

PENERAPAN ALGORITMA RUNUT BALIK DALAM GERAK ANIMASI 3D

Muara Sipahutar¹, Reinhard Sibarani²

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if13064@students.if.itb.ac.id¹, if130128@students.if.itb.ac.id²

Abstrak

Animasi dalam bentuk 3D(3Dimension) telah populer pada zaman sekarang, khususnya dalam bidang 3D Gaming. Setiap karakter bergerak yang diciptakan pada suatu game, memiliki animasi gerak dan tentunya menerapkan algoritma tertentu untuk menentukan jenis pergerakan karakter tersebut. Salah satu algoritma yang cukup berpengaruh dalam gerak (*motion*) dari setiap karakter tersebut ialah gerak runut balik(*backtrack*). Setiap 3D engine mengimplementasikan algoritma ini. Metode runut balik juga memungkinkan karakter tersebut bergerak ke tempatnya semula tanpa adanya *direct command* (perintah langsung dari user). Selain itu metode ini juga bisa membatasi pergerakan karakter. Beberapa contohnya ialah pantulan bola pada dinding atau lantai, adanya *wall* yang membatasi pergerakan karakter, dan sebagainya.

Kata kunci: animasi, *backtrack*, 3D, 3D Engine, *motion*

1. Pendahuluan

Animasi game sudah tidak asing lagi bagi masyarakat zaman sekarang ini. Semakin hari perkembangan game semakin pesat. Mulai dari game yang berbasis 2D dan akhirnya sekarang game 3D telah menjadi populer. Tentunya game-game tersebut telah menggunakan teknik-teknik animasi yang cukup kompleks. Pembuatan navigasi gerakan setiap karakter pun menjadi lebih kompleks, dengan adanya variasi gerakan pada karakter tersebut.

Tentu saja gerakan-gerakan tersebut mengimplementasikan berbagai algoritma agar *motion* karakter sesuai dengan environmentnya dan tentu saja hal tersebut merupakan hal yang cukup sulit. Untuk mempermudah dalam pembuatan animasi tersebut, maka dibuatlah 3D engine, contohnya 3D Max atau Maya. Salah satu algoritma yang berperan utama dalam engine ini adalah algoritma runut balik. Dimana animasi pementalan bola di lantai mengimplementasikan algoritma ini. Dan banyak contoh lain yang membuktikan algoritma runut balik ini sangat dibutuhkan di dalam dunia animasi.

2 Algoritma Runut Balik (Backtracking)

2.1 Pengertian Runut Balik (Backtracking)

Runut Balik (*backtracking*) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan lebih mangkus. Runut balik, yang merupakan perbaikan dari algoritma *brute force*, secara sistematis mencari solusi persoalan di antara semua

kemungkinan solusi yang ada. Namun hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan, sehingga waktu pencarian pun dapat dihemat.

Algoritma *runut balik* ini memperbaiki pencarian solusi secara *exhaustive search* dengan mencari solusi secara sistematis. Untuk memfasilitasi pencarian ini, maka ruang solusi diorganisasikan ke dalam struktur pohon, dimana tiap simpul pohon menyatakan status (*state*) persoalan dan sisi (cabang) dilabeli dengan nilai-nilai xi. Lintasan dari akar ke daun menyatakan solusi-solusi yang mungkin, dan seluruh lintasan dari akar ke daun membentuk ruang solusi. Pengorganisasian pohon ruang solusi diacu sebagai pohon ruang status (*state space tree*).

2.2 Prinsip Pencarian Solusi dengan Metode Runut Balik

Langkah-langkah pencarian solusi dengan metode Runut Balik adalah sebagai berikut:

1. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan yang dipakai adalah mengikuti metode pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan simpul hidup(*live node*), dan simpul hidup yang sedang diperluas dinamakan simpul-E(*Expand node*). Simpul dinomori dari atas ke bawah sesuai dengan kelahirannya.

2. Jika lintasan yang diperluas yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut “dibunuh” sehingga menjadi simpul mati (*dead node*). Simpul yang sudah mati ini tidak akan diperluas lagi.

3. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak lainnya. Bila tidak ada lagi simpul anak yang dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut-balik ke simpul hidup terdekat. Selanjutnya simpul ini menjadi simpul-E yang terbaru.

4. Pencarian dihentikan bila telah ditemukan solusi atau tidak ada lagi simpul hidup untuk runut balik.

2.3 Contoh Runut Balik (Backtracking)

Ada beberapa permasalahan komputasi yang dapat diselesaikan dengan penerapan Algoritma Backtracking ini, diantaranya adalah sebagai berikut:

1. Pewarnaan Graf (*Graph Colouring*)

Masalah pewarnaan graf dapat dibedakan dalam 2 macam yakni pewarnaan simpul dan pewarnaan sisi. Dan penggunaan algoritma *Backtracking* pada Pewarnaan Graf ini membutuhkan waktu asimtotik dalam $O(nm^n)$

2. Sirkuit Hamilton

Sirkuit Hamilton adalah bentuk perjalanan yang mengunjungi semua simpul tepat satu kali dan kembali lagi ke simpul awal.

Banyak contoh persoalan yang meminta menentukan sirkuit Hamilton, diantaranya yang paling terkenal adalah *Traevelling Saleaman Problem*, yakni persoalan mencari sirkuit Hmliton yang mempunyai total bobot minimum.

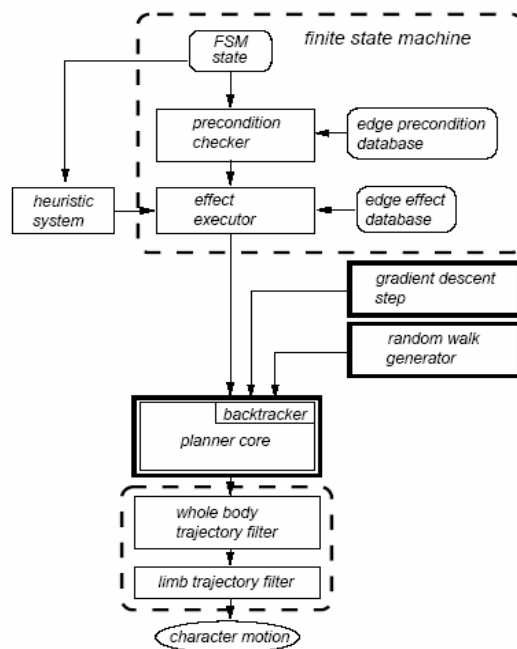
Dan pada tulisan ini, penulis ingin menjelaskan pengaruh algoritma backtracking di dalam dunia Animasi 3D khususnya di dalam penerapan algoritma backtracking pada karakter-karakter bergerak. Dimana di dalam gerak animasi 3D ini, algoritma backtracking dipergunakan untuk membatasi motion suatu karakter dengan *environmentnya* agar karakter tersebut tidak dapat menembus suatu *environment*. Sehingga karakter secara sekilas terlihat diam, namun sebenarnya karakter tersebut melakukan gerakan *backtrack* berulang-ulang dengan delta jarak yang sangat kecil antara karakter dan *environmentnya*.

3. Algoritma dalam Animasi 3D

3.1. Motion Planner

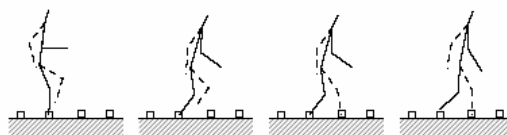
Untuk merealisasikan berbagai gerak dalam animasi 3D, perlu direncanakan metode-metode motion yang digunakan.

Animasi gerakan dalam 3D dapat dibagi dalam beberapa bagian sesuai dengan keterangan skema di bawah ini.



Kali ini yang dibahas adalah backtracking method pada animasi 3D.

Setiap motion dalam animasi memiliki koordinat. Koordinat inilah yang menunjukkan bahwa suatu karakter berpindah dari satu lokasi ke lokasi lainnya. Misalnya suatu karakter berpindah dari koordinat (16, 5) ke (18, 5), dari (18,5) ke (22,5), dan seterusnya akan terlihat karakter bergerak (berpindah tempat).



Beginilah prinsip dasar algoritma dalam animasi. Dengan adanya perubahan dimensi menjadi panjang-lebar-tinggi, maka akan terbentuk animasi 3D.

3. 2. Algoritma Runut Balik dalam Animasi 3D

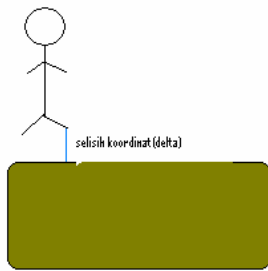
Pada penjelasan sebelumnya tentang algoritma runut balik telah dikatakan bahwa prinsip dasar runut balik adalah “kembali ke simpul atau posisi semula”. Begitu juga runut balik pada animasi 3D, koordinat karakter juga kembali ke posisi sebelumnya.

Pemahaman runut balik dalam gerak berpindah biasa memang telah jelas adanya, namun sebenarnya pemakaian algoritma ini bukan hanya terdapat pada

motion biasa saja. Ketika *environment* karakter tercipta (misalnya dinding, tanah, dan sebagainya), dan karakter berinteraksi di dalamnya, maka algoritma runut balik ini dipakai *engine 3D* untuk merealisasikannya menjadi lebih hidup.



Terlihat pada gambar di atas, karakter yang berupa manusia berinteraksi dengan environmentnya yaitu tanah dan cembungan dinding atas. Yang membuat karakter tersebut dapat berdiri di atas tanah ialah algoritma runut balik antara koordinat tanah dan koordinat kaki karakter. Sebenarnya *3D engine* yang memakai algoritma ini menghitung delta jarak koordinat yang sangat kecil sehingga karakter tersebut memantul-mantul dalam kondisi yang sangat kecil sekali (deltanya mendekati 0) sehingga tidak dapat dilihat mata secara langsung.



Gbr : Selisih koordinat yang sangat kecil

Beberapa *3D engine* tidak membuat jarak ini dengan sangat kecil sehingga karakternya tampak seperti melayang di atas tanah. Efek yang kurang baik ini sebenarnya juga bisa dimanipulasi sehingga karakternya terlihat benar-benar melayang.



Karakter diatas kelihatan melayang, hal ini disebabkan karena adanya delta yang cukup besar antara koordinat tanah dan karakter itu sendiri.

Begitu juga hal ini terjadi ketika karakter bertemu dinding(*wall*). Tentu saja sesuai aturan permainannya, karakter tersebut tidak dapat menembus dinding. Hal ini dimungkinkan karena koordinat dinding dan koordinat karakter dibuat untuk saling backtrack ketika bertemu sehingga karakter tidak mungkin untuk melangkah ke koordinat di luar dinding.

4. Kesimpulan

Algoritma runut balik merupakan salah satu algoritma yang memiliki peranan penting di dalam perkembangan pembuatan Animasi 3D.

Daftar Pustaka

1. Kalisiak, Maciej. 2001. "A grasp base motion planning Algorithm for character planning"
2. Munir, Ir.Rinaldi. 2005 . "Strategi Algoritmik". Lab. Ilmu Rekayasa Komputasi Departemen T. Informatika ITB.
3. Rolfe. Timothy J.. 2002. "An alternative Problem for Backtracking and Bounding"