

KECERDASAN BUATAN DALAM PERMAINAN SNAKE 3D MENGGUNAKAN VISUAL BASIC .NET DAN DIRECTX

Aghus Sofwan
R.Rizal Isnanto
Pasca Maulana

Abstract

Snake3D is a simple game where a snake searches for food and its tail will grow long after getting the food. The goal of the research of this game is to make an artificial intelligence of Snake3D for developing a game using DirectX 3D technology as an API which most programmers use to create a game in Windows platform.

The Methods used in designing this game is UML (Unified Modeling Language). Development of Snake3D game software is using Visual Basic .NET as programming language and DirectX API's as multimedia interface. Artificial intelligence used in this 3D game Snake is A Star (A*) Algorithm. This algorithm is to search a shortest path from the beginning until reach goal. Applying the A Star (A*) Algorithm in this game is to get the food that are placed randomly and the computer will find the path to get it with shortest path to get.

From the results of the testing and analysis it can be concluded that the computer in this game which had been tested is very effective in getting food by using A Star (A*) Algorithm with the shortest path has to get it. The Percentage to get food from shortest path is 100% for the path that which had been pass through but computer does not always win the game, because the computer does not take into account when the food can get imprisoned by their own body, the body of the opponent players which blocks it, or entering the blocked path which has no way out.

Keyword : Artificial Intelligence, Algorithm A Star (A*), Visual Basic .NET, DirectX.

Permainan ular merupakan permainan populer dalam telepon selular beberapa tahun yang lalu. Inti dari permainan ini adalah agar ular yang kita kontrol mendapatkan sebanyak-banyaknya makanan tanpa membentur dinding atau bagian tubuhnya sendiri. Semakin banyak makanan yang ular dapatkan, tubuhnya akan tumbuh sehingga semakin panjang.

Permainan ini sebenarnya merupakan permainan pemain tunggal (*single player game*) atau permainan yang dimainkan sendiri. Dalam permainan ini komputer hanya memunculkan makanan secara acak (*random*) di layar untuk dimakan. Kesuksesan permainan ini bergantung kepada kecepatan dan perhitungan sang pemain agar ular yang dikontrolnya tidak terjebak dinding atau bagian tubuhnya sendiri. Kelakuan seperti inilah yang ingin dicoba dalam permainan ular ini dengan memunculkan karakter kecerdasan buatan komputer pada lawan pemain yang mampu berlaku sebagai pemain umumnya.

Algoritma yang dipilih untuk memodelkan kelakuan ular ini adalah Algoritma A Star (A*) yang merupakan algoritma pencarian jalur terpendek terbaik menuju tujuan.

Algoritma ini sangat cocok karena algoritma ini mendukung perhitungan untuk mencari jalan terpendek menuju makanan sementara menghindari bertabrakan dengan dinding atau bagian tubuhnya atau membuat dirinya sendiri terkurung.

Tujuan dari Tugas Akhir ini adalah untuk membuat kecerdasan buatan menggunakan Algoritma A Star (A*) dalam permainan Snake 3D.

Algoritma A Star (A*)

Algoritma A Star (A*) adalah algoritma pencarian terbaik dalam mencari jalur terpendek dengan perhitungan terkecil pada jalur dengan simpul awal menuju simpul akhir. Algoritma ini pertama kali dideskripsikan pada tahun 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Dalam makalah mereka, disebut dengan Algoritma A. Lalu dengan optimasi heuristik, disebut dengan A Star (A*).

Perhitungan pada Algoritma A Star (A*) dapat ditentukan sebagai berikut :

$$F(x) = G(x) + H(x)$$

Dimana :

- G(x) adalah nilai pada pergerakan simpul awal menuju simpul berikutnya.
- H(x) adalah perkiraan nilai pergerakan simpul awal menuju tujuan akhir simpul. Fungsi ini seringkali disebut dengan fungsi heuristik,

dinamakan heuristik karena perhitungan tersebut berdasarkan perkiraan (guess).

- c. $F(x)$ adalah jumlah nilai dari fungsi $G(x)$ dan $H(x)$. dengan nilai terkecil $F(x)$ adalah jalur terpendek menuju tujuan akhir.

Terdapat ketentuan pada grafik agar algoritma A Star (A^*) ini bila diterapkan akan selalu mendapatkan jalan yang terpendek. ketentuan tersebut yang harus dipenuhi pada grafik yaitu.

- 1) Setiap simpul (node) dalam grafik memiliki jumlah terbatas pada area pencariannya.
- 2) Pada pencarian terdapat jalan yang dilalui untuk mencapai tujuan.
- 3) Fungsi $F(x)$ pada grafik bernilai rendah daripada fungsi $F(x)$ pada pencarian sebelumnya

Visual Basic .NET

Visual Basic .NET merupakan bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual.

Keunggulan bahasa pemrograman ini terletak pada

1. Produktivitas,
2. Kualitas
3. Pengembangan perangkat lunak
4. Kecepatan kompilasi
5. Pola desain yang menarik
6. Pemrograman terstruktur

Keunggulan lain dari Visual Basic .NET adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi yang lain berbasis Windows.

Dengan menggunakan Visual Basic .NET kita bisa membangun berbagai aplikasi Windows (permainan, basisdata, dan lain-lain) dengan cepat dan mudah. Dengan pendekatan visual, akan tercipta aplikasi yang menarik tanpa banyak menuliskan kode. Visual Basic .NET menggunakan bahasa Object Basic sebagai bahasa dasar. Jika telah menguasai Basic, akan dengan mudah memahami program Visual Basic .NET.

Pemrograman di lingkungan Visual Basic .NET dilakukan dengan menggunakan objek. Objek adalah hasil program yang menyatakan paket dari data dan fungsionalitas yang terkait. Di dunia nyata, objek bisa dibayangkan sebagai mobil, mempunyai rangka, badan, roda, pintu, dan lain-lain. Objek bisa terdiri dari objek-objek lain yang lebih kecil.

Objek menyediakan fungsi-fungsi khusus untuk seluruh aplikasi pengguna tanpa perlu mempedulikan proses kerja di objek itu sendiri.

Objek dibuat dari contoh yang disebut sebagai kelas. kembali ke contoh objek mobil di atas, kelas adalah cetak biru (*blueprint*) dari mobil tersebut yang mendefinisikan atribut dan proses dari objek yang

dibuat pada waktu desain, sedangkan objeknya adalah mobil itu sendiri (fisik).

Satu objek harus bisa dibedakan dari objek lainnya dari kelas yang sama. Tanpa karakteristik ini, tidaklah mungkin kita bisa membedakan dua objek, karena hal ini bisa membuat kesulitan besar bagi pengembang. Perbedaan ini bisa saja berupa pengenalan seperti nomor pengenalan (*ID*) yang diberikan ke tiap objek, ataupun setiap objek dibedakan atributnya dari objek lain.

DirectX

DirectX adalah Multimedia API yang menyediakan standar antar-muka untuk berinteraksi dengan kartu grafis dan suara, alat input, dan lainnya. DirectX berisi sejumlah interface untuk menciptakan game dan aplikasi multimedia lainnya, mendukung grafis dua-dimensi (2D) dan tiga-dimensi (3D), efek suara dan musik, alat masukan, dan aplikasi jaringan. Tanpa sejumlah API ini seorang pemrogram harus menulis kode sendiri untuk setiap kombinasi grafis, suara, keyboard, mouse dan joystick.

DirectX mengabstrakkan kode-kode instruksi pada perangkat keras dan menerjemahkan dengan kode yang dapat digunakan langsung pada pemrogram.

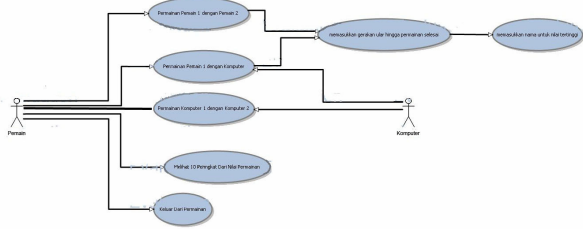
Hampir 10 tahun lamanya, DirectX hanya dapat diakses oleh pemrogram C/C++ , namun pada Desember 2002 Microsoft mengeluarkan DirectX 9, pada keluaran DirectX ini pemrogram C# dan VB .NET dapat mengaksesnya. DirectX 9 dikenal juga sebagai Managed DirectX (MDX).

Beberapa Namespace pada DirectX dan kegunaannya.

1. `Microsoft.DirectX` : top-level namespace DirectX, berisi kelas-kelas dasar juga untuk konstruksi matematik seperti vektor dan matriks.
2. `Microsoft.DirectX.Direct3DX` : sejumlah "pustaka pembantu", dengan banyak fungsi dasar yang digunakan untuk menciptakan aplikasi DirectX3D.
3. `Microsoft.DirectX.Direct3D` : pustaka yang banyak digunakan, berisi kelas-kelas dan struktur untuk membuat grafik 3D dan 2D.
4. `Microsoft.DirectX.DirectDraw` : namespace yang sudah lama, pustaka ini masi digunakan hanya untuk kompatibilitas cadangan agar aplikasi terdahulu masih dapat dijalankan. Merupakan pustaka khusus 2D.
5. `Microsoft.DirectX.DirectInput` : namespace dimana semua input (seperti mouse dan keyboard) dikontrol, termasuk untuk joystick.
6. `Microsoft.DirectX.DirectPlay` : API jaringan untuk game multiplayer, juga berisi kelas-kelas untuk topologi client-server dan komunikasi suara.

Desain Sistem Aplikasi

Diagram *use-case* untuk aktor pada permainan Snake 3D ini ditunjukkan pada Gambar 1.



Gambar 1 Diagram *use-case* untuk aktor pemain dan komputer

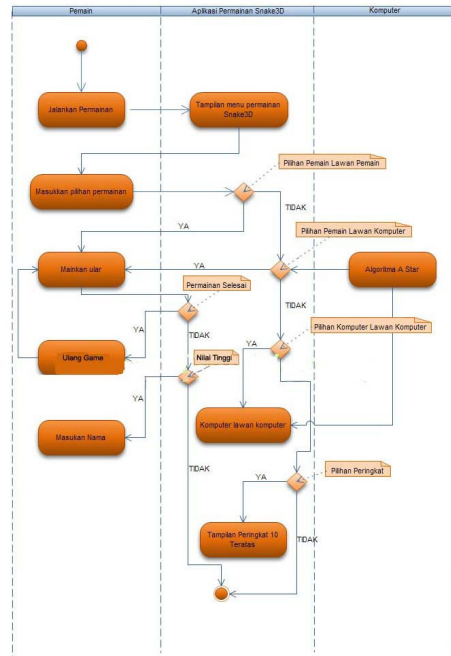
Aktor yang terdapat dalam sistem ada dua, yaitu pemain dan komputer. Pemain akan mendapatkan hasil dari sistem jika memberikan masukkan berupa gerakan untuk ular, baru kemudian setelah permainan selesai bila pemain mendapat nilai tinggi akan memasukkan namanya. Diagram *use-case* untuk aktor pemain dan komputer pada perangkat lunak aplikasi permainan Snake 3D ini ditunjukkan pada Gambar 1.

Seperti yang terlihat pada Gambar 1, tampak bahwa aktor pemain dapat melakukan beberapa proses dalam permainan Snake 3D, antara lain :

1. Memilih permainan pemain 1 dengan pemain 2.
2. Memilih permainan pemain 1 dengan komputer.
3. Memilih permainan komputer 1 dengan komputer 2.
4. Melihat nilai peringkat permainan.
5. Keluar, yaitu pengguna dapat keluar dari sistem.
6. Memasukkan gerakan ular untuk mendapatkan makanan dan menghindari halangan.
7. Memasukkan nama bila mendapat nilai tertinggi dari permainan.

Diagram aktivitas menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti.

Proses aplikasi permainan Snake 3D ini dapat digambarkan dengan diagram aktivitas pada Gambar 2

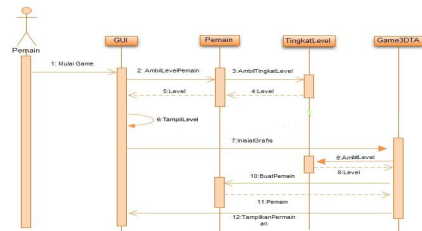


Gambar 2 Diagram Aktivitas Aplikasi Permainan Snake 3D

Diagram urutan menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu.

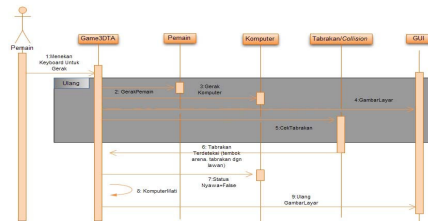
Diagram urutan pada permainan ini secara umum terdiri atas 4 bagian yaitu :

- a) Diagram urutan inialisasi permainan



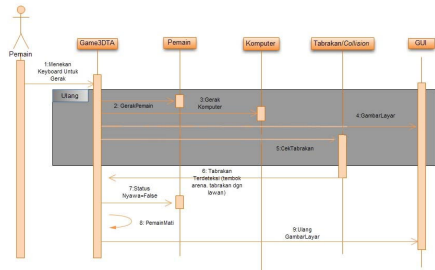
Gambar 3 Diagram urutan inialisasi permainan

- b) Diagram urutan Pemain melawan Komputer dan menang



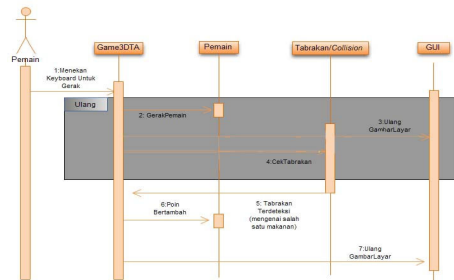
Gambar 4 Diagram urutan pemain melawan komputer dan menang

- c) Diagram urutan Pemain melawan Komputer dan kalah



Gambar 5 Diagram urutan pemain melawan komputer dan kalah

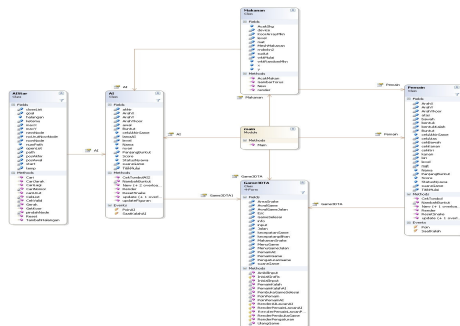
d) Diagram urutan Pemain mendapat poin dari makanan



Gambar 6 Diagram urutan mendapat poin

Diagram kelas adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam perangkat lunak yang sedang kita kembangkan. Diagram kelas menggambarkan struktur dan deskripsi kelas beserta hubungan kelas yang satu dengan yang lain.

Secara garis besar, diagram kelas dari perancangan sistem aplikasi ini dapat dilihat pada Gambar 7. Berdasarkan Gambar 7 pada diagram kelas ini, terlihat kelas **Program Utama (module main)** memiliki relasi dengan beberapa kelas lainnya yaitu **Game3DTA**, **makanan**, **AI**, **AIStar** dan **Pemain**. Kelas **AIStar** termasuk dalam kelas **AI** sehingga berhubungan dengan komputer menggunakan Algoritma A Star (A^*) dalam kelas **AIStar**. Kelas **Pemain** dan **AI** memiliki hubungan dengan kelas **makanan** karena keduanya adalah aktor yang mengakses makanan untuk mendapatkan poin.



Gambar 7 Diagram kelas aplikasi permainan

Snake 3D

Seperti terlihat pada Gambar 7, program ini terdiri dari 4 kelas yang semua terhubung ke kelas ProgramUtama. Hanya satu kelas yang memiliki hubungan dengan kelas AI. Dan juga kelas makanan yang terhubung dengan 2 kelas yaitu AI dan pemain.

Pengujian Proses Penekanan Papan Kunci

Pada pengujian penekanan papan kunci menggunakan metode kotak hitam (*black box*) yang artinya aplikasi dihadapkan pada suatu kondisi kemudian melihat hasilnya. Tujuannya adalah untuk memeriksa respon dari penekanan papan kunci pada aplikasi apakah sudah sesuai atau belum. Tabel 1 menunjukkan hasil pengujian pada penekanan papan kunci.

Tabel 1 Pengujian penekanan papan kunci

Masukan	Keluaran	
	Keluaran diharapkan	Keluaran sistem
Esc	Keluar dari aplikasi	Keluar dari
a	Jalan ke arah kiri	Jalan ke arah kiri
w	Jalan ke arah atas	Jalan ke arah atas
s	Jalan ke arah bawah	Jalan ke arah
d	Jalan ke arah kanan	Jalan ke arah
↑	Jalan ke arah atas	Jalan ke arah atas
↓	Jalan ke arah bawah	Jalan ke arah
←	Jalan ke arah kiri	Jalan ke arah kiri
→	Jalan ke arah kanan	Jalan ke arah
y	Keluar ke menu	Keluar ke menu
t	Lanjut permainan	Lanjut permainan
enter	konfirmasi	konfirmasi

Dari tabel diatas terlihat bahwa hasil penekanan papan kunci sudah sesuai dengan keluaran yang diharapkan.

Pengujian Deteksi Tabrakan (Colission)

Terdapat 4 deteksi tabrakan dalam permainan Snake 3D ini yaitu :

1. Deteksi tabrakan ular dengan batas arena
2. Deteksi tabrakan ular dengan penghalang jalan.
3. Deteksi tabrakan ular dengan tubuhnya sendiri.
4. Deteksi tabrakan ular dengan tubuh lawannya.

Pada Gambar 8 adalah gambar menunjukkan ular berwarna biru mengenai batas arena dan tabrakan terjadi sehingga membuat ular kalah dan menjadi penghalang jalan berwarna abu-abu. Deteksi tabrakan sesuai dengan yang diharapkan.



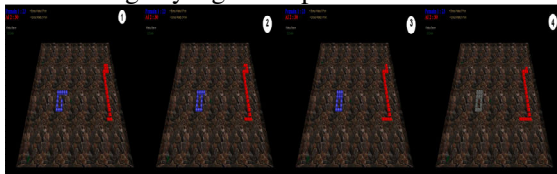
Gambar 8 Deteksi tabrakan ular dengan batas arena

Pada Gambar 9 adalah gambar menunjukkan ular mengenai penghalang jalan. deteksi tabrakan sesuai dengan yang diharapkan.



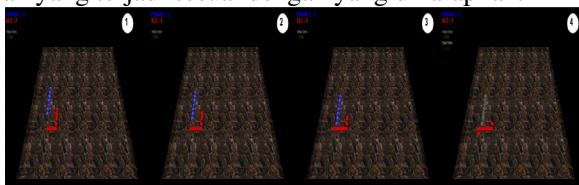
Gambar 9 Deteksi tabrakan ular dengan penghalang jalan

Pada Gambar 10 adalah gambar yang menunjukkan ular mengenai tubuhnya sendiri, dapat terlihat bagian 3 ular terkurung tubuhnya sendiri dan pada bagian 4 tabrakan terjadi . Deteksi tabrakan yang terjadi sesuai dengan yang diharapkan.



Gambar 10 Deteksi tabrakan ular mengenai tubuhnya sendiri

Pada Gambar 11 adalah gambar yang menunjukkan ular mengenai tubuh lawannya. Deteksi tabrakan yang terjadi sesuai dengan yang diharapkan.

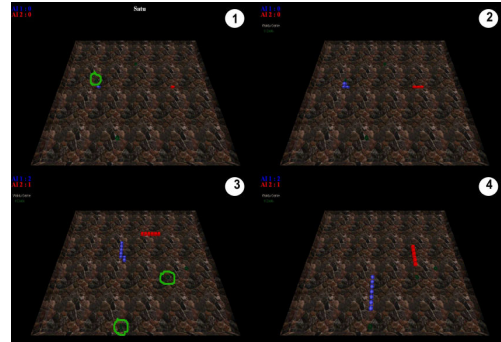


Gambar 11 Deteksi tabrakan ular mengenai tubuh lawannya

Pengujian Algoritma A Star (A*)

Algoritma A Star (A*) merupakan algoritma pencarian dengan jalan terpendek menuju tujuan, dengan algoritma ini komputer akan mendapatkan makanan dengan jalan yang terpendek (cepat dan efisien). Keandalan Algoritma A Star (A*) pada permainan ini dalam hal efisiensi pencarian komputer mendapatkan makanan melalui beberapa pengujian adalah 100%. Persentase mutlak ini adalah benar selama makanan yang dicari terdapat jalan yang dapat

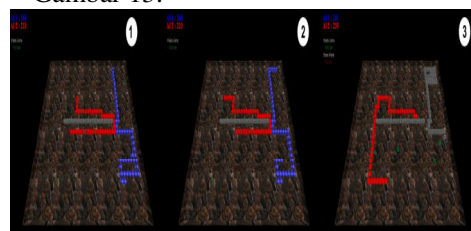
dilalui. Pada Gambar 12 menunjukkan pencarian makanan.



Gambar 12 Pencarian makanan pada permainan Snake 3D

Pada Gambar 12 bagian 1 dan 2 permainan akan segera dimulai dan komputer satu berwarna biru akan melakukan pencarian dengan makanan terdekat sebelah kiri atas ditandai dengan warna hijau. Kemudian pada bagian 3 dan 4 masing-masing komputer satu dan dua melakukan pencarian dengan makan dengan ditandai warna hijau bagaian bawah arena, dapat terlihat masing melakukan perhitungan sendiri dengan mendapatkan makanan. Walaupun dengan Algoritma A Star (A*) komputer dapat mendapatkan makanan secara efisien, bukan berarti permainan pemain melawan komputer ini akan berakhir dengan kemenangan komputer. Beberapa pengujian dalam permainan komputer lawan komputer didapatkan kesalahan komputer yang menyebabkan kekalahan pada permainan ini yaitu.

1. Komputer tidak memperhitungkan pada saat mendapatkan makanan, ular terkurung baik dengan tubuhnya sendiri atau tubuh dari ular lawannya. Kesalahan ini ditunjukkan pada Gambar 13.



Gambar 13 Komputer terkurung pada saat mendapatkan makanan.

Pada Gambar 13 komputer ular berwarna biru telah mendapatkan makanan dengan posisi awal pada bagian 1 lalu makanan melakukan acak dan muncul pada sebelah kanan, komputer tidak memperhitungkan bila mengambil makanan sebelah kanan maka tubuhnya terkurung oleh tubuhnya sendiri dan tidak ada jalan keluar seperti terlihat pada bagian 2 dan 3.

2. Komputer terperangkap pada saat makanan berada pada halangan yang menyebabkan

buntu (tidak ada jalan keluar setelah mendapatkannya) hal ini bergantung pada penghalang permainan.

Pengujian Masalah dan Penanganannya (Troubleshooting)

Dalam permainan Snake 3D ini terdapat 2 masalah yang menyebabkan permainan berhenti dan tidak responsif (*hang / crash*) yaitu.

1. Perhitungan komputer yang berusaha membagi dengan angka nol (*attempted to divide by zero*).
2. Perhitungan komputer yang menyebabkan keluar dari batas (*out of boundary*). Masalah ini terjadi pada saat tampilan menu dalam keadaan dibiarkan (*idle*) dengan tampilan demo ular yang mendapatkan makanan yang diacak dan tampilan arena yang bergerak mendekati dan menjauhi layar terus-menerus

Kesimpulan

Berdasarkan hasil pengujian dan pembahasan dapat disimpulkan beberapa hal sebagai berikut :

1. Komputer menggunakan Algoritma A Star (A*) untuk mencari makanannya. Algoritma A Star (A*) adalah algoritma pencarian terbaik dalam mencari jalur terpendek dengan perhitungan terkecil pada jalur dengan simpul awal menuju simpul akhir.
2. Algoritma A Star (A*) pada komputer dalam permainan ini teruji sangat efektif dalam mendapatkan makanannya dengan jalur terpendek.
3. Persentase komputer mendapatkan makanannya dengan jalur terpendek adalah 100% namun bukan berarti komputer dalam permainan ini akan menang mutlak melawan pemain.
4. Kesalahan yang timbul saat komputer kalah dalam permainan ini adalah pada saat komputer mengambil makanan namun terkurung dalam tubuhnya sendiri.

Saran

Tingkat kesulitan dalam permainan dapat dikembangkan dengan persentase bagaimana ular mendapatkan makanannya. Misalnya 30%: sangat mudah, 50%: mudah, 75%: susah, dan 100%: sangat susah. Algoritma A Star (A*) dapat dikembangkan lebih lanjut agar ular lebih efektif mendapatkan makannya, seperti dapat memperhitungkan bila pada saat mendapatkan makanan apakah jalan menuju makanan tersebut tidak terkurung oleh tubuh ular itu sendiri. Dalam permainan Snake 3D ini sebenarnya permainan berjalan dengan koordinat dua-dimensi

(koordinat x dan y) namun ditampilkan seperti tiga-dimensi. Perlu penelitian lebih lanjut untuk mengembangkan permainan yang berjalan pada bidang tiga-dimensi (koordinat x, y, dan z) dimana ular dapat berjalan ke atas dan mungkin sudut pandang permainan ini adalah orang ke satu seperti permainan tembak menembak (*shooting*).

DAFTAR PUSTAKA

Suharli, S., *Membangun Aplikasi Berbasis Windows dengan Visual Basic .NET*, Penerbit PT Elex Media Komputindo, Jakarta, 2005.

Grungeiger, D., *Programming Visual Basic .NET*, O'Reilly, 2002.

Miller, T., *Managed DirectX® 9 Kick Start: Graphics and Game Programming*, Sams Publishing, 2003.

Morrison, M., *Teach Yourself Game Programming in 24 Hours*, Sams Publishing, 2002.

Buckland, M., *Programming Game AI By Example*, Wordware Publishing, 2005.

Nilsson, N.J., *Artificial Intelligence A New Synthesis*, Morgan Kaufman Publishers, 1998.

---, http://en.wikipedia.org/wiki/A*_search_algorithm, Agustus 2008.

---, <http://www.codeproject.com/KB/directx/mdxsnake.aspx>, Agustus 2008

---, http://www.gamasutra.com/features/19990212/sm_08.htm, Agustus 2008

---, <http://www.policyalmanac.org/games/aStarTutorial.htm>, Agustus 2008

BIODATA



Dilahirkan di Jakarta tanggal 24 Desember 1984. Menempuh pendidikan mulai dari SDN 01 Jakarta, SMPN 2 Jakarta, SMAN 68 Jakarta. Saat ini sedang menyelesaikan studi Strata-1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro konsentrasi Informatika

Komputer.

Pembimbing I Menyetujui,

Aghus Sofwan, ST., MT.
NIP. 132 163 757

Pembimbing II

R. Rizal Isnanto, ST., M.M, MT.
NIP. 132 288 515