IMPLEMENTASI ALGORITMA FINITE STATE MACHINE SEBAGAI PENENTU PERUBAHAN PERILAKU PADA NON-PLAYER CHARACTER DALAM GAME KIKI ADVENTURE

Muhammad Rizky Nafianto

Sistem Informasi, STMIK Widya Cipta Dharma
Jl. M. Yamin No.25, Samarinda, 75123
E-mail : nafianto.rizky@gmail.com.

ABSTRAK

Implementasi Algoritma *Finite State Machine* Pada *Non-player Character* Dalam *Game* Kiki *Adventure* merupakan penelitian yang bertujuan untuk menghasilkan sebuah *game* yang menarik dengan implementasi algoritma *finite state machine.*

Kata Kunci: *Game,Algoritma finite state machine,Android*

---

1. PENDAHULUAN

Game saat ini telah banyak dimainkan oleh banyak orang dari usia muda sampai tua. *Game* jenisnya semakin bervariasi sejalan dengan perkembangan tekhnologi perangkat keras dan perangkat lunak. Perkembangan *game* saat ini berkembang pesat, saat ini tidak hanya dimainkan di komputer, tetapi saat ini juga dapat dimainkan pada handphone android sehingga dapat dimainkan dimana saja.. Pada awalnya *game* identik dengan anak-anak. Para orang dewasa selalu berpikir *game* merupakan suatu kegiatan yang dilakukan oleh anak-anak yang dapat menyenangkan hati mereka.

Dengan kata lain, segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai *game*. Didalam *game* biasanya terdapat musuh yang sering disebut juga dengan *non-player characte*r (NPC) yang merupakan karakter antagonis bagi pemain utama. Tugas musuh tersebut adalah untuk menghalangi *player* untuk mencapai tujuan. *Game* tidak akan menarik bila musuh tidak memiliki gerakan atau perilaku untuk menghalangi *player*. Dengan adanya algoritma *finite state machine* pada musuh akan membuat *game* lebih menarik dan menyenangkan karena musuh dapat berubah perilaku sesuai dengan *input*

Berdasarkan latar belakang tersebut, maka diambil tema skripsi ini dengan judul ”Implementasi Algoritma *Finite State Machine* Sebagai Penentu Perubahan Perilaku Pada *Non-Player Character* Dalam *Game* Kiki *Adventure*”.

2. RUANG LINGKUP PENELITIAN

1. Cakupan permasalahan

Dari permasalahan diatas maka menarik untuk dibuatnya suatu algoritma yang diharapkan dapat menjadikan game semakin menarik maka dalam hal ini rumusan masalah yang dikemukakan adalah “Bagaimana implementasi algoritma finite state machine sebagai penentu perubahan perikau pada non-player character dalam game kiki adventrure?”.

2. Batasan-batasan penelitian

Berdasarkan cakupan masalah yang dibuat, maka ditetapkan batasan-batasan pada sistem yang akan dibangun. Sangat diperlukan batasan masalah yang meliputi sebagai berikut, yaitu:

1. Aplikasi *game* yang akan dibangun bergrafis 2D.
2. Aplikasi permainan ini berjenis *game offline* dan *game* ini hanya dapat dimainkan oleh satu orang atau *single player*.
3. *Game* terdiri dari 4 *level*.
4. Aplikasi *game* ini dibangun menggunakan aplikasi Unity 5.4.2
5. *Target user* pengguna mulai dari usia 6 tahun keatas.
6. *Game* ini hanya dapat dimainkan di perangkat android dengan versi 4.1 keatas

3. BAHAN DAN METODE

Adapun bahan dan metode yang digunakan dalam membangun aplikasi ini yaitu:

3.1 *Game*

Menurut Ismail (2009), *Game* ada dua pengertian. Pertama *game* adalah sebuah aktifitas bermain yang murni mencari kesenangan tanpa mencari menang atau kalah. Kedua, *game* diartikan sebagai aktifitas bermain yang dilakukan dalam rangka mencari kesenangan dan kepuasan namun ditandai pencarian menang-kalah.

3.2 Animasi

Menurut Vaughan (2011), Animasi adalah tindakan membuat sesuatu menjadi hidup. Dengan animasi, serangkaian gambar diubah secara perlahan dan sangat cepat, satu sesudah yang lain sehingga tampak berpadu kedalam ilusi visual gerak. Efek visual seperi *wipe*, *fade*,

zoom, dan *dissolve* merupakan bentuk animasi sederhana. Sebelum video seperti QuickTime dan AVI video menjadi umum, animasi adalah sumber utama aksi dinamis dalam presentasi multimedia..

3.3 Kecerdasan Buatan

Menurut Rich dan Knight (2009), Kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

3.4 Algoritma

Menurut Thomas H. Cormen (2009:5), Algoritma adalah prosedur komputasi yang mengambil beberapa nilai atau kumpulan nilai sebagai *input* kemudian di proses sebagai *output* sehingga algoritma merupakan urutan langkah komputasi yang mengubah input menjadi output.

3.5 Teori Otomata

Menurut Nugroho (2013) Teori otomata adalah kajian mengenai suatu perangkat komputer yang bersifat khayali yang kerap disebut dengan istilah mesin *(machine*). Sebelum lahirnya komputer, tepatnya pada tahun 1930an, Alan Turing telah mempelajari sejenis mesin abstrak yang mempunyai seluruh kemampuan sseperti kemampua komputer dimasa sekarang, setidaknya dalam hal apa yang dapat dikerjakan dan apa yang tidak dapat dikerjakan.

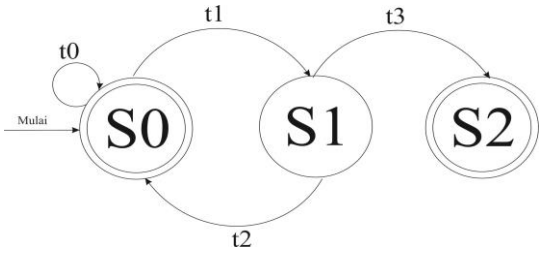
3.6 Finite State Machine

Menurut Rich (2009), *Finite state machine* didefenisikan sebagai perangkat komputasi yang memiliki *input* dan *output* yang merupakan satu dari dua nilai yang dapat di-*accept* dan *reject* . *Finite state machine* dapat mendefinisikan suatu set kondisi yang menentukan kapan suatu bagian harus berubah ke bagian yang lain. Dari *finite state machine* dapat mengetahui bagaimana bagian dari objek *game* berperilaku.

4. RANCANGAN SISTEM

Berikut ini adalah rancangan yang digunakan dalam membangun game kiki adventure:

4.1 *Finite State Machine pada NPC Bat*



Gambar 4.1 *Finite State Machine Pada NPC Bat*.

Keterangan :

S0 : *State* dimana NPC *bat* dalam posisi *stand by/* tidak bergerak

S1 : *State* dimana NPC *bat* mendekati dan menyerang *player*

S2 : *State* dimana NPC *bat* hancur/ menghilang

t0 : Kondisi dimana *player* berada di luar jarak serang NPC *bat*

t1 : Kondisi dimana *player* memasuki jarak serang NPC *bat*

t2 : Kondisi dimana *player* keluar dari jarak serang NPC *bat*

t3 : Kondisi dimana tingkat kesehatan (*health)* NPC Bat bernilai 0

Berdasarkan Gambar 4.1 *deterministic finite state machine* (DFSM) mempunyai karakteristik sebagai berikut :

Q = {S0,S1,S2}

Σ = {t0,t1,t2,t3}

S = S0

F = {S0,S2)

Dan fungsi transisi sebagai berikut :

Tabel 4.1 Fungsi Transisi NPC *Bat*

δ	t0	t1	t2	t3
S0	S0	S1	ε	ε
S1	ε	ε	S0	S2
S2	ε	ε	ε	ε

Maka :

δ(S0,t0 t1 t2 t1 t3) = δ(S0, t1 t2 t1 t3) = δ(S1, t2 t1 t3) = δ(S0,t1 t3) = δ(S1,t3)

= δ S2 Diterima

δ(S0,t0 t1 t3) = δ(S0,t1 t3) = δ(S1, t3) = δ S2 Diterima

δ(S0, t1 t2 t1 t3) = δ(S1, t2 t1 t3) = δ(S0,t1 t3) = δ(S1,t3) = δ S2 Diterima

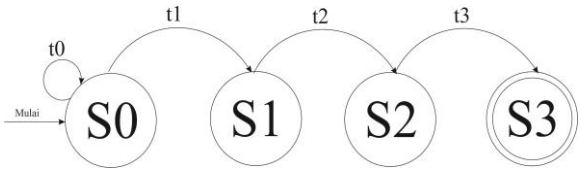
δ(S0,t1 t3) = δ(S1, t3) = δ S2 Diterima

δ(S0,t0) = δ S0 Diterima

δ(S0,t0 t1 t3) = δ(S0,t1 t2) = δ(S1, t2) = δ S0 Diterima

δ(S0, t1 t2) = δ(S1, t2) = δ S0 Diterima

4.2 *Finite State Machine pada NPC Boss*



Gambar 4.2 *Finite State Machine Pada NPC Boss*.

Gambar 4.2 Bagan Deterministic Finite State Machine (DFSM) Pada NPC Boss

Keterangan :

S0 : State dimana NPC Boss melakukan serangan dengan mendekati karakter utama.

S1 : State dimana NPC Boss menciptakan 2 clone dengan indikator darah 100 poin dan kecepatan sebesar 1 poin

S2 : State dimana NPC Boss menciptakan 2 clone dengan indikator darah 50 poin dan kecepatan sebesar 5 poin

S3 : State dimana NPC Boss hancur/ menghilang

t0 : Kondisi dimana NPC Boss memiliki tingkat kesehatan 200 poin

t1 : Kondisi dimana NPC Boss memiliki tingkat kesehatan 150 Poin
t2 : Kondisi dimana NPC Boss memiliki tingkat kesehatan 100 poin
t3 : Kondisi dimana NPC Boss memiliki tingkat kesehatan 0 poin

Berdasarkan Gambar 4.2, deterministic finite state machine (DFSM) mempunyai karakteristik sebagai berikut :

Q = {S0,S1,S2,S3}
Σ = {t0,t1,t2,t3}
S = S0
F = {S3)

Dan fungsi transisi sebagai berikut :

Tabel 4.2 Fungsi Transisi NPC Boss

δ	t0	t1	t2	t3
S0	S0	S1	ε	ε
S1	ε	ε	S2	ε
S2	ε	ε	ε	S3
S3	ε	ε	ε	ε

Maka :

δ(S0, t0 t1 t2 t3) = δ(S0, t1 t2 t3) = δ(S1,t2 t3) = δ(S2,t3) = δ S3 Diterima

δ(S0, t1 t2 t3) = = δ(S1,t2 t3) = δ(S2,t3) = δ S3 Diterima

5. IMPLEMENTASI

1.1 Pembuatan Halaman Menu Utama

Pada halaman menu utama di dalamnya terdapat beberapa tombol yaitu Mulai, Instruksi, Cerita, Tombol Suara dan Keluar . Bila tombol Mulai Ditekan maka akan menuju halaman permainan tingkat pertama. Bila tombol Instruksi ditekan, maka akan menuju halaman instruksi. Bila tombol Cerita ditekan, maka akan menuju halaman cerita.

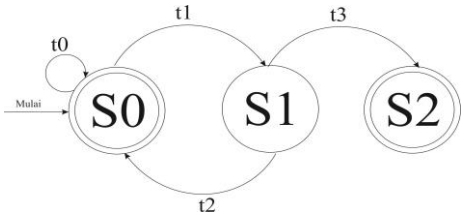
Gambar 5.1 Tampilan Halaman Menu Utama

1.2 Pembuatan Halaman Permainan

Pada halaman permainan ini di dalamnya terdapat beberapa tombol yaitu tombol kiri untuk bergerak ke kiri. Tombol kanan untuk bergerak ke kanan. Tombol lompat untuk melompat.Tombol serang untuk melakukan serangan.

Gambar 5.2 Tampilan Halaman Permainan

1.3 Implementasi Algoritma FSM Pada NPC *Bat*



Gambar 5.2 *Algoritma Finite State Machine* NPC *Bat*

Pada *Flying Enemy Script* terdapat kode :

playerinRange = Physic2D.OverlapCircle (transform.position, playerRange, playerLayer);

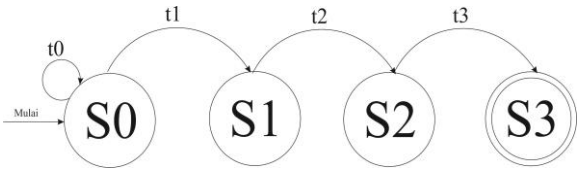
Dimana kode tersebut untuk membuat area deteksi di sekitar NPC *Bat* untuk mendeteksi keberadaan karakter utama. Bila karakter utama berada di luar jangkauan area deteksi NPC tersebut yang merupakan transisi t0, maka NPC tersebut berada pada *state* S0 dimana NPC berada pada keadaan diam.

Jika karakter utama memasuki area deteksi NPC *Bat* yang merupakan transisi t1, maka NPC tersebut akan berpindah *state* dari S0 ke S1 dimana NPC berada pada keadaan mengejar karakter utama untuk menyerangnya. Dan transisi t2 dimana karakter utama keluar ke area deteksi NPC tersebut dan berpindah dari *state* S1 ke S0 dimana NPC dalam keadaan diam. Kondisi tersebut dapat berjalan dengan kode sebagai berikut :

```
if (playerinRange)
{if (thePlayer.transform.position.x > transform.positio
n.
transform.position = Vector3.MoveTowards (transfor
m.position, thePlayer.transform.position, movespeed *
 Time.deltaTime);
 transform.localScale = new Vector3 (0.5f, 0.5f, 1f); }
if (thePlayer.transform.position.x < transform.position
.x) { transform.position = Vector3.MoveTowards (tran
sform.position, thePlayer.transform.position, movespe
ed * Time.deltaTime);
transform.localScale = new Vector3 (-0.5f, 0.5f, 1f) }}
```

Penjelasan dari kode tersebut adalah bila karakter utama memasuki area deteksi NPC *Bat*, maka NPC tersebut akan bergerak kearah karakter utama dan akan berhenti bila karakter utama keluar dari area deteksi NPC Bat.

1.4 Implementasi Algoritma FSM Pada NPC *Boss*



Gambar 5.3 *Algoritma Finite State Machine* NPC *Boss*

Pada transisi t0 yaitu NPC *boss* memiliki indikator darah sebesar 200 poin, maka NPC *boss* akan melakukan serangan pada karakter utama dengan mendekatinya (*state* S0). Pada kode *enemy boss script* ini merupakan kode transisi t1 yaitu *state* S0 berpindah ke S1 dengan kode sebagai berikut :

```
if (currenthealth == 150) {
makeclone ();
 }
```

Dimana bila NPC *boss* memiliki indikator darah sebesar 150 poin atau transisi t2, maka peritah makeclone(); akan dikerjakan dengan kode berisi:

```
 GameObject clone1 = Instantiate(bossprefabs, new V
ector3(transform.position.x + 4f, transform.position.y
+ 3f, transform.position.z), transform.rotation) as Gam
eObject;
GameObject clone2 = Instantiate(bossprefabs, new Ve
ctor3(transform.position.x + 8f, transform.position.y -
3f, transform.position.z),transform.rotation)  as Game
Object;
clone1.transform.localScale = new Vector3(transform.
transform.localScale.z , transform.localScale.y * 0.5f,
transform.localScale.z);
clone1.GetComponent <enemyhealthoss>().enemyhea
lth = 100;
clone1.GetComponent <FlyEnemy1> ().movespeed =
1;
clone2.transform.localScale = new Vector3(transform.
transform.localScale.z , transform.localScale.y * 0.5f,
transform.localScale.z);
clone2.GetComponent <enemyhealthoss>().enemyhea
lth = 100;
clone2.GetComponent <FlyEnemy1> ().movespeed =
1;
}
```

Dimana kode tersebut menjalankan perintah *state* S1 yaitu menciptakan 2 *clone* dari NPC *boss* dengan memiliki indikator darah sebesar 100 poin dengan kecepatan sebesar 1 poin. Didalam *enemy boss script* juga terdapat kode seperti berikut :

```
if (currenthealth == 100) {
makeclone2 ();
 }
```

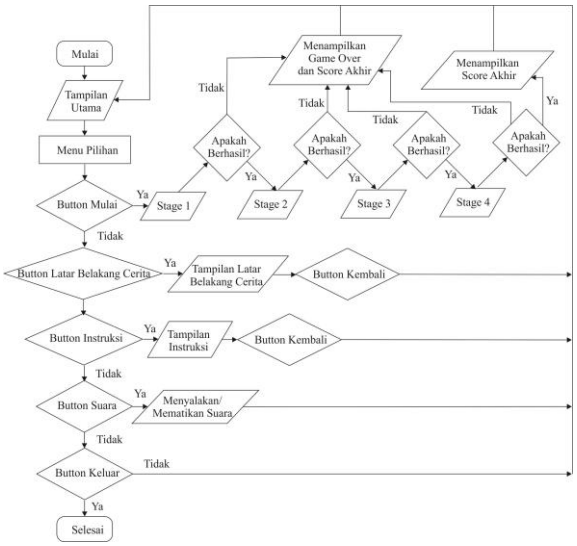
Dimana kode tersebut bila NPC *boss* memiliki indikator darah sebesar 100 poin, atau transisi t2, maka perintah makeclone2(); akan dikerjakan dengan kode berisi:

```
 GameObject clone1 = Instantiate(bossprefabs, new V
ector3(transform.position.x + 4f, transform.position.y
+ 3f, transform.position.z), transform.rotation) as Gam
eObject;
GameObject clone2 = Instantiate(bossprefabs, new Ve
ctor3(transform.position.x + 8f, transform.position.y -
3f, transform.position.z),transform.rotation)  as Game
Object;
clone1.transform.localScale = new Vector3(transform.
transform.localScale.z , transform.localScale.y * 0.5f,
transform.localScale.z);
clone1.GetComponent <enemyhealthoss>().enemyhea
lth = 50;
clone1.GetComponent <FlyEnemy1> ().movespeed =
5;
clone2.transform.localScale = new Vector3(transform.
transform.localScale.z , transform.localScale.y * 0.5f,
transform.localScale.z);
clone2.GetComponent <enemyhealthoss>().enemyhea
lth = 50;
clone2.GetComponent <FlyEnemy1> ().movespeed =
5;
 }
```

Dimana kode tersebut menjalankan perintah *state* S2 yaitu menciptakan 2 *clone* dari NPC *boss* dengan memiliki indikator darah sebesar 50 poin dengan kecepatan sebesar 5 poin. Saat indikator darah NPC *boss* sebesar 0 poin atau transisi t3 maka *state* S3 dilaksanakan yaitu NPC *boss* akan hancur seperti dengan kode Destroy (gameObject);

1.5 *Flowchart* Permainan

. Dalam mengembangkan *game* ini, dibutuhkan *flowchart* untuk mengetahui alur atau jalan *game* yang akan dibuat. *Flowchart* dimulai dari aplikasi *game* dibuka sampai dengan mengakhiri *game*. Berikut adalah *flowchart game* Kiki *adventure*:



Gambar 5.4 *Flowchart* Permainan

1.6 Desain Permainan

Image atau gambar merupakan salah satu pendukung untuk aplikasi *game* Kiki *adventure*. Gambar berupa gambar karakter pemain utama dan musuh, yang ada pada *game* tersebut didesain dengan tampilan yang menggemaskan sehingga tampilan aplikasi tidak membosankan. Berikut ini adalah beberapa gambar yang digunakan dalam *game* Kiki *adventure* :

Tabel 5.1 Gambar Yang Digunakan Dalam *Game* Kiki *Adventure*

No	Gambar	Keterangan
1		Kiki merupakan karakter utama yang bertugas untuk mengalahkan makhluk jahat
2		*Slime* merupakan musuh dari karakter utama yang berjalan didarat dan menyerang pemain dengan mendekatinya
3		*Bat* merupakan musuh karakter utama yang dapat terbang mengikuti karakter utama dan menyerangnya
4		*Boss Bat* merupakan musuh terakhir yang akan berhadapan pada karakter utama
5		*Background* menu merupakan gambar latar belakang yang digunakan pada menu utama
6		Tombol dan navigasi yang digunakan didalam *game* tersebut.

6. KESIMPULAN

Dari permasalahan yang ada penulis memberi kesimpulan sebagai berikut:

1. Aplikasi *game* Kiki *adventure* berbasis *android* dibangun dari tahapan pengembangan multimedia mulai tahap *concept, design, material collecting, assembly, testing, dan distribution.*
2. Impelementasi algoritma *finite state machine* dapat diterapkan pada *game* berbasis *android*.
3. Impelementasi algoritma *finite state machine* dapat diterapkan berdasarkan indikator darah pada NPC dan juga jarak antara NPC dan karakter utama.
4. Perubahan perilaku NPC dengan menerapkan algoritma *finite state machine* terjadi berdasarkan dari *input* yang diberikan oleh karakter utama

7. SARAN

Berdasarkan kesimpulan diatas, maka penulis ingin menyampaikan beberapa saran sebagai berikut :

1. Perlu adanya perbaikan maupun fitur berbasis 3D sehingga *game* tersebut terlihat menarik
2. Permainan ini hanya berbasis *game android offline*. Diharapkan untuk penelitian selanjutnya ada mahasiswa yang mampu mengembangkan *game* menjadi berbasis *online*
3. Diharapkan untuk penelitian selanjutnya mahasiswa dapat mengembangkan game yang bersifat *multiplayer* atau dapat dimainkan lebih dari satu orang.

8. DAFTAR PUSTAKA

Al Fatta, Hanif, 2007, Analisis dan Perancangan Sistem Informasi, Yogyakarta: Andi

Arif, Yunita M., Kurniawan, F., Nugroho Fresi. 2011, Desain Perubahan Pada NPC Game Menggunakan Logika Fuzzy. Laporan Skripsi, Fakultas Sains Dan Teknologi. Malang. Universitas Islam Negri Maulana Malik.

Brent Ellison. 2008. *Defining Dialogue Systems*. http//:Gamasutra.com/, diakses pada 10 November 2016

Hendrawan. 2013. Membangun Game The Kingdom Of Pandawa Berbasis Turn Based Strategy, Laporan Skripsi, Fakultas Teknik Dan Ilmu Komputer. Surabaya. Universitas Komputer Indonesia.

LloydEvarsog, Http://freesound.org/people/LloydEvarsog/sounds/186405 diakses pada 18 Juli 2017

Bumpelsnake, Http://freesound.org/people/bumpelsnake/sounds/267239 diakses pada 18 Juli 2017

Streety, Http://freesound.org/people/streety/sounds/30247 diakses pada 18 Juli 2017

Dheming, Http://freesound.org/people/dheming/sounds/275947 diakses pada 18 Juli 2017

Flo_Rayen,Http://freesound.org/people/Flo_Rayen/sounds/191835 diakses pada 18 Juli 2017

Creative Common Zero, Http://gameart2d.com/the-knight-free-sprites.html diakses pada 18 Juli 2017

Creative Common Zero, Http://gameart2d.com/free-platformer-tiles.html diakses pada 18 Juli 2017

Huda, Ahmad. 2016, Game Edukasi Cepat Tepat Dengan Metode Finite State Machine Dengan Smartphone. Laporan Skripsi, Fakultas Sains Dan Teknologi. Malang. Universitas Islam Negri Maulana Malik.

Ismail, Andang. 2009. Education Games Panduan Praktsi Permainan Yang Menjadikan Anak Anda Cerdas, Kreatif Dan Saleh. Yogyakarta: Pro You Media

Munir, 2008, Kurikulum Berbasis Teknologi Informasi dan Komunikasi.
Bandung : Alabeta.

Nugroho, Adi Sulistyo, 2013, Teori bahasa dan Otomata, Yogyakarta : Graha Ilmu

Rich, E. 2009. Automata, Computability, and Complexity, Theory and Applications. United States of America : Pearson Prentice hall

Rich, Elaine and Knight, Kevin, 2009, Mengenal Artificial Intelegence, new Dehli: Tata McGraw-Hils

Rickman, Roedavan. 2014. Unity Tutorial Game Engine, Bandung: informatika

Rosa dan Shallahudin, 2011, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur Dan Berorientasi Objek), Bandung: Modula

STMIK Widya Cipta Dharma. 2015. Buku Pedoman Penulisan Usulan Proposal Dan Skripsi Jenjang Strata Satu , Samarinda: STMIK Widya Cipta Dharma

Theresia, Arie. 2011. Coredraw x5 & Adobe Photoshop CS5 Untuk Desain Logo Yogyakarta: Andi

Thomas, H Cormen, Charles E Leiserson. 2009. Introduction to Algoritms 3rd Ed. London. The MIT Press

Vaughan, W. 2011. Digital Modeling. New Riders : USA.

Wicaksana, Prima Oktava. 2014, Implementasi Fuzzy State Machine Sebagai Pembangkit Gerak Non-Player Character Pada Game Alibaba Sebagai Media Pembelajaran Tajwid. Laporan Skripsi, Fakultas Sains Dan Teknologi. Malang. Universitas Islam Negri Maulana Malik.