

BAB II

LANDASAN TEORI

Pada Bab ini membahas teori dasar yang berhubungan dengan pembuatan sistem aplikasi. Dasar teori dari literatur dan referensi yang berkaitan dengan permasalahan yang diangkat. Teori tersebut meliputi *virtual reality*, *Google Cardboard*, *Unity*, *Android*, dan *Software* grafis yang digunakan.

2.1 Virtual Reality

Virtual Reality (VR) merupakan teknologi dalam bidang komunikasi dan informasi memungkinkan *user* dapat berinteraksi dengan lingkungan yang disimulasikan oleh komputer sehingga *user* seolah-olah terlibat secara fisik. *Virtual Realty* ini biasanya digunakan untuk pelatihan atau simulasi berbagai kegiatan. Simulasi dengan teknologi VR ini memiliki beberapa keuntungan, menghemat biaya pengeluaran, menghemat waktu, dan menghemat tenaga. Agar dapat mewujudkan suasana yang mendekati atau menyerupai dunia nyata, VR menggunakan perangkat keras khusus seperti kacamata VR sebagai *display* dan *joystick* untuk bergerak dan berinteraksi. Cara kerja sistem *virtual reality* yaitu pemakai melihat suatu dunia semu pada kacamata VR yang sebenarnya berupa gambar-gambar yang bersifat dinamis. “Sebuah computer menghasilkan lingkungan 3D dengan dimana pengguna dapat berpartisipasi secara *real time* dan mengalami sensasi berada disana. Dua hal yang paling penting untuk dipertimbangkan saat membuat atau menggunakan VR adalah *real-time* 3D lingkungan *virtual* dan perangkat antarmuka manusia yang menghubungkan pengguna”. Melalui perangkat *headset* atau *speaker* pengguna dapat mendengar suara yang realistis dan *user* bergerak mengelilingi dunia *virtual* dan berinteraksi dengan menggunakan *joystick*. [1]

Konsep dasar dari VR mencoba membuat dunia dalam komputer. Berbagai alat digunakan oleh pengguna agar dapat menerjemahkan gerakan sehingga dapat memanipulasi objek virtual. Tujuan dari *virtual reality* adalah untuk memungkinkan atau membuat pengguna berinteraksi dengan lingkungan yang tidak nyata dengan disimulasikan agar pengguna seolah berada dalam lingkungan yang nyata. Saat ini, Lingkungan Virtual Reality sebagian besar menghadirkan visual yang ditampilkan pada layer. [1]

2.1.1 Virtual Tour

Sebuah tur virtual adalah simulasi dari lokasi yang ada, biasanya terdiri dari urutan video atau gambar diam. Hal ini juga dapat menggunakan unsur-unsur multimedia lainnya seperti efek suara, musik, narasi, dan teks. Hal ini dibedakan dari penggunaan siaran langsung atau telepariwisata.[3]

Ungkapan "virtual tour" sering digunakan untuk menggambarkan berbagai video dan media fotografi. Panorama menunjukkan pandangan tak terputus, karena panorama dapat berupa serangkaian foto-foto atau rekaman video panning. Namun, "tur panorama" dan "virtual tour" sebagian besar telah dikaitkan dengan wisata virtual yang dibuat menggunakan kamera statis. Wisata virtual tersebut terdiri dari sejumlah foto yang diambil dari sudut pandang tunggal. Kamera dan lensa yang diputar di sekitar atau yang disebut sebagai *paralaks* (titik yang tepat di belakang lensa atau cahaya *konvergen*).[3]

Sebuah tur video adalah motion video penuh pada lokasi. Tidak seperti *wrap-around* statis, tur video adalah video yang berjalan linier melalui lokasi. Menggunakan kamera video, lokasi difilmkan dengan kecepatan berjalan sambil bergerak terus dari satu titik ke titik lain di seluruh lokasi subjek.[3]

Dengan perluasan *video* di internet, wisata virtual berbasis *video* semakin meningkat popularitasnya. Kamera *video* digunakan untuk bergerak dan berjalan-jalan melalui properti subjek seperti nyata. Manfaat dari metode ini adalah bahwa sudut pandang terus berubah sepanjang pergerakan. Namun, menangkap *video* berkualitas tinggi membutuhkan keterampilan teknis secara lebih dan peralatan dari pada mengambil gambar diam digital. *Video* juga menghilangkan kendali penampil dari tur. Mengedit *video* digital memerlukan kemampuan dengan perangkat lunak editing *video* dan memiliki spesifikasi perangkat keras komputer yang handal. Juga, menampilkan *video* melalui internet membutuhkan lebih banyak *bandwidth*. Karena kesulitan-kesulitan ini, tugas membuat tur berbasis *video* sering diserahkan kepada profesional.[3]



Gambar 2.1 Contoh dari *google Expeditions* [2]

Sejarah awal *virtual reality* yaitu saat seorang ilmuwan *University of Utah* Ivan Sutherland dan seorang muridnya Bob Sproull pada tahun 1968 menemukan *Head Mounted Display* yang merupakan jendela dunia *virtual*. Pada tahun 1975, ilmuwan bernama Myron Krueger untuk pertama kalinya menemukan *videoplace* yang memungkinkan penggunanya dapat berinteraksi dengan objek *virtual*. Pada tahun 1990-an istilah *virtual reality* digunakan pada game dan komputer, terutama untuk game *first person shooter* atau game tembak-menembak dari sudut pandang orang pertama. Sistem *virtual reality* ini menggunakan kacamata *stereoscopic* sebagai penyedia gambar 3D dan alat-alat elektronik lain berbasis sensor dipasang pada badan pengguna untuk memberikan efek seolah-olah pengguna bisa berinteraksi dengan semua objek yang tampak di kacamata VR.[3]

program Expeditions memungkinkan setiap sekolah untuk membeli kotak yang berisi satu tablet dan banyak *Google Cardboard*. Nantinya kegiatan belajar mengajar di sekolah antara guru dan murid menjadi lebih interaktif lagi. Dimana guru akan mengendalikan aplikasi *Expeditions* pada tablet dan menentukan salah satu lokasi di seluruh dunia bahkan planet. Selanjutnya, setiap murid akan menggunakan perangkat *VR Google Cardboard* dan merasakan pengalaman 3 dimensi yang seakan-akan sedang berada di lokasi tersebut. Dengan adanya program ini diharapkan murid dapat belajar lebih mendalam dan detail mengenai sebuah lokasi serta lingkungan yang dikunjungi. Kekurangan dari *google Expeditions* adalah masih mahalnya perangkat yang di keluarkan google ini dan hanya dapat di gunakan untuk media pembelajaran di sekolah dan di kendalikan oleh guru mereka saja. Ada satu lagi aplikasi VR yang menjadi bahan observasi, yaitu *VR Forest Animal Adventure*, aplikasi ini mengadaptasi taman safari menjadi sebuah VR. Aplikasi ini sebenarnya menarik karna menampilkan banyak karakter hewan, namun yang di sayangkan aplikasi ini kurang edukatif karna hanya menampilkan karakter hewan tanpa ada keterangan jenis hewan tersebut. [3]



Gambar 2.2 Gambar VR *Forest Animal Adventure* [4]



Gambar 2.3 Contoh Dari Aplikasi *Forest Animal Adventure* [4]



Gambar 2.4 Tampilan Apabila Menggunakan *Google Cardboard* [4]

2.2 Google Cardboard

Google Cardboard merupakan kacamata *virtual reality* ciptaan google yang terbuat dari bahan kardus. Berbeda dengan kacamata *virtual reality* lainnya pada google cardboard tidak terdapat unit *display* khusus yang memproyeksikan gambar 3D ke mata pengguna. Sebagai

gantinya, digunakanlah sebuah ponsel Android biasa dan lensa. Cara menggunakannya yaitu dengan menyisipkan *smartphone* android kedalam cardboard sehingga layarnya menghadap ke lensa, yang akan memproyeksikan tampilan layar itu kemata pengguna.[4]

Untuk merasakan teknologi *virtual reality* tak mesti harus menggunakan perangkat keras yang mahal. Dengan google cardboard ini semua orang bisa merasakan sensasi teknologi *virtual reality* terbuat dari bahan kardus yang memiliki konsep unik kerana *headset* ini terbuat dari potongan kardus berpola harus dirakit terlebih dahulu, dibentuk dan dilipat sedemikian rupa sehingga nantinya menjadi sebetuk kacamata. Google cardboard selain unik karena berbentuk origami yang harus dirakit dan dibentuk terlebih dahulu, perangkat ini menggunakan *smartphone* atau ponsel android sebagai unit display yang nantinya akan memproyeksikan gambar 3D ke mata pengguna, caranya yaitu dengan ponsel android disisipkan di bagian depan kacamata google cardboard dengan layar layar ponsel menghadap ke 2 lensa.[4]



Gambar 2.5 Google Cardboard. [4]

2.2.1 Sensor Gyroscope

Aplikasi google cardboard hanya berfungsi pada ponsel yang memiliki sensor *gyroscope*. Sensor *gyroscope* ini merupakan sensor gerak dengan mendeteksi gerakan sesuai gravitasi dimana sekarang hampir di semua *smartphone* memiliki sensor ini. *Gyroscope* digunakan untuk pengguna agar dapat melihat sekeliling obyek 3D cukup dengan memutar ponsel. Untuk ponsel yang tidak memiliki sensor ini masih bisa menggunakan aplikasi *google cardboard* akan tetapi pandangan pengguna tidak leluasa.[5]

2.2.2 Cardboard SDK

Cardboard SDK merupakan suatu set pengembangan perangkat lunak yang memungkinkan pengguna membuat aplikasi virtual reality pada unity3D dengan lebih mudah. Google cardboard SDK menggunakan OpenGL. Skrip plugin dalam SDK ini memudahkan pengguna untuk memulai sebuah proyek VR *unity*, dapat mengadaptasi aplikasi unity3D ke VR, dan membuat aplikasi yang dapat dengan mudah beralih masuk dan keluar dari mode VR. *Plugin* pada SDK ini juga menyediakan fitur tambahan seperti dapat mensimulasikan gerakan kepala pengguna dalam modus bermain menggunakan tombol *mouse* dan *alt / control* untuk menggeser atau memiringkan kamera, dinamis dapat menyesuaikan tingkat *stereo* untuk mengurangi ketegangan mata, dan fitur lainnya yang tentunya memudahkan pengguna untuk membangun sebuah aplikasi *Virtual reality*. [8]

2.3 Joystick Controller



Gambar 2.6 Joystick Bluetooth [19]

Gambar 4.3 merupakan *joystick (Gamepad)* alat yang untuk mengontrol navigasi user saat berada di lingkungan *virtual reality*. *Joystick* dihubungkan dengan *smartphone* menggunakan *Bluetooth* ataupun kabel *USB*. *Joystick* yang digunakan adalah *joystick bluetooth*. [19]

2.4 Kebun Binatang

Kebun binatang adalah suatu tempat atau wadah yang berbentuk taman dan atau ruang terbuka hijau dan atau jalur hijau yang merupakan tempat untuk mengumpulkan, memelihara kesejahteraan dan memperagakan satwa liar untuk umum dan yang di atur penyelenggaraanyasebagai lembaga konservasi *ex-situ*. Satwa liar yang di kumpulkan dalam

wadah taman satwa adalah satwa liar yang di lindungi dan tidak di lindungi oleh peraturan perundang-undangan, dan akan di pertahankan kemurnian jenisnya dengan cara di pelihara, di tangakarkan di luar habitat aslinya.[6]

2.5 Android

Android adalah sistem operasi untuk telepon seluler seperti *smartphone* atau komputer tablet. Android berbasis Linux sehingga android menyediakan *platform* terbuka untuk para pengembangnya menciptakan aplikasi mereka sendiri. Android memiliki logo berbentuk robot dengan dua antenna dikepalanya ini telah beberapa kali melakukan pembaruan versi. Setiap versi pada android memiliki nama dari makanan, seperti Jelly bean, Kit kat dan yang terbaru bernama lollipop.[7]

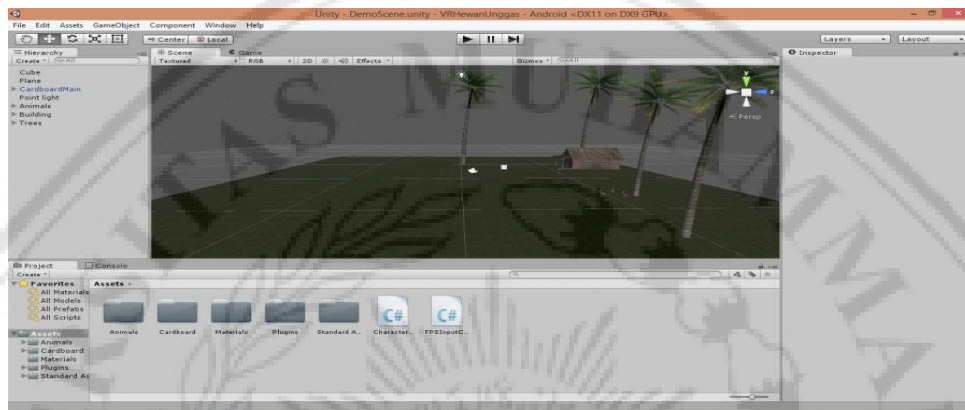
Berawal dari Google Inc membeli Android Inc, pendatangbaru yang membuat perangkat lunak untuk ponsel. Untuk mengembangkan sistem operasi android ini dibentuklah *Open Handset Alliance* persatuan dari 34 perusahaan yang berjalan dalam bidang perangkat keras, perangkat lunak dan telekomunikasi diantaranya Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Invidia. Pada bulan November 2007 android melakukan perilis perdana bersama Open Handset Alliance menyatakan mendukung standar pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis standar terbuka perangkat seluler, kode-kode android dibawah lisensi apache dan sebuah lisensi perangkat lunak. Terdapat dua jenis distributor dari sistem operasi android ini. Pertama yaitu yang mendapat kan dukungan langsung dari Google atau GMS (Google Mail Services). Kedua, dikenal sebagai OHD (Open Handset Distribution) yang benar-benar bebas distribusinya tanpa dukungan langsung dari Google.[7]

2.6 Software Grafis Komputer

Proses pembuatan model rumah dan denah 3D terdapat beberapa langkah menggunakan perangkat lunak. Perangkat lunak untuk pemodelan 3D yang dimaksud disini adalah perangkat lunak untuk membuat objek 3D seperti Sweet Home 3D dan Blender untuk nantinya dimasukan kedalam unity3D. Agar dapat diterima dengan baik pada unity3D, hasil pembuatan model 3D di *export* menjadi jenis file .fbx.

2.6.1 Unity3D

Unity3D suatu *game engine* yang semakin lama semakin berkembang. Unity3D merupakan salah satu *game engine* dengan lisensi *source proprietary*. Namun lisensi pengembangan dibagi menjadi 2, yaitu berbayar dan gratis sesuai dengan perangkat target pengembangan aplikasi. Publikasi aplikasi tidak dibatasi pada *game engine* ini. Pengguna lisensi gratis dapat mempublikasikan aplikasi yang dibangun tanpa harus membayar royalti atau biaya lisensi kepada unity. Namun pada penggunaan versi gratis ada pembatasan beberapa fitur atau modul tertentu ditiadakan yang hanya tersedia bagi pengguna unity berbayar.[8]



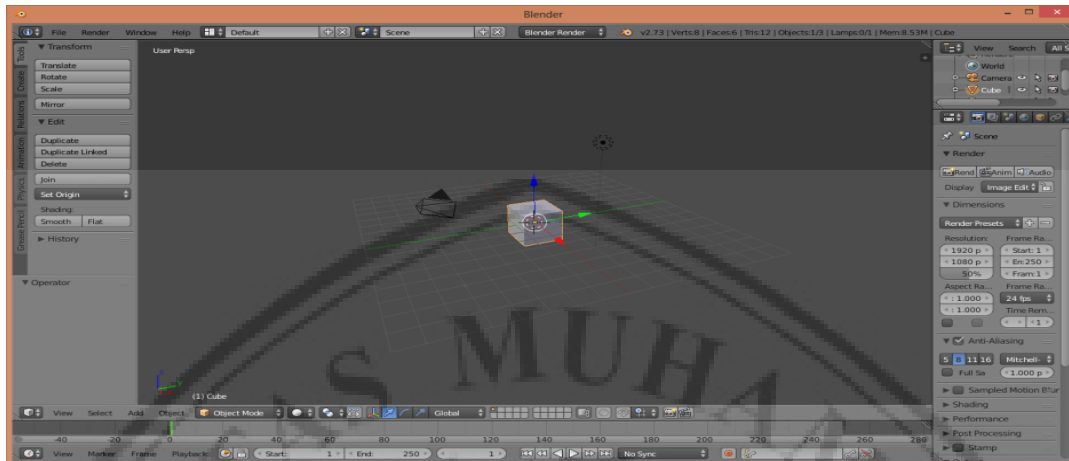
Gambar 2.7 Interface Unity3D [8]

Unity3D merupakan salah satu *game engine* yang populer dan sudah banyak digunakan oleh para *game development* di seluruh dunia terutama yang berbasis perorangan. Selain bisa membangun game PC, unity3D juga dapat membuat game konsol seperti Playstation, Nintendo Wii dan XBOX bahkan dengan unity3D kita dapat membangun game di smartphone berbasis android maupun IOS. Bahasa pemrograman yang bisa diterima unity3D adalah Java Script, CS Script (C#), dan BOO Script. Selain membangun game 3D pengguna juga dapat membangun sebuah game 2D dengan unity 3D bahkan dapat membangun web dengan menambahkan atau menginstall webplayer. Pada unity engine terdapat *framework* lengkap untuk pengembangan professional. Pada tugas akhir ini aplikasi unity3D digunakan sebagai *game engine* untuk membangun aplikasi *virtual reality* pada brosur perumahan ini.[8]

2.6.2 Blender

Blender merupakan program pengolah animasi 3 Dimensi bersifat opensource. Dimana blender ini bebas untuk dikembangkan oleh penggunanya, dapat didistribusikan kembali dan bersifat ilegal. Pada blender terdapat video compositor untuk pembuatan animasi dan *integrated game engine* untuk pembuatan game. Karya yang dihasilkan dengan menggunakan blender ini tidak

bersifat royalti kepada *developer* serta dapat dipublikasikan baik dikomersilkan bahkan dibagikan gratis.[10]



Gambar 2.8 Interface Blender [10]

Blender adalah salah satu dari sekian banyak program *Modeling* 3D dan Animasi, akan tetapi blender memiliki kelebihan sendiri dibandingkan program *modeling* 3D lainnya. Pada blender pengguna dapat membuat game tanpa tambahan lainnya, karena blender sudah memiliki *game engine* sendiri serta menggunakan bahasa pemrograman python. Blender merender grafiknya menggunakan OpenGL yang bisa digunakan berbagai macam sistem operasi seperti Linux, Windows, dan Mac OS. Pada tugas akhir ini aplikasi blender digunakan sebagai pengolah objek 3D binatang dan pohon.[10]

2.8 Perilaku Hewan

Setiap makhluk hidup akan melakukan interaksi dengan lingkungannya sejak pertama kali mereka dilahirkan. Untuk tetap bertahan sebagai makhluk hidup maka mereka harus mampu melakukan adaptasi baik pada level populasi maupun komunitas pada suatu biosfer.

Jika ditelaah berbagai macam interaksi pada makhluk hidup, maka banyak sekali contoh pola-pola perilaku yang diperlihatkan pada hewan dari berbagai taksa. Kajian mengenai perilaku tersebut telah dilakukan sejak lama oleh para peneliti dimana hewan dapat berkomunikasi, bergerak, melakukan interaksi social, dan mencari makanan. Kajian perilaku hewan pada dasarnya mengkaji bagaimana hewan-hewan berperilaku di lingkungan dalam berbagai situasi dengan pola yang spesifik.[13]

2.8.1 Perilaku Gajah

Gajah Sumatera memakan rumput-rumputan, daun, ranting, umbi-umbian dan kadang buah-buahan. Setidaknya terdapat 69 spesies tumbuhan yang bisa dijadikan pakan gajah. Tumbuhan tersebut terdiri dari 29 kelompok rumput-rumputan dan 40 kelompok tanaman non rumput. Gajah Sumatera diketahui lebih menyukai rumput-rumputan. Gajah mengambil tumbuhan yang diperoleh dengan cara mematahkan, menarik, mencabut, atau merenggut.

Efisiensi sistem pencernaan gajah sangat buruk. Hewan ini bisa membuang fesalnya setiap satu jam sekali. Tidaklah heran bila dalam sehari gajah Sumatera memerlukan makanan hingga 230 kg atau setara dengan 5-10% dari bobot tubuhnya. Sedangkan untuk minum dibutuhkan 160 liter air setiap hari. Gajah minum dengan cara menghisap/ menyedot air menggunakan belalainya dan menuangkan ke dalam mulutnya. Sumber air merupakan komponen pendukung kehidupan di habitat gajah. Biasanya sumber air tersebut dalam bentuk air mengalir maupun air yang tergenang. Di musim kemarau gajah Sumatera bisa menggali air di dasar sungai yang mengering hingga kedalaman satu meter.[14]

2.8.2 Perilaku Badak

Badak jawa sangat senang berendam dalam Lumpur, dimana dia dapat diam berdiri tegak didalam kubangan selama 4 (empat) sampai 6 (enam) jam. Tujuan mandi dan berendam dalam lumpur ini adalah untuk mendinginkan suhu badan dan kulit serta mencegah parasit yang sering mengganggu kulitnya. Oleh karena itu, kubangan menjadi sangat penting bagi sang badak untuk berendam, berjemur, bersantai bahkan untuk tidur. Sehingga tidak heran bila sang badak ini akan bertempur habis-habisan dalam mempertahankan atau memperebutkan kubangannya dari satwa lain yang senang berkubang seperti banteng dan babi hutan. Bila ini terjadi dan satwa lain tidak mau mundur, maka sang badak akan bertekad anda atau saya yang mati. Pertempuran yang terjadi akan berakhir dengan kematian disalah satu pihak. [15]

2.8.3 Perilaku Rusa

Secara umum baik rusa timor jantan maupun betina melakukan aktivitas *ingestive* (makan-minum) lebih banyak pada pagi dan sore hari, sedangkan pada siang hari lebih banyak waktu digunakan untuk istirahat. Secara relatif ada perbedaan alokasi waktu yang digunakan untuk aktivitas harian diantara rusa jantan dan betina. Untuk aktivitas makan, terlihat rusa betina relatif menggunakan waktu lebih lama dibanding rusa jantan baik pagi maupun sore hari, begitu pula untuk aktivitas lainnya. Satwa liar memiliki berbagai perilaku dan proses fisiologi untuk

menyesuaikan diri dengan keadaan lingkungan. Dalam mempertahankan hidup, rusa melakukan kegiatan-kegiatan yang agresif, melakukan persaingan, dan bekerjasama untuk mendapat makanan, perlindungan, pasangan untuk kawin, reproduksi dan sebagainya. Semua jenis rusa secara alami memiliki sifat yang selalu waspada. Pada saat rusa terganggu, biasanya mata dan telinga tertuju pada sumber gangguan. Semakin Rusa merasa terancam atau terganggu, maka kaki depan terlihat dihentakkan ke tanah, bulu di sekujur tubuh berdiri, dan diakhiri dengan mengeluarkan suara lengkingan sambil terus melarikan diri. Pada saat ketakutan, rusa timor akan lebih suka melarikan diri dengan sikap kepala yang menyeruduk.[16]

2.9 Finite State Machine

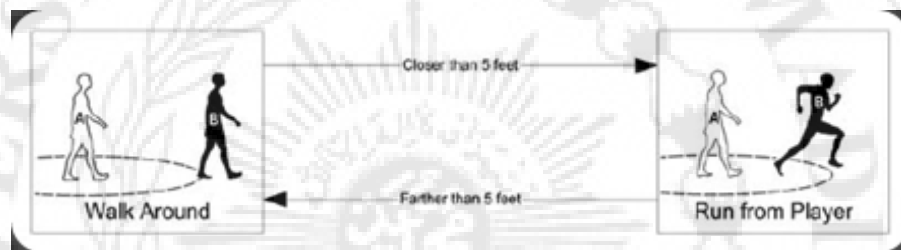
Finite State Machines (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State (Keadaan)*, *Event (kejadian)* dan *Action (aksi)*. Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan 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 (misal interupsi timer). 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 relative kompleks. Berdasarkan sifatnya, metode FSM ini sangat cocok digunakan sebagai basis perancangan perangkat lunak pengendalian yang bersifat reaktif dan real time. Salah satu keuntungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relative besar dengan hanya menggunakan sejumlah kecil item state. Selain untuk bidang kontrol, Penggunaan metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak game, aplikasi WEB dan sebagainya. Dalam bahasa pemrograman prosedural seperti bahasa C, FSM ini umumnya direalisasikan dengan menggunakan statemen kontrol switch case atau/dan if..then. Dengan menggunakan statemen-statement kontrol ini, aliran program secara praktis akan mudah dipahami dan dilacak jika terjadi kesalahan logika.

Finite State Machine di dunia AI Game Programming, merupakan salah satu teknik yang paling sering digunakan. Karena:

1. Implementasinya mudah dan cepat

2. Memudahkan proses debugging. Karena telah dipecah menjadi kepingan yang lebih kecil, proses debugging kalau terjadi behaviour yang tidak semestinya, menjadi lebih mudah
 3. Proses komputasi yg minimal, karena sejatinya FSM hanyalah conditional statement yang dikemas dalam bentuk yang lebih elegan.
 4. Fleksibel, dapat dikombinasikan dengan teknik AI lain misalnya fuzzy logic dan neural network
- Kekurangannya:

1. Behaviour dari agen mudah diprediksi, karena tidak ada searching dan atau learning di dalam agen tersebut
2. Karena mudah diimplementasi, kadang programmer langsung tembak di eksekusi tanpa melakukan desain FSM terlebih dahulu. Biasanya akan terjadi FSM yang terfragmentasi
3. Timbul apa yang dinamakan dengan State Oscillation yaitu ketika batasan antara dua buah state terlalu tipis:



Gambar 2.9 Gambaran *Finite State Machine*[17]

Gambar 2.9 menjelaskan batasan pada objek yang terlalu tipis, dari gambar sebelah kiri objek berjalan sejauh 5 kaki dan gambar sebelah kanan objek berlari 5 kaki. Bentuk Implementasi. Ada beberapa bentuk FSM, diantaranya:

2.9.1 Naive Approach

Menggunakan conditional statement (if-else atau switch-case) tanpa memecah object menjadi objek 2x yang lebih kecil sesuai state nya. Untuk agen yang cuma punya state yang sedikit, metode ini masih memungkinkan. Tapi kalau sudah kompleks, penggunaan metode ini jelas tidak dianjurkan, karena akan membentuk 'spaghetti code' dan monolithic conditional statement. Selain itu juga tidak scalable, tidak fleksibel, dan proses debugging menjadir lebih rumit.

2.9.2. State Transition Table

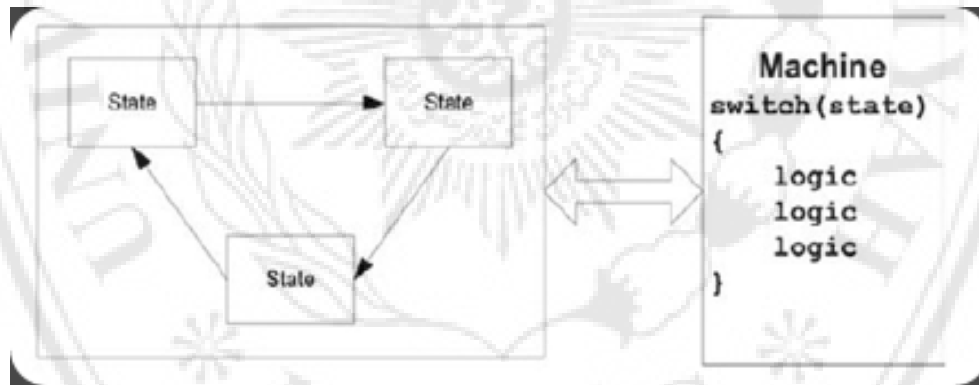
Bentuk ini sudah mengimplementasikan State Pattern, dengan menempatkan transition logic di context. Bentuk ini juga sering disebut sebagai Classic FSM. Dengan metode ini dibuat sebuah tabel yang dikenal dengan State Transition Table, bentuknya seperti ini: *State Transition Table*

Table 2.10 State Transition Table [17]

Current State	Condition	State Transition
Flee	Safe	Patrol
Attack	Health low	Flee
Patrol	See enemies AND Health not low	Attack
Patrol	See enemies AND Health low	Flee

Agen akan melakukan *query* dari tabel tersebut berdasarkan input yang diterima dari environmentnya. Kemudian ketika salah satu kondisi terpenuhi, dia akan mengubah current state menjadi state yang baru sesuai kondisinya.

Dengan begini, maka tentunya akan mempunyai fleksibilitas dan skalabilitas yang jauh lebih baik daripada jika menggunakan naive approach. Dengan drawback akan terbentuk monolithic conditional statements.



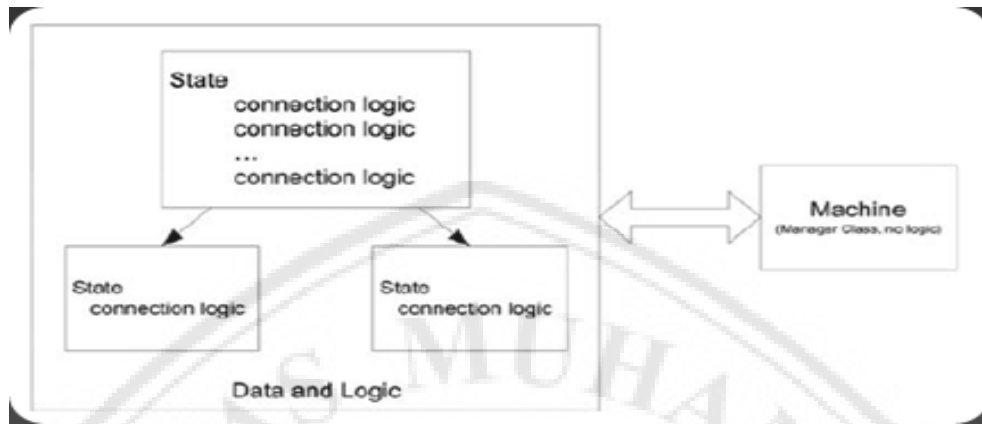
Gambar 2.11 Monolithic Conditional Statements [17]

Conditional Statement menyebabkan komputer mengecek suatu nilai dan memakai hasilnya sebagai dasar untuk memilih di antara beberapa alternatif kegiatan. Disebut juga *IF statement*.

2.9.3. Embedded Rules

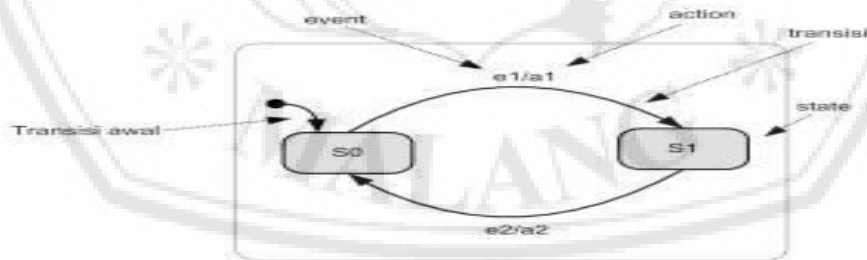
Bentuk ini adalah kebalikan dari bentuk *Classical Approach*, yang berarti *state transition* didefinisikan di *state* itu sendiri. Dan sama dengan *Classical Approach*, bentuk ini juga akan menawarkan fleksibilitas dan skalabilitas yang baik, namun dengan efek samping agak sulit untuk

di-mantain karena aturan-aturan transisi diletakkan di *state* sehingga ketika terjadi penambahan atau pengurangan *state*, maka harus dilakukan *update* juga terhadap *state2x* yang terkait.



Gambar 2.12 Embedded Rules [17]

Diagram keadaan pada dasarnya merupakan salah satu bentuk representasi dari FSM. Diagram ini secara visual menggambarkan tingkah laku yang dimiliki oleh sistem kontrol yang kompleks kedalam bentuk yang lebih sederhana dan relatif mudah dipahami. Dalam diagram ini, *state-state* yang terdapat pada sebuah sistem digambarkan sebagai lingkaran yang diberi label unik, sedangkan transisi *state* yang diakibatkan oleh *event* tertentu direpresentasikan sebagai anak panah yang berasal dari *state* yang ditinggalkan menuju *state* yang aktif. Setiap transisi yang terjadi umumnya juga diikuti oleh aksi yang dilakukan oleh sistem yang dirancang. Secara praktis setiap diagram *state* yang dirancang akan selalu memiliki sebuah transisi awal (inisial) yang menuju salah satu *state* sejak sistem kontrol tersebut mulai dihidupkan. Gambar berikut memperlihatkan contoh penggambaran diagram *state*:



Gambar 2.13 Diagram Finite State Machine [17]

Diagram tersebut memperlihatkan FSM dengan dua buah *state* (S0 dan S1) dan dua buah input (e1 dan e2) serta dua buah aksi (a1 dan a2) *output* yang berbeda : seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *State0*, pada keadaan ini sistem akan menghasilkan *Action2* jika terjadi masukan *Event2*, sedangkan jika terjadi *Event1* maka *Action1* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.[17]