

# Penerapan Algoritma BFS dan DFS dalam Mencari Solusi Permainan Rolling Block

Zakiy Firdaus Alfikri  
13508042

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
if18042@students.if.itb.ac.id

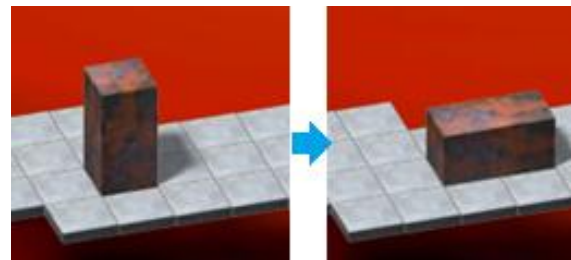
**Abstrak**—Rolling Block merupakan salah satu permainan yang cukup sederhana. Dalam permainan ini ada sebuah balok yang berukuran  $1 \times 1 \times 2$  satuan luas, balok tersebut harus digelindingkan sedemikian rupa hingga bisa masuk ke dalam lubang berukuran  $1 \times 1$  ukuran luas yang merupakan tujuan dari permainan ini. Balok tersebut digelindingkan dalam suatu area permainan. Area permainan berupa papan yang terbentuk dari banyak ubin  $1 \times 1$  ukuran luas. Selama permainan balok harus selalu berada di atas area permainan, jika keluar atau jatuh dari area permainan maka dianggap gagal menyelesaikan permainan.

Salah satu cara yang bisa digunakan dalam mencari solusi permainan ini adalah dengan menggunakan algoritma BFS (Breadth First Search) dan DFS (Deep Search First). Kedua algoritma ini akan melakukan pencarian solusi pada simpul-simpul yang terbentuk dari kemungkinan langkah yang bisa dilakukan oleh balok. BFS akan mencari secara melebar pada pohon simpul yang terbentuk sedangkan DFS akan melakukan pencarian secara mendalam pada pohon simpul yang terbentuk. Solusi yang dicari adalah keadaan di mana balok berdiri dengan alas  $1 \times 1$  pada lubang  $1 \times 1$  yang menjadi tujuan dalam permainan ini.

**Kata kunci**—BFS, DFS, pencarian solusi, Rolling Block

## I. PENDAHULUAN

Rolling Block adalah permainan berjenis puzzle yang secara garis besar bertujuan untuk menemukan langkah-langkah yang tepat hingga balok yang digerakkan bisa masuk ke dalam lubang tujuan. Balok dalam permainan ini berukuran  $1 \times 1 \times 2$  satuan luas dan digerakkan dengan cara digelindingkan. Jika balok digelindingkan maka secara otomatis posisi balok terhadap alas permainan akan berubah, misalnya balok tadinya berdiri dengan sisi alas yang menyentuh alas permainan adalah sisi  $1 \times 1$  jika balok digelindingkan ke kanan maka sisi balok yang menjadi alas menjadi sisi  $2 \times 1$  dan posisi balok akan bergeser lebih ke kanan. Ilustrasi pergerakan ini bisa dilihat pada gambar 1.



Gambar 1. Ilustrasi pergerakan balok

Tujuan dari permainan ini adalah mencari langkah-langkah penggelindingan balok sehingga balok bisa masuk ke dalam lubang yang berukuran  $1 \times 1$  yang ada dalam area permainan. Sisi alas balok harus berada di atas alas permainan jika sebagian atau keseluruhan sisi alas balok berada di luar alas permainan maka balok akan jatuh dan dinyatakan gagal.



Gambar 2. Contoh tampilan permainan Rolling Block

Dengan permasalahan permainan seperti yang sudah dijelaskan, maka dapat dilihat bahwa algoritma BFS maupun DFS bisa digunakan sebagai pencari solusi permasalahan tersebut. Algoritma-algoritma ini bisa digunakan karena dalam permasalahan permainan Rolling Block langkah-langkah yang ada bisa digambarkan dalam suatu pohon ruang status. Dan dari langkah-langkah yang digambarkan dengan pohon ruang

status bisa dicari langkah yang merupakan solusi permainan dengan algoritma BFS ataupun DFS.

## II. METODE

### A. Pencarian Solusi dengan Algoritma BFS

Algoritma BFS, Breadth First Search, adalah salah satu algoritma traversal dalam graf. BFS akan melakukan pencarian yang dimulai dari simpul akar. Setelah itu menelusuri seluruh simpul tetangganya. Lalu menelusuri simpul yang belum dan dikunjungi yang bertetangga dengan simpul yang tadi dikunjungi, demikian seterusnya sampai ditemukan solusi yang dicari di suatu simpul atau sampai semua simpul yang ada sudah selesai ditelusuri.

Algoritma BFS ini memerlukan suatu antrian q untuk menyimpan simpul-simpul yang telah dikunjungi. Tujuannya agar simpul-simpul yang telah dikunjungi suatu saat bisa diketahui sebagai acuan untuk mengunjungi simpul-simpul yang bertetangga dengannya. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali.

Jadi dalam menerapkan algoritma BFS dalam pencarian solusi permainan Rolling Block sebelumnya harus dibentuk pohon ruang status. Pohon ini dibentuk dengan memeriksa ke empat arah kemungkinan balok bisa digelindingkan lalu dari kemungkinan-kemungkinan ini dibuat lagi kemungkinan arahnya yang akan menjadi simpul-simpul baru di bawah simpul sebelumnya, dan begitu seterusnya.

Agar pembuatan pohon menjadi lebih mudah, sebelumnya area permainan direpresentasikan menjadi sebuah papan (tabel) yang berisi keterangan permainan. K merupakan keterangan tidak ada alas permainan di mana posisi ini tidak bisa ditempati oleh balok, B merupakan keterangan di mana balok berada dan posisinya, jika hanya ada satu B maka berarti balok dalam posisi berdiri dengan alas 1x1, jika ada dua B yang berdampingan maka balok sedang dalam posisi tidur dengan alas sesuai keterangan B dalam tabel. T merupakan lubang tujuan. Balok harus berada pada posisi T dengan posisi berdiri (dengan alas balok 1x1) untuk menyelesaikan permainan ini.

y\x	1	2	3	4	5	6
1	K	K	K	K	K	K
2		B	K	K	K	K
3	K					K
4						
5	K			T		
6	K	K				K

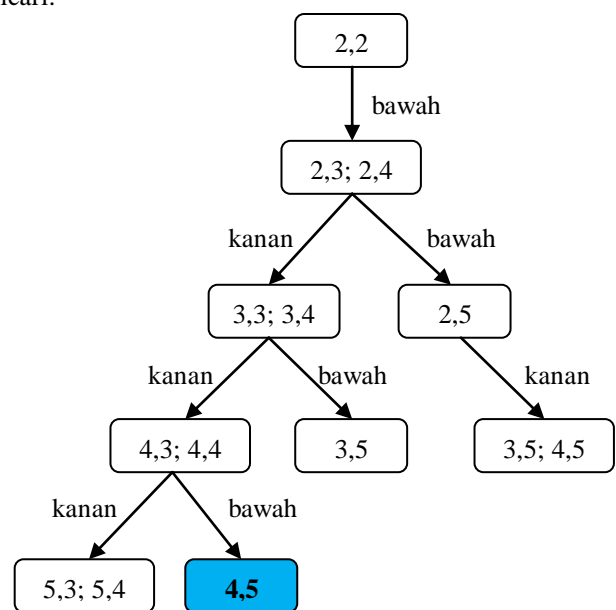
Gambar 3. Representasi area permainan dalam tabel

Ada beberapa elemen yang diperlukan dalam

penyelesaian menggunakan algoritma BFS yaitu antrian untuk menyimpan posisi dan matriks untuk menyimpan keterangan apakah suatu posisi sudah pernah dikunjungi atau belum. Algoritma BFS yang akan digunakan secara garis besar adalah sebagai berikut.

1. Masukkan posisi awal balok permainan pada antrian. Tandai posisi ini pada matriks sebagai posisi yang sudah dikunjungi. Jika posisi awal berhimpitan dengan posisi T maka solusi ditemukan.
2. Coba gelindingkan balok dengan posisi sebelumnya adalah simpul paling awal dalam antrian ke empat arah: atas, kanan, bawah, dan kiri. Hapus simpul ini dari antrian. Jika balok bisa digelindingkan dalam arti posisi setelah digelindingkan masih berada di alas permainan (kotak kosong dalam tabel) tidak berada di kotak K atau di luar tabel selain itu posisi setelah digelindingkan tidak boleh merupakan posisi yang sudah dikunjungi seperti dicatat dalam matriks, maka masukkan posisi balok setelah digelindingkan ke dalam antrian. Dan tandai posisinya pada matriks sebagai posisi yang sudah dikunjungi.
3. Ulangi pemeriksaan langkah 2 sampai posisi B (hanya satu B, alas 1x1, bukan BB, alas 2x1) berhimpitan dengan posisi T yang artinya solusi permainan sudah ditemukan atau sampai semua posisi yang mungkin sudah diperiksa.

Algoritma di atas jika diterapkan dalam pencarian solusi permainan Rolling Block akan membentuk sebuah pohon ruang status. Berikut ini adalah ilustrasi pohon ruang status yang terbentuk dengan algoritma yang telah dijelaskan. Status berwarna biru merupakan solusi yang dicari.



Gambar 4. Pohon yang terbentuk dengan algoritma BFS

Pada ilustrasi pohon di atas posisi balok yang tidak dimungkinkan tidak dituliskan begitu pun dengan posisi yang sudah pernah dikunjungi. Hal ini sudah dijelaskan pada langkah-langkah algoritma yang sudah dituliskan. Begitu posisi yang merupakan solusi ditemukan, algoritma menghentikan proses pembuatan pohon.

### B. Pencarian Solusi dengan Algoritma DFS

Algoritma DFS, Deep First Search, juga merupakan salah satu algoritma traversal dalam graf. DFS akan melakukan pencarian dimulai dari simpul akar. Lalu berikutnya akan dilakukan di simpul berikutnya yang merupakan simpul pertama yang bertetangga dengan simpul akar, lalu pencarian mendalam dimulai lagi secara rekursif dari simpul tersebut. Ketika mencapai simpul *s* sedemikian rupa sehingga semua simpul yang bertetangga dengannya telah dikunjungi, pencarian akan dirunut balik ke simpul terakhir yang dikunjungi sebelumnya mempunyai suatu simpul tetangga *t* yang belum dikunjungi. Pencarian mendalam akan dimulai lagi dari *t*. Pencarian akan berakhir bila tidak ada lagi simpul belum dikunjungi yang dapat dicapai dari simpul yang telah dikunjungi.

Algoritma DFS memerlukan tabel tambahan untuk menyimpan simpul-simpul yang telah dikunjungi. Tabel ini akan digunakan untuk merunut balik pada saat mencapai simpul *s* sedemikian rupa sehingga semua simpul yang bertetangga dengannya telah dikunjungi.

Seperti pada penerapan algoritma BFS, pada penerapan algoritma DFS juga dibutuhkan tabel yang merepresentasikan area permainan agar pembuatan pohon ruang status menjadi lebih mudah. Tabel yang merepresentasikan area permainan beserta seluruh keterangan yang diperlukan dalam permainan sama dengan tabel yang digunakan pada penerapan algoritma BFS, lihat Gambar 3.

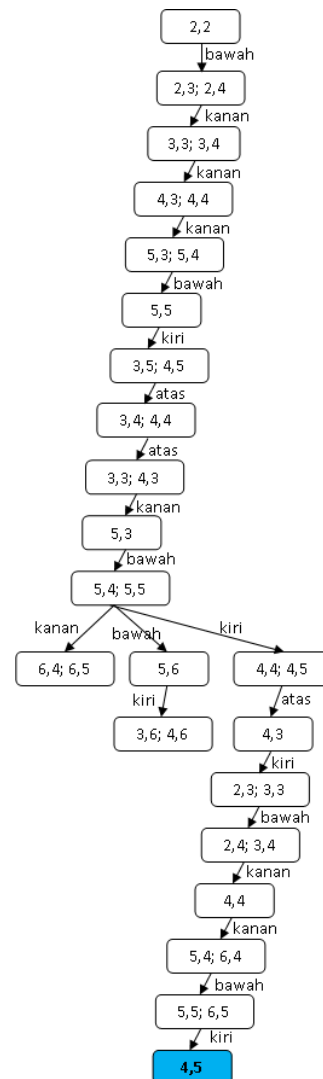
Ada beberapa elemen yang dibutuhkan pada penerapan algoritma DFS pada pencarian solusi permainan Rolling Block ini yaitu matriks sebagai pengganti tabel yang menyimpan simpul-simpul yang telah dikunjungi. Matriks ini akan mencatat posisi yang sudah dikunjungi. Lalu antrian untuk menyimpan langkah penyelesaian. Algoritma DFS yang akan diterapkan secara garis besar adalah sebagai berikut.

1. Masukkan posisi balok awal pada antrian. Jika posisi balok berhimpitan dengan T maka solusi ditemukan.
2. Untuk suatu arah (urutan arah yang dipilih adalah atas, kanan, bawah, lalu kiri) yang dipilih periksa apakah balok mungkin digelindingkan ke arah tersebut. Jika mungkin tandai posisi pada matriks sebagai posisi yang sudah dikunjungi. Tambahkan posisi balok yang baru digelindingkan ke dalam antrian. Apabila posisi balok yang baru

digelindingkan merupakan posisi yang sudah dikunjungi maka posisi tidak akan dimasukkan ke dalam antrian.

3. Jika sampai pada suatu simpul di mana dari keempat arah sudah tidak ada langkah yang memungkinkan dan balok belum sampai ke T, maka pemeriksaan mundur ke posisi pada antrian yang memiliki arah yang belum dicek. Posisi-posisi yang sudah diperiksa saat melakukan pemeriksaan mundur pada antrian dihapus dari antrian.
4. Ulangi pemeriksaan mulai langkah 2 sampai 3 secara rekursif sampai posisi B (hanya satu B dalam tabel, alas 1x1, bukan BB, alas 2x1) berhimpitan dengan posisi T yang artinya solusi permainan sudah ditemukan atau sampai semua posisi yang mungkin sudah diperiksa.

Algoritma di atas jika diterapkan dalam pencarian solusi permainan Rolling Block akan membentuk sebuah pohon ruang status. Berikut ini adalah ilustrasi pohon ruang status yang terbentuk dengan algoritma yang telah dijelaskan. Status berwarna biru merupakan solusi yang dicari.



Gambar 4. Pohon yang terbentuk dengan algoritma DFS

Pada ilustrasi pohon di atas posisi balok yang tidak dimungkinkan tidak dituliskan begitu pun dengan posisi yang sudah pernah dikunjungi. Hal ini sudah dijelaskan pada langkah-langkah algoritma yang sudah dituliskan. Begitu posisi yang merupakan solusi ditemukan, algoritma menghentikan proses pembuatan pohon.

### III. ANALISIS

#### A. Analisis Pencarian Solusi dengan Algoritma BFS

Pencarian solusi permainan Rolling Block menggunakan algoritma BFS yang diterapkan pada pohon ruang status yang dibentuk mempunyai rata-rata performansi yang baik. Penggunaan algoritma ini akan selalu menghasilkan langkah-langkah solusi yang merupakan langkah-langkah paling pendek untuk mendapatkan solusi.

Langkah-langkah solusi yang dihasilkan algoritma BFS merupakan langkah paling pendek yang mungkin karena algoritma BFS akan melakukan pencarian ke semua cabang penyelesaian terlebih dahulu sebelum melanjutkan mencari di cabang-cabang penyelesaian dari cabang penyelesaian yang dia masuki. Algoritma BFS akan melakukan pencarian seperti itu sampai menemukan solusi yang dicari. Jadi algoritma BFS akan menemukan solusi di cabang terendah yang digambarkan dengan jumlah langkah paling sedikit yang mungkin.

Akan tetapi jika dilihat dari rata-rata kasus permainan Rolling Block yang umum yaitu posisi balok berada di bagian kiri area permainan dan posisi lubang tujuan berada di bagian kanan area permainan, dan jarak dari ujung kiri ke ujung kanan permainan biasanya jauh, maka performansi algoritma BFS tidak terlalu bagus untuk diterapkan karena algoritma BFS akan melakukan pencarian secara melebar pada semua kemungkinan langkah yang bisa diambil dari awal. Cara ini tidak efektif dan efisien karena banyak usaha pencarian yang dilakukan sebelum menemukan solusi.

Tetapi kasus ini hanya untuk kondisi umum di mana letak lubang tujuan dan letak balok sudah bisa diprediksi. Walau begitu algoritma BFS akan tetap menghasilkan langkah-langkah solusi yang merupakan langkah terpendek yang mungkin dalam mendapatkan solusi.

#### B. Analisis Pencarian Solusi dengan Algoritma DFS

Pencarian solusi permainan Rolling Block menggunakan algoritma DFS yang diterapkan pada pohon ruang status yang dibentuk mempunyai performansi yang baik apabila lubang tujuan berada di atas kanan posisi balok dan akan mempunyai

performansi yang kurang bagus apabila lubang tujuan berada di kiri bawah posisi balok. Jadi performansinya tergantung pada letak solusi pada pohon ruang status. Penggunaan algoritma ini secara rata-rata menemukan solusi dari pohon ruang status dengan cepat.

Performansi algoritma DFS akan bagus apabila lubang tujuan dalam permainan ini berada di sebelah atas kanan karena kemungkinan langkah yang dicari pertama kali adalah kemungkinan langkah ke atas kemudian ke kanan baru ke bawah dan terakhir ke kiri. Dan algoritma DFS adalah algoritma yang akan mencari secara mendalam pada simpul tetangga pertama dari akar yaitu simpul langkah yang mengarah ke atas pada pohon ruang status permainan ini. Pencarian secara mendalam pada simpul-simpul yang merepresentasikan langkah gerakan ke kiri dilakukan terakhir kali setelah semua kemungkinan langkah pada simpul-simpul sebelumnya yaitu simpul-simpul langkah yang mengarah ke atas, kanan, dan bawah selesai diperiksa. Jadi untuk kasus lubang tujuan ada di sebelah kiri posisi balok performansi yang ditunjukkan algoritma ini tidak terlalu baik.

Pencarian secara rata-rata cepat dibanding algoritma lain karena secara rata-rata lubang tujuan dalam permainan ini letaknya berjauhan dari posisi awal balok, rata-rata kondisi awal dalam permainan ini adalah posisi awal balok ada di bagian ujung kiri area permainan dan posisi lubang tujuan berada di ujung sebelah kanan area permainan. Pencarian secara rata-rata cepat karena selain letak lubang tujuan di kanan adalah karena pencarian secara mendalam pada suatu simpul sehingga untuk kasus solusi yang dicari berada jauh di bagian bawah pohon ruang status pencarian akan lebih baik jika dilakukan pencarian yang langsung mendalam pada suatu simpul dan rata-rata langkah yang menuju lubang tujuan berada di bagian simpul-simpul pertama tetangga akar (posisi awal balok), yaitu simpul-simpul yang mengarah atas kanan.

#### C. Perbandingan Pencarian Solusi Menggunakan Algoritma BFS dengan Algoritma DFS

Dalam pencarian langkah-langkah solusi permainan Rolling Block seperti sudah dijelaskan sebelumnya bisa dipakai algoritma BFS dan juga DFS. Kedua algoritma memiliki pendekatan yang berbeda dalam melakukan pencarian dalam pohon ruang status. Algoritma BFS melakukan pencarian secara melebar sedangkan algoritma DFS melakukan pencarian secara mendalam.

Kedua pendekatan berbeda yang dilakukan kedua algoritma ini juga mempunyai kelebihan dan kekurangan masing-masing. Algoritma BFS mempunyai kelebihan seperti akan selalu menghasilkan langkah-langkah solusi yang paling pendek dari semua kemungkinan langkah-langkah solusi lain yang mungkin, algoritma DFS tidak selalu menemukan solusi dengan langkah-langkah terpendek karena solusi yang dihasilkan algoritma DFS

adalah kemungkinan solusi pertama yang didapat dari cabang paling awal yang mempunyai simpul solusi sehingga tidak selalu mendapat solusi terpendek.

Pencarian rata-rata yang dilakukan algoritma DFS lebih cepat dibanding algoritma BFS karena algoritma DFS langsung mencari secara mendalam di suatu cabang penyelesaian. Hal ini akan menyebabkan solusi lebih cepat ditemukan walaupun letak solusi tidak bisa dijamin sebagai solusi terbaik dan bisa saja merupakan solusi terburuk yaitu solusi terpanjang.

Lalu algoritma BFS akan lebih baik apabila simpul solusi berada di bagian-bagian atas dari pohon ruang status yang dibentuk. Sedangkan algoritma DFS akan lebih baik apabila simpul solusi berada di bagian kiri dari pohon ruang status yang dibentuk. Jadi dalam permainan Rolling Block ini algoritma BFS akan lebih baik apabila posisi lubang tujuan berdekatan dengan posisi awal balok, walau kondisi posisi lubang di sebelah atas, kanan, bawah, maupun kiri dari posisi awal balok, performa algoritma BFS akan lebih baik jika posisi lubang tujuan berdekatan dengan posisi awal balok. Algoritma DFS akan lebih baik jika posisi lubang tujuan berada di sebelah atas kanan dari posisi awal balok, walau posisi lubang tujuan dengan posisi awal balok berjauhan, performa algoritma DFS akan lebih baik jika posisi lubang tujuan berada di sebelah atas kanan dari posisi awal balok.

#### IV. KESIMPULAN

Permainan Rolling Block adalah sebuah permainan yang berjenis *puzzle* yang langkah-langkah solusinya bisa dicari dengan menggunakan algoritma BFS (Breadth First Search) dan DFS (Deep First Search). Kedua algoritma ini mempunyai pendekatan pencarian solusi yang berbeda, BFS mencari secara melebar sedangkan DFS mencari secara mendalam. Algoritma BFS dan DFS mempunyai kelebihan dan kekurangan masing-masing dalam melakukan pencarian pada berbagai kondisi yang dimungkinkan dalam permainan Rolling Block

#### REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika, 2009.
- [2] <http://www.freeonlinegames.com/game/block-n-roll.html>  
Waktu akses: Selasa, 7 Desember 2010 pukul 06.15
- [3] <http://www.simplegamereviews.com/>  
Waktu akses: Selasa, 7 Desember 2010 pukul 07.08
- [4] <http://www.cse.ohio-state.edu/~gurari/course/cis680/cis680Ch14.html>  
Waktu akses: Selasa, 7 Desember 2010 pukul 07.08
- [5] <http://www.crypto.cs.mcgill.ca/~crepeau/CS250/2004/28.BFS1.pdf>  
Waktu akses: Selasa, 7 Desember 2010 pukul 07.08
- [6] <http://www.ics.uci.edu/~epstein/161/960215.html>  
Waktu akses: Selasa, 7 Desember 2010 pukul 07.08

#### 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, 7 Desember 2010



Zakiy Firdaus Alfikri  
13508042