

Solusi Runut Balik untuk Persoalan Lintasan Hamilton dalam Perjalanan Kuda Catur

Hardani Maulana¹, Andoko Gunawan², Aldo Juwoti Yahya³

Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if13077@students.if.itb.ac.id¹, if13083@students.if.itb.ac.id²,
if13085@students.if.itb.ac.id³

Abstrak

Algoritma runut-balik (*backtracking*) adalah salah satu algoritma yang dapat mencari solusi persoalan yang lebih mangkus. Algoritma ini mencari solusi persoalan tanpa memeriksa semua kemungkinan yang ada, melainkan hanya kemungkinan pencarian solusi ke arah yang dipertimbangkan. Saat ini algoritma runut-balik banyak diterapkan untuk program games dan masalah-masalah pada bidang kecerdasan buatan (*artificial intelligence*). Salah satu persoalan yang kami bahas di sini adalah penggunaan algoritma runut-balik pada salah satu permainan yang menggunakan papan catur (Kami terinspirasi dari persoalan N-Ratu). Persoalan itu adalah sebagai berikut : Diberikan setengah papan catur (ukuran 4 x 8). Kuda diletakkan pada petak di salah satu sudut papan. Sisa petak diisi dengan bidak pion. Kuda harus bisa memakan habis seluruh pion dengan syarat tidak berjalan ke tempat yang kosong (tidak ada pion), dalam artian sudah dilalui sebelumnya. Bisa dimisalkan petak pada papan catur adalah simpul graf. Sisi diartikan kuda dapat bergerak dari salah satu simpul pada ujung sisi ke simpul pada ujung sisi yang lain. Sehingga kuda harus bisa berjalan dengan membentuk lintasan Hamilton yang menghubungkan seluruh simpul. Adakah solusi dari persoalan ini?

Kata kunci: runut-balik, papan catur, kuda, lintasan hamilton

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.
- Dengan metode runut-balik, 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 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*).

Properti Umum Metode Runut-balik

1. Solusi persoalan.
Solusi dinyatakan sebagai vektor dengan n -tuple:

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

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

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

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. Penyelesaian Persoalan Lintasan Hamilton dalam Perjalanan Kuda Catur

Salah satu persoalan yang kami bahas di sini adalah penggunaan algoritma runut-balik pada salah satu permainan yang menggunakan papan catur (Kami terinspirasi dari persoalan N-Ratu). Persoalan itu adalah sebagai berikut :

Diberikan setengah papan catur (ukuran 4×8). Kuda diletakkan pada petak di salah satu sudut papan. Sisa petak diisi dengan bidak pion. Kuda harus bisa memakan habis seluruh pion dengan syarat tidak berjalan ke tempat yang kosong (tidak ada pion), dalam artian sudah dilalui sebelumnya. Bisa dimisalkan petak pada papan catur adalah simpul graf. Sisi diartikan kuda dapat bergerak dari salah satu simpul pada ujung sisi ke simpul pada ujung sisi yang lain. Sehingga kuda harus bisa berjalan dengan membentuk lintasan Hamilton yang menghubungkan seluruh simpul.

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 32 tuple yang berisi urutan dari petak yang dilalui oleh kuda. Posisi petak dituliskan dalam bentuk dua digit integer. Misal : 21 artinya baris 2 kolom 1, 35 artinya baris 3 kolom 5. Solusi dicapai bila seluruh pion dalam papan catur sudah dimakan oleh kuda.

Contoh solusi:

$S = (11, 23, 31, \dots, \text{dst})$

Dinyatakan solusi telah ditemukan bila kuda tidak bisa berjalan lagi dan seluruh pion telah dimakan.

Jika ternyata tidak ada solusi maka jumlah tuple adalah lebih kecil dari jumlah seharusnya, yaitu 32.

2. Fungsi pembangkit

Fungsi $\text{jalanKuda}(k)$ membangkitkan petak berikutnya yang dilalui kuda dan memasukkan petak tersebut ke dalam solusi persoalan. Petak berikutnya didapat dengan memilih salah satu elemen dari list yang dihasilkan fungsi $\text{jalanBerikut}(k)$. Dari list ini dipilih posisi pion dengan jumlah elemen $\text{jalanBerikut}(k+1)$ yang terkecil.

Fungsi – fungsi bantuan:

Fungsi $\text{jalanBerikut}(k)$ menghasilkan ListJalan yang berisi list dari posisi pion berikutnya yang bisa dimakan oleh kuda, serta jumlah elemen dari ListJalan pada $\text{jalanBerikut}(k+1)$.

Contoh:

	1	2	3	4
1	K			
2				
3				
4				

Fungsi $\text{jalanBerikut}(11)$ menghasilkan

ListJalan : $\{(23,3), (32,3)\}$.

Artinya kuda dari 11 (baris 1 kolom 1) bisa berjalan ke 23 (baris 2 kolom 3) dengan kemungkinan berikutnya 3 alternatif jalan dan 32 (baris 3 kolom 2) dengan kemungkinan berikutnya 3 kali jalan. Dari list tersebut dipilih petak dengan alternatif jalan berikutnya yang paling kecil, jika sama maka pilih salah satu dan petak yang lain dimasukkan ke dalam alternatif jalan bila diharuskan melakukan runut-balik.

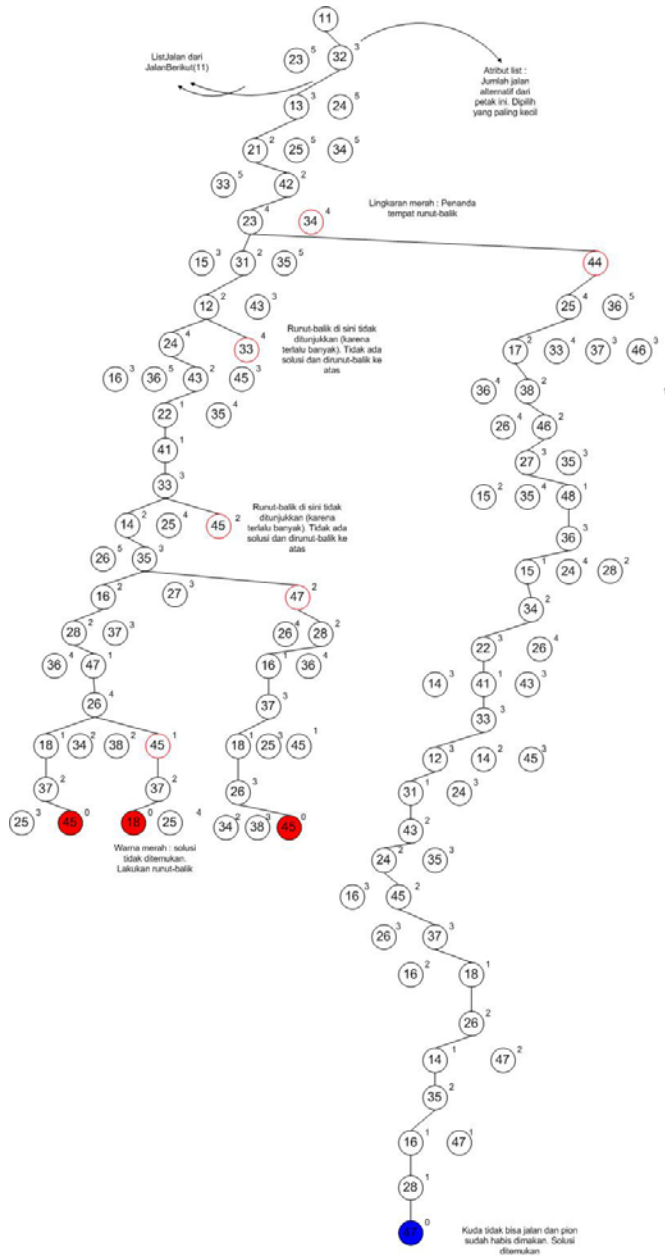
3. Fungsi pembatas

Bila kuda tidak bisa berjalan lagi tetapi masih ada pion yang belum dimakan maka simpul tersebut dibunuh dan melakukan runut-balik ke penanda runut balik terdekat.

Aturan yang berlaku agar bisa menemukan solusi :

1. 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.
2. Bila langkah yang dipilih ternyata tidak menemukan solusi, maka runut-balik ke penanda runut-balik yang terdekat.
3. Lakukan langkah tersebut hingga ditemukan solusi

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



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*.

Daftar Pustaka

- [1] Rinaldi Munir. *Diktat Kuliah Strategi Algoritmik*. Departemen teknik Informatika ITB : 2004.
- [2] Rinaldi Munir. *Diktat Kuliah Matematika Diskrit*. Departemen teknik Informatika ITB : 2004.