

# ANALISIS ALGORITMA BACKTRACKING PADA GAME KUDA MENYEBERANG JEMBATAN DENGAN LINTASAN HAMILTON

Sahrul Arif<sup>1</sup>, Helmi Fajar A<sup>2</sup>, Yanuar Asshidqi<sup>3</sup>

Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Telkom  
Jl. Telekomunikasi, Bandung

[sahrularif@yahoo.co.id](mailto:sahrularif@yahoo.co.id), [lhypsync@gmail.com](mailto:lhypsync@gmail.com), [korness@gmail.com](mailto:korness@gmail.com)

---

## Abstrak

Algoritma Backtracking merupakan salah satu solusi algoritma dalam pencarian solusi yang lebih baik dari algoritma lainnya. Algoritma ini mencari solusi persoalan tanpa memeriksa semua kemungkinan yang ada, melainkan hanya kemungkinan solusi ke arah yang dipertimbangkan. Untuk sekarang ini, algoritma backtrack ini merupakan salah satu algoritma yang diterapkan pada program games dam masalah-masalah pada bidang kecerdasan buatan. Salah satu persoalan yang akan kami bahas yakni pada permainan kuda untuk menyeberang dengan aturan harus menghabiskan balok yang ada diantara dua jembatan dan kuda harus kembali ke posisi semula agar bisa menyeberang dengan selamat. Jadi, kuda harus bisa berjalan menurut Lintasan Hamilton yakni terakhir kuda kembali pada posisi awal. Mengapa pada persoalan ini kami menggunakan algoritma backtracking dengan lintasan hamilton? Karena menurut kami penggunaan algoritma ini lebih mangkus dan lebih mudah dipahami dari algoritma – algoritma yang lainnya.

**Kata kunci:** Algoritma Backtracking, Kuda, lintasan Hamilton

---

## Abstract

Backtracking Algorithm is a better solution algorithm. It can find the solution without checking all the possibilities, but only the solution which satisfy the problem. Nowadays, backtracking algorithm used in many game programs and artificial intelligence. One of the problem which will focused is a flash game, a horse which will across the bridge, used Hamilton circuit. Why did we use backtracking algorithm for hamilton circuit to this problem? Because, in uor humble opinion, it gives efficient step and easier to understand than the others.

**Keywords :** Backtracking algorithm, Horse, Hamilton circuit

---

## 1. Pendahuluan

Runut-balik (backtracking) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus. Runut-balik, yang merupakan perbaikan dari algoritma brute-force, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Runut-balik merupakan bentuk tipikal dari algoritma rekursif. Saat ini algoritma runut-balik banyak diterapkan untuk program games (seperti permainan tic-tac-toe, menemukan jalan keluar dalam sebuah labirin, catur, dll) dan masalah-masalah pada bidang kecerdasan buatan (artificial intelligence).

Persoalan yang muncul pada game kuda menyeberang jembatan adalah bagaimana menghabiskan balok / kotak dan kuda kembali ke posisi awal. Setiap kali berpindah balok, maka balok tersebut akan jatuh, ini berarti kuda tidak dapat menggunakan lintasan yang sama dalam perjalanannya.

Untuk persoalan seperti ini banyak cara yang dapat digunakan untuk mendapatkan solusinya,

metode coba-coba merupakan cara paling sederhana yang dapat digunakan, selain itu kita dapat menggunakan tree / pohon. Penggunaan pohon memudahkan kita untuk melakukan runut balik jika langkah yang kita ambil salah. Selain itu pohon juga memberikan beberapa solusi langkah selanjutnya dari posisi sekarang.

## 2. Pembahasan

### 2.1 Properti umum Metode Runut-balik

Properti Umum Metode Runut-balik

1. Solusi persoalan.

Solusi dinyatakan sebagai vektor dengan n-tuple:

$X = (x_1, x_2, \dots, x_n)$ ,  $x_i \in$  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 vektor solusi.

3. Fungsi pembatas (pada beberapa persoalan fungsi ini dinamakan fungsi kriteria)

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, tetapi jika tidak, maka  $(x_1, x_2, \dots, x_k)$  dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

## 2.2 Penyelesaian Lintasan Hamilton

Salah satu permasalahan dalam penyelesaian lintasan Hamilton pada game kuda menyebrang jembatan yakni :

Diberikan sebuah papan yang berisikan kotak – kotak yang berjumlah 14 kotak. Kuda diletakan pada posisi awal yakni posisi bebas (misal : kita taruh posisi kuda pada kotak 1 ), kuda harus bisa melewati semua kotak tanpa harus balik lagi dengan aturan jalan (membentuk huruf L) sampai semua kotak itu habis dan kuda harus balik ke posisi awal yakni posisi 1 agar kuda dapat menyebrangi jembatan dengan selamat.

Persoalan perjalanan kuda catur ini merupakan permainan kecerdasan logika di mana pemain harus memikirkan strategi dalam setiap langkah kuda.

Penyelesaian:

1. Solusi persoalan

Solusi persoalan dituliskan dalam vektor dengan 14 kotak / tuple yang berisi urutan dari kotak yang dilalui oleh kuda. Posisi kotak dituliskan dalam bentuk dua digit integer. Misal : posisi kuda di nomor 1 maka dia bisa melangkah / jalan ke nomor 7,8,10. Solusi dicapai apabila seluruh kotak dalam papan sudah dilewati oleh kuda.

Contoh solusi:

$$S = (12,23,31,..dst)$$

	12	13	14
8	9	10	11
4	5	6	7
	1	2	3

Dikatakan solusi apabila kuda telah mengambil langkah seluruh kotak / tuple dan tidak kembali lagi ke kotak semula sampai dia kembali ke posisi awal pertama kali kuda mengambil posisi.

2. Fungsi pembangkit

Fungsi  $JalanKuda(k)$  membangkitkan kotak berikutnya yang dilalui kuda dan memasukkan kotak tersebut ke dalam solusi persoalan. Kotak berikutnya didapat dengan memilih salah satu elemen dari list yang dihasilkan fungsi  $JalanBerikut(k)$ . Dari list tersebut dipilih posisi dengan jumlah elemen  $JalanBerikut(k+1)$  yang terkecil.

Fungsi – fungsi bantuan:

Fungsi  $JalanBerikut(k)$  menghasilkan ListJalan yang berisi list dari posisi kotak berikutnya yang dapat dilewati oleh kuda, serta jumlah elemen dari ListJalan pada  $JalanBerikut(k+1)$ .

Contoh :

8		10	
			7
	1		

Fungsi  $JalanBerikut(1)$  menghasilkan

ListJalan :  $\{(7,8,10)\}$ .

Artinya kuda dari kotak 1 bisa berjalan

ke kotak 7 atau 8 atau 10 sehingga kuda mempunyai 3 kemungkinan, demikian juga setelah kuda berjalan ke kotak berikutnya maka kuda mempunyai beberapa kemungkinan tergantung kuda tersebut mempunyai kemungkinan untuk melangkah. Dari list tersebut dipilih kotak dengan alternatif jalan berikutnya yang paling kecil, jika sama maka pilih salah satu dan kotak yang lain dimasukkan ke dalam alternatif jalan bila diharuskan melakukan runut-balik.

3. Fungsi pembatas

Bila kuda tidak bisa berjalan lagi tetapi masih ada kotak yang belum dilewati maka kotak tersebut dilewati dan melakukan runut-balik ke penanda runut balik terdekat.

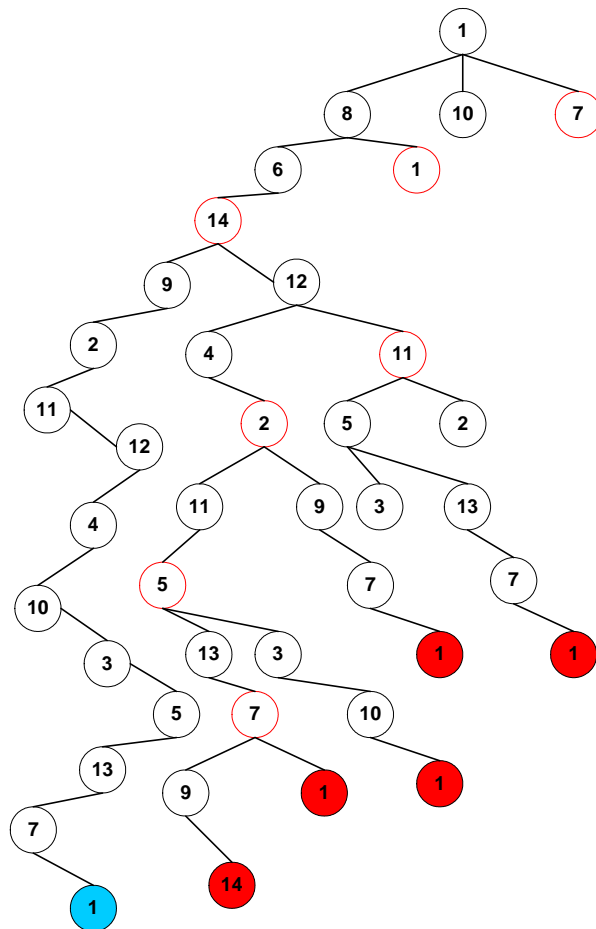
Aturan yang berlaku agar bisa menemukan solusi :

- Langkah berikutnya harus mempunyai jumlah kemungkinan langkah selanjutnya yang lebih kecil dibandingkan langkah yang lain. Bila ada jumlah langkah selanjutnya yang sama, maka menjadi penanda bila terjadi runut-balik.
- Bila langkah yang dipilih ternyata tidak menemukan solusi, maka runut-balik ke penanda runut-balik yang terdekat.
- Lakukan langkah tersebut hingga ditemukan solusi.

Analisa penggunaan Teknik Backtracking pada lintasan Hamilton :

1. Keunggulan menggunakan algoritma backtracking yakni mudah merunut balik apa yang kita kerjakan, apabila kita salah melangkah maka kita akan kembali pada posisi yang mempunyai solusi permasalahan
2. kelemahan menggunakan algoritma backtracking yakni kita belum dapat / belum mengetahui apakah langkah yang kita ambil merupakan langkah yang terbaik, sehingga memungkinkan terjadi terlalu banyak backtracking yang harus dilakukan. Hal ini dapat diminimalisasi dengan menggunakan fungsi Heuristik untuk menentukan langkah yang lebih optimal.

Penggambaran proses pencarian solusi menggunakan algoritma runut-balik dengan pohon graf alur perjalanan kuda :



Runut balik disini ditunjukkan (karena terlalu banyak) Tidak ditemukan solusi dan runut balik ke atas

KETERANGAN :

- solusi ditemukan
- solusi tidak ditemukan , lakukan runut balik
- runut balik disini

#### 4. Kesimpulan

Untuk berbagai macam persoalan yang memiliki banyak kemungkinan langkah untuk menemukan solusi, runut-balik dapat menyelesaikan dengan cukup baik. Hal yang paling penting dalam pengoptimalan algoritma runut-balik dalam pencarian solusi adalah penentuan fungsi pembangkit dan fungsi pembatas. Kedua fungsi inilah yang akan menentukan aturan langkah mana yang harus diambil dalam pencarian solusi.

Melalui analisis yang didapat dari penggunaan algoritma runut-balik ini, masih menunjukkan kerumitan dalam penyelesaian masalah. Terutama untuk masalah dengan jumlah solusi yang cukup banyak. Namun untuk saat ini algoritma runut-balik merupakan algoritma yang bisa menyelesaikan banyak persoalan, karena itulah sering dipakai dalam pembuatan *games*.

Agar penggunaan backtracking lebih mangkus maka sebaiknya ditambahkan fungsi heuristic pada algoritma backtracking.

#### **Daftar Pustaka**

- [1] Rinaldi Munir. *Diktat Kuliah Strategi Algoritmik*. Departemen teknik Informatika ITB : 2004.
- [2] Rinaldi Munir. *Matematika Diskrit edisi Kedua*. Informatika, Bandung : 2003.
- [3] Suyanto, *Diktat Kuliah Intelijensia Buatan*. Jurusan Teknik Informatika STTTelkom : 2002.