

Metode Pathfinding pada map 2D Menggunakan Algoritma Diagonal & Bidirectional BFS

IF3051 Strategi Algoritma

Fitra Haryadi 13509093¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

¹13509093@std.stei.itb.ac.id

Abstrak—Pathfinding atau pathing adalah metode untuk mencari jalur atau menemukan rute terpendek antara dua titik yang akan dilalui. Pathfinding dibutuhkan pada game 2D/3D terutama game yang menggunakan map sebagai tempat player bernavigasi. Pada makalah ini akan dijelaskan lebih lanjut tentang metode pathfinding menggunakan algoritma BFS dengan pencarian diagonal dan bidirectional yang mempersingkat waktu pencarian dengan signifikan.

Kata Kunci—Pathfinding, BFS, diagonal, bidirectional, dan path.

I. PENDAHULUAN

Pathfinding dalam konteks games adalah cara menggerakkan pemain melalui suatu jalan terpendek tanpa menabrak rintangan. Penggunaan metode pathfinding paling sering adalah pada game real-time (berorientasi waktu nyata) dimana seorang pemain pada titik awal menentukan titik akhir/tujuan.

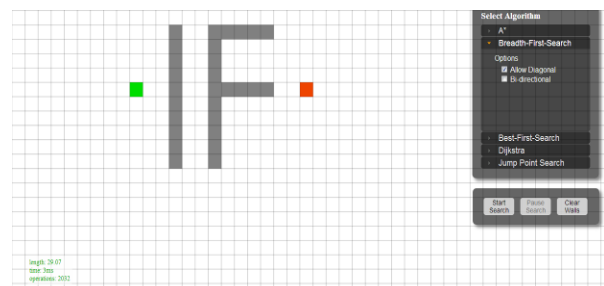
Pathfinding saat ini telah menjadi elemen penting di dunia games. Dimana semua games 2D/3D pasti memiliki metode pathfinding yang berbeda. Sesuai dengan tingkat kerumitan dari games 2D/3D yang diusung. Semakin rumit map dari suatu game maka semakin rumit pula metode pathfinding yang digunakan.

Pada permainan ber-genre real-time strategy seperti Dota dan Dota 2, metode pathfinding merupakan metode yang otomatis dijalankan setiap kali pemain berpindah tempat. Pemain sangat dimudahkan, karena ia tidak perlu mengatur pergerakan karakter. Pemain hanya perlu menentukan titik akhir/final yang akan dituju. Sedangkan titik start diperoleh dari posisi karakter pemain yang terkini.

Pada kesempatan kali ini, penulis tidak akan membahas semua metode pathfinding yang digunakan di dunia games 2D/3D. penulis hanya akan membahas dua metode pathfinding yang digunakan di dunia games pada umumnya, yaitu pathfinding dengan algoritma BFS dan A* yang dapat menelusuri map secara diagonal dan bi-directional, sehingga sangat mempercepat proses pathing

dalam games.

Metode pathfinding dibahas pada makalah ini menggunakan map yang tersusun dari kumpulan balok/tiles yang memiliki empat komponen utama. Komponen utama itu adalah titik awal/start, titik akhir/finish, penghalang/dinding/wall, dan path. Path merupakan jalur yang dihasilkan setelah pathfinding dijalankan.



Gambar 1. Ilustrasi map yang digunakan.

II. ALGORITMA BFS

Pada sistem graph, breadth-first search (BFS) adalah strategi pencarian node tertentu pada graph dimana pencarian dibatasi hanya pada dua operasi, yaitu : (a) kunjungi dan periksa/tandai sebuah node pada graph; (b) kunjungi dan periksa node tetangga yang belum pernah dikunjungi.

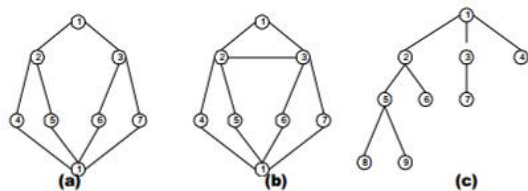
Misalkan BFS dimulai dari node akar/ node paling tinggi pada graph kemudian menelusuri semua tetangga yang belum pernah dikunjungi dari node akar. Setelah itu tetangga dari setiap tetangga dari node akar akan dikunjungi dst. Pencarian akan berhenti apabila node yang dicari telah ditemukan atau pencarian telah mencapai semua node daun dari graph yang digunakan.

Algoritma BFS menggunakan antrian/queue sebagai struktur data yang digunakan untuk menyimpan hasil/node yang telah dikunjungi. Berikut contoh penggunaan struktur data antrian pada BFS:

1. Masukkan node akar pada antrian

2. Keluarkan node dari antrian dan periksa
 - Jika elemen yang dicari ditemukan pada node ini, maka hentikan pencarian dan kembalikan hasil
 - Jika elemen tidak ditemukan, maka masukkan node tetangga pada antrian dan teruskan pencarian
3. Jika antrian kosong ,setiap node telah ditelusuri, kembalikan hasil tidak ditemukan
4. Jika antrian tidak kosong maka ulangi dari langkah dua.

Contoh 1: (misalkan traversal dimulai dari simpul 1)



Gambar (a) BFS(1): 1, 2, 3, 4, 5, 6, 7, 8.

Gambar (b) BFS(1): 1, 2, 3, 4, 5, 6, 7, 8

Gambar (c) BFS(1): 1, 2, 3, 4, 5, 6, 7, 8, 9

Gambar 2. Contoh pencarian BFS pada tiga buah graph

Pseudocode atau kode semu adalah deskripsi informal tingkat tinggi dari suatu program komputer atau algoritma yang mendeskripsikan algoritma tersebut dengan sederhana. Berikut merupakan pseudocode dari algoritma BFS.

```

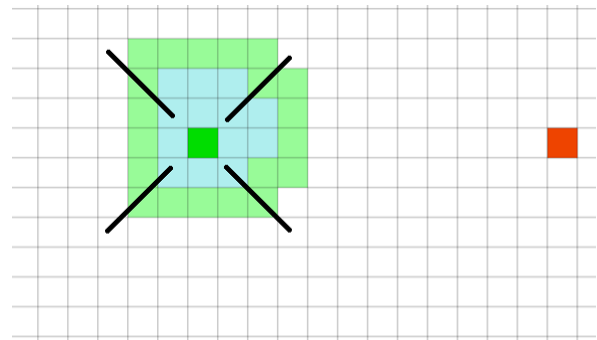
1  procedure BFS( $G, v$ ) :
2      create a queue  $Q$ 
3      enqueue  $v$  onto  $Q$ 
4      mark  $v$ 
5      while  $Q$  is not empty:
6           $t \leftarrow Q.dequeue()$ 
7          if  $t$  is what we are
looking for:
8              return  $t$ 
9          for all edges  $e$  in
 $G.adjacentEdges(t)$  do
12              $u \leftarrow$ 
 $G.adjacentVertex(t, e)$ 
13             if  $u$  is not marked:
14                 mark  $u$ 
15                 enqueue  $u$  onto

```

Gambar 3. Pseudocode dari algoritma BFS

III. ALGORITMA BFS DIAGONAL & BI-DIRECTIONAL

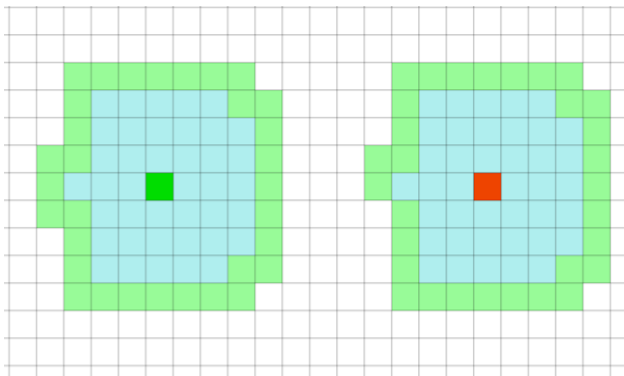
Implementasi algoritma BFS untuk pathfinding memiliki banyak variasi pencarian. Variasi pencarian paling sederhana yaitu pencarian ke arah kanan-kiri dan atas-bawah yang berakhir pada titik yang dicari atau target. Variasi selanjutnya adalah menggunakan metode pencarian secara diagonal yang tentunya dapat diimplementasikan di map. Metode ini sangat berguna apabila range pencarian memiliki ukuran yang sangat besar.



Gambar 3. Pencarian BFS dengan diagonal search

Gambar di atas menunjukkan metode pencarian BFS dengan memanfaatkan diagonal search. Diagonal search memanfaatkan semua elemen yang berada di sekitar titik hijau / titik start untuk mencari titik finish. Diagonal search berarti memasukkan elemen yg berada di garis diagonal dari titik start kedalam antrian.

Metode lainnya yang membantu mempercepat waktu pathfinding adalah bidirectional search. Dimana bidirectional search adalah pencarian dua arah yang menemukan jalur terpendek dari titik start ke titik finish. Metode ini merupakan metode yang bersifat dua pencarian simultan dimana titik start melakukan pencarian maju dari tujuan dan titik finish melakukan pencarian mundur. Kemudian kedua antrian pencarian ini akan bertemu di tengah-tengah. Alasan menggunakan metode ini adalah untuk memotong waktu pencarian hingga menjadi setengah dari waktu sebelumnya. Penjelasan dengan model pencarian masalah yang telah disederhanakan adalah kedua titik membuat sebuah expand tree dengan percabangan b dan d sebagai jarak dari titik start ke titik finish. Dimana setiap pencarian expand tree tersebut memiliki kompleksitas $O(b^{d/2})$ dan jumlah kompleksitas dari kedua expand tree ini akan memiliki nilai kurang dari $O(b^d)$ yang merupakan nilai kompleksitas pencarian tanpa bidirectional search.

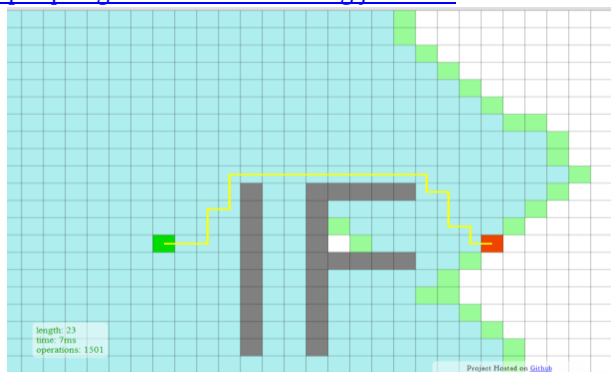


Gambar 4. Pathfinding diagonal dan bidirectional search.

Gambar diatas mendemonstrasikan pathfinding dengan metode diagonal dan bidirectional search yang mempercepat waktu pencarian. Dimana perbandingan waktu pencarian akan dilakukan pada analisis.

IV. ANALISIS

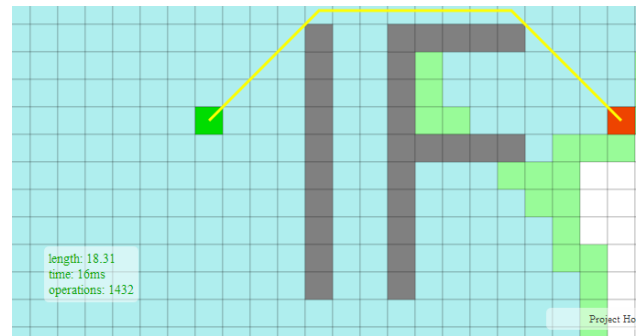
Pathfinding merupakan metode pencarian rute terpendek dan tercepat di antara dua titik, start dan finish, yang memanfaatkan berbagai metode untuk menghemat waktu pencarian. Semakin pendek dan cepat rute yang ditemukan maka semakin baik pula metode pathfinding yang digunakan. Metode pathfinding menggunakan diagonal dan bidirectional search diharapkan dapat mempercepat waktu pencarian. Berikut perbandingan panjang rute, waktu, dan jumlah operasi yang dilakukan oleh pathfinding diagonal search dan pathfinding diagonal-bidirectional search. Program yang digunakan adalah program javascript pathfinding demo milik Xueqiao Xu yang dapat ditemukan di <http://qiao.github.com/PathFinding.js/visual/>.



Gambar 5. Percobaan BFS.

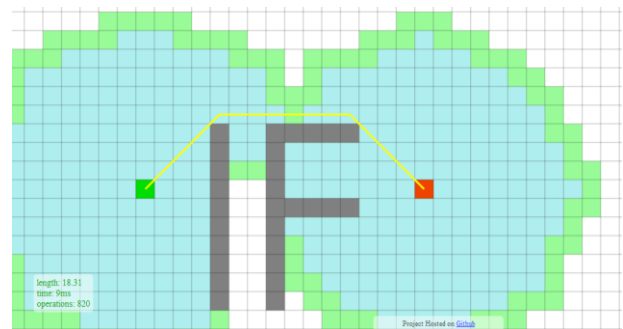
Berikut adalah percobaan pathfinding dengan algoritma BFS tanpa diagonal dan bidirectional search. Hasil dari percobaan di atas adalah panjang rute 23 kotak, waktu 7 ms, dan jumlah operasi yang dilakukan adalah 1501. Pada percobaan di atas dapat dibuktikan bahwa rute yang digunakan untuk pathfinding hanya bersifat ke kanan dan atas-bawah. Rute ini tidak memanfaatkan kotak diagonal sebagai antrian pencarian yang ditelusuri. Sedangkan waktu pencarian dihitung dari dimulainya pencarian dari titik start ke titik finish. Faktor yang dapat mempengaruhi

waktu pencarian adalah jumlah operasi, dimana jumlah operasi adalah jumlah node tetangga dan node yang telah dikunjungi.



Gambar 6. Percobaan diagonal BFS

Berikut adalah percobaan pathfinding dengan algoritma diagonal BFS dengan penghalang. Hasil dari percobaan ini adalah panjang rute 18.31 kotak, waktu 16ms, dan jumlah operasi yang dilakukan adalah 1432. Pada percobaan di atas dapat dibuktikan bahwa diagonal search sangat membantu untuk menemukan rute tercepat dimana rute yang ditemui bernilai 4,69 kotak lebih pendek atau 20.4 % lebih kecil daripada pathfinding dengan BFS, dimana panjang rute dinilai dari pusat kotak start ke pusat kotak finish. Sedangkan waktu pencarian menjadi jauh lebih lama yaitu menjadi 2.3 kalinya dari waktu semula. Akan tetapi, jumlah operasi yang dilakukan lebih sedikit yaitu 1432.



Gambar 7. Percobaan diagonal-bidirectional BFS

Berikut adalah percobaan pathfinding dengan algoritma diagonal & bidirectional BFS. Hasil dari percobaan di atas adalah panjang rute 18,31, waktu 9ms, dan jumlah operasi yang dilakukan adalah 820. Pada percobaan di atas dapat dibuktikan bahwa diagonal dan bidirectional membuat pathfinding lebih efisien. Dimana rute yang dihasilkan 20,4 % lebih pendek dan operasi yang dilakukan 681 lebih sedikit.

V. KESIMPULAN

BFS adalah algoritma yang sangat baik dalam menemukan rute terpendek pada map 2D karena selalu berhasil menemukan rute terpendek yang sebenarnya. Walaupun BFS melakukan banyak sekali operasi, tetapi hal tersebut dapat dibantu

dengan diagonal search yang memperpendek rute pencarian dan bidirectional search yang menghemat operasi yang dilakukan.

REFERENCES

- [1] Munir, Rinaldi. Diktat Kuliah IF3051 Strategi Algoritma. Bandung: STEI ITB.
- [2] http://www.youtube.com/watch?v=DINCL5cd_w0
diakses pada 13.00 tanggal 21 Desember 2012
- [3] <http://qiao.github.com/PathFinding.js/visual/>
Diakses pada 13.00 tanggal 21 Desember 2012

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, 15 Desember 2012



Fitra Haryadi
13509093