

OPTIMASI ALGORITMA BREADTH FIRST SEARCH PADA GAME ENGINE 3D THIRD PERSON SHOOTER MAZE BERBASIS AGEN CERDAS ANDROID

(SEARCH THE BLIND BREADTH FIRST SEARCH ALGORITHM IN 3D GAME ENGINE MAZE THIRD PERSON SHOOTER ANDROID BASED ON INTELLIGENT AGENT)

Astrid Novita Putri

Fakultas Teknologi Informasi dan Komunikasi Program Studi Teknik Informatika
Universitas Semarang
astrid@usm.ac.id

Abstract- *Game is currently very popular in the community at large, one of which is the game third person shooter (TPS) which can be run through a mobile phone or computer, making it very easy and affordable, one thrid person shooter game 3D maze. The labyrinth is a game to find the right path to achieve the objectives which the way players experience many obstacles to destination, so spend a lot of time, then in need of a settlement in order to facilitate the player in completing the levels on every obstacle, in need of a algorithn Breadth First Search for ease in completing permainan. Cara employment levels every algorithn Breadth First Search is a search method that starts with the roots off the road to the next. This search is done by looking at all the nodes or vertices have the same level to determine the final outcome at that level, if they do not find the will to move to the next level. so that the process backtrack to re-find the right path to achieve goals the appropriate time.*

Keywords - Games, Third person, Shooter, Maze, Breadth First Search.

I. PENDAHULUAN

Game sangat terkenal di kalangan masyarakat, dari anak-anak sampai dewasa seperti suatu bentuk hiburan yang mendukung hubungan interaksi sosial. Dengan perkembangan teknologi informasi dan komunikasi serta komputer dan perangkat keras yang canggih, telah menunjukkan potensi untuk membuat game di dalam komputer sangat baik. Salah Satu jenis permainan yang sangat populer di semua kalangan masyarakat adalah *Third Person Shooter* labirin (disingkat *TPS*) pada Game 3D Unity ini berisi suatu permainan yang berpusat pada senjata dan proyek memerangi senjata melalui sudut pandang orang pertama. Dalam sebuah game, termasuk dalam *Third Person Shotter* labirin diperlukan suatu penyelesaian menyelesaikan masalah pada game *Third Person Shotter* melewati jalan labirin yang bertujuan untuk

memotivasi pemain agar lebih terpacu untuk menyelesaikan permainan hingga selesai.

Pada Penelitian ini peneliti akan berfokus mengenai *Game Engine* pada Player dan belum adanya penelitian yang membahas tema ini, pada *Game Engine Third Person Shooter* labirin, Bagaimana cara untuk menyelesaikan game ini yang menggunakan pola labirin, maka di butuhkan suatu agen cerdas Algoritma *Artifical Intelligece* pencarian melebar pertama (*Breadth First Search*), untuk mempermudah pemain menyelesaikan misi, sehingga menantang pemain untuk menyelesaikan misi.

II. LANDASAN TEORI

A. Game Engine

Game engine adalah sebuah sistem perangkat lunak (*software*) yang di rancang sebagai pembuatan atau pengembangan suatu video game. Game Engine yang memberikan suatu kemudahan dalam menciptakan konsep pada suatu game yang akan di buat. Mulai dari sistem rendering, physics, arsitektur, suara, scripting, A.I., dan bahkan dalam sistem *networking*.

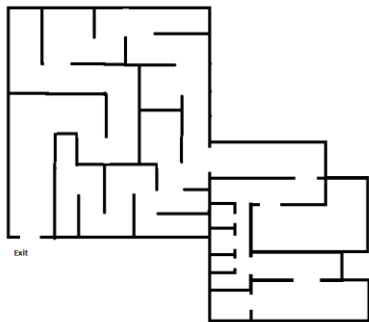
Contoh *game engine* sebagai berikut :

1. *Construct Classic* adalah *Construct* merupakan *software* untuk membuat game berbasis pemrograman C++ dengan menggunakan event base yang mempermudah *gamer* untuk membuat *game*
 2. *Game Maker* merupakan suatu game berbasis pemrograman C++ dengan menggunakan *event block*.
 3. *Unity Engine* sebuah game engine unity yang memungkinkan developer untuk membuat game berbasis 2d maupun 3d.
 4. *Unreal Engine* adalah sebuah software untuk mendvelop game terbaru yang tergolong sangat powerfull untuk membuat suatu game.
- [1]

B. Game Labirin

Permainan game labirin adalah suatu permainan yang terlihat sangat sederhana tetapi mampu untuk menyelesaikannya suatu teka-teki logika dimana menyelesaikannya dan tampilannya dari sudut pandang 3D, tingkatan game ini tergantung pada penyelesaian rute perjalanan yang harus di lalui dari jalur yang tepat. Dengan object dan keyboard menekan tombol panah atas dan bawah, ke depan dan belakang, ke kanan dan kiri. Hingga menemukan jalan keluar, batas waktu yang di berikan waktu adalah 5 menit untuk menyelesaikan, jika belum selesai maka *timeout* dan mengulang dari awal kembali.

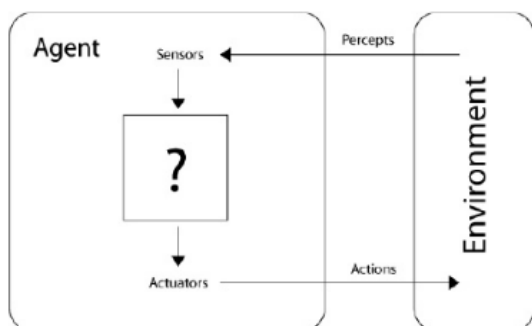
Permainan labirin adalah permainan mencari jalan keluar, dari beberapa jalur pada suatu area permainan. Template permainan berbentuk persegi atau persegi panjang yang ukurannya dapat diatur sesuai dengan keinginan user. Di dalamnya terdapat serangkaian jalur berupa labirin yang bercabang, namun tidak setiap cabang labirin tersebut merupakan jalan keluar karena ada yang terhalang oleh tembok-tembok penghalang. [2]



Gambar 1 Game 3D Labirin

C. Agen Cerdas

Permainan atau aplikasi yang dirancang menggunakan agen cerdas sebagai otak untuk melawan manusia. Agen adalah sesuatu yang dapat mengesan lingkungannya melalui sensors dan mengambil indakan terhadap lingkungannya melalui actuators. Agen yang berinteraksi dengan lingkungan melalui *actuators*. Agen yang berinteraksi dengan lingkungan melalui *sensors* dan *actuators* dapat dilihat pada gambar 1.

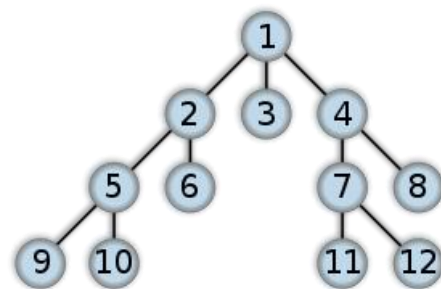


Gambar 2 Agen yang berinteraksi dengan lingkungan melalui *sensors* dan *actuators*

Definisi agen rasional adalah untuk setiap deretan persepsi yang mungkin, sebuah agen rasional hendaklah memilih satu tindakan yang diharapkan memaksimalkan ukuran *performance*-nya dengan adanya bukti yang di berikan oleh deretan presepsio apapun pengetahuan terpasang yang dimiliki agen itu. Empat agen dasar yaitu *simple reflex agents*, *model-based reflex agents*, *goal-based agents* dan *unity-based agents*. [3]

D. Algoritma Breadth First Search (BFS)

Metode pencarian yang di mulai dengan akar (Level 0) ke level selanjutnya. Pencarian ini di lakukan dengan mencari pada semua node atau simpul yang mempunyai level atau tingkatan sama sampai mennetukan hasil akhir (*Goal*) pada level tersebut, bila tidak aka akan pindah ke level selanjutnya. [4]



Gambar 3 Algoritma Breadth First Search

Langkah-Langkah Algoritma Breadth First Search

:

1. Masukkan node akar ke dalam Queue.
2. Ambil node awal Queue, lalu cek apakah node merupakan solusi.
3. Jika node merupakan solusi, pencarian selesai dan hasil di kembalikan.
4. Jika node untuk solusi , masukkan seluruh node anak ke dalam Queue.
5. Jika Queue kosong dan setiap node sudah di cek, pencarian selesai.
6. Jika Queue tidak kosong, ulangi pencarian mulai dari poin 2. [4]

E. Unity 3D

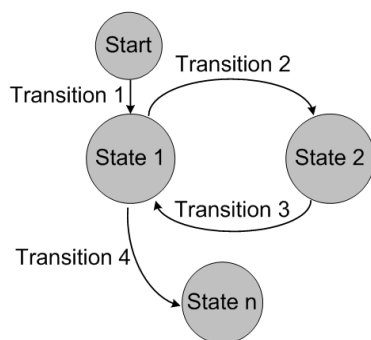
Unity 3D adalah sebuah software development yang terintegrasi untuk menciptakan video game atau konten lainnya seperti visualisasi arsitektur atau real-time animasi 3D. Unity 3D dapat digunakan pada microsoft Windows dan MAC OS X, dan permainan yang dihasilkan dapat dijalankan pada Windows, MAC, Xbox 360, OlayStation 3, Wii, iPad, iPhone, Android dan Linux. Unity 3D juga dapat menghasilkan permainan untuk browser dengan menggunakan plugin Unity Web Player. Unity 3D juga memiliki kemampuan untuk mengeksplor permainan yang dibangun untuk fungsionalitas Adobe Flash 3D. [5]

F. Finite State Machine

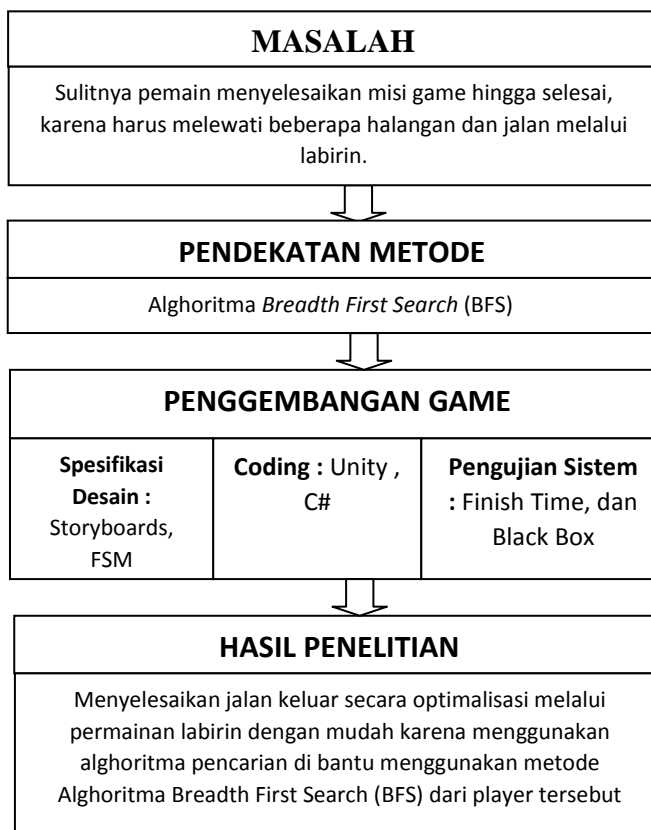
Finite State Machine (FSM) merupakan pemodelan dari perilaku (*behavior*) sebuah sistem atau obyek yang kompleks dengan beberapa kondisi atau mode yang terdefiniskan dimana mode transisi berubah sesuai dengan keadaan. *Finite State Machine* terdiri dari empat elemen utama :

- State* yang mendefinisikan kelakuan dan mungkin menghasilkan aksi.
- Transisi *state* dimana merupakan perpindahan dari satu *state* ke *state* lain.
- Aturan atau kondisi yang harus dipenuhi supaya ada transisi *state* kejadian (*events*).
- Input yang terjadi baik *internal* maupun *external*, yang memungkinkan *trigger* aturan dan mengacu ke transisi *state*. [6] [7].

Alur pada diagram state dapat dilihat pada gambar dibawah ini pada gambar 4 terdapat contoh alur state *Finite State Machine*.



Gambar 4 Contoh Diagram *State* Sederhana



Gambar 5 Kerangka Pikir Penelitian

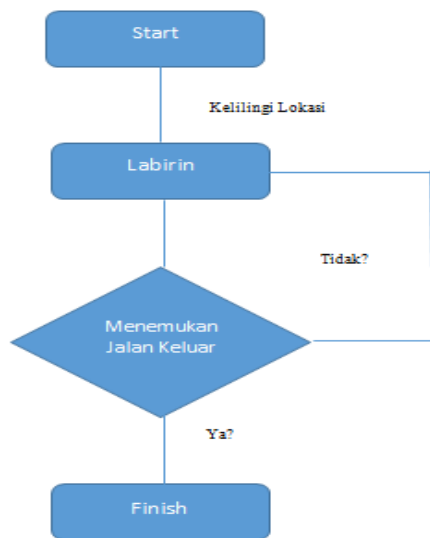
Penelitian yang akan dilakukan peneliti adalah penggunaan *artificial intelligent* dengan Algoritma *Breadth First Search* (BFS) untuk membantu *Player* dalam menemukan jalan keluar labirin sehingga mempermudah player untuk menyelesaikan misi.

III. PENELITIAN TERKAIT

No	Nama Penelitian	Jurnal	Hasil	Kekurangan
1.	Penerapan Algoritma Backtracking Pada Permainan Math Maze [10]	Pelita Informatika Budi Darma, Volume VII, Nomor : 3, Agustus 2014.	Penerapan algoritma ditetapkan dengan angka sisi kiri dan sisi atas didalam permainan math maze.	Penerapan alghoritma di tentukan harus melewati jalur yang di tetapkan, hanya menggunakan visual basic, belum berbentuk 3D
2	Penggunaan Metode Kecerdasan Buatan Runtut Maju Dalam Memecahkan Permasalahan Game Labirin [11]	Jurnal Ilmu Komputer-Volume 5-No.1-April 2012 Universitas Udaya	Hasil eksekusi program yang dibuat berdasarkan konsep kecerdasan & model pemikiran memiliki urutan jalan yang sama dengan hasil penyelesaian apabila masalah tersebut dicari secara manual	Belum di implemetasikan ke bahasa pemrograman.

Tabel 1 Penelitian Terkait

Berdasarkan Beberapa Penelitian di atas dapat disimpulkan bahwa [10] [11] belum jelasnya penelitian mengenai game labirin, dan belum terselesaikannya masalah menyelesaikan game labirin dari titik awal hingga akhir dari titik akhir menggunakan alghoritma dalam bahasa pemrograman dan game nyata 3D sehingga mempermudah pemain dalam menyelesaikan misinya.



Gambar 6 Flowchart *Game Labirin Third Person Shotter*

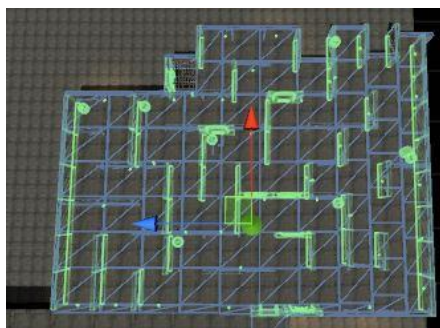
a. Tampilan *Game Labirin Third Person Shotter*

Perkembangan industry game sekarang ini sangat pesat dari segi grafis dan *Artifical Intelligent*-nya. Penelitian ini akan meneliti mengedepankan pada menyelesaikan game labirin



Gambar 7 Tampilan *Game* Ketika Player Menghampiri Setelah Memburu Jalan Keluar

Pada gambar di bawah ini adalah jalan labirin pada game labirin *Third Person Shooter* di lihat dari sudut y .



Gambar 8 Tampilan *Game* Labirin *Third Person Shooter*

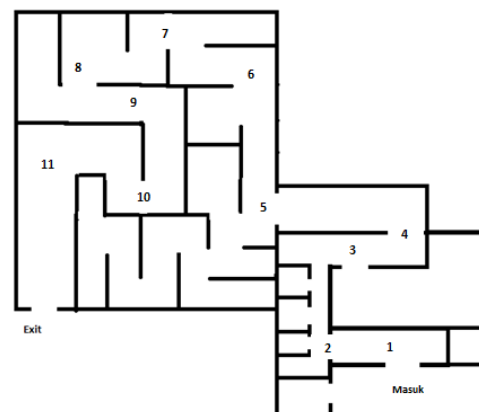
Pada gambar selanjutnya adalah gambar tampilan *design game* Labirin *Third Person Shooter* di desain di tampilan di dalam ruangan dalam posisi kurang cahaya sehingga memberikan kesan horor dan di lengkapi dengan efek api cahaya pada penerangan setiap jalan dalam sudut labirin.



Gambar 9 Tampilan Pintu Keluar *Game Third Person Shooter*

IV. HASIL PENELITIAN DAN PEMBAHASAN

a. Analisa Algoritma *Breadth First Search* pada *Game Third Person Shooter* (BFS)

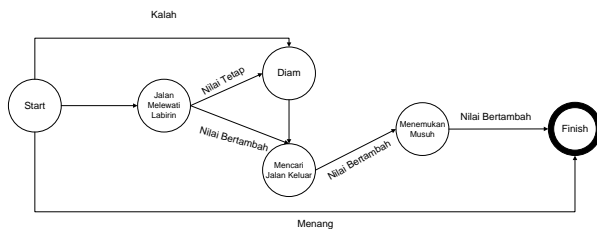


Gambar 10 Analisa Penyelesaian pada game menggunakan metode *Algoritma Breadth First Search*.

Pada penjelasan di atas di atas dapat di simpulkan bahwa *Algoritma Breadth First Search*, pencarian buta pada semua node pada level n dan di kunjungi terlebih dahulu sebelum mengunjungi node node pada level n+1. Pencarian di mulai dari node masuk ke node *exit*, kemudian berpindah ke level berikutnya, demikian pula dari kanan ke kiri hingga menemukan goal.

b. *Finite State Machine*

Kriteria yang ada di dalam *Game* Labirin *Third Person Shooter* Dengan *Algoritma Breadth First Search* dapat dijelaskan pada *Finite State Machine* sebagai berikut :



Gambar 11 FSM Game Labirin *Third Person Shooter* Dengan Algoritma *Breadth First Search*.

c. Implementasi Pada Game Labirin *Third Person Shooter* dengan menggunakan Unity 3D

Listing Coding Algoritma *Breadth First Search* pada Game *Third Person Shooter*

```

1. private void ProgressStepCycle(float speed){
2.   if (m_CharacterController.velocity.sqrMagnitude > 0 && (m_Input.x != 0 || m_Input.y != 0))
3.     {m_StepCycle += (m_CharacterController.velocity.magnitude + (speed*(m_IsWalking ? 1f : m_RunstepLengthen)))*Time.fixedDeltaTime;}
4.   if (!(m_StepCycle > m_NextStep)){ return;}
5.   m_NextStep = m_StepCycle + m_StepInterval};
6. Private void bfs(int x, int z, int depth, float jump, float previousHeight)
7. {int resistance=1;
8.   if (depth >=0) {CheckTile(x,z).GetComponentInChildren(CheckIfClicked()).Selected();
9.   if (x+1 < 25){CheckTile(x+1, z, depth, jump, previous Height);}
10.  if (x-1 >= 0){
11.    CheckTile(x-1, z, depth, jump, previousHeight);}
12.  if (z+1 < 25) {
13.    CheckTile(x, z+1, depth, jump, previousHeight);}
14.  if (z-1 >=0){
15.    CheckTile(x, z-1, depth, jump,previousHeight);}}
16. private void CheckTile(int x, int z, int depth, float jump, float previousHeight){
17.   float tileHeight = tiles(x, z).GetComponent(TileDimensions()).height;
18.   float difference = tileHeight - previousHeight;
19.   if (difference<0) difference *=-1;
20.   if CheckTile(x, z).GetComponentInChildren.CheckIfClicked().occupied && difference(jump);{
21.     int resistance = tiles(x, z).GetComponent(TileDimensions()).getResistance();
22.     if (resistance<0) resistance =1;BFS(x, z, depth-resistance, jump, tileHeight);}}
  
```

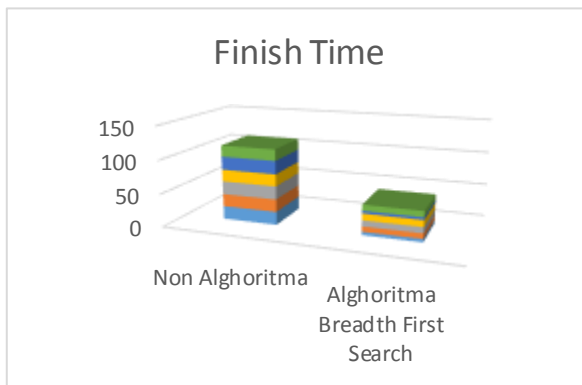
V. KESIMPULAN

Hasil dari eksperimen penelitian mengenai *Game Third Person Shooter* labirin maka dapat di simpulkan sebagai berikut : implementasi game *Third Person Shooter* labirin menggunakan *Algoritma Breadth First Search* dapat menyelesaikan hasil dengan titik yang di gunakan untuk mempermudah jalannya player untuk menyelesaikan game tersebut. Berdasarkan hasil testing, ada beberapa pengujian yang digunakan dalam *game* ini yaitu Pengujian interface yang menguji fungsi – fungsi *game* secara interface ke pemain berhasil diterima dengan baik. Kemudian ada pengujian skenario dari 50 kali percobaan *game* dapat melakukan skenario yang diujikan menggunakan metode *Algoritma Breadth First Search*, sedangkan dengan metode lain atau tanpa metode ada beberapa yang tidak bisa dilakukan sesuai dengan harapan. Dan akurasi yang di dapat adalah 85 % pada pengujian testing dan pada akhirnya dapat disimpulkan bahwa metode *Algoritma Breadth First Search* dalam masalah *finish time* lebih lama tetapi akan memudahkan pemain untuk menyelesaikan dengan cepat.

Saran :

- a. Pada penelitian ini di fokuskan pada *Game Third Person Shooter* labirin, sedangkan jika tema *labirin* sangat luas yaitu Penerapan Algoritma *Backtracking* Pada Permainan Maze, Penggunaan Metode Kecerdasan Buatan Runut Maju Dalam Memecahkan Permasalahan Game Labirin, dan sebagainya menggunakan Mungkin untuk penelitian selanjutnya dapat menambahkan kekurangan pada peneliti sekarang.
- b. *Environment game* dapat dikembangkan lagi dengan menambahkan berapa *effect* sehingga *game* lebih menarik para pemain, sehingga *game* yang dimainkan lebih realistis.
 1. Alur cerita *game* dapat dikembangkan sehingga dapat memperbanyak level *game* yang semakin menantang.
 2. Untuk labiirin *game* dapat di tambahkan parameter dengan metode lain sehingga ada perbandingan banyak metode.

Untuk keterangan gambar di bawah ini yang pada bagian kiri adalah tanpa menggunakan algoritma, sehingga dalam menyelesaikan permainan lebih lama, sedangkan dibandingkan dengan menggunakan *Algoritma Breadth First Search* pada bagian kanan menunjukan bahwa *Algoritma Breadth First Search* lebih cepat menyelesaikan permainan dibandingkan dengan tanpa menggunakan algoritma, sehingga mempermudah pemain dalam menyelesaikan permainan game labirin *thrid person shooter*.



Gambar 11 Hasil Pengujian antara Metode *Algoritma Breadth First Search* dan tanpa metode dengan Waktu Penyelesaian Permainan.

$$\text{Rumus Akurasi} = \frac{\text{Akurasi keberhasilan}}{\text{Jumlah Pengujian}} \times 100 \%$$

$$= \frac{28}{30} \times 100 \% = 93,34 \%$$

DAFTAR PUSTAKA

1. Fairuz Moch. Shiratudin Walid Thabet, "Utilizing a 3D Game Engine to Develop a Virtual Review System", *Jurnal of information Technology in Construction*, January 2011, ISSN : 1874-4752.
2. Santi I Gede, "Penggunaan Metode Kecerdasan Runut Maju Dalam Memecahkan Permasalahan Game Labirin," *Jurnal Ilmu Komputer* Volume 5 No 1-April 2012.
3. Stuart, Russel and Peter Norvig., "Artificial Intelligence A Modern Approach.", 2 Edition. United States Of America Prentice Hall, 2005
4. Sutojo,dkk, "Kecerdasan Buatan". Penerbit Andi, 2011.
5. Novita Astrid ,dkk, "Game Scoring Supporting objects Menggunakan Agen Cerdas Berbasis Fuzzy Mamdani". *Jurnal Transformatika*. 2016
6. Nendya Bhakti Matahari, "Pemetaan perilaku non-playable character pada permainan berbasis role playing game menggunakan metode finite state machine", *Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana Yogyakarta*, 2011.
7. Kim, C.H., Jeong, S.M., Hur, G.T., dan Kim, B.G. (2006), "Verification of FSM using Attributes Definition of NPCs Models", *IJCSNS International Journal of Computer Science and Network Security*, VOL.6 No.7A, July 2006.168-174.

8. Nasution Rahmad, "Penerapan Algoritma Backtracking Pada Permainan Math Maze", *Jurnal Pelita Informatika Budi Darma*, Volume VII, Nomor : 3, Agustus 2014.
9. Santi I Gede Astawa, "Penggunaan Metode Kecerdasan Buatan Runut Maju Dalam Memecahkan Permasalahan Game Labirin", *Jurnal Ilmu Komputer*-Volume 5- No.1-April 2012 Universitas Udaya