

MEMBANGUN “RPG (ROLE PLAYING GAME) “ADVENTURE OF REN : EDENWEID ISLAND DAN PENGEMBANGAN GAME AGENT BERBASIS FINITE STATE MACHINE”

Hj.Ekawati Yulsilviana¹⁾, Reza Andrea²⁾, Teguh Sanjaya³⁾

^{1,2,3}Teknik Informatika, STMIK Widya Cipta Dharma

^{1,2,3}Jl. M. Yamin No.25, Samarinda, 75123

E-mail : ekawicida@gmail.com¹⁾, reza@bibirdesign.com²⁾, sanjayateguhh@gmail.com³⁾

ABSTRAK

Game atau permainan sangat di sukai semua kalangan dan merupakan aplikasi yang paling banyak digunakan dan dinikmati para pengguna media elektronik sebagai media hiburan. *Game* yang menjuru ke Petualangan juga termasuk salah satu jenis *game* paling di cintai di dunia dan penulis berharap *game* ini bisa diterima di para fans *Game* RPG.

Dalam pembuatan RPG (*Role Playing Game*) “*Adventure Of Ren : Edenweid Island*” dan Pengembangan *Game Agent* Berbasis *Finite State Machine*”. Aplikasi yang digunakan dalam membuat permainan adalah *RPG Maker MV* diharapkan permainan ini tidak menjadi monoton dan pemain juga tidak merasa bosan memainkannya, lalu pada *game agent*nya menggunakan metode *finite state machine* (FSM), *game agent* akan menunjukkan bagaimana interaksi antara player dan juga NPC (*Non-Playable Character*) pada *game*.

Pengujian *game* ini menggunakan, *beta testing* hasil dari pengujian sistem menyimpulkan bahwa *game* ini baik sekali pada kecepatan proses kerja/*loading times* dan fungsi semua menu pada permainan yang digunakan baik sekali. Hasil dari pembuatan *Game RPG* ini berupa .Apk yang dapat di jalankan pada *Smartphone Android*.

Kata Kunci: Role Playing Game, Adventure Of Ren : Edenweid Island, Game agent, Finite State Machine

1. PENDAHULUAN

Game merupakan suatu jenis model permainan atau pertandingan. *Game* bisa diartikan sebagai aktivitas terstruktur atau semi terstruktur, yang biasanya dilakukan untuk bersenang-senang dan kadang digunakan sebagai alat pembelajaran (Suyanto, 2003). Permainan saat ini sangat bervariasi, mulai dari fighting game, racing game dan game kecerdasan seperti tebak soal baik kata maupun bergambar dan lain lain. Permainan tersebut pastinya memiliki tujuannya masing masing. Salah satu *game* yang paling diminati sekarang adalah *game* petualangan atau RPG (*Role Playing Game*). Dewasa ini, banyak sekali permainan yang menghibur dan menantang. Seperti halnya permainan petualangan yang sedang berkembang saat ini, dengan tampilan dan cerita yang menarik membuat para gamer penasaran ingin mencoba permainan tersebut.

Berdasarkan uraian di atas terlihat bahwa permainan modern saat ini memiliki sisi menghibur di dalamnya, hal tersebut dapat menjadi inspirasi dalam pembuatan Membangun RPG (*Role Playing Game*) “*Adventure Of Ren : Edenweid Island*” dan Pengembangan *Game Agent* Berbasis *Finite State Machine*”.

Permainan *Adventure of Ren* adalah permainan petualangan dimana di dalamnya menceritakan seorang pemuda yang bernama “Ren” sedang tersesat di dunia lain yang bernama “*Eden World*” dan sedang melakukan perjalanan kembali menuju dunia asalnya yaitu bumi tapi didalam perjalanan sang karakter utama banyak

mendapatkan rintangan dan juga karakter akan mendapatkan teman perjalanan di dalam *game* tersebut.

Dengan dibuatnya Permainan *Adventure of Ren : Edenweid Island*, diharapkan agar permainan tersebut menjadi wadah hiburan menarik yang bertujuan untuk berpetualang bersama karakter utama. Kenapa menggunakan android agar para *user* bisa memainkan *game* ini di *smartphone* mereka di mana saja dan kapan saja.

2. RUANG LINGKUP PENELITIAN

Permasalahan difokuskan pada :

1. Permainan ini menggunakan sistem *battle* untuk menaikkan *level* para karakternya.
2. Jenis *game* RPG yg dibuat adalah *game* 2 dimensi
3. *Game* ini di bangun untuk *android*
4. *Game agent* pada permainan ini dirancang dengan metode *Finite State Machine* (FSM)
5. Permainan ini menggunakan metode Pengembangan Multimedia.

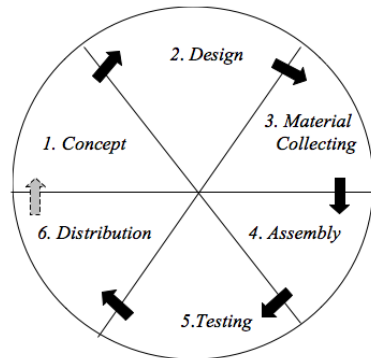
3. BAHAN DAN METODE

Adapun bahan dan metode yang gunakan dalam membangun *game* ini yaitu:

3.1 Tahapan Pengembangan Multimedia

Menurut Binanto (2010), metodologi pengembangan multimedia terdiri dari enam tahap, yaitu

concept (pengonsepan), *design* (pendesainan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Keenam tahap ini tidak dapat bertukar posisi. Meskipun begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan.



Gambar 3.1 Tahapan Pengembangan Multimedia

Tahapan Pengembangan Multimedia Meliputi :

1. *Concept*

Tahapan *concept* (pengonsepan) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi *audiens*). Tujuan dan pengguna akhir program berpengaruh pada nuansa multimedia sebagai pencerminan dari identitas organisasi yang menginginkan informasi sampai pada pengguna akhir. Karakteristik pengguna termasuk kemampuan pengguna juga perlu dipertimbangkan karena dapat memengaruhi pembuatan desain.

Selain itu, tahap ini juga akan menentukan jenis aplikasi (presentasi, interaktif, dan lain-lain) dan tujuan aplikasi (hiburan, pelatihan, pembelajaran dan lain-lain). Dasar aturan untuk perancangan juga ditentukan pada tahap ini, misalnya ukuran aplikasi, target, dan lain-lain. *Output* dari tahap ini biasanya berupa dokumen yang bersifat naratif untuk mengungkapkan tujuan proyek yang ingin dicapai.

2. *Design*

Design (perancangan) adalah tahap pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan, dan kebutuhan material/bahan untuk program. Spesifikasi dibuat serinci mungkin sehingga pada tahap berikutnya, yaitu *material collecting* dan *assembly*, pengambil keputusan baru tidak diperlukan lagi, cukup menggunakan keputusan yang sudah ditentukan pada tahap ini. Meskipun demikian, pada prakteknya, pekerjaan proyek pada tahap awal masih akan sering mengalami penambahan bahan atau pengurangan bagian aplikasi, atau perubahan-perubahan lain.

3. *Material Collecting*

Material Collecting adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan. Bahan-bahan tersebut, antara lain gambar *clip art*, foto, animasi, *video*, *audio*, dan lain-lain yang dapat diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangannya. Tahap ini dapat dikerjakan secara

paralel dengan tahap *assembly*. Namun, pada beberapa kasus, tahap *material collecting* dan tahap *assembly* akan dikerjakan secara linear dan tidak paralel.

4. *Assembly*

Tahap *Assembly* adalah tahap pembuatan semua objek atau bahan multimedia. Pembuatan aplikasi didasarkan pada tahap *design*, bagan alir, dan /atau struktur navigasi.

5. *Testing*

Tahap *Testing* (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan melihatnya apakah ada kesalahan atau tidak. Tahap pertama pada tahap ini disebut tahap pengujian *alpha* (*alpha test*) yang pengujiannya dilakukan oleh pembuat atau lingkungan pembuatnya sendiri. Setelah lolos dari pengujian *alpha*, pengujian *beta* yang melibatkan penggunaan akhir akan dilakukan.

6. *Distribution*

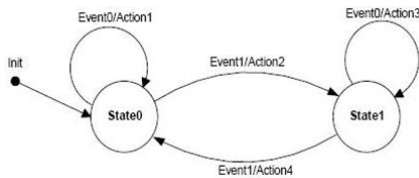
Pada tahap ini, aplikasi akan disimpan dalam suatu media penyimpanan. Jika media penyimpanan tidak cukup untuk menampung aplikasinya, kompresi terhadap aplikasi tersebut akan dilakukan. Tahap ini juga dapat disebut tahap evaluasi untuk pengembangan produk yang sudah jadi supaya menjadi lebih baik. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap *concept* pada produk selanjutnya.

3.2 *Finite State Machine*

Ada beberapa definisi mengenai *Finite State Machine* (FSM) atau sering juga disebut dengan *Finite State Automata* (FSA).

1. FSM didefinisikan sebagai perangkat komputasi yang memiliki *input* berupa *string* dan *output* yang merupakan satu dari dua nilai yang dapat di-*accept* dan *reject*. (Rich: 2009)
2. *Finite Automata* adalah model matematika sistem dengan masukan dan keluaran diskrit. Sistem dapat berada di salah satu dari sejumlah berhingga konfigurasi internal disebut *state*. (Hariyanto: 2004)
3. FSM 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). 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 kompleks. (Setiawan: 2006)

Diagram tersebut memperlihatkan FSM 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.

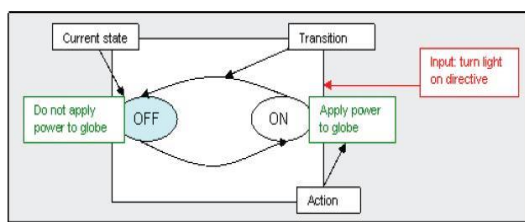


Gambar 3.2 Contoh Diagram state sederhana

FSM terdiri dari dua jenis, yaitu FSM ber-*output* dan FSM tidak ber-*output*. FSM tidak ber-*output* digunakan untuk pengenalan bahasa dalam komputer, dengan *input* yang dimasukkan akan diperoleh apakah *input* tersebut dikenal oleh bahasa komputer atau tidak. Salah satu penggunaan FSM tidak ber-*output* adalah program *compiler*, yaitu program untuk memeriksa apakah perintah yang digunakan pengguna benar atau salah. Sementara untuk FSM ber-*output* digunakan untuk merancang mesin atau sistem. Dan FSM yang akan digunakan dalam penelitian ini adalah FSM ber-*output*, dan untuk selanjutnya akan dituliskan dengan FSM saja.

Ada dua metode utama untuk memperlakukan FSM untuk. Yaitu *Moore Machine* dan *Mearly Machine* yang dinamakan berdasarkan penemunya.

1. *Moore Machine*

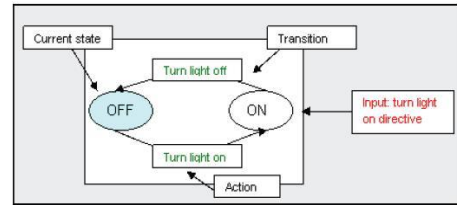


Gambar 3.3 Moore State Machine.

Sumber : (Brownlee, 2010)

Moore Machine adalah tipe dari FSM dimana *output* dihasilkan dari *state*. Pada gambar diatas mencontohkan dimana *state* mendefenisikan apa yang harus dilakukan. Keluaran pada *Moore Machine* diasosiasikan sebagai *state*.

2. *Mearly Machine*



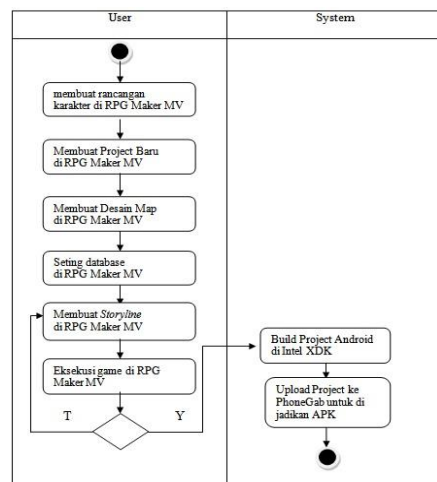
Gambar 3.4 Mearly State Machine
(Sumber: Brownlee, 2010).

Mearly Machine berbeda dengan *Moore Machine* dimana keluarannya merupakan hasil dari transisi antar *state*. Keluaran pada *Mearly Machine* diasosiasikan sebagai transisi.

4. RANCANGAN SISTEM/APLIKASI

Membangun RPG (*Role Playing Game*) “*Adventure Of Ren : Edenweid Island*” dan Pengembangan *Game Agent Berbasis Finite State Machine*”. ini menggunakan alat bantu UML sebagai salah satu cara untuk mempermudah dalam pembuatan aplikasi ini.

Alur UML pembuatan aplikasi dimulai dengan membuat rancangan Karakter di RPG Maker MV. Kemudian Membuat rancangan Project Baru di RPG Maker MV. Setelah itu lanjutkan dengan membuat *Design Map* RPG Maker MV, lalu *Setting Database* di RPG Maker MV, kemudian membuat *Storyline* di aplikasi RPG Maker MV. Eksekusi game RPG Maker MV tapi jika belum selaras maka kembali ke *RPG Maker MV* untuk menyelaraskannya. Build Project Android di Intel XDK maka langkah selanjutnya *Upload project* ke website *PhoneGab* Seperti yang terlihat pada gambar 4.1.



Gambar 4.1 UML pembuatan aplikasi

5. IMPLEMENTASI

Hasil implementasi berdasarkan analisis dan perancangan adalah sebagai berikut :

1. Tampilan Menu Aplikasi Adventure of Ren : Edenweid Island.. Dapat dilihat pada gambar.



Gambar 5.1 Tampilan Menu Adventure of Ren : Edenweid Island

2. Tampilan Scene awal Cerita Dapat dilihat pada gambar.



Gambar 5.2 Tampilan Scene awal game

3. Tampilan scene quest kakek. Dapat dilihat pada gambar.



Gambar 5.3 Scene Quest Kakek

4. Tampilan Scene Dungeon Cerita Dapat dilihat pada gambar.



Gambar 5.4 Tampilan Scene Dungeon game

5. Tampilan Scene Battle Game Dapat dilihat pada gambar.



Gambar 5.5 Tampilan Scene Dungeon game

6. Tampilan Scene Dungeon Hutan Dapat dilihat pada gambar.



Gambar 5.6 Tampilan Scene Dungeon Hutan game

7. Tampilan Scene Final Battle Dapat dilihat pada gambar.



Gambar 5.7 Tampilan Scene Final Battle game

8. Tampilan Scene Akhir game Dapat dilihat pada gambar.



Gambar 5.8 Tampilan Scene Final Battle game

8. KESIMPULAN

Berdasarkan uraian dari masing-masing bab dan hasil pembahasan maka dapat disimpulkan bahwa pembuatan Game RPG "Adventure Of Ren : Edenweid Island" sebagai berikut:

1. Konsep RPG (Role Playing Game) "Adventure Of Ren : Edenweid Island" adalah permainan atau game yang dapat memberikan kita petualangan ke dunia lain, sehingga dibuatlah aplikasi game ini untuk semua kalangan dan juga untuk para pecinta game RPG klasik seperti ini.
2. Game "Adventure Of Ren : Edenweid Island" ini mengimplementasi *game agent* (agen cerdas) yang menggunakan metode *Finite State Machine*. Dengan manfaat agar dalam bermain tidak terasa monoton atau tidak terasa membosankan, dan dapat menghibur pengguna saat memainkan *game* ini.
3. Dengan adanya hasil presentasi dari responden yaitu didapat presentasinya 83% maka *RPG (Role Playing Game) "Adventure Of Ren : Edenweid Island"* ini dapat di terima dan digunakan oleh semua kalangan, karena banyak yang menilai permainan ini baik jika di mainkan oleh pengguna lain.

9. SARAN

Berdasarkan hasil dari penelitian ini ada beberapa saran, yaitu sebagai berikut.:

1. Memperbanyak lagi ekspresi pada *game agent* (agen cerdas) agar lebih berinteraksi pada pengguna.
2. Perlu adanya perbaikan maupun penambahan fitur tutorial sehingga *game* ini bisa dimainkan oleh banyak kalangan yang bukan pecinta *game* RPG.
3. Permainan ini hanya berbasis *android*, diharapkan ada pihak yang mampu mengembangkan menjadi berbasis *platform* lainya seperti *iOS*.
4. Diharapkan agar permainan ini dapat dijadikan sebagai acuan atau referensi bagi mahasiswa informatika dalam membuat tugas akhir.

10. DAFTAR PUSTAKA

- Anton, Prasetya, 2011, Pembuatan Game RPG *The Miracle Of The Stone "Peter In Avatica"* Menggunakan RPG Maker VX
- Arif, Hariadi, Mardi, 2010, *Integrasi Hierarchy Finite State Machine Dan Logika Fuzzy Untuk Desain Strategi NPC Game*. Surabaya : ITS.
- Beny, 2010, Pengembangan Software RPG Menggunakan aplikasi RPG Maker VX
- Binanto, Iwan (2010), *Multimedia Digital –Dasar Teori dan Pengembangannya*. Yogyakarta: Andi.
- Brownlee, J, 2011, *Finite State Machines in Game*. <url:http://ai-depot.com/ Finite State Machines (FSM).html>. diakses : 26 april.
- Febrian, Jack.2004, *Kamus Komputer dan Teknologi Informasi*.
- Fadjar, Rasjid. 2014, *Android : Sistem operasi pada smartphone*.
http://www.ubaya.ac.id/2014/content/articles_detail/7/Android--Sistem-Operasi-pada-Smartphone.html
diakses : 25 july 2017
- Hend, 2006, *Fokus Bangun Dasar Perancangan Sistem Dengan UML*. Yogyakarta : Gramedia Pustaka Utama
- Hariyanto, B. 2004. *Teori Bahasa, Otomata, dan Komputasi serta terapannya*. Bandung : Informatika
- Kamus Besar Bahasa Indonesia, 2008, id.m.wikipedia.org/wiki/istimewa :
History/Kamus_Besar_Bahasa_Indonesia.
Diakses Pada Tanggal 21 September 2011.

- M. Suyanto, 2005, *Multimedia Alat untuk Meningkatkan Keunggulan Bersaing*. Yogyakarta : Penerbit Andi
- Rich, E. 2009. *Automata, Computability, and Complexity, Theory and Applications*. United States of America : Pearson Prentice Hall
- Setiawan, I. 2006. *Perancangan Software Embedded System Berbasis FSM*. Semarang: Universitas Diponegoro
- Suyanto, M. 2003. *Multimedia Alat Untuk Meningkatkan Keunggulan Bersaing*. Andi : Yogyakarta
- Sutarman , 2009, *Pengantar Teknologi Informasi*, Yogyakarta, Bumi Aksara.
- Tito Bimantoro (2016), *Pemodelan Perilaku Musuh Menggunakan Finite State Machine(FSM) Pada Game Pengenalan Unsur Kimia*.
- Vaughan, Tay, 2011. *Multimedia : Making It Work 8th Edition*. New York : McGrawHill.
- Wahana, 2013 *Membuat Game RPG dengan RPG Maker*, Yogyakarta: Andi Offset
- Zen, Revaldo. 2008. *Permodelan Sistem Kontrol Elevator*. Jakarta: FMIPA UI.