

MEMBANGUN EDUGAME “VIGIE GARDEN” PENGENALAN NAMA SAYURAN DENGAN GAME AGENT FINITE STATE MACHINE BERBASIS ANDROID

Reza Andrea¹⁾, M. Irwan Ukkas²⁾, Wiwin³⁾

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

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

E-mail : reza@bibirdesign@gmail.com¹⁾, irwan212@yahoo.com²⁾, wiwin6602@gmail.com³⁾

ABSTRAK

Game edukasi berbasis simulasi didesain untuk mensimulasikan permasalahan yang ada sehingga diperoleh ilmu yang dapat digunakan untuk menyelesaikan permasalahan tersebut. *Game* simulasi dengan tujuan edukasi ini dapat digunakan sebagai salah satu media edukasi yang memiliki pola pembelajaran *learning by doing*. Berdasarkan pola yang dimiliki oleh *game* tersebut, pemain dituntut untuk belajar sehingga dapat menyelesaikan permasalahan yang ada.

Dalam pembuatan *edugame* “Membangun *Edugame Vigie Garden* Pengenalan Nama Sayuran Dengan *Finite State Machine* Berbasis *Android*”. Aplikasi yang digunakan dalam membuat permainan adalah *Swish max4* dan dengan algoritma *shuffle random* diharapkan permainan ini tidak menjadi monoton dan pemain tidak dapat mengingat posisi gelembung pada permainan, lalu pada *game agentnya* menggunakan metode *finite state machine* (FSM), *game agent* akan memberikan pemberitahuan kepada para pemain ketika menjawab salah atau benar dalam permainan.

Kata Kunci: Nama Sayuran, *Edugame*, *Finite State Machine*, *Android*.

1. PENDAHULUAN

Game edukasi sangat menarik untuk dikembangkan. Ada beberapa kelebihan dari *game* edukasi dibandingkan dengan metode edukasi konvensional. Salah satu kelebihan utama *game* edukasi adalah visualisasi dari permasalahan nyata. *Game* edukasi berbasis simulasi didesain untuk mensimulasikan permasalahan yang ada sehingga diperoleh ilmu yang dapat digunakan untuk menyelesaikan permasalahan tersebut.

Untuk itu dalam penelitian ini akan menerapkan *Edugame “Vigie Garden”* untuk pembelajaran mengenal jenis-jenis dan manfaat sayuran juga permainan *drag and drop*. Penelitian ini menggunakan algoritma pengacakan posisi (*shuffle random*) agar permainan tidak monoton, dan tidak mudah untuk ditebak tentang soal yang akan diberikan nantinya.

Pada permainan ini *game agent* (agen cerdas) akan memberikan aksi-reaksi, mengamati, dan bertindak pada suatu kondisi sehingga terlihat seperti diri sendiri, dalam *game* ini setiap tindakan atau aktivitas dibangun oleh agen untuk memenuhi kondisi lingkungannya. Maka disini *game agent* akan diimplementasikan dengan metode *finite state machine* (FSM), yang mana FSM ini adalah sebuah

metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan *state* (keadaan), *event* (kejadian), dan *action* (aksi). *Game agent* yang dilibatkan berupa sebuah animasi yang berekspresi dan mengomentari setiap langkah pemain. Jika langkah pemain salah maka animasi akan berekspresi dan berkomentar dengan nada kecewa, marah, cemas, dan begitu juga sebaliknya.

2. RUANG LINGKUP PENELITIAN

Permasalahan difokuskan pada :

1. *Edugame* diperuntukkan untuk anak usia dini (3 – 6) tahun.
2. *Edugame* ini belajar dan bermain tentang nama dan ejaan sayuran.
3. *Edugame* ini memiliki permainan yang terdiri dari 10 *level*.
4. *Edugame* ini mengenalkan 14 jenis sayuran.
5. Pembuatan *game* menggunakan *SwishMax4*.
6. *Game agent* menerapkan metode *Finite State Machine* (FSM).
7. Algoritma yang digunakan pada permainan ini yaitu algoritma logika pengacakan posisi objek (*shuffle random*) yang berfungsi untuk mengacak soal atau objek

3. BAHAN DAN METODE

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

3.7 Game

Menurut Ismail (2009), *Game* merupakan salah satu multimedia yang banyak digemari oleh masyarakat. *Game* dimainkan oleh semua kalangan, dari usia remaja hingga dewasa. *Game* mempunyai peranan penting bagi perkembangan teknologi manusia. Dari segi tampilan, musik dan penerapan hiburan yang populer untuk semua kalangan usia. Sejak pertama kali ditemukan sampai saat sekarang, teknologi *game* telah mengalami kemajuan yang terbilang pesat. Hal ini ditandai dengan berkembangnya jenis *game*, produk, alat dan jenis interaksi *game* dengan penggunaan yang semakin beragam bentuknya.

4.7 Algoritma Pengacakan Posisi

Menurut Reza Andrea (2015), *Shuffle random* adalah pengacakan urutan indeks dari sebuah record atau array. Pengacakan ini diibaratkan pengocokan pada dek kartu, dimana semua kartu dikocok sehingga susunannya teracak. Contoh lain misalkan A adalah array 5 x 1, $A = [1\ 2\ 3\ 4\ 5]$ maka proses *shuffle random* akan mengacak susunan indeks dari array A menjadi $A1 = [5\ 1\ 3\ 2\ 4]$ ataupun menjadi susunan array yang lain. Dalam bahasa pemrograman fungsi *shuffle random* tidak hanya dapat mengacak angka, tetapi juga dapat mengacak array string ataupun campuran string dan angka.

3.3 Agent Game (Intelligent Agent)

Menurut Kristanto (2007), Dalam kecerdasan buatan, *intelligent agent* adalah sebuah entitas otonom yang mengamati dan bertindak atas lingkungan, yaitu membutuhkan agen dan mengarahkan aktivitasnya untuk mencapai tujuan yaitu rasional. *Intelligent agent* juga dapat belajar atau menggunakan pengetahuan untuk mencapai tujuan mereka. Menurut Nikola Kasabov, *Agent Intelligent* harus menunjukkan karakteristik berikut.:

1. Mampu menganalisis sendiri dalam hal perilaku, kesalahan dan kesuksesan.
2. Belajar dan meningkatkan interaksi dengan lingkungan (perwujudan).
3. Belajar dengan cepat dari sejumlah besar data.

Dari karakteristik ini dapat disimpulkan bahwa *agent* adalah suatu fungsi yang dapat memberikan suatu aksi atau tindakan dalam suatu kondisi sehingga terlihat seperti sendiri dalam menentukan suatu tindakan atau member keputusan dalam menjalankan fungsinya dalam sebuah entitas otonom yang mengamati dan bertindak atas lingkungannya.

3.4 Finite State Machine (FSM)

Bahasa formal dapat dipandang sebagai entitas abstrak, yaitu sekumpulan *string-string* simbol alphabet tertentu. Namun bahasa juga dapat dipandang sebagai

entitas-entitas abstrak yang dapat dikenali atau dibangkitkan melalui suatu mesin komputasi. Mesin yang dapat mengenali bahasa kelas ini adalah *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*.
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*.
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.

3.5 Android

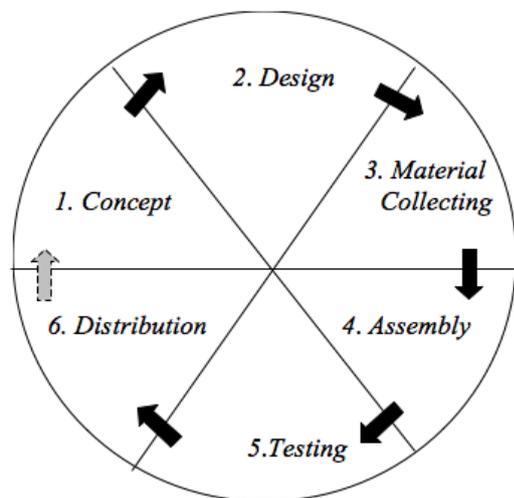
Menurut Safaat (2012), *Android* merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi inti yang di-release oleh Google. Sedangkan *Android SDK* (*Software Development Kit*) menyediakan *Tools* dan *API* (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform *Android* dengan menggunakan bahasa pemrograman *Java*. Dikembangkan bersama antara *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, *NVIDIA* yang tergabung dalam OHA (*Open Handset Alliance*) dengan tujuan membuat sebuah standar terbuka untuk perangkat bergerak (*mobile device*).

3.6 SwishMax4

SwishMax4 merupakan pengembangan dari Program *Swish v.3*, yang kini telah memiliki 230 *built-in* efek seperti efek *Explode*, *Vortex*, *3D Spin*, *Snake* dan banyak lainnya. Seperti halnya *Swish*, *SwishMax* juga memiliki alat bantu untuk membuat garis, kotak, *elips*, *kurva bazier*, gerak animasi, *sprite*, tombol *roll over* dan lainnya. Format dasar *SwishMax* adalah *swi file*, namun dapat juga diekspor kedalam *file flash* (*swf*), *movie* (*avi*) ataupun *execute* (*exe*) program yang dapat dijalankan berdiri sendiri. Sehingga animasi *Swishmax* dapat diletakkan langsung di *web*, ataupun diikutkan dalam presentasi *Microsoft Powerpoint* dan *Microsoft Word*.

3.7 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 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.

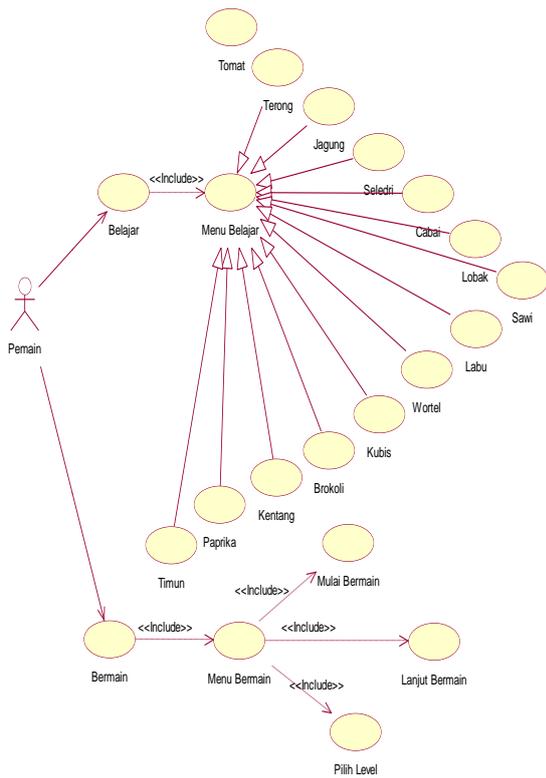
6. *Distribution*

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.

4. RANCANGAN APLIKASI

Dalam pembuatan game *vigie garden* semua konsep mulai dari Perancangan *edugame Vigie Garden* pengenalan nama sayuran *Unified Modeling Language* (UML). Terdapat 3 diagram yang digunakan pada system ini yaitu *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagrams* sampai dengan perancangan.

1. *Use Case Diagram* *Edugame "Vigie Garden"* Perancangan *use case diagram* akan menggambarkan bagaimana pemain berinteraksi dengan *use case* yang ada pada sistem. Gambar 2 berikut ini adalah gambaran *use case diagram*.



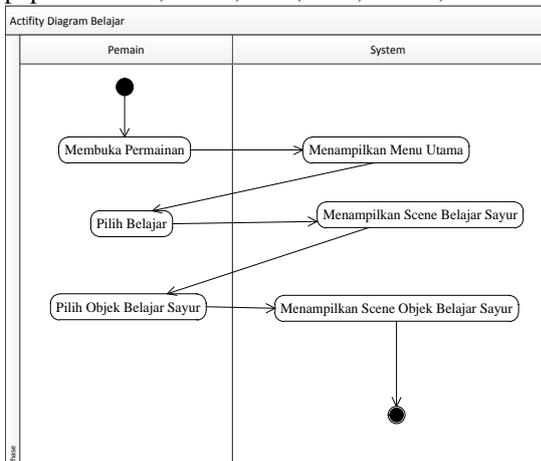
Gambar 2. Use Case Diagram

2. Activity Diagram Edugame “Vigie Garden”.

Untuk mengenal proses dari suatu sistem digunakan diagram Activity Diagram. Dengan diagram Activity Diagram ini dapat diketahui proses yang terjadi pada aplikasi. Gambar Activity Diagram bisa dilihat di bawah ini

1) Activity Diagram Belajar Edugame “Vigie Garden”.

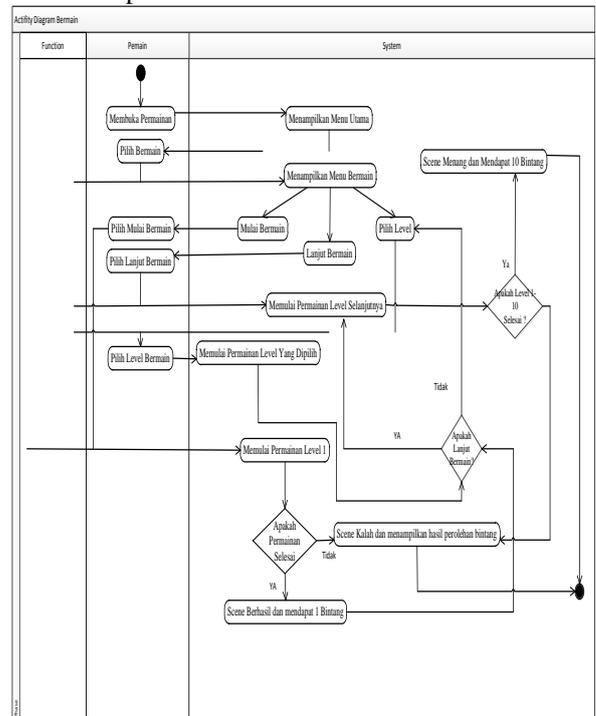
Alur aktifitas yang terjadi disaat pemain memilih “Belajar” pada menu utama. Saat pemain memilih “Belajar”, maka sistem akan langsung menampilkan scene “Belajar” dan kita dapat memulai belajar mengenal 14 nama sayuran yaitu tomat, terong, jagung, seledri, cabai, kacang hijau, brokoli, kentang, paprika merah, timun, sawi, labu, wortel, dan kubis.



Gambar 3. Activity Diagram Belajar

2) Activity Diagram Bermain “Vigie Garden”

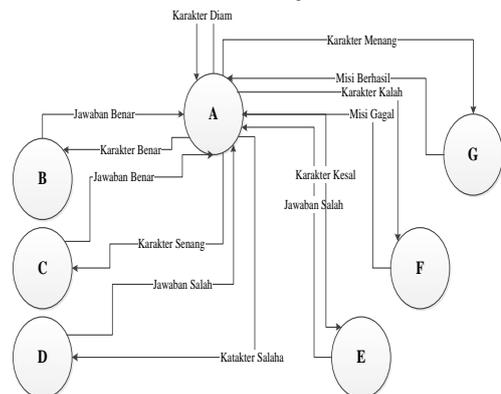
Alur aktifitas yang terjadi disaat pemain memilih “Bermain” pada menu utama. Saat pemain memilih “Bermain”, sistem akan langsung menampilkan menu bermain, jika pemain memilih mulai bermain maka sistem akan menampilkan permainan level pertama. Jika pemain berhasil memenangkan permainan level pertama maka akan masuk pada scene behasil dan mendapatkan 1 bintang tapi jika tidak berhasil akan masuk pada scene kalah dan mendapatkan bitang. Jika pemain berhasil memenangkan permainan level pertama pemain bisa memilih lanjut pada level selanjutnya hingga level 10 dan menang dan mendapatkan 10 bintang, tetapi jika tidak berhasil memenangkan hingga level 10 makan pemain akan masuk pada scene kalah.



Gambar 4. Activity Diagram Bermain

3. Konsep Finite State Machine (FSM)

FSM adalah sebuah metodologi perancangan sistem control yang menggambarkan tingkah laku atau prinsip kerja sistem (game agent) dengan menggunakan tiga hal yaitu state (keadaan), event (kejadian), action (aksi).



Gambar 5. Rancangan FSM (Finite State Machine).

Keterangan :

A = Karakter (*game agent*) Homing “Besiap / Menunggu”.

B = Karakter (*game agent*) “Benar”.

C = Karakter (*game agent*) “Senang”.

D = Karakter (*game agent*) ”Salah”

E = Karakter (*game agent*) ”Kesal”

F = Karakter (*game agent*) ”Kalah”

G = Karakter (*game agent*) ”Menang”

Pada gambar 5 menunjukkan bahwa setiap kondisi yang ada didalam *game agent*, yang dihasilkan dari aksi pemain akan menyebabkan perpindahan *state* dalam FSM. Aksi yang diberikan pemain akan menentukan apa yang akan dilakukan oleh *game agent*.

5. IMPLEMENTASI

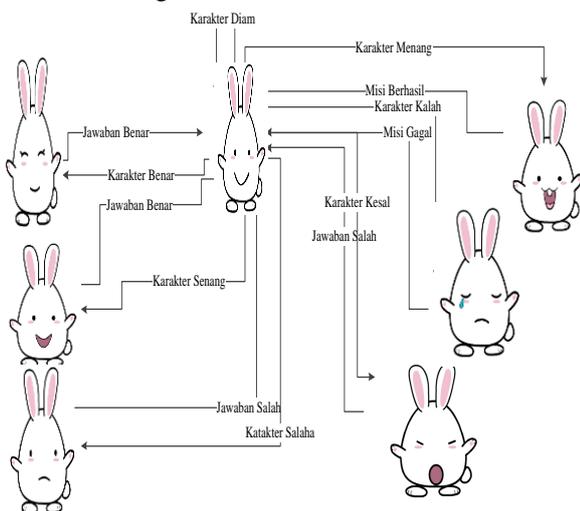
Hasil implementasi berdasarkan analisis dan perancangan adalah sebagai berikut :

1. Assembly Finite State Machine “Game Vigie Garden Mengenal Nama Sayuran”

FSM adalah sebuah metodologi perancangan sistem control yang menggambarkan tingkah laku atau prinsip kerja sistem (*game agent*) dengan menggunakan tiga hal yaitu *state* (keadaan), *event* (kejadian), *action* (aksi).

Pada gambar 4.28 menunjukkan bahwa setiap kondisi yang ada didalam *game agent*, yang dihasilkan dari aksi pemain akan menyebabkan perpindahan *state* dalam FSM. Aksi yang diberikan pemain akan menentukan apa yang akan dilakukan oleh *game agent*.

Pada *edugame* ini terdapat tujuh karakter fsm, yaitu karakter homing atau diam, karakter benar dan senang yang akan muncul ketika pemain menjawab permainan dengan benar, karakter salah dan kesal yang akan muncul ketika pemain menjawab permainan salah, dan karakter menang



Gambar 6 scene Assembly Finite State Machine “Game Vigie Garden Mengenal Nama Sayuran”

2. Tampilan Menu Utama

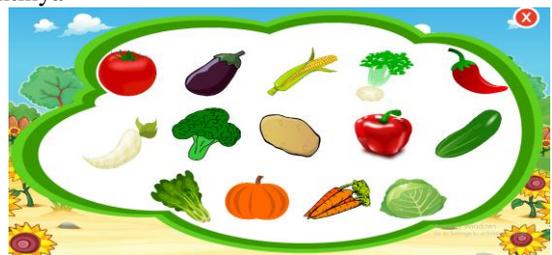


Gambar 7. Menu Utama

Seperti yang terlihat pada gambar 7, Terdapat nama atau judul dari *Edugame* “*Vigie Garden Mengenal Nama Sayuran*”. Setiap tombol pada *scene* menu utama memiliki fungsi masing-masing, saat kita mengklik *button* belajar maka akan ditampilkan *scene* belajar, saat kita mengklik *button* bermain maka akan ditampilkan *scene* bermain, dan ketika kita mengklik *button* keluar maka permainan akan keluar dari aplikasi.

3. Tampilan Scene Belajar

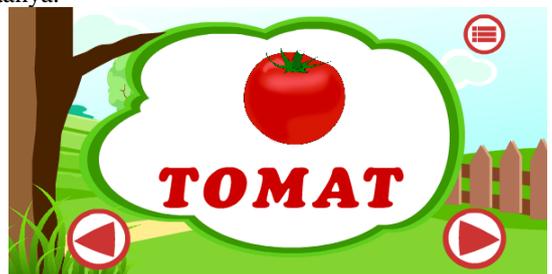
Pada gambar 8 *Scene* Belajar adalah *scene* dimana pemain dapat melihat, memahami dan mempelajari isi materi yang ada, sehingga memudahkan pemain dalam menyelesaikan misi yang di berikan pada *scene* bermain nantinya



Gambar 8. Tampilan Belajar

4. Tampilan Scene Tomat

Pada gambar 9 *Scene* Tomat adalah *scene* dimana pemain dapat mempelajari nama sayur tomat dan ejaannya.



Gambar 9. Tampilan Scene Tomat

6. Tampilan Scene Bermain

Pada gambar 10 dan 11 *Scene* Bermain adalah *scene* dimana pemain bermain dengan cara menyelesaikan misi sesuai dengan pertanyaan sebelum waktu selesai, pemain harus menyelesaikan sepuluh level dengan ketentuan pemain tidak boleh gagal dalam menyelesaikan permainan pada setiap level, jika gagal maka pemain harus mengulang permainan dari level pertama.



Gambar 4.10 Scene Bermain.



Gambar 11 Scene Bermain.

7. Tampilan Scene Level

Pada gambar 12 Scene Level adalah scene dimana pemain dapat memulai permainan dari level 1 atau memilih level jika permainan dilanjutkan.



Gambar 12 Scene Level.

8. Tampilan Scene Misi Berhasil

Pada gambar 13 Scene Misi Berhasil adalah scene yang akan tampil jika pemain berhasil menyelesaikan misi pada satu level.



Gambar 13 Scene Misi Berhasil.

9. Tampilan Scene Gagal

Pada gambar 14 Scene gagal adalah scene yang akan menampilkan hasil akhir atau hasil dari level yang sudah terselesaikan.



Gambar 15 scene Gagal.

10. Tampilan Scene Menang

Pada gambar 16 Scene Menang adalah scene yang akan tampil jika pemain berhasil menyelesaikan semua level.



Gambar 4.27 scene Menang.

5. KESIMPULAN

Berdasarkan uraian dari masing-masing bab dan hasil pembahasan maka dapat disimpulkan bahwa pembuatan Edugame "Vigie Garden" Pengenalan Nama Sayuran sebagai berikut :

1. Konsep Edugame "Vigie Garden" Pengenalan Nama Sayuran adalah permainan atau game edukasi yang dapat membantu anak-anak mengenal nama-nama sayuran beserta ejaannya, sehingga dibuatlah edugame yang bersifat multimedia.
2. Edugame ini menggunakan algoritma pengacakan posisi objek permainan atau bisa disebut *shuffle random* yang berfungsi untuk mengacak objek dengan tujuan agar pemain tidak dapat mengingat letak objek dalam permainan.
3. Edugame 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.

6. SARAN

Berdasarkan hasil dari penelitian ini ada beberapa saran, yaitu :

1. Memperbanyak lagi ekspresi pada *game agent* (agen cerdas) agar lebih berinteraksi pada pengguna.

2. Perlu adanya perbaikan maupun penambahan fitur sehingga *game* ini dapat terlihat lebih menarik.
3. Dapat mengembangkan game berbasis *iOS*
4. Dapat mengembangkan game menjadi game *online*
5. Diharapkan agar permainan ini dapat dijadikan sebagai acuan atau referensi bagi mahasiswa informatika dalam membuat tugas akhir.

7. DAFTAR PUSTAKA

- Adnyana, MA. 2011, Modul Swishmax. [http : // ilmukomputer.org / 2008/11/ 25 /animasi-flashdengan-swishmax-2/](http://ilmukomputer.org/2008/11/25/animasi-flashdengan-swishmax-2/). diakses pada tanggal 8 September 2012.
- Andrea, Reza. 2013, *Teknik Pengacakan Posisi – Find Me The Game Prosiding Senaik*. Samarinda : Unmul Press.
- Binanto, Iwan. 2010, *Multimedia Digital-Dasar Teori dan Pengembangannya*. Yogyakarta : Andi
- Brownlee, J. 2011, *Finite State Machines in Game*. [http://ai-depot.com/ Finite State Machines \(FSM\).html](http://ai-depot.com/Finite%20State%20Machines%20(FSM).html)>. diakses : 26 april.
- Elliani. 2014, *Membangun Edugame Smart&Fun Hijaiyah Berbasis Android*. Samarinda : STMIK Widya Cipta Dharma.
- Hakim. Zainal. 2012, “Sejarah Adobe Photoshop”. Jakarta : PT. Gramedia Pustaka Umum.
- Handriyantini. 2009, *Permainan Eduktif (Educational Games) Berbasis Komputer untuk Siswa Sekolah Dasar*. Jurnal Sekolah Tinggi Informatika & Komputer : Malang.
- Ismail, Andang. 2009, *Education Games*. Jakarta : Proumedia.
- Kristanto. Andri. 2007, *Kecerdasan Buatan*. Yogyakarta : Graha Ilmu.
- Muhammad, Shalahuddin. 2011, *Modul Pembelajaran: Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*, Modula Bandung.
- Nazruddin, Safaat. 2012, (Edisi Revisi). *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika. Bandung.
- Nugroho, Adi. 2010, *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP*, Yogyakarta : Penerbit Andi Offset.
- Pressman, R. 2010, *Software Engineering: A Practitioner's Approach*. McGraw-Hill.
- Setiawati, 2008. *Education Games*. Jakarta : Proumedia.
- Rachman, Suharto, Purwanti. 2014, *Agent Cerdas Animasi Wajah Untuk Game Tebak Kata*. Semarang : UDINUS