

BAB 2

LANDASAN TEORI

2. 1. Teori Umum

2. 1. 1. Pengertian Game

Game sudah banyak dikenal oleh orang sejak dahulu, hampir semua orang pernah memainkan *game* dikarenakan saat ini *game* merupakan salah satu kebutuhan bagi manusia, berikut adalah beberapa definisi daripada *game* sebagai berikut :

1. Menurut gamedev (<http://www.gamedev.net>), *game* adalah sesuatu yang bersifat interaktif, memiliki peraturan sendiri dimana mengandung tantangan dan suatu kondisi menang yang mendefinisikan suatu realitas dengan tujuan untuk menghibur.
2. Menurut wikipedia (<http://en.wikipedia.org>), *game* adalah aktivitas yang melibatkan satu atau lebih pemain. *Game* dapat pula diartikan sebagai tujuan yang ingin dicapai pemain atau sekumpulan aturan yang menandakan apa yang dilakukan pemain dan yang tidak dapat dilakukan. *Game* dimainkan terutama untuk hiburan, kesenangan, tetapi juga dapat berfungsi sebagai sarana latihan, pendidikan dan simulasi.
3. Menurut Encarta Encyclopedia 2006, *game* adalah aktivitas atau kontes yang diatur oleh sekumpulan aturan tertentu. *Game* ditujukan

sebagai sarana rekreasi atau untuk mengembangkan kemampuan mental atau fisik. *Game* memiliki banyak variasi. *Game* dapat memiliki jumlah pemain tertentu dan dapat dimainkan dalam bentuk kompetisi atau kooperatif.

2. 1. 2. Pengertian Personal Computer (PC) Game

Menurut webopedia (<http://www.webopedia.com>), *Personal Computer* lebih biasa disebut dengan *computer games* atau *PC games*. Orang-orang bermain di *personal computer* dengan komponen komputer yang standar yaitu *keyboard* dan *mouse*, atau *joystick* atau *gamepad*. *Video* balik yang dapat diterima oleh *gamer* didapatkan melalui *monitor*, dan *sound* melalui *speakers* atau *headphones*.

2. 1. 3. Jenis (Genre) Game

Menurut Andrew Rollings dan Dave Morris (2004.p12), *game* memiliki *genre* yang luas, antara lain :

- a. *Sports – Type game* ini terdiri dari *game* yang menciptakan tiruan dari *sports* asli jadi pemain bisa merasakan seperti bermain *sports* asli hanya dengan memainkan *game*-nya
- b. *Action – Type game* ini menyediakan pemain dengan beberapa rintangan dan musuh yang harus dikalahkan dengan menggunakan reaksi cepat sang pemain dan biasanya tidak memberikan banyak teka-teki pada pemain.

- c. *Adventure – Type game* ini terdiri dari *game* yang ada beberapa tantangan untuk pemain selama pemain mengendalikan 1 atau lebih karakter yang berkelana tempat yang disediakan.
- d. *Strategy* – Pemain diberi kesempatan untuk memikirkan strategi untuk menyelesaikan masalah atau mendapatkan gol, biasanya *setting*-nya tentang peperangan.
- e. *Simulation – Type* ini menyediakan simulasi dari aktifitas asli, jadi pemain diharapkan untuk mempunyai pengalaman yang sama tanpa melibatkan aktifitas yang asli.
- f. *Puzzle – Game* ini tentang pemain mencoba untuk menyelesaikan teka-teki yang di berikan oleh *game*.
- g. *Role Playing Game (RPG)* – Dalam jenis ini pemain diharapkan untuk mengikuti cerita dan diberikan 1 karakter atau lebih untuk di kendalikan, karakter akan di berikan perjalanan dan akan mendapatkan pengalaman dengan mengalahkan musuh.
- h. *Toys* – *game* yang sangat menarik, seperti tetris.
- i. *Educational* – Belajar sambil melaksanakan permainan

Beberapa *genre game* juga dapat dikombinasikan dalam tiap-tiap *game* karena *game* sangat kaya atas daya tampung dan tidak bisa dimasukan 1 *genre*.

2. 1. 4. Pengertian Engine Game

Menurut Kenneth C. Finney (2004), *engine game* menyediakan fitur yang paling penting dari dunia pemrograman *game* : perenderan 3D, jaringan, grafik, dan *scripting*. *Engine game* juga menyediakan perenderan dari lingkungan *game*. Setiap *game* menggunakan *system* yang berbeda untuk mengatur bagaimana aspek *visual* dari *game* akan dibentuk.

Dengan menciptakan lingkungan grafik yang konsisten dan mempopulasikan lingkungan itu dengan objek-objek yang mematuhi hukum fisika, *engine game* memungkinkan *game* untuk dapat berkembang sesuai dengan garis produksi dan jalan cerita yang masuk akal. *Engine game* mengenkapsulasi karakteristik dari dunia nyata, seperti waktu, gerakan, efek gravitasi, dan hukum fisika lainnya.

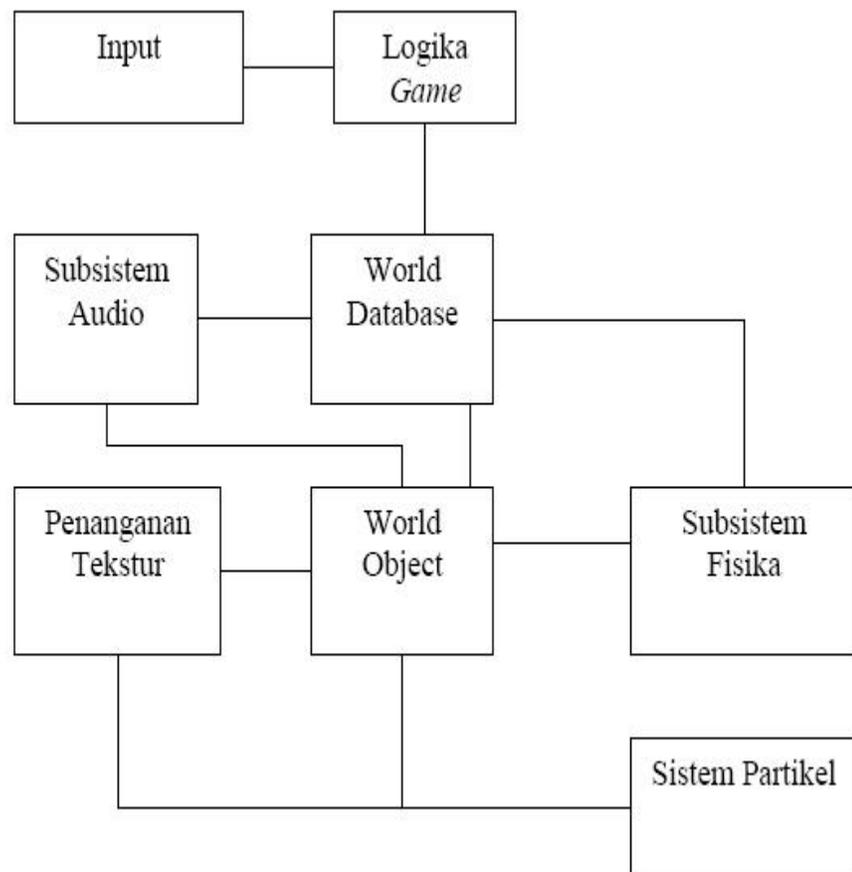
Menurut Michael Morrison (2002), *engine game* berguna di dalam situasi ketika anda berencana membuat lebih dari satu *game*, dan anda tak perlu menemukan roda baru lagi (membuat *background*, *layer* judul, musik latar) setiap kali anda akan membuat *game*.

Keuntungan menggunakan *engine game* untuk *game* Windows adalah *engine game* dapat menyembunyikan kode-kode Windows yang *relative* sulit, yang sebenarnya tidak terlalu berhubungan dengan *game* yang akan dibuat sehingga anda dapat *focus* pada kode *game* anda.

Engine game juga mewakili organisasi dari kode *game* sehingga tugas aplikasi umum terpisah dari tugas yang berhubungan dengan *game*.

2. 1. 5. Komponen-komponen Engine Game

Menurut Kevin Hawkins dan Dave Astle (2001,p708), *engine game* yang sederhana terdiri dari komponen-komponen berikut ini :



Gambar.2.1. Komponen Engine Game menurut Kevin Hawkins dan Dave Astle

Berdasarkan gambar diatas, pada dasarnya *engine game* menerima *input* melalui subsistem *Input* dan mengirimkan sebuah pesan kepada subsistem Logika *Game*, yang kemudian menangani pesan dan mengeksekusi *game cycle*. Dalam sebuah *game cycle*, subsistem Logika *Game* menanggapi *Input*, melakukan perhitungan fisika yang dibutuhkan pada objek *game*, menangani deteksi dan respon benturan, *load* dan menghancurkan objek, menggerakkan kamera, dan memainkan bunyi yang dibutuhkan selama *game* berjalan.

2. 1. 6. Jenis Engine Game

Jenis-jenis Engine Game yang saat ini sering digunakan adalah:

a. Torque

Torque adalah *engine game* kelas AAA yang dapat bejalan di berbagai *platform* seperti Linux, Windows, Mac dengan harga yang murah. Fitur-fitur yang dimiliki *engine* Torque adalah kode jaringan *multi-player*, *engine* perenderan untuk di dalam maupun luar ruangan, animasi skeletal, pembuatan GUI dengan cara *drag and drop*, *world editor* yang sudah terintegrasi, dan bahasa *scripting* yang mirip bahasa C.

b. 3D Game Studio

3D *GameStudio*, sering dikenal sebagai *Gamestudio* atau 3DGS adalah pengembangan permainan 3D sistem komputer yang memungkinkan pengguna untuk membuat *game* 3D dan aplikasi *virtual reality* yang lain, dan mempublikasikan mereka bebas royalti. Termasuk model / daerah *editor*, *level editor*, *script editor* / *debugger* dan dilengkapi dengan sebuah kumpulan besar textures, model dan karya seni, serta permainan sistem template yang memungkinkan penciptaan dasar shooter games atau *RPGs* tanpa pemrograman. Untuk permainan kompleks atau aplikasi lain yang dapat anda gunakan salah satu bahasa scripting yang terintegrasi bernama *Lite-c* atau eksternal pengembangan bahasa seperti *Visual C + +* atau *Delphi*.

c. True Vision 3D

TV3D SDK adalah multi-bahasa mesin 3D, terutama yang ditulis dalam C + + dengan dukungan untuk Visual Basic 6, Visual Basic.Net, C #, Delphi, Python, dan C + +. Dibangun di atas platform DirectX padat, Anda dapat dengan cepat dan mudah prototipe kompleks game 3D dan aplikasi dalam sampai 80% lebih sedikit waktu.

d. XNA (untuk XBOX)

XNA Framework yang didasarkan pada pelaksanaan asli. NET *Compact Framework* 2,0 untuk Xbox 360. Termasuk yang ekstensif set *class libraries*, permainan khusus untuk pembangunan, untuk meningkatkan maksimum *code reuse* di target platform. Framework berjalan pada versi *Common Language Runtime* yang dioptimalkan untuk bermain game untuk menyediakan lingkungan eksekusi dikelola. *Runtime* yang tersedia untuk Windows XP, Windows Vista, dan Xbox 360. Sejak permainan XNA ditulis untuk runtime, mereka dapat berjalan pada platform apapun yang mendukung XNA *Framework* dengan sedikit atau tanpa modifikasi. Game yang berjalan di *framework* dapat ditulis teknis dalam bahasa NET-compliant apapun, tetapi hanya C # dan XNA Game Studio *Express* IDE dan semua versi *Visual Studio* 2005 yang secara resmi didukung.

e. RPG Maker

RPG Maker adalah program yang memungkinkan pengguna untuk membuat *role-playing games*. Kebanyakan versi termasuk *tile set* ditetapkan berdasarkan *map editor*, (*tilesets* dipanggil *chipset* dalam versi pra-XP), *scripting language* sederhana untuk *scripting events*, dan *battle editor*. Termasuk semua versi awal premade *tilesets*, karakter, dan acara yang dapat digunakan dalam menciptakan permainan baru. Fitur yang menarik dari program RPG Maker Versi PC

adalah pengguna dapat membuat tilesets baru dan karakter, dan menambahkan grafis baru yang dia mau. Beberapa situs telah dikembangkan, seperti *Gaming World* dan *RPG Maker Network*, yang didedikasikan untuk membantu pengguna berbagi kreasi mereka.

2. 1. 7. Pengertian Unified Modelling Language (UML)

Menurut Bernd Brügge dan Allen H Dutoit (2000, p24), notasi membantu kita menyampaikan ide kompleks secara ringkas dan tepat. Dalam proyek yang melibatkan banyak anggota, seringkali dalam latar belakang kebudayaan dan teknik yang berbeda, ketepatan dan kejelasan adalah aspek yang sangat penting mengingat begitu mudahnya terjadi salah pengertian. Agar suatu notasi dapat menjadi alat komunikasi yang akurat, notasi tersebut harus memiliki semantik yang telah didefinisikan dengan baik, dapat merepresentasikan aspek dari sistem secara tepat, dan dapat dimengerti dengan baik oleh para anggota tim.

Menurut Whitten, Bentley, dan Dittman (2004, p430), *Unified Modeling Language*TM (UML) adalah sebuah pendekatan untuk (1) mempelajari objek-objek yang ada untuk melihat apakah objek tersebut dapat digunakan kembali atau dimodifikasikan untuk kegunaan baru, dan (2) mendefinisikan objek baru atau yang telah di modifikasi yang akan digabungkan dengan objek yang ada untuk membuat aplikasi bisnis

Menurut Timothy C. Lethbridge dan Robert Laganiere (2002, p151), UML adalah standar bahasa grafis untuk memodelkan *software* berorientasi objek. UML dikembangkan pada pertengahan tahun 1990an oleh James Rumbaugh, Grady Booch, dan Ivar Jacobson dimana mereka telah mengembangkan notasi mereka masing-masing pada awal 1990an.

Pada November 1997, UML diresmikan sebagai standar untuk pemodelan objek oleh Object Management Group (OMG).

Menurut Martin Fowler dan Kendall Scott (2000, p13), UML adalah bahasa pemodelan, bukan suatu metode. UML tidak memiliki notasi atas process yang merupakan bagian penting dari metode.

2. 1. 8. Jenis-jenis UML

Menurut Whitten, Bentley, dan Dittman (2004, p441), UML terdiri atas sembilan diagram yang dikelompokkan dalam lima katagori berdasarkan sudut pandangannya, yaitu :

Kategori 1 : Use – Case Model Diagram

Use-case diagram menggambarkan interaksi antara sistem dengan luar sistem dan sistem dengan *user*. Dengan kata lain, *use case diagram* secara grafis menggambarkan siapa yang akan menggunakan sistem dan dalam cara apa *user* ingin berinteraksi dengan sistem.

Kategori 2 : Static Structure Diagram

UML menawarkan dua diagram untuk memodelkan struktur statis dari sistem informasi, yaitu :

a. Class Diagram

Class diagram melukiskan struktur sistem dalam bentuk objek. Disini digambarkan objek *class* yang membangun system beserta hubungan antar *class*.

b. Object Diagram

Object diagram serupa dengan *class diagram*, tetapi disamping menggambarkan object *class*, digambarkan juga objek *instance* yang menampilkan nilai atribut dari *instance*. Diagram ini dapat digunakan untuk membantu tim pengembangan memahami struktur sistem lebih baik.

Kategori 3 : Interaction Diagram

Interaction diagram memodelkan interaksi, terdiri atas sekumpulan object, hubungan, dan pesan yang dikirimkan antara objek tersebut. Diagram ini memodelkan aspek dinamis dari sistem.

UML memiliki dua diagram untuk tujuan ini, yaitu :

a. Sequence Diagram

Sequence diagram secara grafis menggambarkan bagaimana objek saling berinteraksi melalui pesan dalam melakukan suatu operasi atau melakukan *use case*.

b. Collaboration Diagram

Collaboration diagram serupa dengan *sequence diagram*, tetapi yang difokuskan disini bukanlah urutan (*sequence*) melainkan interaksi antar objek dalam format jaringan.

Kategori 4 : State Diagram

State diagram juga memodelkan aspek dinamis dari sistem. UML memiliki diagram untuk memodelkan perilaku kompleks dari objek dan diagram untuk memodelkan perilaku dari *use case* atau metode.

Diagram tersebut adalah:

a. Statechart Diagram

Statechart Diagram digunakan untuk memodelkan aspek dinamis dari suatu objek. Pada *statechart diagram* diilustrasikan daur hidup objek, berbagai keadaan objek, dan peristiwa yang menyebabkan transisi dari keadaan yang satu ke keadaan yang lain.

b. Activity Diagram

Activity diagram digunakan untuk menggambarkan urutan aktivitas secara berurutan dari proses bisnis atau *use case*.

Kategori 5 : Implementation Diagram

Implementation diagram juga memodelkan struktur dari sistem informasi.

Yang termasuk dalam *implementation diagram* adalah :

a. Component Diagram

Component diagram digunakan untuk menggambarkan organisasi dari sistem dan ketergantungan dari komponen *software* dalam sistem. *Component diagram* dapat juga digunakan untuk menunjukkan bagaimana kode program dibagi menjadi modul-modul (atau komponen).

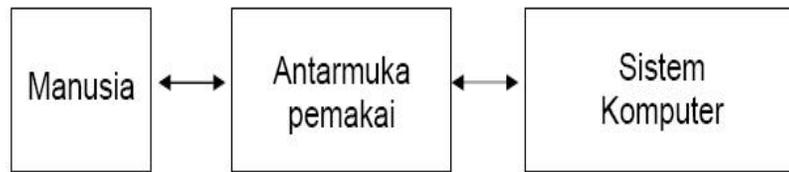
b. Deployment Diagram

Deployment Diagram mendeskripsikan arsitektur fisik dalam 'node' untuk *hardware* dan *software* dalam sistem. Disini digambarkan konfigurasi dari komponen *software*, *processor*, dan peralatan lain yang membangun arsitektur sistem secara *run-time*.

2. 1. 9. Pengertian Interaksi Manusia Komputer (IMK)

Menurut Ben shneiderman (1997), Interaksi manusia dan computer (IMK) atau *Human-Computer Interaction* (HCI) adalah disiplin ilmu yang berhubungan dengan perancangan, evaluasi, dan implementasi sistem komputer interaktif untuk digunakan oleh manusia, serta studi fenomena-fenomena besar yang berhubungan dengannya.

Fokus pada IMK adalah perancangan dan evaluasi antarmuka pemakai (*user interface*). Antarmuka pemakai adalah bagian sistem komputer yang memungkinkan manusia berinteraksi dengan komputer.



Gambar. 2. 2. Fokus Pada Interaksi Manusia dan Komputer (IMK)

2. 1. 10. Konsep Perancangan Antar Muka Pemakai

Dalam IMK, terdapat 8 (delapan) aturan (*golden rules*) yang digunakan dalam perancangan antarmuka pemakai yaitu :

- a. Berusaha untuk konsisten
- b. Memungkinkan *frequent users* menggunakan *shortcuts*
- c. Memberikan umpan balik yang informative
- d. Merancang dialog yang memberikan penutupan (keadaan akhir)
- e. Memberikan pencegahan kesalahan dan penanganan yang sederhana
- f. Memungkinkan pembalikan aksi yang mudah
- g. Mendukung pusat kendali internal (*internal locus of control*)
- h. Mengurangi beban ingatan jangka pendek

2. 1. 11. Pengertian Multimedia

Menurut PC Webopaedia. Multimedia adalah penggunaan komputer untuk menampilkan teks, grafik, video, animasi, dan suara dalam bentuk terpadu.

Menurut Wikipedia (<http://en.wikipedia.org/wiki/Multimedia>), multimedia adalah penggunaan beberapa media yang berbeda

menyampaikan informasi (teks, audio, grafik, animasi, video, dan interaktivitas). Multimedia juga mengacu kepada alat penyimpanan komputer, terutama yang digunakan untuk menyimpan isi multimedia.

2. 1. 12. Elemen Multimedia

Elemen-elemen yang terdapat dalam multimedia adalah :

a. Teks

Merupakan dasar penyampaian informasi, juga merupakan media paling sederhana yang direpresentasikan dengan jenis huruf (*typeface*) yang beragam agar harmonis dengan elemen media lainnya.

b. Citra Diam (gambar)

Merupakan representasi spesial dari objek yang disusun sebagai matriks nilai numerik yang merepresentasikan setiap titik / *pixel* serta diciptakan dengan program *paint / image editing*.

c. Suara

Merupakan fenomena fisik yang dihasilkan oleh pergetaran materi.

Audio memiliki 3 kategori yaitu :

- Ucapan (*speech*) : suara orang berbicara
- Musik (*music*) : hasil pendengaran alat musik
- Efek Suara (*sound effect*) : suara lainnya seperti tembakan, gelas pecah, halilintar.

d. Animasi

Merupakan penayangan *frame-frame* gambar secara cepat untuk menghasilkan kesan gerakan.

e. Video

Video sama seperti animasi, tetapi disimpan dalam format khusus yang dapat menyimpan adegan dunia nyata atau rekaan dengan komputer.

Video adalah elemen yang paling kompleks, dan paling memerlukan persyaratan *hardware* yang tinggi.

2. 1. 13. Artificial Intelligence (AI)

Menurut Sri Kusumadewi (2003,p1), artificial intelligence merupakan salah satu bagian ilmu computer yang membuat agar mesin (computer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia.

Kecerdasan buatan atau “Artificial Intelligence” itu sendiri dimunculkan oleh seorang professor dari Massachusetts Institute of Technology yang bernama John McCarthy pada tahun 1956 pada Darmouth Conference yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu : mengetahui dan memodelkan proses-proses berfikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut.

2. 1. 14. Evaluasi Game

Evaluasi *Game* yang biasa diterapkan pada setiap *game* yang beredar secara umum adalah sebagai berikut:

1. Kuesioner
2. 8 Aturan Emas
3. 5 Elemen Multimedia
4. Perbandingan *game* sejenis

2. 2. Teori Khusus

2. 2. 1. Engine Torque

Menurut Edward F. Maurina III (p3), Torque adalah *engine game* kelas AAA yang dapat berjalan di berbagai *platform* seperti Linux, Windows, Mac dengan harga yang murah. Fitur-fitur yang dimiliki *engine* Torque adalah kode jaringan *multi-player*, *engine* perenderan untuk di dalam maupun luar ruangan, animasi skeletal, pembuatan GUI dengan cara *drag and drop*, *world editor* yang sudah terintegrasi, dan bahasa *scripting* yang mirip bahasa C.

2. 2. 2. Jenis-jenis Game Engine Torque

Pada kenyataannya, Jenis-jenis *game* yang telah berhasil dikembangkan oleh Torque antara lain seperti FPS, Sport, Adventure dan RPG. Torque adalah teknologi asli yang digunakan dalam pembuatan *game* Tribes, Starsiege, dan Tribes 2. Torque sudah terbukti di bidang

industri dan sekarang adalah bagian dari *game-game* indie yang sedang digemari seperti Marble Blast, Orbz, dan ThinkTanks.

2. 2. 3. Pengertian Dimensi

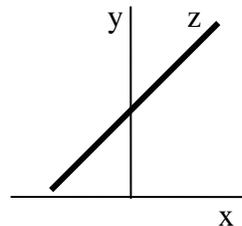
Menurut Foley, Van Dam, Feiner dan Hughes (1995, p105), dimensi dapat diartikan sebagai banyak cara untuk menentukan posisi sebuah benda yang didasarkan terhadap acuan tertentu. Sebuah benda dikatakan berdimensi 1 (satu) jika posisinya dapat ditentukan dengan angka. misalnya, sebuah kurva / garis (kurva atau garis merupakan sekumpulan titik-titik) dimana posisinya dapat ditentukan oleh sebuah angka yang menyatakan jarak suatu titik terhadap titik awal. Benda berdimensi 2 memiliki posisi yang dapat ditentukan oleh 2 angka, misalnya suatu permukaan bola dapat diukur dengan angka derajat lintang dan angka derajat bujur.

Menurut Dick T (2002), dimensi juga dapat diartikan sebagai banyaknya arah yang bersifat bebas (tidak saling memotong) yang dapat digunakan untuk menentukan posisi benda, contohnya atas-bawah, kanan-kiri, depan-belakang. Sebuah benda berdimensi 2 (dua) dapat ditentukan dengan arah kiri-kanan dan atas-bawah.

2. 2. 4. Pengertian 3 Dimensi

Dunia 3 Dimensi (3D) tidaklah sesederhana dunia 2 dimensi yang hanya mempunyai 2 koordinat saja, melainkan memiliki 3 buah koordinat atau axis, yaitu x, y, dan z. Axis x adalah axis mendatar atau horizontal, axis y adalah axis tegak atau vertikal, sedang axis z adalah axis yang menembus layar monitor atau ke dalam.

Untuk lebih jelasnya, koordinat atau axis dalam 3D dijelaskan pada gambar berikut:



Gambar 2.3. Bidang Cartesian 3 Dimensi

Dalam penglihatan matematisnya, objek di dalam ruang 3 dimensi mempunyai kerumitan tersendiri, diantaranya:

- a. Setiap objek bergerak mengikuti arah mata angin
- b. Digunakan teknik penghitungan tabrakan (*collision*) antara koordiant x, y, dan z

- c. Pergerakan objek di dalam 3D harus dilakukan dalam perhitungan yang cepat untuk mendapatkan tampilan gerakan yang bagus di layar.

Yang menjadi permasalahan adalah layar monitor merupakan layar 2D. Untuk dapat menampilkan objek-objek 3D pada layar 2D sehingga terlihat 3D perlu dilakukan proyeksi 3D, yaitu memproyeksikan koordiant 3D (x, y, z) dari suatu objek sehingga menjadi koordiant 2D (x, y) pada layar monitor.

2. 2. 5. Konsep Umum Grafik 3 Dimensi

Koordinat adalah konsep dasar dalam grafik 3 dimensi. Tampilan objek 3 dimensi adalah tampilan yang dibagi berdasarkan koordinat suatu titik benda dalam jarak koordinat tertentu. Sistem grafik 3 dimensi didasarkan pada jarak koordinat (x, y, z) , yang kadang menimbulkan keraguan atau kebingungan dalam melihat letak titik di layar.

2. 2. 6. Model 3 Dimensi

Model 3D adalah bentuk konstruksi untuk mensimulasi dan menolong dalam memahami suatu konsep model geometri yang berisi informasi deskripsi objek. Objek 3D digambarkan ke dalam layar untuk menciptakan gambaran dari keseluruhan dunia buatan ke dalam simulasi dunia nyata. Objek dalam layar ditransformasikan ke titik-titik koordinat dan dialokasikan untuk membentuk imajinasi sebuah dunia 3D.

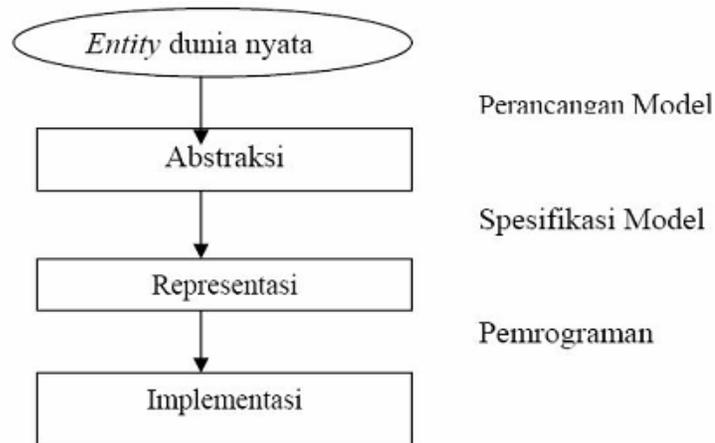
Bagian-bagian dari model :

- a. Representasi : mendefinisikan suatu bentuk
- b. Penggunaan *interface* : mendefinisikan spesifikasi bentuk
- c. Peralatan : memanipulasi bentuk

Sebagai contoh, objek 3D bisa diwakilkan dengan batas permukaan bidang vertex ataupun titik untuk dapat melihat tampilan dalam dunia yang sesungguhnya. Model harus digambarkan atau di-*render* untuk membedakan model dengan tiruannya. Model mendeskripsikan objek dan alat-alatnya sedangkan *rendering* mengubah bentuk model ke dalam gambar di layar. Akhirnya objek dan alat yang berinteraksi akan memanipulasi model setelah proses. Sebagai pelengkap, digunakan efek cahaya dan kamera yang posisinya dapat diatur sehingga memberikan pandangan pada model.

Fasilitas lain untuk membuat model terlihat nyata adalah teknik perwarnaan. Teknik ini menghasilkan spesifikasi warna yang *solid* yang dilakukan dengan pencampuran warna. Disediakan juga pendekatan *visual* atau teknik analisa warna untuk pengembangan imajinasi warna sebelum diterapkan pada objek. Struktur logika untuk mendiskusikan model adalah penelusuran yang dimulai dari abstraksi, direpresentasikan, dan diimplementasikan.

Menurut Banzal (Academic press, 1992), struktur model digambarkan seperti berikut:



Gambar.2. 4. Struktur Model

2. 2. 7. **Komponen-komponen Game**

Menurut Andrew Rollings dan Dave Morris (2004,p42), *game* memiliki 5 komponen terpenting, yaitu :

a. **Fitur**

Fitur merupakan salah satu yang membedakan setiap *game* yang ada. Fitur disini dapat merupakan fitur yang dapat membantu menggambarkan jalan cerita *game* ke dalam bentuk-bentuk yang dapat dilihat maupun dirasakan.

b. **Gameplay**

Gameplay dapat membantu para *developer* untuk mengetahui cara kerja suatu *game*, dimana fitur yang dibuat akan membentuk suatu *gameplay*.

c. Interface

Interface merupakan setiap tampilan yang ada di dalam *game*. *Interface* yang baik adalah dengan icon yang sangat minimum dan tidak membosankan para pemain *game*.

d. Aturan

Aturan merupakan kumpulan peraturan-peraturan dalam *game*.

e. Desain level

Desain *level* disini mencakup *style*, *background*, dan jalan cerita dari *game*.

2. 2. 8. Game 3D

Menurus Andrew Rollings dan Dave Morris (2004,p515-521), industri *game* selalu berusaha untuk mengikuti perkembangan teknologi yang ada di dunia ini. Ketika perangkat-perangkat *computer* seperti *processor*, *graphic card* versi baru mulai muncul di pasaran, para *developer game* selalu berusaha untuk mengikuti perkembangan tersebut.

Pada saat kemampuan proses pada *computer* semakin cepat, para *developer* juga senantiasa menciptakan *game* yang semakin canggih sehingga munculah *engine* dengan grafik 3D (3 Dimensi). Maka dari itu, *game* 3D dengan hitungan *polygon* yang sangat besar dan pencahayaan yang sudah canggih, juga *textur mapping* mulai diproduksi. *Game* 3D mempresentasikan objek dalam bentuk 3 dimensi sehingga objek akan terlihat lebih nyata seperti dalam kehidupan nyata.

2. 2. 9. Game Balance

Menurut Andrew Rollings dan Dave Morris (2004,p105-112), *game balance* adalah hal yang penting dalam perancangan *game*. Jika suatu *game* tidak seimbang (tidak *balance*), maka ada beberapa bagian dari *game* yang sangat jarang digunakan sehingga membuat pengembangan *game* menjadi sia-sia.

Ada 3 jenis *game balance*, yaitu :

a. Player/Player Balance

Jika *game* termasuk dalam *multiplayer game*, masing-masing *player* memiliki keuntungan yang sama kecuali keahlian *player*.

b. Player/Game Balance

Pada *player/gameplay balance*, keberhasilan *player* dalam tiap-tiap *level game* akan mendapatkan imbalan yang sesuai dengan keberhasilan yang dicapai.

c. Gameplay/Gameplay Balance

Fitur-fitur yang ada dalam *game* haruslah seimbang, contohnya jika sebuah pedang dalam *game* dapat menghancurkan dua kali lipat daripada pedang lainnya, maka pedang tersebut adalah pedang yang harus didapatkan dengan melewati *level game* yang lebih tinggi.

2. 2. 10. Teknik *Artificial Intelligence* pada *Game RPG*

Ada banyak teknik yang digunakan oleh game desainer untuk *Artificial Intelligence* dari karakter. Beberapa teknik yang populer adalah:

- *Randomization*, dalam teknik ini adalah tindakan yang diberi label atau nomor lalu *Artificial Intelligence* akan secara acak memilih tindakan.
- *Simple Rules*, teknik ini di dalam *Artificial Intelligence* memilih tindakan berdasarkan aturan yang diberikan oleh perancang. Aturan-aturan yang diberikan untuk menentukan tindakan yang harus diambil oleh *Artificial Intelligence* jika kondisi tertentu terjadi.
- *Pattern*, dalam teknik ini *Artificial Intelligence* yang diberikan pola oleh perancang. Kemudian *Artificial Intelligence* akan bertindak berdasarkan pola. Jadi, ini merupakan tindakan berulang yang dilakukan oleh *Artificial Intelligence*.

Ini adalah teknik yang populer digunakan pada *Role Playing Game*. Ada banyak teknik lainnya, namun teknik di atas sebagian besar ditemukan pada *Role Playing Game*.