

PENGEMBANGAN PERILAKU KARAKTER ULAT PADA GAME MERAWAT TANAMAN BERBASIS SISTEM *MULTI AGENT* MENGUNAKAN ALGORITMA *FLOYD-WARSHALL*

DEVELOPMENT OF CHARACTER CATERPILLAR BASE ON MULTI AGENT SYSTEM WITH FLOYD-WARSHALL ALGORITHM

¹Farid Reza Sukma, ²Ir. Burhannuddin Dirgantoro, M.T., ³Dr. Purba Daru Kusuma, S.T., M.T.

^{1,2,3}Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

Abstrak

Sejak usia dini sangatlah penting untuk mengenalkan kepada anak bagaimana caranya memelihara lingkungan sekitar seperti merawat tanaman guna menambah pengetahuan dasar mereka. Maka dari itu sebagai daya tarik di butuhkan tampilan *visual* yang menggambarkan lingkungan sekitar dengan bentuk 3D. Sehingga User khusus nya anak – anak tertarik untuk mencoba *game* dan mengimplementasikan di kehidupan nyata.

Tujuan pengembangan perilaku ulat adalah untuk mencari jarak terdekat untuk berpindah dari pohon satu ke pohon lain. *Floyd-Warshall* merupakan salah satu algoritma yang dapat digunakan sebagai metode pencarian *path* terpendek dengan menggunakan *cost* terendah.

Kata kunci: Unity, *multi agent*, *Floyd-Warshall*, ulat

Abstract

Since an early age it is very important to introduce children how to care surrounding environment such as caring for plants to increase their basic knowledge. Therefore, as an attraction, it requires a visual display that describes the surrounding environment in 3D. So expecially children are interested to trying the game and implementing in real life.

The purpose of developing caterpillar behavior is to find the closest distance to move from one tree to another. Floyd-Warshall is one algorithm that can be used as a search method for the shortest for caterpillar to find the shortest path with the lowest cost. Also using features in the Unity engine to make this game. By playing this game, expecially children can find out how to care for the environment by caring for plants.

Keywords: Unity, *multi agent*, *Floyd-Warshall*, *caterpillar*

1. Pendahuluan

Game merupakan hal yang tidak bisa dipisahkan dari perkembangan seorang anak. Persentase anak – anak usia 2 – 7 tahun dalam Useran komputer diluar sekolah sekitar 26% dan untuk remaja umur 14 – 18 tahun sekitar 44%. Pada kenyataannya, dunia pertanian telah dieksplorasi menjadi suatu tema permainan komputer yang menarik saat ini. Permainan *digital* berbasis pembelajaran dapat menjadi sarana yang menjanjikan untuk memotivasi dan meningkatkan pembelajaran serta pengetahuan. Saat ini *game* ber *genre* edukasi masih kurang salah satunya edukasi dalam merawat tanaman dimana ulat akan menjadi hama perusak [1].

Teknologi dapat memperluas berbagai peluang bagi anak-anak untuk belajar tentang dunia, untuk mengembangkan kemampuan komunikatif dan *learn to learn* [2]. Pengalaman anak-anak dalam bermain dan belajar dengan teknologi dapat mendukung proses belajar mereka. Jika kita menggunakan *system multi agent* pada *game* kita, akan dapat membantu menyediakan desainer *game*, sarana untuk mencapai *game* secara kreatif dan membangun lebih kompleks pengembangannya [3].

Berdasarkan data tersebut penulis membuat *game* yang berjudul “Edu Farm” dengan *genre role play gaming* (RPG). Pengembangan *game* ini dimana ulat sebagai *Non Player Character* (NPC) di rancang dengan koordinasi *Multi Agent*. Pergerakan ulat kantong dari pohon satu ke pohon lainnya menggunakan jalur yang efektif atau jalur terpendek. Untuk menentukan jalur terpendek diperlukan sebuah solusi, solusi yang dipakai untuk menentukan jalur terpendek adalah dengan menerapkan algoritma *Floyd-Warshall* pada *game* ini.

2. Dasar Teori

2.1 Game

Menurut Clark (2006) *game* adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan dibatasi oleh konteks tertentu. Menurut Arief S. Sadiman (2010) *game* adalah kompetisi antara para pemain yang berinteraksi satu sama lain dengan menggunakan aturan-aturan tertentu untuk mencapai tujuan-tujuan tertentu pula. Dalam sebuah permainan harus ada kompetisi agar pemain terangsang untuk terus bermain, kompetisi tersebut dapat berwujud menang dan kalah. Pemain harus bisa menemukan strategi atau cara untuk memecahkan masalah sehingga dapat memenangkan *game* tersebut.

Game adalah permainan yang menggunakan media elektronik, merupakan sebuah hiburan berbentuk multimedia yang di buat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasan batin. Bermain *game* merupakan salah satu sarana pembelajaran. *Game* lebih sering dimainkan oleh anak-anak, akan tetapi pada zaman sekarang orang dewasa juga suka bermain *game* dan mengikuti perkembangan *game-game* yang ada sekarang. Jenis *game* sangatlah tergantung dari perkembangan zaman. Jika dilihat dari grafis yang digunakan dalam aplikasi permainan, maka aplikasi permainan dapat digolongkan menjadi dua jenis, yaitu aplikasi permainan 2D (dua dimensi) dan 3D (tiga dimensi).

2.2 Ulat

Hama pada tanaman atau tumbuhan memang sangat banyak dan berbagai jenis, mulai dari serangga yang menyerang pada daun dan bahkan pada bagian akar yang mengakibatkan tanaman akan cepat mati. Namun banyak para petani yang mengalami gangguan atau masalah tanaman dengan diserangnya hama ulat. Hama ulat memang terdengar sangat biasanya, namun sebenarnya akan mengakibatkan tanaman menjadi abnormal dan bahkan mengalami kelayuan yang sangat parah [4].

Ulat pemakan daun terdiri atas ulat api, ulat kantong, dan ulat bulu. Jenis yang paling sering menimbulkan kerugian di perkebunan antara lain: Ulat Api *Setothosea asigna*, *Setora nitens*, *Darna (Ploneta) bradleyi* dan *Birthisia bisura*; Ulat Kantong *Mahasena corbetti* dan *Metisa plana*; serta Ulat Bulu *Dasychira inclusa*, *D.mendosa* dan *Amathusia phidippus*. Serangan hama seperti ulat kantong ditandai dengan mengeringnya daun tanaman seperti terbakar.

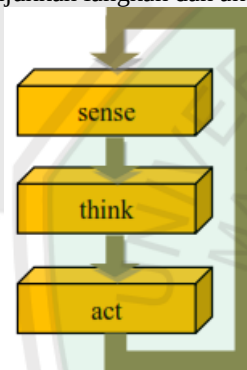
Karakter ulat di dalam *game* ini akan di desain sebagaimana aslinya yang dimana selalu menyerang pohon dan dia akan berpindah tempat untuk merusak pohon lainnya [5].

2.3 Non Playable Character (NPC)

Autonomous character adalah jenis *autonomous agent* yang ditujukan untuk Useran komputer animasi dan media interaktif seperti *games*, simulasi dan *virtual reality*. *Agent* ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi yang tindakannya ditulis di muka, dan untuk “*avatar*” dalam sebuah permainan atau *virtual reality*, tindakan yang diarahkan secara *real time* oleh pemain. Dalam permainan, karakter otonom biasanya disebut NPC (*Non Playable Character*) [6].

Kedudukan NPC dalam suatu *game* adalah sebagai komponen pelengkap yang dapat menjadikan *game* tersebut tidak membosankan. Biasanya dalam sebuah *game*, perilaku NPC ditentukan sebagai musuh atau pengganggu dari karakter yang dimainkan User. Banyak teknik yang digunakan untuk membuat NPC yang dapat berperilaku realistis dan bervariasi.

Perilaku NPC pada dasarnya akan berperilaku dengan mengulangi tiga tahap yaitu *sense*, *think*, dan *act*. Seperti yang diperlihatkan pada Gambar 2.1 yang menunjukkan langkah dan aktivitas perilaku NPC [7].

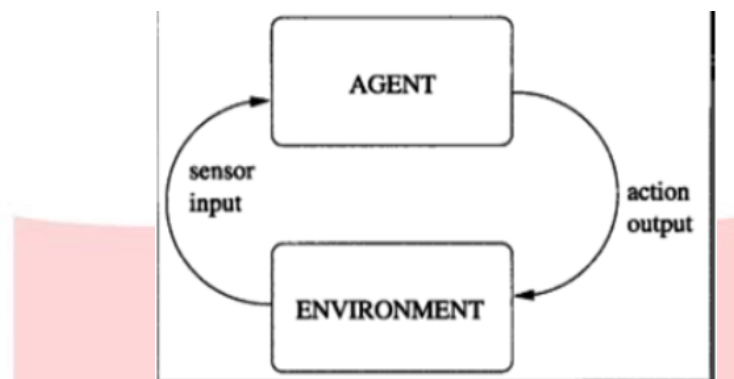


Gambar 2.1 Tahap perilaku NPC [7]

Sense : memahami situasi melalui yang terjadi pada saat pada sistem
Think : menentukan aksi apa yang tepat dan sesuai dengan aturan
Act : memerintah sistem untuk melakukan aksi yang telah ditentukan

2.4 Multi Agent

Sistem *multi agent* berasal dari *distributed artificial intelligence* yang merupakan salah satu cabang ilmu *artificial intelligence* (AI). Dalam kerangka perkembangan teknologi informasi yang pesat saat ini membawa dampak yang baik bagi perkembangan teknologi *agent* dan *multi agent* sehingga telah dapat dengan mudah diimplementasikan secara nyata. Banyak sekali contoh aplikasi yang menerapkan sistem *agent* dan *multi agent* ini seperti ACORN (*Agent-based Community Oriented Routing Network*), FIPA-OS yang dibuat oleh *Foundation for Intelligent Physical Agent*. Perkembangan dari sistem *multi agent* juga tidak terlepas dari kelebihan yang dimilikinya seperti karakteristiknya yang otonomi, proaktif, *intelligent* (cerdas), *corporate* (mampu bekerja sama dengan *agent* lain) [8].



Gambar 2.2 Interaksi *Agent* dengan lingkungan [8]

Gambar 2.2 menjelaskan bahwa sebuah *agent* mampu menerima masukan dari lingkungan di sekitarnya dan memberikan keluaran sesuai dengan masukan tersebut secara otomatis seperti yang telah ada di gambar di atas. *Agent* tidak selamanya beroperasi sendirian tapi kadang-kadang bisa melibatkan *agent-agent* lain. Interaksi-interaksi antar *agent* kadang-kadang sangatlah kompleks untuk ditangani secara individual sehingga kadang-kadang penanganan *agent* bisa dilakukan secara kolektif [9].

2.5 Algoritma Floyd-Warshall

Algoritma yang ditemukan oleh Warshall untuk mencari rute terpendek merupakan algoritma yang sederhana dan mudah implementasinya. Algoritma *Floyd-Warshall* memiliki input graf berarah dan berbobot (V, E), yang berupa daftar titik (*node* / titik V) dan daftar sisi (sisi E). Bobot garis e dapat diberi simbol $w(e)$. Jumlah bobot sisi-sisi pada sebuah jalur adalah total bobot jalur tersebut. Sisi pada E diperbolehkan memiliki bobot negatif, akan tetapi tidak diperbolehkan bagi graf W_{ij} untuk memiliki siklus dengan bobot negatif. Algoritma ini menghitung bobot terkecil dari semua jalur yang menghubungkan sebuah pasangan titik, dan melakukannya sekaligus untuk semua pasangan titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang paling minimum.

Dalam iterasinya untuk mencari lintasan terpendek, algoritma *Floyd-Warshall* membentuk n matriks, sesuai dengan iterasi- k . Ini akan menyebabkan prosesnya lambat, terutama untuk nilai n yang besar. Meskipun waktu prosesnya bukanlah yang tercepat, algoritma *Floyd-Warshall* sering dipergunakan untuk menghitung lintasan terpendek karena kesederhanaannya. Di samping itu, implementasi algoritma *Floyd-Warshall* sangat mudah dibuat [10].

Matriks keterhubungan W yang digunakan untuk menyatakan graf berarah berbobot sama dengan matriks yang digunakan untuk menyatakan graf berbobot, yaitu elemen-elemennya menyatakan bobot garis. Secara umum matriks keterhubungan untuk menyatakan graf berarah berbobot tidaklah simetris karena bobot garis dari titik v_i ke v_j ($W_{i,j}$) tidak sama dengan bobot garis dari titik v_j ke v_i ($= W_{j,i}$) dan $W_{i,i} = \infty$ untuk semua i [11].

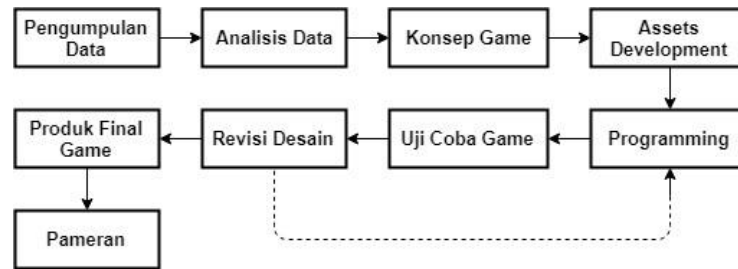
3. Pembahasan

3.1 Rancangan Umum Game

Game ini bertemakan tentang petani yang ini membasmi hama perusak tanaman. Pada game ini petani memiliki musuh yaitu ulat dan siput, sehingga memberikan tantangan untuk membasmi hama sebelum pohon habis. Peran dari siput dan ulat adalah memakan pohon hingga habis. Di game ini user memiliki tantangan lain seperti timer. Dalam pengambilan sistem poin, petani (user) harus membasmi ulat dan siput dengan timer yang telah ditentukan untuk menambahkan poin skornya. Dan sasaran User game ini nantinya adalah semua umur.

3.2 Perancangan Sistem

Agar game EduFarm dapat menghasilkan keluaran yang diharapkan dan berfungsi dengan baik, maka rancangan game yang akan dibuat adalah



Gambar 3.1 Perancangan Sistem

Gambar 3.1 menunjukkan Perancangan Sistem game EduFarm dengan penjelasan skema rancangan sistem sebagai berikut :

- Pengumpulan data untuk mendapatkan data yang *valid* sehingga hasil kesimpulan dan penelitian tidak akan di ragukan kebenarannya dan berguna sebagai bahan analisis
- Analisis untuk menentukan jenis karakter ulat dan algoritma yang akan digunakan.
- Konsep game untuk menentukan jalan cerita *game*
- *Asset Development* untuk mengembangkan *game* yang sudah ada dan memperbaharui game menjadi lebih baik
- *Programming* untuk memulai pembuatan *game* berdasarkan algoritma masing-masing
- Uji coba *game* untuk mengetahui kekurangan dan kelebihan *game* yang sudah di kembangkan
- Revisi Desain dibuat dengan menggunakan *unity game engine* dan memasukkan karakter petani, dan siput.
- Produk *final game*
- Untuk percobaan *game* kepada semua kalangan umur

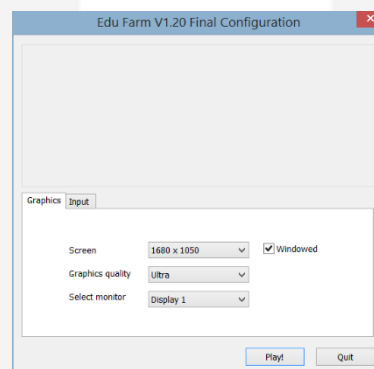
4. Pengujian dan Analisis

4.1 Implementasi

Permainan EduFarm dibuat dengan menggunakan *Unity* dan menggunakan gambar – gambar yang diubah menjadi sebuah *game* berbasis komputer. Berikut adalah implementasi *game* EduFarm:

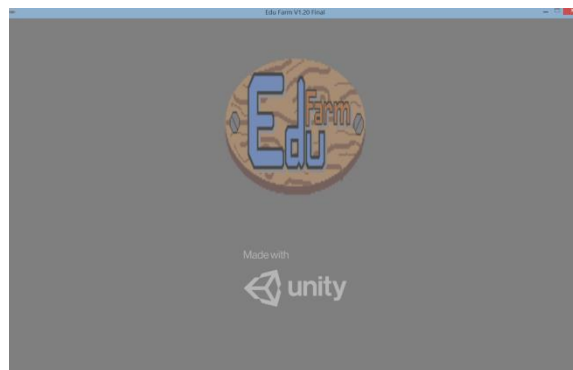
4.1.1. Tampilan Pertama pada *Unity*

Tampilan pertama pada *unity* terdapat *setting* sebelum di game berburu koi di play yaitu screen resulation , graphics qulity, select monitor.



Gambar 4.1 Tampilan Configuration

4.1.2. Tampilan *Splash Screen Unity*



Gambar 4.2 Tampilan *Splash Screen Game* EduFarm
Pada Gambar 4.2. adalah tampilan *Splash Screen* saat masuk kedalam Game EduFarm.

4.1.3. Tampilan Utama pada *Game*

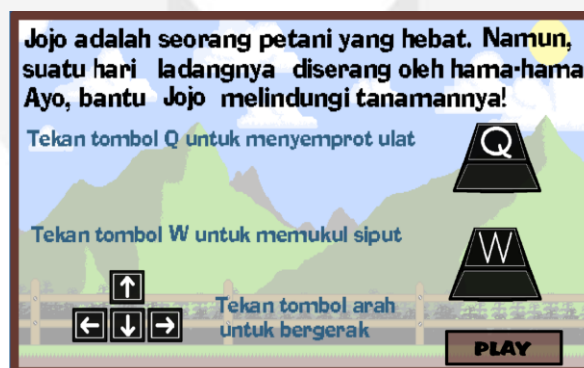


Gambar 4.3 Menu Utama

Pada gambar 4.3 adalah tampilan menu utama game EduFarm yang terdapat 3 *button* yaitu *button play*, *tutorial*, dan *quit*.

- *Button play* : untuk memulai *game*
- *Button tutorial* : untuk melihat cara memainkan *game*
- *Button quit* : untuk keluar dari *game*

4.1.4. Tampilan *Tutorial*



Gambar 4.4 *Tutorial*

Pada gambar 4.4 adalah tampilan *tutorial* yang berisikan langkah-langkah untuk bermain game EduFarm dan terdapat *button play* yang berfungsi untuk masuk ke dalam permainan.

4.1.5. Tampilan Level



Gambar 4.5 Tampilan Level 1

Pada gambar 4.5 adalah tampilan level yaitu level dalam game EduFarm dan terdapat beberapa *asset* di dalam level pertama yaitu :

- *Timer* : target waktu untuk memainkan game pada level pertama berada di sebelah tengah
- papan *score* : jumlah ulat dan siput yang di basmi petani *user* , berada di sebelah kanan tampilan

4.1.6. Tampilan Game Over



Gambar 4.8 Game Over

Pada gambar 4.8 adalah tampilan *game over* yang berfungsi ketika petani *user* gagal melewati tantangan tiap *level*

4.2 Pengujian Black box

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Berikut ini adalah hasil pengujian dari aplikasi yang dibangun :

Table 4.1 Deskripsi Data Hasil Uji

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Keluaran yang diharapkan	Hasil yang Didapat	Hasil uji	
					Diterima	Ditolak
A.1	Tombol play	Menekan tombol play	Aktif	Play aktif	✓	
A.2	Tombol quit	Menekan tombol quit	Aktif	Quit aktif	✓	
A.3	Tombol tutorial	Menekan tombol tutorial	Aktif	Tutorial aktif	✓	
A.5	Tombol menu ketika game selesai	Menekan tombol menu	Aktif	Kembali ke menu utama	✓	
A.5.5	Tombol level 2 ketika game selesai	Menekan tombol level 2	Aktif	Game berjalan	✓	
A.6	Tombol menu ketika game	Menekan tombol	Aktif	Kembali ke menu	✓	

ID Pengujian	Deskripsi Pengujian	Prosedur Pengujian	Keluaran yang diharapkan	Hasil yang Didapat	Hasil uji	
					Diterima	Ditolak
	selesai	menu		utama		
A.6.6	Tombol level 3 ketika game selesai	Menekan tombol level 3	Aktif	Game berjalan	✓	
A.7	Tombol back to menu di scene game over	Menekan tombol back to menu	Aktif	Kembali ke menu utama	✓	

4.3 Pengujian Beta

Pengujian beta dilakukan pada 30 responden selama 4 hari pada tanggal 8 sampai 11 juli 2019. Dari kuisioner yang telah diberikan kepada responden, data dianalisa menggunakan model skala likert yaitu:

Rumus Index % = Total skor/Y*100

dimana,

Y = Skor tertinggi likert * jumlah responden

X = Skor terendah * jumlah responden

Total skor = T * Pn

dimana,

T = Total jumlah responden yang memilih

Pn = Pilihan angka skor likert

Tabel 4.2 Presentase Nilai

Jawaban	Keterangan
0%-24,99%	Tidak Setuju
25%-49,99%	Cukup Setuju
50%-74,99%	Setuju
75%-99,99%	Sangat Setuju

Rekap hasil kuesioner sebagai berikut :

Tabel 4.3 Rekap Hasil Kuesioner

No.	Pertanyaan	Presentase Nilai (%)	Keterangan
Aspek Sistem			
1	Apakah tampilan game EduFarm menarik?	74.16%	Setuju
2	Apakah kombinasi desain dan warna pada game EduFarm menarik?	79.16%	Setuju
3	Apakah sound sudah tepat?	76.6%	Sangat Setuju
4	Apakah game beburu koi layak dipublikasi?	70%	Setuju
Aspek Pengguna			
1	Apakah instruction pada game EduFarm mudah dipahami?	77.5%	Sangat Setuju
2	Apakah tingkat kesulitan game EduFarm cukup?	70.83%	Setuju
3	Apakah misi tiap sudah cukup?	67.5%	Setuju
4	Apakah game EduFarm sangat menghibur?	77.5%	Sangat Setuju
Aspek Interaksi			
1	Apakah menu pada game EduFarm mudah di pahami?	50%	Setuju
2	Apakah karakter game EduFarm menarik?	73.33%	Setuju
3	Apakah kontrol dari game mudah di pahami?	75.8%	Sangat Setuju

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil tugas akhir ini dan pengujian yang dilakukan maka dapat diambil kesimpulan :

1. Karakter Ulat (NPC) dapat berinteraksi dengan karakter lain yang ada di dalam game
2. Game EduFarm sudah dapat berjalan baik dan dapat dimainkan di computer atau laptop
3. Game EduFarm berhasil menjadi game yang menghibur penggunanya

5.2 Saran

Saran yang dapat diberikan penulis untuk pengembangan selanjutnya:

1. Sistem didalam *game* masih terdapat kekurangan diantaranya ulat dan siput masih terkadang terpental keluar dari area, karakter ulat dan siput ideal pergerakannya masih belum berfungsi dengan baik. Maka dari itu masih diperlukan pengembangan lebih lanjut.
2. Kedepannya *game* EduFarm dapat dikembangkan dalam bentuk *mobile*.

DAFTAR PUSTAKA

- [1]Albir Darama M, “Dampak Game Terhadap Perkembangan Anak”, Fakultas Ilmu Pendidikan, Universitas Negeri Semarang, Desember 2013.
- [2]M.S. Purohit, “ *Game theoretic multi-agent approach to traffic flow control*”, Juni 2013.
- [3]YangBo, “*Evolutionary simulation game based on the module of Complex Multi-Agent System*”, Oktober 2009.
- [4]Pamuju Ronny, Bambang Tri Rahardjo, Hagus Tarno, “ POPULASI DAN SERANGAN HAMA ULAT KANTUNG *Metisa plana Walker* (Lepidoptera; Psychidae) SERTA PARASITOIDNYA DI PERKEBUNAN KELAPA SAWIT KABUPATEN DONGGALA, SULAWESI TENGAH”, Jurnal HPT Volume 1 Nomor 2, Juni 2013.
- [5]Suholid Dede, “EFEKTIVITAS PENGENDALIAN HAMA ULAT KANTUNG (*Mahasena corbetti*) PADA TANAMAN KELAPA SAWIT SECARA KIMIAWI DI KEBUN MARIHAT PT. PERKEBUNAN NUSANTARA IV, Sekolah tinggi ilmu pertanian, Oktober 2016.
- [6]Gansala Fauzadin, “ PERILAKU *NON PLAYABLE CHARACTER* (NPC) MUSUH PADA GAME SEPEDA MENGGUNAKAN *FUZZY STATE MACHINE* (FuSM)”, Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang, 9 Juni 2016.
- [7]Pesona Wildan Nazar, “ DESAIN *NON PLAYABLE CHARACTER* SEBAGAI MUSUH PADA GAME SEPEDA MENGGUNAKAN METODE *MARKOV STATE MACHINE*.” Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang, 8 Juni 2016.
- [8]Teguh Irawan Fathony, Eriq Muh. Adams Jonemaro, Muhammad Aminul Akbar, “ Implementasi Multi-Agent Path Finding Menggunakan Algoritma Conflict – Based Search Pada Game Bergenre Adventure – Puzzle”, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 10, hlm. 3827-3834, Oktober 2018.
- [9]Hollins Wray Kyle, Akshat Kumar, Shlomo Zillberstein, “ Integrated Cooperation and Competition in Multi – Agent Decision – Making”, Collage of Information and Computer Sciences, University of Massachusetts, Amherst, MA, USA, School of Information System, Singapore Management University, Singapore, April 2013.
- [10]Zani Wibowo M, Lutfi Pratama, Iis Pradesan, “Rancang Bangun Aplikasi Permainan “*Help Your Mom*” Menggunakan Algoritma *Floyd-Warshall*”, STMIK GI MDP Program Studi Teknik Informatika, STMIK GI MDP Palembang, April 2013.
- [11]Widya Ningrum Friska dan Andrasto Tatyantoro, “ Penerapan Algoritma *Floyd-Warshall* Dalam Menentukan Rute Terpendek pada Pemodelan Jaringan Pariwisata di Kota Semarang”, Jurnal Teknik Elektro Vol. 8 No.1, Januari – Juni 2016.