

KECERDASAN BUATAN MENGGUNAKAN ALGORITMA A STAR (A*) DALAM PERMAINAN ULAR TANGGA (SNAKE 3D)

Kikik Chomsan Maaruf

Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
Email : kikik.m@students.amikom.ac.id

Abstrak

Ular Tangga adalah permainan sederhana di mana ular melakukan pencarian makanan dan ekornya akan tumbuh panjang setelah mendapatkan makanan. Tujuan dari penelitian permainan ini adalah untuk membuat kecerdasan buatan dari permainan ular tangga dengan menggunakan Unity 3D yang programmer gunakan untuk membuat game Android.

Metode yang digunakan dalam merancang game ini adalah UML (Unified Modeling Language). Pengembangan perangkat lunak permainan ular tangga menggunakan Unity 3D. Kecerdasan buatan yang digunakan dalam game 3D ini adalah Algoritma A Star (A*). Algoritma ini adalah untuk mencari jalan terpendek dari awal sampai tujuan jangkauan. Menerapkan Algoritma A Star (A*) dalam game ini adalah untuk mendapatkan makanan yang ditempatkan secara acak dan komputer akan menemukan jalan untuk mendapatkannya dengan jalan terpendek.

Dari hasil pengujian dan analisis dapat disimpulkan bahwa komputer dalam game ini telah diuji sangat efektif dalam mendapatkan makanan dengan menggunakan Algoritma A Star (A*). Persentase untuk mendapatkan makanan dari jalur terpendek adalah 100% untuk jalan yang telah melewati, tetapi komputer tidak selalu memenangkan pertandingan, karena komputer tidak memperhitungkan ketika makanan bisa terkurung oleh tubuh mereka sendiri, tubuh pemain lawan yang menghambat, atau memasuki jalur yang diblokir.

Kata Kunci: Kecerdasan Buatan, Algoritma A Star, Permainan.

1. Pendahuluan

Latar Belakang

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.^[1] 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. 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.^[2]

Rumusan Masalah

1. Bagaiman membuat AI pada permainan Snake 3D menggunakan Algoritma A Star?

Tujuan

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

Tinjauan Pustaka

Penelitian ini mengacu pada penelitian yang dilakukan

Oleh Sunarto yang berjudul "Pembuatan Game 3 Dimensi "Me Vs Aliens" Dengan Menggunakan Unity 3d Game". Perbedaan penelitian yang dilakukan oleh Sunarto dan penulis yaitu dari algoritma kecerdasan buatan dan genre game tersebut. Penelitian sebelumnya menggunakan Algoritma AI untuk kecerdasan buatan dan action sebagai genre game, sedangkan penulis menggunakan Algoritma A Star kecerdasan

buatan.^[3] Referensi yang kedua mengacu pada penelitian yang dilakukan oleh Michello Pratama Tjahyadi yang berjudul "Prototipe Game Musik Bambu Menggunakan Engine Unity 3D". Pada penelitian tersebut telah dibuat prototipe game musik bambu yang mengandung nilai-nilai pendidikan mengenai kebudayaan daerah khususnya alat musik bambu Minahasa.^[4] Perbedaan penelitian yang dilakukan oleh Michello Pratama Tjahyadi dengan penulis yaitu pada Algoritma yang digunakan. Peneliti sebelumnya menggunakan Algoritma AI, sedangkan penulis menggunakan Algoritma A Star.

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*).

Heuristik adalah penilai yang memberi harga pada tiap simpul yang memandu A* mendapatkan solusi yang diinginkan. Dengan heuristik yang benar, maka A* pasti akan mendapatkan solusi (jika memang ada solusinya) yang dicari. Dengan kata lain, heuristik adalah fungsi optimasi yang menjadikan algoritma A* lebih baik dari pada algoritma lainnya. Perhitungan pada Algoritma A Star (A*) dapat ditentukan sebagai berikut :

$$F(x) = G(x) + H(x) \dots (1)$$

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).
- 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

Game Engine adalah sebuah sistem perangkat lunak yang dirancang untuk menciptakan atau mengembangkan video game. Ada banyak game engine yang dirancang untuk bekerja pada konsol video game dan sistem desktop seperti Microsoft Windows, Linux, dan Mac OS. Fungsional inti yang disediakan oleh game engine biasanya mencakup mesin render untuk merender 2D atau 3D grafis, physics engine dan deteksi tabrakan, suara, script, animasi, kecerdasan buatan (AI), jaringan, streaming, manajemen memory, threading, dan grafik animasi.

Unity Game Engine merupakan software yang digunakan untuk membuat video game 3D atau konten yang interaktif lainnya seperti, visual arsitektur dan real-time 3D animasi. Unity Game Engine tidak hanya merupakan sebuah game engine, tapi juga merupakan sebuah editor. Unity Game Engine mirip dengan game engine lainnya seperti, Director, Blender game engine, Virtools, Torque Game Builder atau Gamestudio.

Unity Game Engine tidak dirancang untuk proses desain atau modelling, dikarenakan unity bukan tool untuk mendesain. Jika ingin mendesain, pergunakan 3D editor lain seperti 3dsmax atau Blender. Banyak hal yang bisa dilakukan dengan unity, ada fitur audio reverb zone, particle effect, dan Sky Box untuk menambahkan langit.

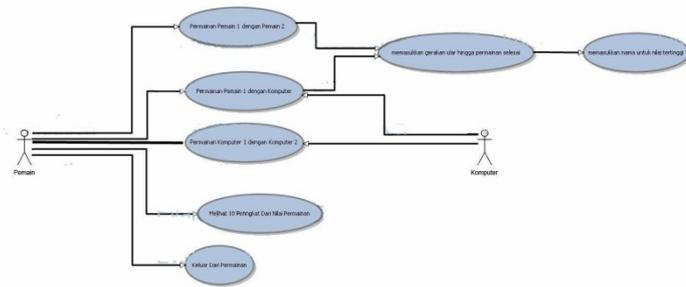
Kelebihan dari Unity Game Engine yaitu multi platform, Unity Game Engine dapat dioperasikan pada platform Windows dan Mac Os dan dapat menghasilkan game untuk Windows, Mac, Linux, Wii, iPad, iPhone, dan platform Android. Unity Game Engine juga dapat menghasilkan game browser, untuk menjalankan pada web browser kita memerlukan sebuah plugin yaitu Unity web player plugin. Untuk saat ini Unity Game Engine sedang dikembangkan untuk platform Xbox 360 dan PlayStation 3.

Fitur scripting yang disediakan, mendukung 3 bahasa pemrograman, JavaScript, C#, dan Boo. Flexible and Easy Moving, rotating, dan scaling objects hanya perlu sebaris kode. Begitu juga dengan Duplicating, removing, dan changing properties. Visual Properties Variables yang di definisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di drag and drop, bisa memilih warna dengan color picker. Berbasis .NET. Artinya penjalanan program dilakukan dengan Open Source .NET platform, Mono.

2. Pembahasan

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

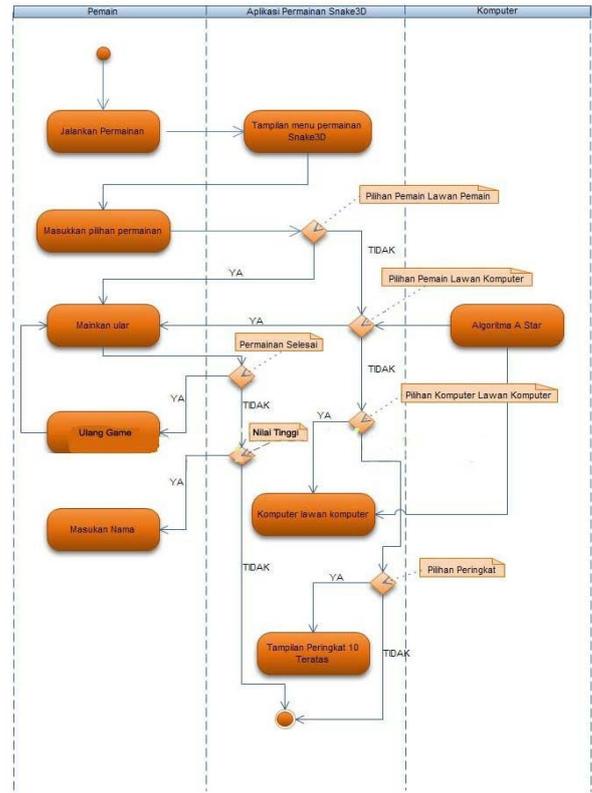
Aktor yang terdapat dalam sistem ada dua, yaitu pemain dan komputer. Pemain akan mendapatkan hasil dari sistem jika memberikan masukkan berupa gerakan untuk ular, baru kemudian setelah permainan selesai bila pemain mendapat nilai tinggi akan memasukkan namanya. Diagram *use-case* untuk aktor pemain dan 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

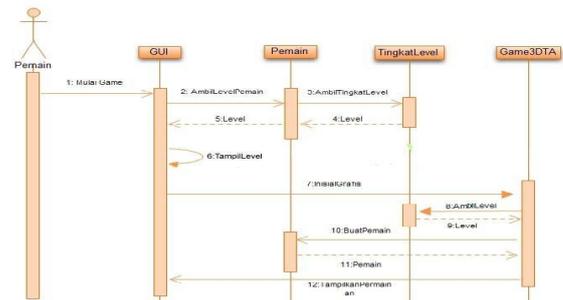


Gambar 2 Diagram Aktivitas Aplikasi Permainan Snake 3D

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

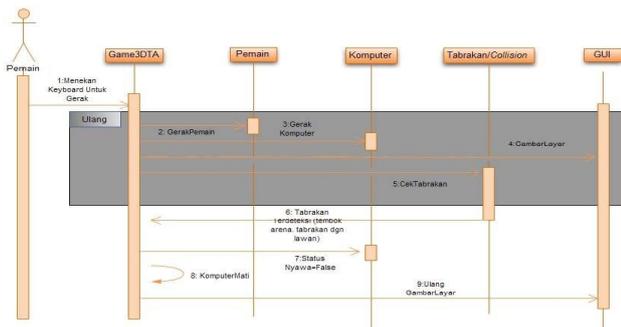
Diagram urutan pada permainan ini secara umum terdiri atas 4 bagian yaitu :

- a) Diagram urutan inialisasi permainan



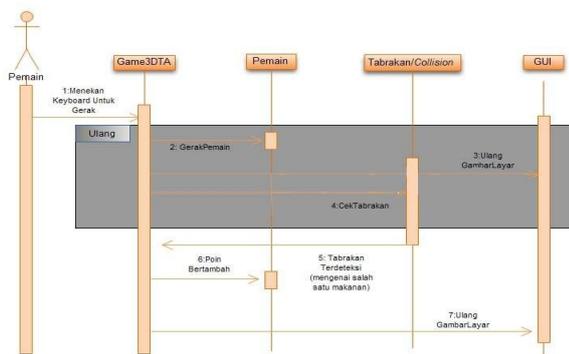
Gambar 3 Diagram urutan inialisasi permainan

- b) Diagram urutan Pemain melawan Komputer dan menang



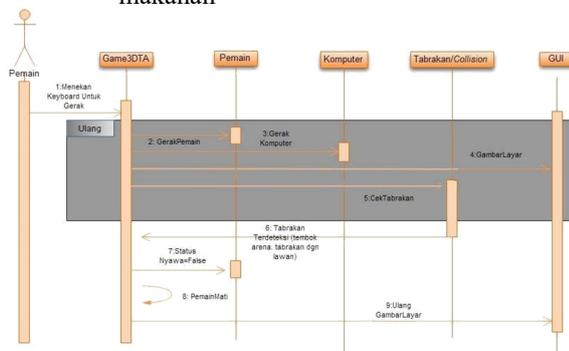
Gambar 4 Diagram urutan pemain melawan komputer dan menang

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

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 7 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.



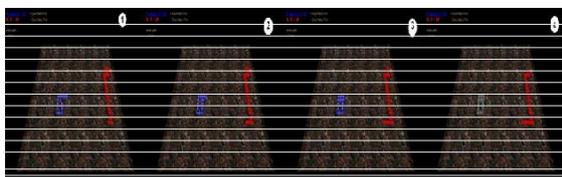
Gambar 7 Deteksi tabrakan ular dengan batas arena

Pada Gambar 8 adalah gambar menunjukkan ular mengenai penghalang jalan. deteksi tabrakan sesuai dengan yang diharapkan.



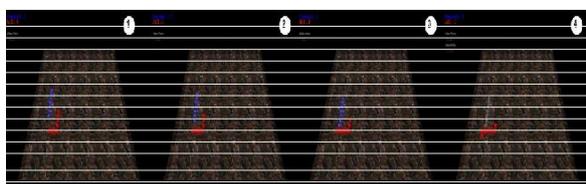
Gambar 8 Deteksi tabrakan ular dengan penghalang jalan

Pada Gambar 9 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 9 Deteksi tabrakan ular mengenai tubuhnya sendiri

Pada Gambar 10 adalah gambar yang menunjukkan ular mengenai tubuh lawannya. Deteksi tabrakan yang terjadi sesuai dengan yang diharapkan.

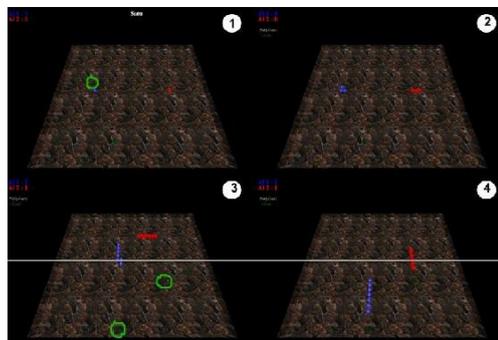


Gambar 10 Deteksi tabrakan ular mengenai tubuh lawannya

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

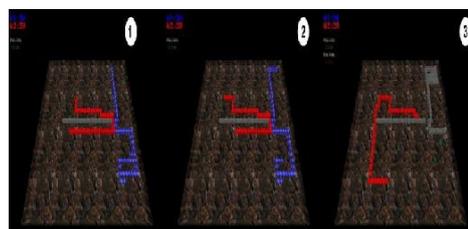
komputer mendapatkan makanan melalui beberapa pengujian adalah 100%. Persentase mutlak ini adalah benar selama makanan yang dicari terdapat jalan yang dapat dilalui. Pada Gambar 12 menunjukkan pencarian makanan.



Gambar 11 Pencarian makanan pada permainan Snake 3D

Pada Gambar 11 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 12.



Gambar 12 Komputer terkurung pada saat mendapatkan makanan.

Pada Gambar 12 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 penelitian dan pembahasan yang penulis lakukan mengenai “Kecerdasan Buatan Menggunakan Algoritma A Star Dalam Permainan Ular Tangga (Snake 3D)”. Maka penulis pada akhirnya mengambil beberapa kesimpulan yaitu :

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.

DAFTAR PUSTAKA.

- [1] Morrison, M., Teach Yourself Game Programming in 24 Hours, Sams Publishing, 2002
- [2] Kusumadewi, Sri, 2003, Artificial Intelligence (Teknik dan Aplikasinya), Yogyakarta : Graha Ilmu.
- [3] Buckland, M., Programming Game AI By Example, Wordware Publishing, 2005.
- [4] Nilsson, N.J., Artificial Intelligence A New Synthesis, Morgan Kaufman Publishers, 1998.

Biodata Penulis

Kikik Chomsan Maaruf, sedang menjalani program studi S1 Teknik Informatika semester 5 di STMIK AMIKOM Yogyakarta.