dapat dinaiki oleh dua orang, pemain harus mengatur strategi agar siapa-siapa saja yang akan naik perahu dan siapa yang akan kembali untuk menjemput orang yang ingin menyebrang juga. Perampok dan Pengelana sama-sama dapat naik perahu, namun perampok dapat membunuh pengelana jika jumlah perampok di suatu daratan melebihi jumlah pengelana pada suatu daratan tersebut. Tujuan dari permainan ini adalah pemain harus dapat menyeberangkan ketiga pengelana ke daratan B dengan selamat sedangkan ketiga perampok tetap tinggal di daratan A.

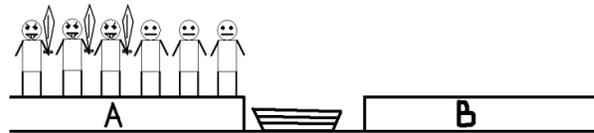
Kata kunci: Runut-balik, *Backtracking*, Permainan Tiga Pengelana bersama Tiga Perampok.

1. PENDAHULUAN

Sejalan dengan pesatnya perkembangan teknologi informasi, banyak Permainan-permainan (*games*) yang bermunculan ditengah-tengah masyarakat. Tidak semua permainan (*game*) yang mengandalkan ketangkasan pemain dalam menekan tombol *keyboard* atau

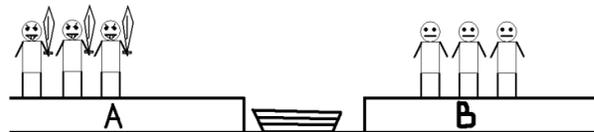
mengerakkan *mouse*. Terdapat permainan (*game*) yang menuntut kemampuan logika pemain untuk menemukan solusi suatu persoalan.

Permainan Tiga Pengelana bersama Tiga Perampok adalah salah satu jenis permainan (*game*) yang mengandalkan logika pemainnya. Pada awal permainan, akan kita jumpai tiga orang pengelana dan tiga orang perampok pada suatu daratan A dimana tiga orang pengelana hendak menyebrang ke daratan B. Untuk bisa sampai ke daratan B, pengelana harus memanfaatkan perahu yang hanya bisa dinaiki oleh 2 orang.



Gambar 1. Permainan 3 Pengelana & 3 Perampok.

Karena perahu hanya dapat dinaiki oleh dua orang, pemain harus mengatur strategi agar siapa-siapa saja yang akan naik perahu dan siapa yang akan kembali untuk menjemput orang yang ingin menyebrang juga. Perampok dan Pengelana sama-sama dapat naik perahu, namun perampok dapat membunuh pengelana jika jumlah perampok di suatu daratan melebihi jumlah pengelana pada suatu daratan tersebut. Pemain harus dapat menyeberangkan ketiga pengelana ke daratan B dengan selamat sedangkan ketiga perampok tetap tinggal di daratan A.



Gambar 2. Pengelana telah sampai di daratan B dengan selamat.

Untuk menyelesaikan persoalan ini, dapat digunakan Algoritma runut balik (*backtracking*). Algoritma ini berbasis DFS (*Depth First Search*), dimana solusi dapat ditemukan dengan penelusuran yang lebih sedikit dan dapat mencari solusi permasalahan secara lebih mangkus.

Algoritma runut-balik sendiri merupakan perbaikan dari algoritma brute force dan DFS.

2. METODE

2.1 Strategi Umum Penyelesaian Permainan Tiga Pengelana dan Tiga Perampok.

Secara umum, untuk menyelesaikan Permainan Tiga Pengelana dan Tiga Perampok kita dapat mencoba semua kemungkinan-kemungkinan agar ketiga pengelana dapat sampai di daratan B. Sebelum mengantarkan pengelana atau perampok ke daratan B atau sebaliknya, kita harus memeriksa apakah jumlah perampok di suatu daratan akan menjadi lebih banyak dari jumlah pengelana. Jika jumlah perampok lebih banyak, maka pengelana akan dibunuh oleh perampok dan permainan berakhir.



Gambar 3. Pengelana di daratan A akan dibunuh oleh perampok.

2.2 Pemodelan Masalah

Untuk memodelkan masalah ini, kita misalkan daratan A, daratan B, dan Perahu adalah tabel yang akan diisi oleh Pengelana dan Perampok. Tabel A menunjukkan daratan A, Tabel P menunjukkan Perahu (A pada tabel P menunjukkan tempat perahu melabuh), dan Tabel B menunjukkan daratan B. PR1, PR2 dan PR3 menunjukkan perampok sedangkan PL1, PL2 dan PL3 menunjukkan pengelana.

- a. Situasi pada awal permainan

Tabel A (Daratan A)

PR1	PR2	PR3	PL1	PL2	PL3
-----	-----	-----	-----	-----	-----

Tabel P (Perahu)

		A
--	--	---

Tabel B (Daratan B)

--	--	--	--	--	--

Pada awal permainan, kita dapat memilih diantara PR1,PR2,PR3, PL1, PL2 dan PL3 yang akan menaiki perahu. Jika kita memilih PL1 dan PL2 untuk naik perahu, saat perahu meninggalkan daratan A maka PL3 akan dibunuh oleh perampok PR1,PR2 dan PR3 di daratan A

karena jumlah perampok ada 3 orang sedangkan pengelana sendirian di daratan A.

- b. Situasi saat PL1 dan PL2 menaiki perahu

Tabel A (Daratan A)

PR1	PR2	PR3			PL3
-----	-----	-----	--	--	-----

Tabel P (Perahu)

PL1	PL2	A
-----	-----	---

Tabel B (Daratan B)

--	--	--	--	--	--

- c. Situasi saat perahu sampai di daratan B

Tabel A (Daratan A)

PR1	PR2	PR3			PL3
-----	-----	-----	--	--	-----

(PL3 akan dibunuh oleh perampok)

Tabel P (Perahu)

PL1	PL2	B
-----	-----	---

Tabel B (Daratan B)

--	--	--	--	--	--

Untuk mengubah status perahu dari A ke B atau sebaliknya diperlukan sedikitnya satu orang untuk mengemudikan perahu ke daratan A atau B. Perampok dan Pengelana di daratan A dapat menaiki/menuruni perahu jika status perahu saat itu adalah A, sedangkan Perampok dan Pengelana di daratan B dapat menaiki/menuruni perahu jika status perahu saat itu adalah B.

2.3 Gambaran Umum Algoritma Runut-balik (*backtracking*)

Runut-balik (*backtracking*) adalah algoritma berbasis DFS dapat mencari solusi persoalan secara lebih mangkus. Runut-balik yang merupakan perbaikan algoritma *brute-force* secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan metode ini, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan. Akibatnya waktu pencarian dapat dihemat. Runut balik lebih alami dinyatakan dalam algoritma rekursif. Kadang-kadang disebutkan pula bahwa Runut-balik merupakan bentuk tipikal dari algoritma rekursif.

2.4 Prinsip Pencarian Algoritma Runut-balik (*Backtracking*)

Pada sub bab ini, kita hanya akan meninjau pencarian solusi pada pohon ruang status yang dibangun secara dinamis. Langkah-langkah pencarian solusi adalah sebagai berikut :

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti metode pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan simpul hidup (*live node*). Simpul hidup yang sedang diperluas dinamakan simpul-E (*expand node*). Simpul diberi nomor dari atas ke bawah sesuai dengan urutan kelahirannya.
2. Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut dibunuh sehingga menjadi simpul mati (*dead node*). Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan fungsi pembatas (*bounding function*). Simpul yang telah mati tidak akan pernah diperluas lagi.
3. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak yang lainnya. Bila tidak ada lagi simpul anak yang dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan Runut-balik ke simpul hidup terdekat (simpul bapak). Selanjutnya, simpul ini menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.
4. Pencarian dihentikan bila tidak menemukan solusi atau tidak ada lagi simpul hidup untuk runut-balik

2.5 Penerapan Algoritma Runut-balik dalam Permainan Tiga Pengelana dan Tiga Perampok.

Secara garis besar, algoritma runut balik untuk menyelesaikan Permainan ini diberikan sebagai berikut :

```

procedure SelesaikanPermainan()
{
if isTerselesaikan() then
    exit()
    {permainan telah terselesaikan,
    program selesai}
else
    naikPerahu()
    {Pengelana atau Perampok atau
    keduanya naik perahu}
    if BisaTerbunuh then
    {periksa apakah pengelana dapat
    terbunuh}
        backtrack()
    else

```

```

    jalanKeB()
    {perahu jalan ke daratan B}
    turunPerahu()
    if BisaTerbunuh() then
    {Periksa lagi apakah
    pengelana dapat terbunuh}
        backtrack()
    else
    naikPerahu()
    jalanKeA()
    turunPerahu()
    if BisaTerbunuh() then
    {Periksa lagi apakah
    pengelana dapat terbunuh}
        backtrack()
    else
        SelesaikanPermainan()
        {rekursif}
}

```

Sebagai gambaran dari algoritma diatas, dapat kita lihat dari contoh penyelesaian permainan dengan algoritma diatas:

- a. Situasi Pertama (Awal)

Tabel A

PR1	PR2	PR3	PL1	PL2	PL3
-----	-----	-----	-----	-----	-----

Tabel P

		A
--	--	---

Tabel B

--	--	--	--	--	--

- b. Situasi Kedua (PR1 & PL1 naik Perahu)

Tabel A

	PR2	PR3		PL2	PL3
--	-----	-----	--	-----	-----

Tabel P

PR1	PL1	A
-----	-----	---

Tabel B

--	--	--	--	--	--

- c. Situasi Ketiga (Perahu sampai di daratan B)

Tabel A

	PR2	PR3		PL2	PL3
--	-----	-----	--	-----	-----

Tabel P

PR1	PL2	B
-----	-----	---

Tabel B

--	--	--	--	--	--

d. Situasi Keempat (PL1 turun perahu)

Tabel A

	PR2	PR3		PL2	PL3
--	-----	-----	--	-----	-----

Tabel P

	PR1	B
--	-----	---

Tabel B

PL1					
-----	--	--	--	--	--

e. Situasi Keempat (PR1 naik perahu kembali ke A)

Tabel A

	PR2	PR3		PL2	PL3
--	-----	-----	--	-----	-----

Tabel P

	PR1	A
--	-----	---

Tabel B

PL1					
-----	--	--	--	--	--

Pada situasi ini, PL2 dan PL3 akan dibunuh karena jumlah perampok ada tiga orang sedangkan pengelana hanya dua orang, maka lakukan Runut-balik (*backtrack*) ke Situasi Keempat. Pada situasi keempat bukan PL1 yang turun dari perahu, melainkan PR1. Demikian seterusnya sampai solusi yang benar ditemukan.

3. KESIMPULAN

Salah satu alternatif solusi dalam menyelesaikan Permainan Tiga Pengelana dan Tiga Perampok dapat digunakan algoritma Runut-balik (*backtracking*). Pemilihan algoritma Runut-balik (*backtracking*) dalam menyelesaikan Permainan Tiga Pengelana dan Tiga Perampok dapat dibilang baik jika dibandingkan dengan algoritma *brute force*. Hal ini dikarenakan pada algoritma Runut-balik, semua kemungkinan solusi tidak diperiksa, berbeda dengan algoritma *brute force* yang mencari semua kemungkinan solusi. Algoritma Runut-balik tidak memeriksa semua kemungkinan solusi karena simpul-simpul yang tidak mengarah ke solusi akan dimatikan dan tidak akan pernah diperiksa lagi

REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF2251 Strategi Algoritmik", Bandung, 2007
- [2] <http://en.wikipedia.org> diakses pada Selasa 22 Mei 2007