

# Penerapan Algoritma Runut-balik Dalam Pembuatan *Artificial Intelligence* Pada *Game Turn-base-system*

Ignatius Alriana Haryadi Moel - 13513051<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

<sup>1</sup>alrianahm@students.itb.ac.id

**Abstrak**—Pada *game turn-base-system* biasanya pemain akan melawan sebuah komputer yang sering kita sebut sebagai AI atau *Artificial Intelligence*. Pergerakan AI sendiri didasari pada sebuah algoritma. Bagaimana dia tahu petak yang tidak boleh dilewati, posisi musuh atau jangkauan serangan musuh. Dengan algoritma runut-balik, kita akan mengetahui bagaimana AI dapat bergerak dengan baik.

**Keywords**— TBS, *Backtracking* dalam game, AI game

## I. PENDAHULUAN

### A. *Turn-base-system Game*

*Turn-base-system Game* adalah sebuah tipe permainan (*video game*) yang membagi alur permainan ke dalam beberapa giliran (*turns*). Secara umum terdapat dua giliran dalam *Turn-base-system Game* yaitu giliran pemain (*player's turn*) dan giliran lawan (*enemy's turn*). Secara bergantian permainan akan masuk ke dalam giliran pemain dan giliran lawan.

Pada giliran pemain, pemain dapat melakukan aksi untuk setiap karakter yang ada seperti *move*, *attack*, *defend*. Dalam setiap gilirannya setiap karakter hanya bisa digerakan sekali dengan beberapa aksi (*move*, *attack*) atau (*defend*). Selain menggerakkan karakter yang ada pemain juga dapat melakukan aksi pada sistem permainan seperti membuat karakter dan mengakhiri giliran.

Pada giliran musuh, jika kita melawan pemain lain maka hal yang akan dilakukan pada giliran ini akan sama dengan giliran pemain hanya saja dengan sisi yang berbeda. Jika pemain melawan komputer, maka komputer akan bergerak dengan algoritma AI-nya. Jika terdapat tingkatan musuh, maka tingkat musuh menentukan algoritma apa yang dipakai. Pergerakan komputer ini meliputi pergerakan (*movement*), perilaku musuh apakah menyerang atau bertahan, apakah akan membuat karakter atau mengakhiri giliran. Untuk mengatur semua itu diperlukan algoritma yang cukup rumit. Dalam hal ini yang akan dibahas hanyalah algoritma pergerakan (*movement*), agar musuh dapat bergerak dengan baik, dapat mendekati karakter pemain untuk menyerang karakter pemain dan dapat jauh karakter pemain.

Berikut adalah beberapa contoh *Game Turn-base-system*:



Advanced Wars 2 nintendo(2003) / Gameboy



Fire Emblem "The Binding Blade" nintendo(2002) / Gameboy



Disgaea2 / PS2



Final Fantasy Tactics / Gameboy

Sumber:

[http://vignette3.wikia.nocookie.net/disgaea/images/0/08/Disgaea\\_2\\_PSP\\_Logo.jpg/revision/latest?cb=20110306225942&path-prefix=en](http://vignette3.wikia.nocookie.net/disgaea/images/0/08/Disgaea_2_PSP_Logo.jpg/revision/latest?cb=20110306225942&path-prefix=en)  
<http://coolrom.com/screenshots/gba/Final%20Fantasy%20Tactics%20Advance.jpg>

### B. Artificial Intelligence in Game

Artificial Intelligence atau sering disebut sebagai AI adalah studi dan pengembangan sistem komputer yang menyerupai kecerdasan manusia. Dengan kata lain komputer akan berperilaku seperti atau menggantikan manusia. Beberapa jenis kecerdasan buatan yaitu algoritma pencarian seperti BFS, DFS, runut-balik, A\*.

Penerapan AI dalam *game* bertujuan untuk membuat *game* lebih menantang dan lebih menarik. AI dalam *game* akan mencoba mempelajari pemain untuk mengalahkan pemain dengan melihat langkah-langkah yang dilakukan oleh pemain dan melihat beberapa langkah ke depan. Sehingga aksi atau langkah yang diambil oleh komputer akan mendekati langkah optimal untuk mengalahkan pemain.

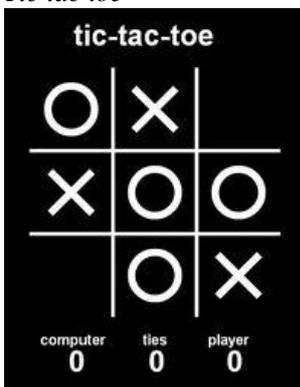
## II. DASAR TEORI

Runut-balik dapat dipandang dari 2 sisi:

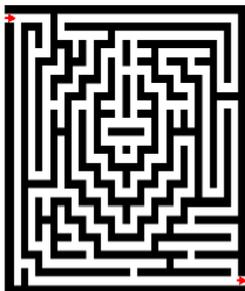
1. Sebagai bagian di dalam algoritma traversal *Breadth First Search*.
2. Sebagai sebuah metode pemecahan masalah yang terstruktur dan sistematis.

Runut-balik adalah algoritma pencarian solusi yang berbasis pada DFS. Algoritma ini sering diterapkan untuk programs *games*:

1. *Tic-tac-toe*



2. *Maze*



3. Catur



4. Sudoku

5	3			7			
6			1	9	5		
	9	8					6
8				6			3
4			8		3		1
7				2			6
	6					2	8
			4	1	9		5
				8			7
						7	9

Algoritma runut-balik memperbaiki kerja dari algoritma *brute-force*. Pada *brute-force*, semua kemungkinan solusi ditelusuri satu persatu. Dengan algoritma runut-balik, pilihan yang ditelusuri hanyalah pilihan yang mengarah ke solusi. Pilihan yang tidak mengarah ke solusi tidak akan diperhitungkan lagi dengan cara memotong langkah berikutnya.

Proerti umum metode runut-balik:

1. Solusi persoalan  
Solusi berisi langkah langkah yang harus diambil dari himpunan langkah-langkah  $\{X_i..X_n\}$
2. Fungsi Pembangkit  
Menginisiasi nilai yang merupakan komponen dari solusi  $X_k$ .
3. Fungsi Pembatas  
Bergantung pada fungsi pembangkit. Fungsi pembatas akan bernilai benar jika mengarah ke solusi, bernilai salah jika tidak mengarah ke solusi. Jika fungsi pembatas bernilai benar maka fungsi pembangkit akan menjalankan langkah selanjutnya. Jika fungsi pembangkit bernilai salah maka semua langkah sampai ke k akan dibuang.

Prinsip Pencarian solusi dengan algoritma runut-balik:

1. Solusi dicari dengan membentuk lintasan dari akar ke daun mengikuti aturan pembuatan pohon ruang DFS.
2. Simpul yang sudah dibangkitkan dinamakan simpul hidup.
3. Simpul yang sedang diperluas dinamakan simpul-E.
4. Setiap simpul-E diperluas maka lintasan yang dibangun akan bertambah.
5. Jika lintasa yang sedang dibentuk tidak mengarah ke solus, maka simpul-E akan dibatasi dan menjadi simpul mati.

6. Fungsi yang membatasi sebuah simpul adalah fungsi pembatas.
7. Simpul mati tidak akan diperluas lagi.
8. Jika pembentukan lintasan berujung pada simpul mati, proses pencarian akan *backtrack* ke simpul di atasnya.
9. Bangkitkan kembali simpul anak yang lain. Simpul anak ini akan menjadi simpul-E yang baru.
10. Pencarian dihentikan saat kita sampai pada daun solusi.

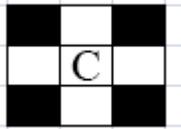
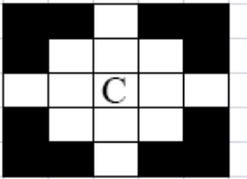
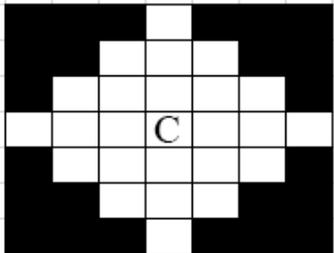
### III. PENERAPAN ALGORITMA RUNUT-BALIK

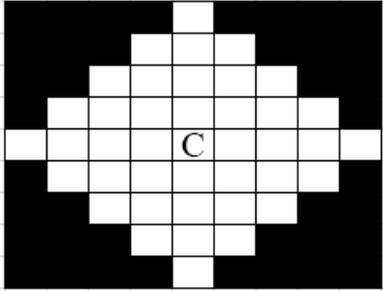
Dalam *Turn-base-system Game*, setiap karakter memiliki beberapa atribut. Dalam tahap mendesain kita mulai dengan permainan yang sederhana. Misalkan sebuah karakter memiliki 2 atribut yaitu:

1. HP : Nilai nyawa yang dimiliki
2. MOV : Jangkauan pergerakan (n)

Karakter dapat menyerang dengan jarak dekat yang berarti posisi karakter dengan karakter yang ingin diserang bersebelahan. Dan juga jarak jauh, yang berarti posisi karakter tidak harus bersebelahan untuk dapat menyerang, tapi posisi bergantung pada jangkauan serangnya. Namun untuk desain yang sederhana kita hanya perlu karakter dengan tipe jarak dekat.

Pergerakan karakter dalam *Turn-base-system Game* digambarkan dalam bentuk kotak-kotak (*grid*). Untuk  $MOV = n$ ,  $n=1..4$ . Jumlah kotak yang mungkin dituju oleh karakter (C) adalah  $2n^2+2n$ .

<i>Movement Range</i>	N	Jumlah kotak
	1	4
	2	12
	3	24

	4	40
------------------------------------------------------------------------------------	---	----

Tabel 1 *Movement Range*

Informasi tambahan yang dibuat saat kita *generate map* yaitu kotak akan bernilai -1 jika tidak bisa dilewati, 0 jika bisa dilewati (direpresentasikan kosong agar tidak membuat bingung),  $P_i$  Jika merupakan lawan.  $i$  menandakan jumlah lawan yang terdeteksi dalam range. Pendeteksian lawan tidak dibahas dalam makalah ini, namun dapat dengan mudah diselesaikan dengan algoritma BFS.

Properti algoritma runut-balik dalam pembahasan kali ini:

Ada 3 kasus pergerakan yang akan kita bahas menggunakan algoritma runut-balik yaitu:

1. Bergerak saja.
2. Bergerak untuk menyerang lawan.
3. Bergerak untuk menghindari lawan.

Kita akan mencoba kasus pertama dimana tidak ada lawan yang terdeteksi dalam range. Misalkan karakter C memiliki atribut :

HP : 100

MOV : 2

Yang berarti kita menggunakan pergerakan dengan  $n=2$  pada tabel *movement range*.

	-1			
	-1	C		
			-1	

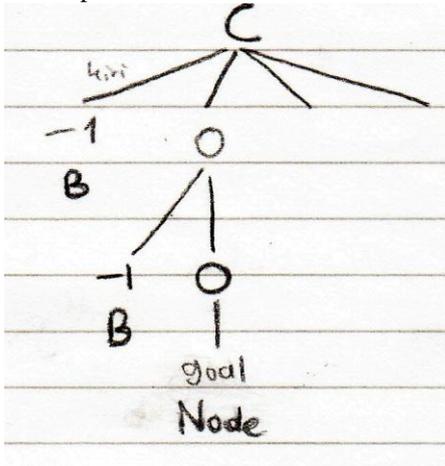
*Movement range* setelah dilakukan pendeteksian

Properti algoritma runut-balik dalam kasus kali ini:

1. Solusi persoalan  
 $S\{\text{kiri,atas,kanan,bawah}\}$
2. Fungsi Pembangkit( $T(k)$ )  
Mengembalikan nilai dari suatu kotak.
3. Fungsi Pembatas  
 $T(k)=0 \rightarrow \text{true}$ .

$T(k) == -1 \rightarrow \text{false}$ .

Pohon ruang status dibangun dengan bergerak kiri, atas, kanan, bawah. Urutan ini bergantung kepada posisi awal dengan markas musuh. Kotak yang sudah dilewati oleh simpul hidup akan bernilai -1.



Pohon ruang status kasus satu.

Solusi = atas  $\rightarrow$  atas

Selanjutnya adalah kasus kedua dan ketiga. Terdapat musuh dalam jangkauan gerak karakter. Pada awalnya kita memiliki kondisi yang membolehkan sebuah karakter untuk menyerang yaitu jika HP karakter  $\geq$  HP lawan. Karena kita akan menyerang, jangkauan gerak lawan tidak kita pedulikan.

Atribut karakter C1

HP : 88

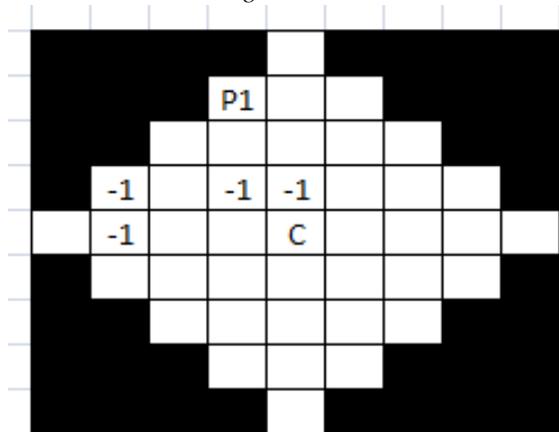
MOV : 4

Atribut karakter lawan

HP : 51

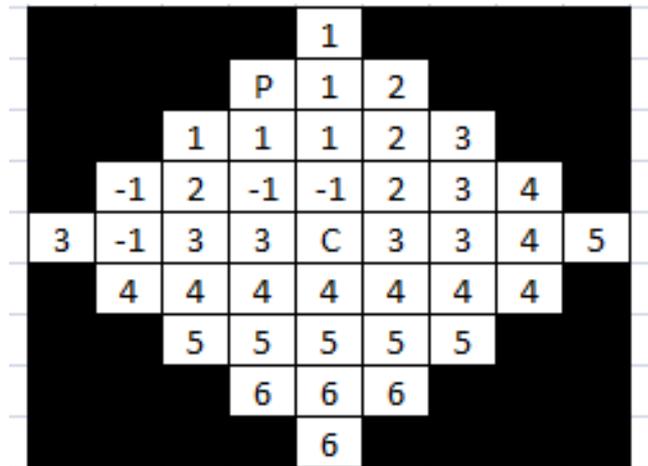
MOV : 2

Yang berat kita menggunakan pergerakan dengan  $n=3$  pada tabel *movement range*.



*Movement range* setelah dilakukan pendeteksian

Untuk menyelesaikan kasus ini dibutuhkan fungsi pembantu untuk menghitung jarak tiap kotak relatif terhadap lawan.

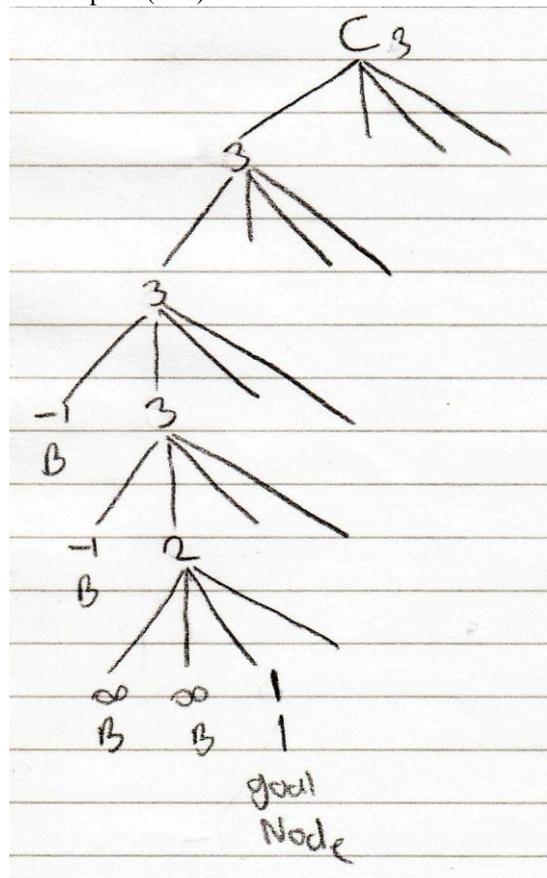


*Movement range* setelah dilakukan fungsi pembantu

Properti algoritma runut-balik dalam kasus kali ini:

1. Solusi persoalan  
S{kiri,atas,kanan,bawah}
2. Fungsi Pembangkit(T(k))  
Mengembalikan nilai dari suatu kotak.
3. Fungsi Pembatas  
 $T(k) \leq T(k-1)$  and  $T(k) \neq -1 \rightarrow \text{true}$ .  
 $T(k) == -1 \rightarrow \text{false}$ .

Pohon ruang status dibangun dengan bergerak kiri, atas, kanan, bawah. Urutan ini bergantung kepada posisi awal dengan markas musuh. Kotak yang sudah dilewati oleh simpul hidup akan bernilai -1. Algoritma akan berhenti sampai  $T(k+1) = P$ .



Pohon ruang status kasus dua.

Solusi: C → kiri → kiri → atas → atas → atas → kanan.

Jika HP karakter < HP lawan. Karena kita akan menghindari, jangkauan gerak lawan kita pedulikan.

Atribut karakter C1

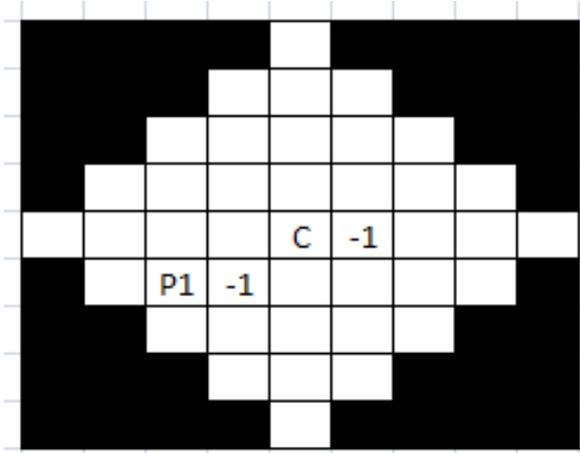
HP : 88

MOV : 4

Atribut karakter lawan

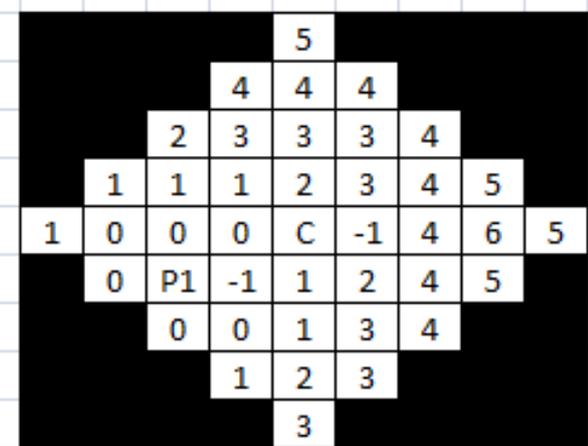
HP : 100

MOV : 2



Movement range setelah dilakukan pendeteksian

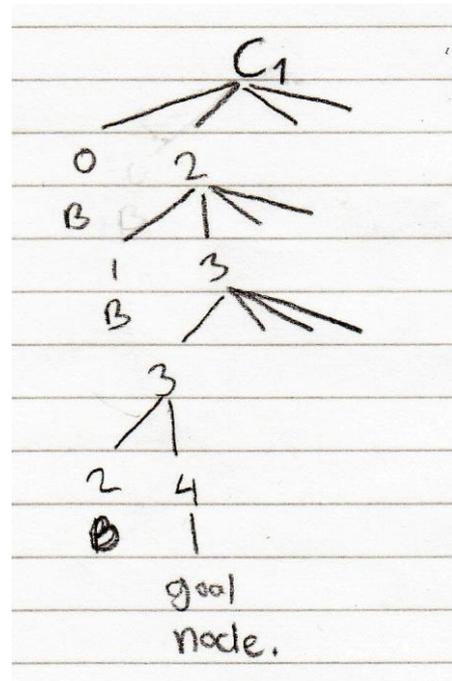
Untuk menyelesaikan kasus ini dibutuhkan fungsi pembantu untuk menghitung jarak tiap kotak relatif terhadap lawan. Jika kotak berada dalam *movement range+1* lawan maka isi kotak dikurang 1. Hal ini untuk menghindari karakter berhenti di *movement range* lawan.



Movement range setelah dilakukan fungsi pembantu

Properti algoritma runut-balik dalam kasus kali ini:

4. Solusi persoalan  
S {kiri,atas,kanan,bawah}
5. Fungsi Pembangkit(T(k))  
Mengembalikan nilai dari suatu kotak.
6. Fungsi Pembatas  
T(k)>T(k-1) and T(k)!=-1 → true.  
T(k)=-1 → false.



Pohon ruang status kasus dua.

Solusi: C → atas → atas → kiri → kiri → atas

#### IV. KESIMPULAN

Game Turn-base-system membutuhkan AI lebih dari apa yang dibahas di dalam makalah ini. AI dapat membuat pergerakan komputer lebih menyerupai manusia walaupun mungkin berbeda dengan cara pikir kita sendiri, karena setiap orang pasti memiliki gaya bermain tersendiri.

Penerapan AI akan sangat diperlukan dalam dunia *game* bukan hanya dalam pergerakan saja namun akan mencapai pola pikir manusia secara heuristic bahkan komputer suatu saat akan memiliki gaya bermain tersendiri

#### V. UCAPAN TERIMA KASIH

Saya mengucapkan terimakasih kepada Tuhan Yesus atas berkat dan kasih-Nya, Bapak Rinaldi Munir dan Ibu Ulfa sebagai dosen IF2211 Strategi Algoritma yang telah memberikan ilmu pengetahuan sebagai dasar dibuatnya makalah ini, dan juga kepada keluarga saya yang selalu memberikan dorongan untuk terus belajar.

#### REFERENSI

- [1]. Cawsey A. 1998. The Essence of Artificial Intelligence. Europe, Prentice Hall.
- [2]. Munir, Rinaldi. "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika ITB, 2009.
- [3]. <http://jasapembuatantesisilmukomputer.blogspot.com/2013/07/menerapkan-kecerdasan-buatan-algoritma.html>  
Tanggal akses 3 Mei 2015 pukul 02:10
- [4]. [http://www.gamasutra.com/view/feature/129959/designing\\_ai\\_algorithms\\_for\\_php](http://www.gamasutra.com/view/feature/129959/designing_ai_algorithms_for_php)  
Tanggal akses 3 Mei 2015 pukul 02:20
- [5]. <http://www.mobygames.com/glossary/genres> Definitions.htm

- Tanggal akses 4 Mei 2015 pukul 15:28
- [6]. <http://www-formal.stanford.edu/jmc/whatisai/node3.html>  
Tanggal akses 4 Mei 2015 pukul 15:25
- [7]. <http://www.imada.sdu.dk/~marco/Teaching/AY2014-2015/DM842/Slides/dm842-p2-lec1.pdf>  
Tanggal akses 4 Mei 2015 pukul 11:18

### 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, 4 Mei 2015



Ignatius Alriana Haryadi Moel - 13513051