

PERANCANGAN DAN IMPLEMENTASI *FINITE STATE MACHINE* PADA GAME “CASTLE OF ILLUSION”

Achmad Badruddin

Program Studi Teknik Informatika S1, Fakultas Teknologi Industri
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia
Badruddin283@gmail.com

ABSTRAK

Game merupakan salah satu hiburan yang sangat digemari oleh semua kalangan baik dari anak-anak hingga orang dewasa sebagai sarana refreshing untuk mengisi waktu luang. *Game* dikembangkan dalam berbagai jenis salah satu yang paling populer adalah permainan petualangan, dimana permainan petualangan memiliki jalan cerita yang bermacam-macam, dan juga pada permainan petualangan selalu membuat para pemainnya penasaran akan akhir cerita dari *game* petualangan tersebut. *Game adventure Castle Of Illusion* merupakan permainan petualangan dimana pemain akan menjalankan karakter seorang kesatria yang harus menyelamatkan hasil bumi yg telah di curi oleh seorang penyihir jahat jahat dan juga para pengikutnya.

Dalam *game* ini menggunakan *game engine* Unity3D, dengan menerapkan kecerdasan buatan FSM (*Finite State Machine*). FSM (*Finite State Machine*) digunakan pada karakter NPC (*Non Playable Character*) yaitu karakter yang digerakan oleh kecerdasan buatan yang digunakan untuk mendukung *game* tersebut. Seperti karakter *Enemy* dan *Boss* yang menggunakan tiga hal yaitu : Keadaan, Kejadian, dan Aksi.

Dari pengujian yang telah dilakukan hasil dari kecerdasan buatan FSM (*Finite State Machine*) pada *game Adventure “Castle Of Illusion”* ini karakter NPC (*Non Playable Character*) dapat mendeteksi keberadaan *player* untuk menentukan aksi serangan dari musuh tanpa melibatkan pengguna *game*, dari hasil pengujian yang dilakukan pada 10 orang yang berbeda maka mendapatkan hasil sebagai berikut, 2 orang menyatakan cukup, 6 orang menyatakan baik, 2 orang menyatakan baik sekali.

Kata kunci : *game adventure, Castle Of Illusion, finite state machine, Unity3D*

1. PENDAHULUAN

1.1 Latar Belakang

Pada saat ini *video game* telah berkembang pesat, *game* adalah merupakan salah satu sarana refreshing, *game* merupakan salah satu sarana hiburan yang banyak diminati baik anak-anak hingga orang dewasa. Sebuah *game* harus memiliki jalan cerita dan goal yang dicapai, ada banyak jenis *game* berdasarkan tipenya antara lain *Adventure, Action, Real time strategy, Role playing game, Puzzle game*. *Adventure game* adalah *game* yang menuntut kemampuan pemainnya untuk berfikir dan menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter, menggunakan benda-benda yang tepat dan diletakkan ditempat yang tepat. *Gameplay* jenis ini adalah keharusan *player* memecahkan bermacam-macam *puzzle* melalui interaksi dengan orang di lingkungan dalam *game* tersebut. Berikut ini adalah contoh dari beberapa *game adventure* saat ini seperti *Super Mario, Kontra, dan Metal Slug*.

Ada banyak media atau perangkat yang dapat digunakan untuk memainkan sebuah *game* diantaranya *smartphone, komputer, dan konsol game*. Komputer adalah salah satu terget media yang paling banyak digunakan para pengembang di industri *game* untuk meluncurkan produknya. Hal ini dikarenakan komputer merupakan media yang paling cepat

berkembang dan memungkinkan *developer game* untuk memaksimalkan kinerja *game* mereka, hal ini dikarenakan dari sisi *hardware* yang cepat berkembang, dan makin banyaknya *game engine* yang dapat digunakan secara gratis.

Metode yang digunakan dalam *game “Castle Of Illusion”* ini menggunakan metode FSM (*Finite State Machine*), dimana metode tersebut berfungsi untuk pengambilan keputusan pada karakter NPC (*Non Playable Character*) yaitu karakter yang digerakkan oleh kecerdasan buatan yang digunakan untuk mendukung *game* tersebut seperti karakter musuh agar gerakan dan aksi dari musuh tanpa melibatkan pengguna *game*. Dari uraian diatas penulis ingin mengimplementasikan penggunaan metode FSM (*Finite State Machine*) untuk melakukan pembuatan *game* dengan judul “*Castle Of Illusion*” yang merupakan *game* 2D dengan *genre Side Scroller, Adventure*, menggunakan *Unity* sebagai *game engine*.

1.2 Rumusan Masalah

Rumusan masalah pada pembuatan *game Adventure “Castle Of Illusion”* ini adalah bagaimana menerapkan metode *Finite State Machine (FSM)* pada musuh untuk berinteraksi pada *player* agar dapat menyerang dengan aksi musuh tanpa melibatkan pemain dengan gerakan yang sudah dibuat secara otomatis.

1.3 Batasan Masalah

Agar penulisan dalam penelitian yang dibuat tidak keluar dari topik yang ditentukan, maka penelitian ini terdapat batasan permasalahan sebagai berikut:

1. Game “Castle Of Illusion” ini dibuat dengan menggunakan *game engine* Unity.
2. Game “Castle Of Illusion” menerapkan tipe *side scroller game*.
3. Game “Castle Of Illusion” ini bergenre *Adventure game*.
4. Game “Castle Of Illusion” ini memiliki 3 level yang berbeda dimana disetiap level terdapat bos level.
5. Karakter utama pada Game “Castle Of Illusion” ini adalah seorang cowboys yang misterius.
6. Kecerdasan dalam pembuatan *game* ini adalah FSM (*Finite State Machine*) untuk digunakan sebagai kondisi tindakan pada musuh.
7. Game ini dibuat untuk berjalan pada sistem operasi Windows.
8. Cerita dari *game* ini merupakan cerita fiksi yang dibuat oleh penulis.

1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang *game* “Castle Of Illusion” berbasis desktop
2. Mengimplementasikan Metode *Finite State Machine* untuk NPC (*Non Playable Character*) biasa pada *game* “Castle Of Illusion”

2. TINJAUAN PUSTAKA

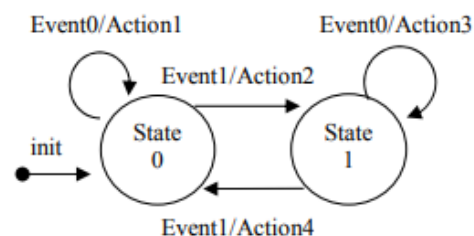
2.1. Game

Game pada dasarnya bersifat hiburan karena jika pengguna memainkan *game* maka akan terasa senang. Dalam era saat ini, *game* disajikan dengan kualitas visualisasi yang cukup canggih karena didukung oleh teknologi sehingga pemain lebih interaktif sesuai kemaunnya sendiri dan pemain terasa hidup dalam *game* tersebut. Maka bisa disebutkan bahwa *game* berkembang beriringan dengan teknologi.[2]

NPC (*Non-Playable Character*) adalah obyek bergerak atau karakter pada dunia *game* yang dijalankan oleh komputer dan bisa berinteraksi dengan pemain. Penelitian ini mengenai penggunaan *Finite State Machine* untuk mendapatkan variasi respon NPC dengan pemain pada *game*. Dengan adanya variasi respon NPC pada *game* tersebut diharapkan *game* akan menjadi lebih menarik untuk dimainkan karena respon NPC lebih sulit untuk diprediksi.[1]

2.2. FSM (Finite State Machine)

Finite State Machine adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut, *State* (Keadaan), *Event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif. Contoh diagram *state* sederhana ditunjukkan pada Gambar 2.1.



Gambar 2.1 Contoh Diagram State Sederhana.[4]

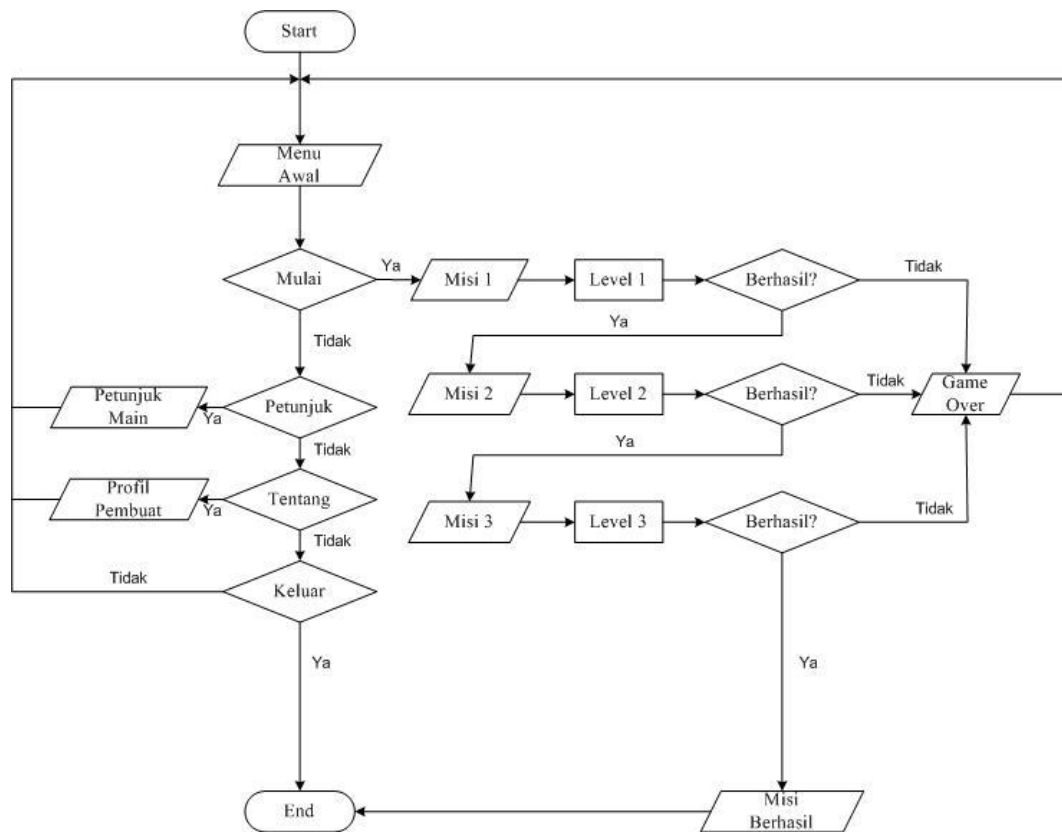
Diagram tersebut memperlihatkan FSM (*Finite State Machine*) dengan dua buah *state* dan dua buah *input* serta empat buah aksi output yang berbeda: seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*, sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

Salah satu alternatif implementasi FSM adalah menggunakan pemrograman berorientasi objek (*Object Oriented Programming*) atau yang sering disingkat sebagai OOP. Kelebihan penggunaan OOP pada FSM adalah fleksibilitasnya yang tinggi dan pemeliharannya yang mudah baik pada sistem yang sederhana, menengah, maupun sistem yang kompleks. Selain itu juga mendapatkan manfaat dari salah satu kelebihan OOP yaitu penggunaan kembali kode yang telah diketik (*code reusability*) sehingga pengetikan kode menjadi lebih sedikit.[3]

3. METODE PENELITIAN

3.1 Flowchart Game

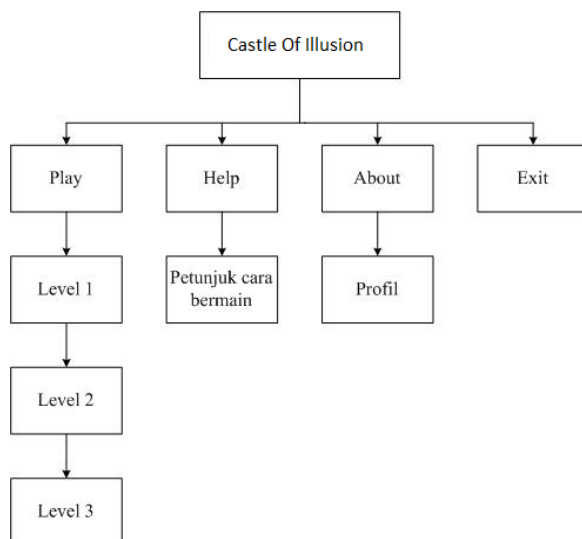
Pada perancangan *flowchart game* berfungsi untuk mengetahui alur proses dari alur program dimulai dari mulai *game* hingga selesai seperti gambar 3.1



Gambar 3.1 Flowchart Game

3.2 Struktur Menu

Pada game Adventure “Castle Of Illusion” terdiri dari beberapa menu yaitu *Play Game*, *Instruction*, *About*, dan *Exit*. Diagram struktur menu seperti Gambar 3.2

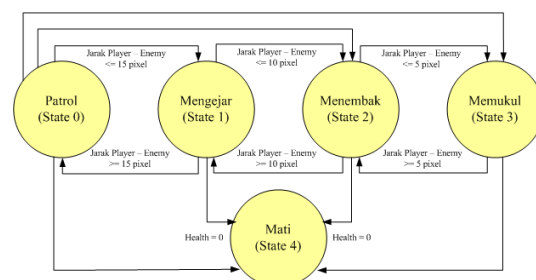


Gambar 3.2 Struktur Menu pada game

3.3 Penerapan FSM pada game

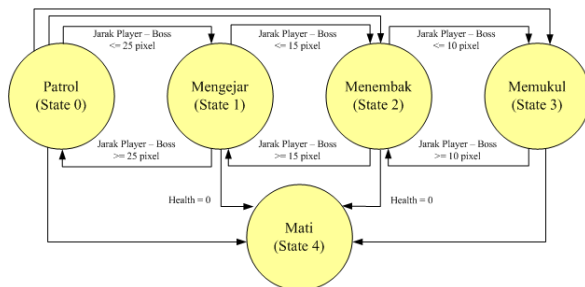
Finite State Machine adalah sebuah metodologi perancangan *system control* yang menggambarkan tingkah laku atau prinsip kerja dengan menggunakan tiga hal yang meliputi *State* (keadaan), *Event* (kejadian), dan *Action* (aksi). Pada satu saat dalam

periode waktu yang cukup signifikan, *system* akan berada pada salah satu *state* yang aktif. *System* dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri.



Gambar 3.3 FSM pada enemy

Penerapan alur FSM pada karakter *Enemy* Level memiliki sifat jika karakter *Player* mendekat pada jarak $< 15 \text{ pixel}$ dari *Enemy*, maka *Enemy* akan mengejar, jika *Player* mendekat lagi pada jarak $< 10 \text{ pixel}$ maka *Enemy* akan menembak *Player*, dan jika *Player* mendekat pada jarak $< 5 \text{ pixel}$ maka *Enemy* akan menyerang *Player*, apabila *player* menjauh dari *Enemy* dengan jarak $> 15 \text{ pixel}$ maka *Enemy* akan kembali *Patrol* pada tempat semula *Enemy* berada.



Gambar 3.4 FSM pada boss

Penerapan alur FSM pada karakter *Boss Level* memiliki sifat jika karakter *Player* mendekat pada jarak $< 25 \text{ pixel}$ dari *Boss*, maka *Boss* akan mengejar, jika *Player* mendekat lagi pada jarak $< 15 \text{ pixel}$ maka *Boss* akan menembak *Player*, dan jika *Player* mendekat pada jarak $< 10 \text{ pixel}$ maka *Boss* akan menyerang *Player*, apabila *player* menjauh dari *Boss* dengan jarak $> 25 \text{ pixel}$ maka *Boss* akan kembali *Patrol* pada tempat semula *Boss* berada.

4. HASIL DAN PEMBAHASAN

4.1. Tampilan Main Menu

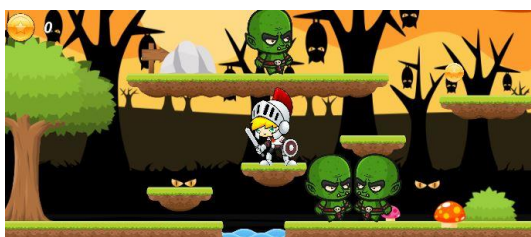
Tampilan awal yang akan muncul pada saat pemain menjalankan *Game Adventure Castle Of Illusion*. seperti Gambar 4.1



Gambar 4.1 Tampilan keseluruhan game

4.2. Tampilan Player Idle

Player akan berfungsi dengan menekan tombol A untuk berjalan ke kiri dan D untuk berjalan ke kanan seperti pada Gambar 4.4



Gambar 4.2 Tampilan Player Idle

4.3. Tampilan Player Shooter

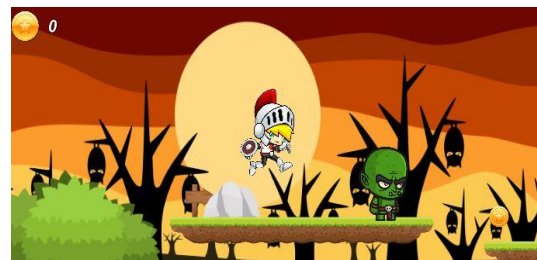
Shoot akan berfungsi dengan menekan tombol V seperti pada Gambar 4.4



Gambar 4.3 Tampilan Player Shoot

4.4. Tampilan Player Jump

Jump akan berfungsi dengan menekan tombol space seperti pada Gambar 4.4



Gambar 4.4 Tampilan Player Jump

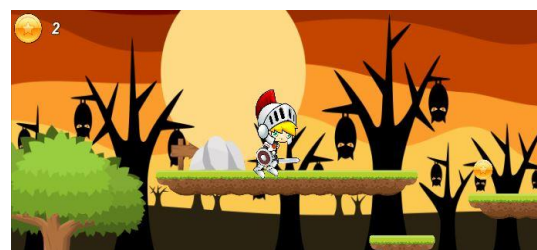
4.5. Tampilan Play Game

Attack akan berfungsi dengan menekan tombol shift seperti pada Gambar 4.4



Gambar 4.5 Tampilan Player Attack

Pada Gambar 4.6 Jika Player membunuh enemy akan menambah coin seperti pada Gambar 4.6



Gambar 4.6 Tampilan Coin Plus

Pada Gambar 4.7 Jika player tidak terkena sight enemy akan patrol seperti pada Gambar 4.7



Gambar 4.7 tampilan enemy patrol

Pada Gambar 4.8 menggambarkan enemy menyerang player karena player mengenai zona sight dari enemy seperti pada Gambar 4.8



Gambar 4.8 Tampilan enemy menyerang player

Gambar 4.9 menggambarkan ketika karakter player menyerang enemy dan health bar enemy habis maka enemy akan mati.



Gambar 4.9 Karakter Enemy Mati

Gambar 4.9 menggambarkan ketika karakter player terkena serangan dari enemy dan health bar player habis maka player akan mati.

4.6. Pengujian AI (Artificial Intelligence)

Pengujian *artificial intelligence* adalah pengujian mengenai fungsi yang berkaitan dengan *artificial intelligence* yang ada pada *Game Castle Of Illusion*. Hasil pengujian dari *artificial intelligence* pada game ini seperti pada table 1

Tabel 1 Pengujian artificial intelligence

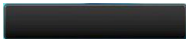
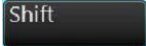
No	State	Event	Action	Hasil
1	Patrol State	Jika <i>enemy</i> tidak menjangkau sight <i>player</i>	Maka <i>enemy</i> pada kondisi patrol state	Sesuai
2	Patrol State	Jika <i>enemy</i> tidak menjangkau sight <i>player</i>	Maka <i>enemy</i> tetap pada posisi	Sesuai

		dalam jangka waktu 5 detik	patrol state	
3	Condition State	Jika <i>enemy</i> menjangkau sight dengan jarak < 10 pixel karakter <i>player</i>	Maka <i>enemy</i> akan melakukan tindakan mengejar <i>player</i>	Sesuai
4	Action State	Jika <i>enemy</i> menjangkau sight dengan jarak < 5 pixel karakter <i>player</i>	Maka <i>enemy</i> akan melakukan tindakan menyerang karakter <i>player</i>	Sesuai
5	Patrol State	Jika <i>player</i> menjauhi jarak pada jangkauan > 10 pada <i>enemy</i>	Maka <i>enemy</i> akan kembali pada posisi idle state dan patrol state	Sesuai

4.7. Pengujian Control Player

Pengujian *control player* adalah pengujian setiap fungsi dari tombol yang sudah diterapkan untuk menggerakkan karakter utama. Hasil pengujian *player* pada table 2

Tabel 2 Pengujian control player

No	Tombol Keyboard	Fungsi	Hasil
1		Space Button pada keyboard berfungsi untuk <i>player</i> melompat	Sesuai
2		V button pada keyboard berfungsi agar <i>player</i> dapat menembak	Sesuai
3		Shift Button pada keyboard berfungsi untuk <i>player</i> melakukan Attack	Sesuai
4		<i>Player</i> bergerak ke kiri	Sesuai
5		<i>Player</i> bergerak ke kanan	Sesuai

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah pembuatan *Game Adventure "Castle Of Illusion"*, maka penulis dapat mengambil kesimpulan:

1. Hasil dari pengujian Kecerdasan Buatan yang diterapkan pada *enemy* berjalan dengan baik dan sesuai dengan tingkat prosentase 100%.
2. Hasil dari pengujian Kecerdasan Buatan yang diterapkan pada *Boss Level* berjalan dengan baik dan sesuai dengan tingkat prosentase 100%.
3. Hasil dari pengujian kontrol pada *player* berfungsi dengan baik dan sesuai dengan tingkat prosentase 100%.
4. Hasil dari pengujian Fungsional yang dilakukan pada 4 sistem operasi yang berbeda yaitu *Windows 7*, *Windows 8*, *Windows 8.1* dan *Windows 10*, berjalan dengan baik dan sesuai dengan tingkat prosentase 100%.

5.2. Saran

Setelah dilakukan pengujian terhadap *game adventure "Castle Of Illusion"* maka masih ada kekurangan sehingga untuk pengembangan lebih lanjut disarankan :

1. Harapan penulis semoga aplikasi ini dapat di kembangkan dengan penambahan fitur- fitur yang lebih menarik.

2. *Game adventure "Castle Of Illusion"* dapat dijalankan pada berbagai macam *platform* yang berbeda.
3. Dapat mengembangkan lagi *game adventure "Castle Of Illusion"* dengan menambah level, *rank point*, *save game*, dan *multiplayer game*

DAFTAR PUSTAKA

- [1] Adi Wijaya, Surya., Susi Juniastuti, Supeno Mardi SN, dan Moch. Hariadi. 2009. *Desain Fuzzy State Machine Untuk Menghasilkan Variasi Respon NPC (Non-Playable Character) Pada Sebuah Game*. Program Studi MMT-ITS.
- [2] Irawan, J.D., Prasetyo, S., Wibowo, S.A. and Pranoto, Y.A., 2016. Pelatihan Pembuatan Game Menggunakan Greenfoot. *Industri Inovatif Jurnal Teknik Industri*, 5(2).
- [3] Pratama, W., 2014. *Game Adventure Misteri Kotak Pandora*. *Telematika*, 7(2).
- [4] Rahadian, M.F., Suyatno, A. and Maharani, S., 2017. PENERAPAN METODE FINITE STATE MACHINE PADA GAME "THE RELATIONSHIP".
- [5] Setiawan, I. 2006. *Perancangan Software Embedded System Berbasis FSM*. Semarang: Universitas Diponegoro.