BAB II
TINJAUAN PUSTAKA

2.1 Hutan

Hutan adalah suatu wilayah yang memiliki banyak tumbuh-tumbuhan lebat seperti pohon, semak, paku-pakuan, rumput, jamur dan lain sebagainya serta menempati daerah yang cukup luas. Hutan berfungsi sebagai penampung karbon dioksida (*carbon dioxide sink*), habitat hewan, modulator arus hidrologika, dan pelestari tanah serta merupakan salah satu aspek biosfer bumi yang paling penting. Hutan adalah bentuk kehidupan yang tersebar di seluruh dunia. Kita dapat menemukan hutan baik di daerah tropis maupun daerah beriklim dingin, di dataran rendah maupun di pegunungan, di pulau kecil maupun di benua besar.

Menurut Undang-undang Nomor 41 Tahun 1999 tentang Kehutanan, pengertian hutan adalah suatu kesatuan ekosistem berupa hamparan lahan berisi sumberdaya alam hayati yang didominasi pepohonan dalam persekutuan alam lingkungan, yang satu dengan yang lainnya tidak dapat dipisahkan. Definisi hutan yang disebutkan di atas, terdapat unsur-unsur yang meliputi: [11]

1. Suatu kesatuan ekosistem
2. Berupa hamparan lahan
3. Berisi sumberdaya alam hayati beserta alam lingkungannya yang tidak dapat dipisahkan satu dengan yang lainnya.
4. Mampu memberi manfaat secara lestari

Keempat ciri pokok dimiliki suatu wilayah yang dinamakan hutan, merupakan rangkaian kesatuan komponen yang utuh dan saling ketergantungan terhadap fungsi ekosistem di bumi. Eksistensi hutan sebagai subekosistem global menenpatikan posisi penting sebagai paru-paru dunia.

Unsur pokok yang terkandung di dalam definisi kawasan hutan, dijadikan dasar pertimbangan ditetapkannya wilayah-wilayah tertentu sebagai kawasan hutan. Kemudian, untuk menjamin diperolehnya manfaat yang sebesar-besarnya dari hutan dan berdasarkan kebutuhan sosial ekonomi masyarakat serta berbagai faktor pertimbangan fisik, hidrologi dan ekosistem, maka luas wilayah yang minimal harus dipertahankan sebagai kawasan hutan adalah 30% dari luas daratan [11].

Berdasarkan kriteria pertimbangan pentingnya kawasan hutan, maka sesuai dengan peruntukannya menteri menetapkan kawasan hutan menjadi:

1. Wilayah yang berhutan yang perlu dipertahankan sebagai hutan tetap
2. Wilayah tidak berhutan yang perlu dihutankan kembali dan dipertahankan sebagai hutan tetap

Pembagian kawasan hutan berdasarkan fungsi-fungsinya dengan kriteria dan pertimbangan tertentu, ditetapkan dalam Peraturan Pemerintah RI No. 34 tahun 2002 tentang Tata Hutan dan Penyusunan Rencana Pengelolaan Hutan, Pemanfaatan Hutan dan Penggunaan Kawasan Hutan Pasal 5 ayat (2), sebagai berikut : [12]

1. Kawasan hutan konservasi yang terdiri dari kawasan suaka alam (cagar alam dan suaka margasatwa), kawasan pelestarian alam (taman nasional, taman hutan raya, dan taman wisata alam), dan taman buru.
2. Hutan Lindung
3. Hutan Produksi

2.1.1 Jenis-jenis Hutan Di Indonesia

Indonesia adalah salah satu negara yang memiliki hutan yang luas di dunia. Luas hutan tersebut dulu mencapai 113 juta hektar dan terus berkurang drastis akibat kebodohan oknum pemerintah dan penjahat yang selalu haus uang dengan membabat dan menggunduli hutan demi mendapat keuntungan yang besar tanpa melihat dampak bagi lingkungan global. Brikut di bawah ini adalah pembagian macam-macam / jenis-jenis hutan yang ada di Negara Kesatuan Republik Indonesia disertai arti definisi dan pengertian : [12]

1. **Hutan Bakau**

   Hutan bakau adalah hutan yang tumbuh di pantai-pantai landai dan berlumpur yang terkena pasang surut. Hutan bakau sangat penting karena menjadi tempat bagi berbagai jenis ikan dan udang. Hutan bakau juga dapat melindungi daratan dari pengaruh abrasi dan dapat menjadi penampung banjir dari pedalaman daratan. Hutan bakau dapat ditemui di Pantai Papua, Sumatra bagian timur, dan sepanjang pesisir Kalimantan.

2. **Hutan Rawa**

   Hutan rawa adalah hutan yang meliputi daerah rawa-rawa dengan berbagai jenis tumbuhan seperti beluntas, pandan, dan ketapang. Jenis hutan ini banyak terdapat di pantai timur Sumatra, Kalimantan Barat, dan Kalimantan Tengah.

3. **Hutan Sabana**

   Hutan sabana adalah padang rumput yang diselingi pepohonan dan banyak terdapat semak belukar. Sabana umumnya dijumpai di Nusa Tenggara.

4. **Hutan Musim**

   Hutan musim adalah hutah yang memiliki perbedaan kondisi pada musim hujan dan kemarau yang cukup mencolok. Tumbuhan yang ada di hutan musim pada musim kewarau biasanya akan meranggas dan pada musim hujan akan tumbuh lebat kembali. Tumbuhan yang mengalami peristiwa ini di antaranya pohon jati dan pohon kapok. Hutan ini biasanya terdapat di daerah bertemperatur tinggi. Hutan musim banyak terdapat di Jawa Tengah, Jawa Timur, sampai Nusa Tenggara.

5. **Hutan Hujan Tropis**

   Hutan Hujan Tropis adalah jenis hutan yang paling terkenal di Indonesia yang negara tropis ini. Kepulauan Indonesia yang beriklim tropis banyak memperoleh sinar matahari, memiliki curah hujan yang tinggi, dan temperatur rata-rata tinggi. Oleh karena itulah tumbuhan dari berbagai macam jenis dapat tumbuh dengan mudah di daratan Indonesia dan membentuk hutan hujan tropis. Hutan hujan tropis memiliki keragaman hayati yang melimpah, dan menjadikannya paru-paru

dunia yang utama (dunia berutang pada Indonesia karena hal ini) sekaligus tempat hidup berbagai spesies hewan dan tumbuhan.

2.2 Bertahan Hidup (*Survival*)

Survival berasal dari kata *"SURVIVE"* yang artinya berjuang untuk hidup. Faktor keberhasilan dalam *survival* adalah faktor "Mental" karena hanya berkat mental yang tinggi kita bisa keluar dari segala kesulitan. Latihan *survival,* dapat juga dikatakan sebagai latihan "membajakan Mental Manusia" (membuat mental seseorang sekeras baja).

Secara umum, *"Survival"* adalah tindakan yang paling awal bagi mahluk hidup untuk mempertahankan hidupnya dari berbagai ancaman, dimana makluk hidup melakukan sesuatu tindakan untuk melindungi dirinya dari berbagai ancaman agar tetap hidup.

Keadaan tidak menentu (*Survival*) ini bisa terjadi pada setiap orang dan pendaki gunung atau siapa saja yang sedang melakukan perjalanan ataupun penjelajahan dialam bebas. Sehingga pengetahuan *Survival* merupakan bagian dari pengetahuan dasar tehnik hidup di alam bebas yang harus di pahami dan di kuasai oleh setiap pecinta alam dan pendaki gunung. Dengan memahami dan menguasai pengetahuan *Survival* paling tidak telah mempunyai suatu gambaran, tindakan apa yang harus dilakukan apabila pada suatu saat di hadapkan pada kondisi tidak menentu ini [9].

2.2.1 Pengertian *Survival* Hutan

Survival hutan adalah suatu tindakan mempertahankan hidup dari berbagai ancaman binatang buas dan kondisi alam untuk kelangsungan hidup dalam situasi dan kondisi terdesak. Binatang buas pada *survival* hutan ini seperti ular, kalajengking, babi hutan, serigala dan beruang.

2.2.2 *Survival* Individu

Survival Individu merupakan keadaan *Survival* seorang diri selain menghadapi masalah teknis juga akan menghadapi masalah kejiwaan yang kadang-kadang lebih berat dari masalah teknisnya sendiri. Kesepian dan bosan adalah suatu

keadaan yang tidak akan datang secara tiba-tiba, sangat lain dengan rasa takut dan panik. Kesepian dan bosan akan datang secara perlahan dan tidak terduga, biasanya akan timbul pada saat semua pekerjaan teknis *survival* telah dikerjakan dan kebutuhan untuk *survival* sudah didapatkan seperti air, makanan dan tempat berlindung.

2.2.3 *Survival* Kelompok

Berkelompok pada saat *survival* lebih baik dari pada *survival* perorangan, karena pada *survival* perorangan seluruh bahaya dan resiko akan di hadapi sendiri, namun ketika berkelompok maka akan tersedia banyak tangan untuk melakukan pekerjaan dengan adanya teman berkomunikasi. Teman berkomunikasi ini merupakan salah satu faktor penting agar dapat mengatasi rasa sepi dan tertekan oleh rasa kesendirian. Salah satu cara untuk menjaga kebersamaan tetap terkontrol pada saat *survival* kelompok yaitu seluruh anggota harus segera memilih seorang pemimpin dan setiap anggota harus selalu didalam kelompok.

2.2.4 *Survival* Kit

Survival kit adalah satu set peralatan atau satu kotak/tas peralatan *survival* yang umumnya dapat digunakan untuk semua jenis daerah seperti gunung, hutan, padang pasir, pantai serta laut. *Survival* kit biasanya digunakan para penerbang pesawat militer. Untuk seorang penjelajah ataupun petualang *survival* kit adalah salah satu perlengkapan dasar yang harus dimiliki. *Survival* kit dapat dibuat sendiri sesuai dengan tujuan perjalanan dan kebutuhan pribadi [9].

2.2.5 Faktor-faktor Penyebab *Survival*

Berikut ini merupakan faktor-faktor yang dapat menjadi penyebab terjadinya survival, yaitu diantaranya :

1. Kehabisan perlengkapan/perbekalan dari suatu perjalanan.
2. Kecelakaan dalam suatu perjalanan dengan pesawat udara atau kapal laut.
3. Tersesat di suatu daerah asing atau rawan.
4. Meloloskan diri di suatu daerah yang belum dikenal.

5. Lingkungan suatu daerah yang belum dikenal.
6. Hal-hal lain yang belum pasti (kekurangan pangan, kekurangan oksigen, dll).
7. Kondisi terjepit oleh ancaman binatang buas dan kondisi alam.
8. Tersesat di suatu pantai.
9. Keterbatasan perbekalan / perlengkapan.
10. Lingkungan yang belum dikenal.

2.2.6 Kebutuhan Seorang *Survivor*

Berhasil atau tidaknya seorang *survivor* keluar dari keadaan tidak menentu ini, semua tergantung pada diri kita sendiri. Awal dari keberhasilan kita adalah menanamkan atau menumbuhkan dari semangat "HARUS HIDUP". Tanpa semangat itu, kecil kemungkinan dapat keluar dari keadaan ini. Setelah mendapatkan semangat "HARUS HIDUP" maka kebutuhan yang harus dimiliki seorang *survivor* adalah : [14]

1. Sikap mental yang mendukung *survival* diantaranya ; semangat, percaya diri, akal sehat, disiplin dan rencana kegiatan yang matang, serta kemampuan belajar dari pengalaman.
2. Pengetahuan, terutama pengetahuan yang berhubungan dengan tehnik *survival* yaitu ; cara membuat tempat perlindungan (bivoak), pengetahuan cara memperoleh air dan makanan, membuat api, orientasi medan dan lain-lainPengalaman dan Latihan, *Survival* adalah seni dan perlu kreativitas untuk menjalaninya, semakin kreatif seseorang maka semakin besar peluang orang tersebut untuk tetap hidup bahkan bisa menolong nyawa orang lain. Oleh karena itu pengalaman dan latihan sangat menentukan keberhasilan.
3. Makanan adalah sumber energi bagi semua mahluk hidup entah di manapun kita berada termasuk saat kita berpetualang di hutan atau alam bebas. ada beberapa Jenis tumbuhan dan hewan yang bisa dimakanan bagi petualang seprti pramuka ketika berada di alam bebas " hutan, gunung" adalah bekal

pengetahuan penting yang harus di miliki oleh setiap individu pramuka atau *survival* yang sering berpetualang menyambangi hutan liar dan gunung.

4. Peralatan atau *Survival* Kit, biasakan SELALU membawa *survival* kit dalam setiap perjalanan. Karena dengan memiliki *survival* kit, satu set perlengkapan sudah dimiliki untuk keadaan darurat. Isi kotak *survival* kit diantaranya ; korek api kedap udara, lilin, kaca pembesar, cermin, jarum dan benang, kail dan senarnya, sol sepatu dan benangnya, kompas, senter kecil, dan obat-obatan. bahaya tersebut karena;
    a. Mengandung bisa / racun. Bukan berarti kita tidak memakan jenis hewan ini, akan tetapi perlu diperhatikan bahayanya bagi tubuh kita. Apabila unsur racun / bisa dalam tubuh binatang ini bisa kita hilangkan maka kita dapat mengkonsumsinya. Binatang yang berbahaya tersebut diantaranya adalah; Nyamuk malaria, semut api, tawon atau lebah, kelabang dan atau scorpio/kalajengking, pacet, harimau, buaya, ular, ikan lepu batu, ikan pari dan lain-lain.
    b. Menyebarkan bau yang khas / busuk. Binatang tertentu tidak dapat dimakan karena mempunyai kelenjar bau yang menyebar secara khas (busuk). Ini dimunkinkan karena bau busuk tersebut berfungsi sebagai senjata untuk melindungi dari predator. Contoh binatang ini adalah tikus busuk atau cecurut. Satwa sebagai Sumber makanan yaitu:
        1. Molusca, contohnya siput dan kerang.
        2. Annelida, contohnya cacing tanah dan sondari (Pheretima sp) dan lintah (Hirudinaria sp).
        3. Insecta, contohnya belalang (Palanga sp)
        4. Crustacea, contohnya udang dan kepiting
        5. Pisces, semua ikan dapat dimakan.
        6. Amphibia, contohnya katak Rana sp
        7. Reptilia, contohnya ular, kadal, cecak dan lain-lain.

8. Mamalia, contohnya kelinci, rusa, tikus dan lain-lain.
9. Aves, contohnya ayam hutan Gallus gallus

Hal-hal yang perlu diperhatikan dalam keadaan darurat yaitu jangan memakan tumbuhan yang tidak umum, sebaiknya memakan tumbuhan yang tidak mengandung banyak getah dan tidak kecut. Berikut ini cara-cara pemanfaatan tumbuhan yang dapat menjadi bahan makanan ketika terjadinya *survival*, yaitu: [14]

1. Tumbuhan yang dapat dimakan langsung, biasanya Tumbuhan yang dimanfaatkan pada bagian buah serta daun atau pucuk. Contoh : Rasamala, gelagah, sintrong, bunut, putat dan lain-lain.
2. Tumbuhan yang harus dimasak terlebih dahulu, Biasanya dimasak dengan cara direbus, dibakar atau digoreng. Contoh : Pucuk puring, umbut palem-paleman rotan, buah saninten, pasang, umbi talas dan lain- lain
3. Tumbuhan yang harus diolah lalu dimasak, Biasanya dilakukan pengolahan terlebih dahulu seperti perendaman selama berhari-hari atau diberi campuran bahan penetral sebelum dimasak, atau dalam memasaknya harus sering mengganti air, karena apabila dimakan langsung dapat mengakibatkan gatal atau memabukan. Contoh : Umbi Acung, Suweg, gadung dll.

Contoh tumbuhan yang dapat dimakan :

a. Umbi : talas, kentang, besusu, paku tanah
b. Batang : umbut muda pisang, sagu, rumput madu, rebung.
c. Buah : kelapa, arbei, *strawbery hutan*, nipah, dll
d. Biji : padi, jagung, rumput teki Madura
e. Bunga *: turi, pisang*
f. Daun *: rasamala, melinjo, tangkai dan daun begonia ,daun selada air (arnong)*
g. Tunas */ pucuk : cemara.*

2.3 Game

Video *game* merupakan salah satu media hiburan yang paling popular untuk semua kalangan usia. Sejak pertama kali ditemukan sampai saat ini, teknologi *game* telah mengalami kemajuan yang pesat. Hal ini ditandai dengan berkembangnya jenis, produk, alat, dan jenis interaksi gamedengan penggunaan yang semakin beragam bentuknya [6].

2.3.1 Pengertian Game

Game berasal dari kata bahasa inggris yang memiliki arti dasar Permainan. Permainan dalam hal ini merujuk pada pengertian "kelincahan intelektual" (*intellectual playability*).*Game* juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana *game* itu menarik untuk dimainkan secara maksimal. *Game* merupakan sebuah bentuk seni dimana penggunanya, disebut dengan pemain (*player*), diharuskan membuat keputusan-keputusan dengan tujuan untuk mengelola sumber daya yang diperoleh dari kesempatan-kesempatan bermain (token) miliknya, untuk mencapai suatu tujuan tertentu.

Video *game* adalah bentuk *game* yang interaksi utamanya melibatkan media video (dan biasanya melibatkan audio). Berdasarkan representasi visualnya, *game* dapat dibedakan menjadi 2 jenis, yaitu game 2 dimensi (2D) dan 3 dimensi (3D).*Game* 2D adalah game, yang secara matematis, hanya melibatkan 2 elemen koordinat kartesius, yaitu xdan y, sehingga konsep kamera pada game 2D hanya menentukan "gambar" mana pada game yang dapat dilihat oleh pemain. Sedangkan game 3D adalah game yang selain melibatkan elemen xdan y, juga melibatkan elemen zpada perhitungannya, sehingga konsep kamera pada game 3D benar-benar menyerupai konsep kamera pada kehidupan nyata, yaitu selain digeser (seperti pada game 2D), juga dapat diputar dengan sumbu tertentu.

Menariknya akan tantangan maupun visualisasi yang ada pada game sering mengakibatkan ketergantungan terhadap game tersebut (*addicted)*. Ini merupakan salah satu akibat buruk yang disebabkan oleh game (*game addicted)*[6].

2.3.2 Sejarah Game Komputer

Game komputer adalah sebuah permainan interaktif yang dijalankan menggunakan komputer. Program komputer menerima input dari si pemain melalui pengendali dan menampilkan lingkungan buatan melalui TV atau layar monitor. Sejarah teknologi *game* komputer secara langsung berhubungan dengan perkembangan komputer itu sendiri.

Game komputer generasi pertama, tahun 1952, adalah sebuah permainan OXO (tic-tac-toe) dalam versi grafik yang diciptakan oleh seorang mahasiswa Universitas Cambridge bernama A.S. Gouglas. *Game* ini diprogram dengan memakai komputer EDVAC vacuum tube yang memiliki layer berupa *cathode ray tube* (CRT). Pada tahun 1958. William Haginbotham menciptakan *video game* pertamanya, dan *game* komputer pertama yang benar-benar diciptakan menggunakan komputer itu sendiri adalah Spacewar oleh Steve Russel yang terinspirasi dari kisah fiksi ilmiah berjudul Skylark karangan Edward E Smith. Steve Russel bekerja menggunakan sebuah komputer mainframe bernama MIT PDP-1 yang biasa dipakai untuk perhitungan statistic pada tahun 1961.

Game generasi kedua, di tahun 1971, Nolan Bushnell bersama Ted Dabney menciptakan *game* bergenre *arcade* yang pertama yang diberi judul Komputer Space, yang didasari oleh Spacewar. Tahun 1972, Nolan dan Ted memulai Atari komputer. Mereka mengembangkan *game* berjudul Pong yaitu *game* pertama yang tersedia untuk publik. Karena sebelumnya *game* hanya ada didalam komputer mainframe yang digunakan untuk kesenangan sendiri. Asal usul Pong dimulai saat Nolan ingin membuat *game* sederhana dan mudah dimengerti. Menggunakan memory dan micro processor kelas rendah, dengan kemampuan grafis yang sederhana, akhirnya dia membuat versi elektronik dari permainan ping pong yang kemudian menjadi Pong.

Pong kemudian berevolusi menjadi sebuah *game* bernama Breakout. *Game* itu diciptakan oleh Steve Jobs untuk Atari dan bersama Steve Wozniak mulai berpikir untuk menciptakan sebuah PC. Mereka meminjam semua peralatan yang dipakai dalam proyek Breakout untuk membuat sebuah prototype bernama Apple I. dan itu adalah cikal bakal komputer Apple Macintosh yang ada sekarang.

Pada tahun 1980, Atari mengeluarkan *game* berjudul Asteroid dan Lunar Lander. Kedua *game* tersebut adalah *game* pertama yang didaftarkan kepada perusahaan untuk mendapatkan hak cipta yang paten. Asteroid merupakan *game* yang penuh inovasi baru dalam grafisnya, dengan menggunakan metode raster, *game* ini merupakan *grafis vector line* seperti yang ada pada *osciloscope*. Pengenalan Atari Video komputer *System* dengan CPU biasa dan slot untuk kasetnya, yang menjadi suatu era baru dalam dunia *game*. Di tahun 1980 itu juga menunjukkan adanya peningkatan penjualan PC yang biasa dipakai untuk *game*.

Kini komputer bersaing dengan mesin *console* seperti PS atau Xbox. Komputer tidak hanya sebagai mesin untuk kepentingan bisnis semata, tapi juga untuk hiburan seperti *game* [6].

2.3.3 Elemen Dasar *Game*

Elemen Dasar Game merupakan suatu *game* terdapat beberapa elemen yang menyusun *game* tersebut, seperti jenis *game*, karakter *game, background*, elemen sound/suara, dan gerakan-gerakan dasar.

2.3.4 Jenis "*Platform"/* Alat yang digunakan

Beberapa alat yang digunakan untuk menjalankan suatu *game*, diantaranya adalah :[6]

1. Arcade Games

   Game yang dijalankan pada mesin dengan input dan output audio visual yang telah terintegrasi dan tersedia ditempat-tempat umum, seperti mal, bandara dan sebagainya. Contoh permainan ding-dong.

2. PC Games

   Game yang dimainkan pada PC (Personal Computer) yang memiliki kelebihan yaitu memiliki tampilan antarmuka yang baik untuk I/O, output visual kualitas tinggi karena layar computer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televisi biasa.

3. Console Games

   Game yang dijalankan pada suatu mesin spesifik yang biasanya tersedia di rumah pribadi, seperti Playstation, Microsoft Xbox 360, Nintendo Wii, Sega dan sebagainya.

4. Mobile Games

   Game yang dapat dimainkan khusus untuk mobile phone atau PDA.

5. Handheld Games

   Game yang dimainkan di console khusus *video game* yang dapat dibawa kemana-mana, contoh *GameBoy*, Nintendo DS dan Sony PSP.

2.3.5 Jenis-jenis *Game* Komputer

Dengan perkembangan informasi seperti sekarang, *genre* dari *game* masih terus berkembang, sebagai contoh simulasi menari (*dance simulation*) yang diperkenalkan oleh desainer *game* dari Jepang. Namun secara umum *game* komputer dapat dibagi menjadi beberapa jenis berdasarkan *genre* yang diterapkannya, yaitu:

A. *Role Playing Game* (RPG)

Role Playing Game (RPG) adalah salah satu *game* yg mengandung unsur *experience* atau *leveling* dalam *gameplay*nya. Biasanya dalam *game* ini memiliki kebebasan untuk menjelajah dunia *game* tersebut, dan kadang kala dalam beberapa *game*, dapat menentukan ending dari *game* tersebut RPG terbagi 2 : *Action* RPG & *Turn Based* RPG (Menunggu giliran untuk berjalan seperti Monopoly).

B. *First Person Shooting* (FPS)

*First Person Shooting (*FPS*)* adalah *game* yang tembak menembak yang memiliki ciri utamanya adalah penggunaan sudut pandang orang pertama dibuat dibelakang senjata.

C. *Third Person Shooter* (TPS*)*

Third Person Shooter (TPS) adalah *game* yang mirip dengan FPS yaitu memiliki *gameplay* tembak menembak hanya saja sudut pandang yg digunakan dalam *game* ini adalah orang ketiga.

D. *Strategy*

Strategy adalah genre *game* yang memiliki *gameplay* untuk mengatur suatu unit atau pasukan untuk menyerang markas musuh dalam rangka memenangkan permainan. biasanya di dalam *game* Strategy, dituntut untuk mencari *gold* untuk membiayai pasukan. *Games Strategy* dibagi 2 :

1. *Real Time Strategy* (RTS).

   Pada *game RTS*, dapat dikendalikan pasukan secara langsung, dari mencari sumber daya, hingga menghancurkan musuh.Semua pertempuran ini dapat disaksikan secara langsung.Contoh : Warcraft 3

2. *Turn Based Strategy* (TBS).

   Sistemnya seperti Turn Based RPG, tetapi disini selain mengendalikan character utama, dikendalikannya pasukan dan kota untuk memenangkan pertarungan. Biasanya memainkan game nya di atas peta. Contoh : Civilization

E. *Simulation*

Simulation adalah genre yang mementingkan realisme. Segala faktor pada *game* ini sangat diperhatikan agar semirip didunia nyata. Segala nilai, material, referensi, dan faktor lainnya adalah berdasarkan dunia nyata. Cara memainkannya juga berbeda, karena biasanya kontrol yang dimiliki cukup rumit. Genre simulasi meliputi *game* racing, *flight*, sampai militer.

F. *Tycoon*

Tycoon adalah *game* yang menjadikannya sebagai seorang *businessman* yang akan mengembangkan sesuatu properti untuk dikembangkan hingga laku di pasaran.

G. *Racing*

Racing Game adalah *game* berjenis balapan yang memungkinkan untuk mengendalikan sebuah kendaraan untuk memenangkan sebuah balapan.

H. *Action Adventure*

Action Adventure adalah *game* berupa petualangan salah seorang karakter yang penuh dengan penuh aksi yg akan terus ada hingga *game* tersebut tamat. (Biasanya Action dimasukan kategori RPG)

I. *Arcade*

Arcade game adalah genre *game* yang tidak terfokus pada cerita, melainkan hanya dimainkan "*just for fun*" atau untuk kejar-mengejar point / *highscore*.

J. *Fighting Game*

Fighting adalah genre *game* bertarung. Seperti dalam *arcade*, pemain dapat mengeluarkan jurus-jurus ampuh dalam pertarungannya. Genre *fighting* biasanya *one on one* dalam sebuah arena yang sempit.

K. *Sports*

Sport Adalah genre *game* bertema permainan olahraga. Sistem permainan akan berbeda-beda tergantung jenis olahraga yang menjadi tema *game* tersebut [6].

2.4 Game Edukasi

Game edukasi adalah *game* yang khusus dirancang untuk mengajarkan *user* terhadap suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya [7].

2.4.1 Kriteria *Game Edukasi*

Menurut *Hurd* dan *Jenuings*, perancangan yang baik haruslah memenuhi kriteria dari *education game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education* game, yaitu:

1. Nilai Keseluruhan (*Overall Value*)

   Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur timer.

2. Dapat Digunakan *(Usability)*

   Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Aplikasi ini merancang sistem dengan *interface* yang *user friendly* sehingga *user* dengan mudah dapat mengakses aplikasi.

3. Keakuratan *(Accuracy)*

   Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian *(Appropriateness)*

   Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan *user* dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan *(Relevance)*

   Relevan artinya dapat mengaplikasikan isi *game* ke target *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak-anak maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Objektifitas *(Objectives)*

   Objektifitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikasi ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik *(Feedback)*

   Untuk membantu pemahaman *user* bahwa permainan (performance) mereka sesuai dengan objek *game* atau tidak, *feedback* harus disediakan. Aplikasi ini menyajikan *animasi* dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan [7].

2.4.2 Poin-Poin pada *Game* Edukasi

Hal-hal yang harus ada pada *game edukasi* ini adalah sebagai berikut:

1. Produktif

   Game edukasi harus dapat mengembangkan sikap produktif dari para pemain *game.* Jadi inti dari *game* tersebut akan tersimpan di memori sebagai pemain sehingga suatu saat bisa menginovasi dan merangsang kreativitas dalam menciptakan sesuatu yang baru.

2. Aktif

   Game edukasi perlu dapat mengembangkan sikap aktif sehingga secara tidak langsung mengembangkan syaraf-syaraf motorik kasar dan motorik halus pada pemain.

3. Efektif dan Efisien

   Sebuah *game edukasi* tentunya perlu punya dampak dalam kehidupan para pemainnya, misal pada *Brain Challenge*, lama kelamaan pemain dapat menjadi makin cepat berpikir, atau makin analitif. Selain itu, dari sisi waktu, pemain yang ingin segera belajar tidak harus menunggu kelas dimulai, bisa dimainkan kapan saja dan dimana saja sehingga *game edukasi* bisa dikatakan efisien.

4. Kreatif

   Melalui *game edukasi*, setidaknya pemain bisa memikirkan beberapa cara lain untuk menyelesaikan masalah, dan tidak harus selalu menggunakan satu macam

cara, sehingga merangsang pemain untuk kreatif dan tidak hanya menunggu orang lain menyelesaikannya dan menirunya.

5. Menyenangkan

   Game edukasi tetap adalah *game*, jangan sampai kehilangan unsur menyenangkan dalam *game* tersebut karena padatnya isi materi dan monotonnya *gameplay*. Jadi *game* sebisa mungkin bisa membuat pemain makin pandai perlahan tapi pasti tanpa meninggalkan kesenangan bermain. Jadi sebaiknya *game edukasi* tanpa disadari bermanfaat untuk mengembangkan *IQ*, *EQ* dan *SQ* [7].

2.4.3 Manfaat *Game* Edukasi

Ada beberapa manfaat yang terdapat dalam game edukasi ini, Salah satu contoh manfaat dari *Game Edukasi* ini yaitu:

1. **Mampu melatih konsentrasi**

   Semakin tinggi usia pemain maka semakin terbatas daya tangkapnya. Oleh karena itu permainan dan pengajaran yang menggunakan alat dan media yang baik dan tepat akan membantu mempertahankan daya tangkapnya

2. **Mengajarkan sesuatu dengan lebih cepat dengan waktu relatif singkat**

   Bila pelajaran hanya disampaikan dengan kata-kata saja, mungkin bisa tersampaikan atau salah paham. Namun dengan bantuan alat dan media yang baik seperti *game edukasi*, guru bisa menjelaskan dalam waktu cepat dan mencapai indikator kebnerhasilan belajar lebih cepat.

3. **Menambah daya paham dan ingatan**

   Dalam menjelaskan sesuatu jika menggunakan media yang tepat tentu akan lebih mudah dimengerti dan memperdalam pengalaman belajar serta ingatan anak. Melalui indera penglihatan dan pendengaran anak yang berusia dini dapat memahami perbedaan arti, warna, serta bentuk melalui *game edukasi*. Ini salah satu kegunaan *game edukasi* bagi para pemain yang berusia muda.

4. **Membuat proses belajar menyenangkan**

   Cara mengajar yang monoton tentu akan membosankan. Tetapi bila disampaikan dalam bentuk yang berbeda, media yang berbeda tentu akan menyenangkan dan mampu membangkitkan motivasi belajar.

5. **Membangkitkan emosi**

   Menyampaikan suatu materi dengan media-media yang menarik tentunya akan lebih berhasil daripada menggunakan ceramah saja. Dengan media yang menarik tentu akan membangkitkan emosi , perhatian pada materi dan juga pada media tersebut. Misal ada *game edukasi* yang memiliki kisah dan cerita, tentu akan lebih mudah menyerap inti yang ingin diajarkan.

6. **Mampu Mengatasi Keterbatasan Bahasa**

   Perbedaan kebudayaan sering menimbulkan kesalah pahaman, namun dengan media mampu mengatasi kesalahpahaman akan keterbatasan untuk mengerti suatu bahasa. Beberapa *game edukasi* juga diarahkan untuk mempelajari bahasa.

7. **Meningkatkan kemampuan berkomunikasi**

   Dengan permainan edukatif pasti merangsang para pemain untuk berhubungan dan berkomunikasi dengan orang lain. Paling tidak dengan permainan *edukasi* akan menimbulkan banyak pertanyaan dan imajinasi yang tentunya akan ditanyakan pada guru, orang tua atau teman sesama pemain [7].

2.5 Algoritma

Algoritma adalah urutan langkah-langkah untuk memecahkan suatu masalah. Berikut ini adalah beberapa contoh dan penjelasan dari algoritma.

2.5.1 Algoritma Fuzzy Logic

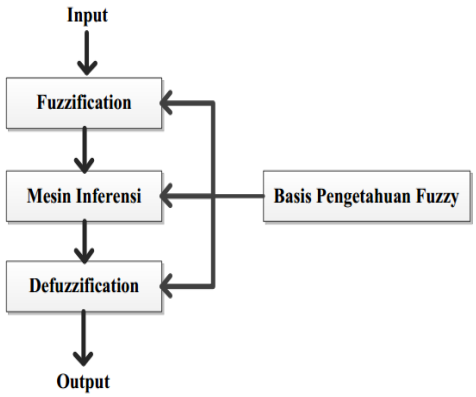
Konsep tentang *fuzzy* diperkenalkan oleh *Prof. Lotfi Astor Zadeh* pada 1962. Logika *fuzzy* adalah metodologi sistem control pemecahan masalah, yang cocok diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system,* jaringan PC, *multi-channel* atau *workstation* berbasis akurasi data dan sistem control. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat

biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, dan lain-lain.Oleh karena itu semua ini mempunyai nilai keanggotaan 0 dan 1. Akan tetapi, dalam logika *fuzzy* memungkinkan nilai keanggotaan berada diantara 0 dan 1.

Konsep himpunan *fuzzy* memiliki 2 atribut, yaitu: [13]

1. Linguistik, yaitu nama suatu kelompok yang me wakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya dingin, sejuk, panas mewakili variable temperatur.
2. Numeris, yaitu suatu nilai yang menunjukan ukuran dari suatu variabel, misalnya 10, 35, 40 dan sebagainya.

   Gambar 2.1 menggambarkan cara kerja logika *fuzzy* sebagai struktur elemen dasar sistem *inferensi fuzzy*.



Gambar 2. 1 Struktur Sistem *Interferensi Fuzzy*

Keterangan :

1. Basis Pengetahuan *Fuzzy* merupakan kumpulan rule-rule *fuzzy* dalam bentuk pernyataan *IF…THEN*.
2. Fuzzyfikasi adalah proses untuk mengubah input sistem yang mempunyai nilai tegas menjadi variabel linguistic menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.

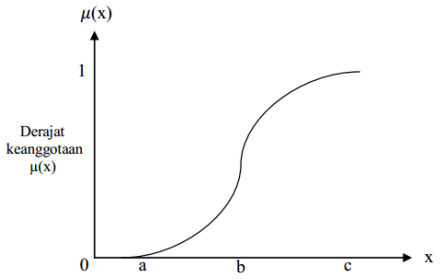
3. Mesin Inferensi merupakan proses untuk mengubah input *fuzzy* dengan cara mengikuti aturan-aturan (*IF -THEN Rules*) yang telah ditetapkan pada basis pengetahuan *fuzzy*.
4. *DeFuzzyfikasi* merupakan proses mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan *fuzzyfikasi*.

2.5.1.1 Fungsi Keanggotaan (Member function)

Pada fuzzy system, fungsi keanggotaan memainkan peran yang sangat penting untuk merepresentasikan masalah dan menghasilkan keputusan yang akurat. Terdapat banyak sekali fungsi keanggotaan yang dapat digunakan, diantaranya adalah: [10]

1. Fungsi *sigmoid*

   Sesuai dengan namanya, fungsi ini berbentuk kurva *sigmoidala* seperti huruf S. Setiap nilai x (anggota *crips set*) dipetakan ke dalam interval [0,1]. Grafik dan notasi matematika untuk fungsi ini adalah sebagai berikut :



Gambar 2. 2 Grafik Fungsi Sigmoid

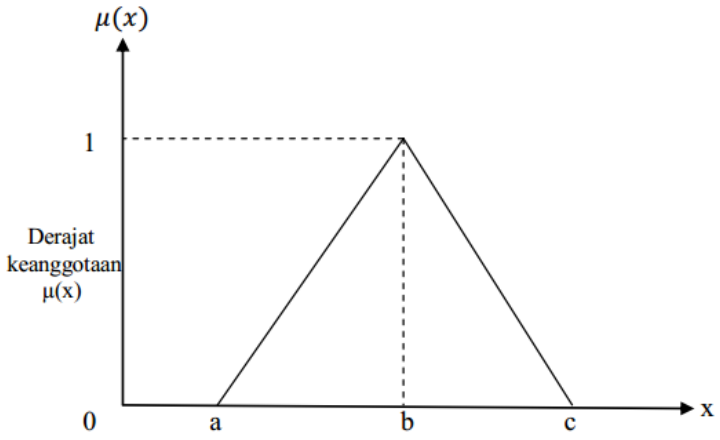
Fungsi keanggotaan :

$$Sigmoid(x,a,b,c) = \begin{cases} 0; x \le a \\ 2(x-a)/(c-a)^2; a \le x \le b \\ 1-2((c-x)/(c-a))^2; b \le x \le c \\ 1; x \ge c \end{cases}$$

…… 2.2

2. Fungsi Segitiga

Fungsi segitiga hanya terdapat satu nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika x = b. Tetapi, nilai-nilai disekitar b memiliki derajat keanggotaan yang turun cukup tajam (menjauhi 1) . Grafik dan notasi matematika dari fungsi segitiga adalah seperti pada gambar 2.3 :



Gambar 2. 3 Grafik Fungsi Segitiga

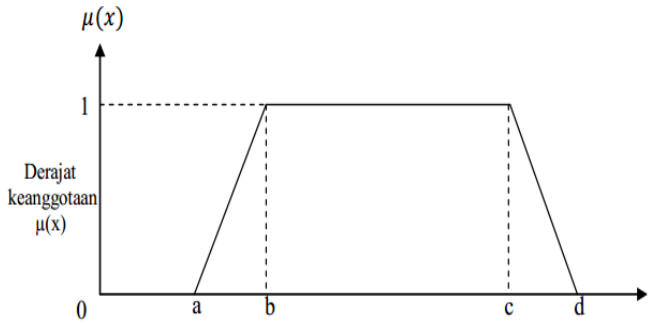
Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0; x \leq a \; atau \; x \geq c \\ (x-a)/(b-a); a \leq x \leq b \\ (b-x)/(c-b); b \leq x \leq c \end{cases}$$

…… 2.3

3. Fungsi Trapesium

Berbeda dengan fungsi segitiga, pada fungsi ini terdapat beberapa nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika $b \leq x \leq c$. Tetapi, derajat keanggotaan untuk $c \leq x \leq d$ memiliki karakteristik yang sama dengan fungsi segitiga. Grafik dan notasi matematika dari fungsi Trapesium adalah seperti pada gambar 2.4 :



Gambar 2. 4 Grafik Fungsi Trapesium

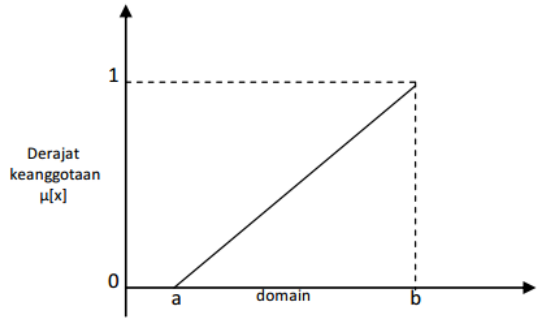
Fungsi Keanggotaan :

$$\mu(x) = \begin{cases} 0; & x \le a \text{ atau } x \ge d \\ (x-a)/(b-a); & a \le x \le b \\ 1; & \\ (d-x)/(d-c); & b \le x \le c \end{cases}$$

……2.4

4. Fungsi Linear

Pada fungsi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk kurva ini paling sederhana dan menjadi pilihanyang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaanhimpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi. Keadaan tersebut digambarkan seperti pada gambar 2.5.



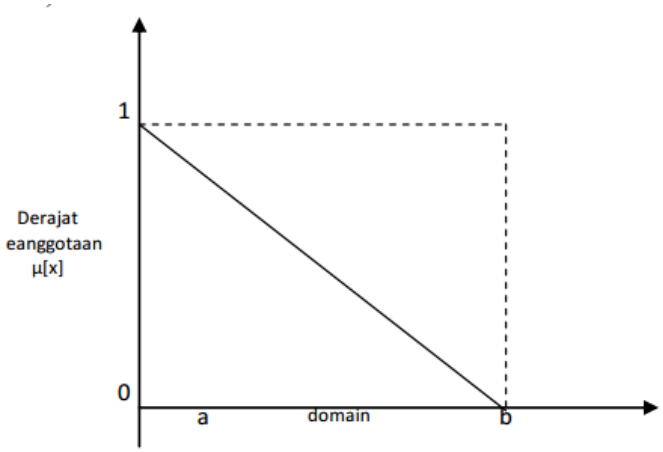
Gambar 2.5 Representasi Linear Naik[10]

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x < a \\ (x-a)/(b-a); & a < x \le b \\ 1; & x \ge b \end{cases}$$

……..... 2.5

Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah. Keadaan tersebut digambarkan seperti pada gambar 2.6.



Gambar 2.6 Representasi Linear Turun[10]

Fungsi keanggotaan :

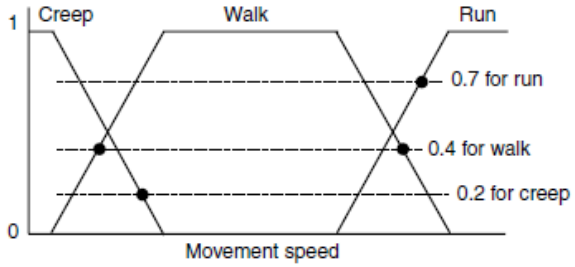
$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \le x < b \\ 0; & x \ge b \end{cases}$$

…….. 2.6

2.5.1.2 Defuzzifikasi

Setelah menerapkan apa logika *fuzzy* yang dibutuhkan, biarkan dengan seperangkat keanggotaan nilai untuk *fuzzy* set. Untuk mengubahnya kembali menjadi data yang berguna, perlu menggunakan sebuah teknik *defuzzifikasi*. Teknik *fuzzifikasi*

melihat di bagian terakhir adalah cukup jelas dan hampir di mana-mana..Ada beberapa teknik *defuzzifikasi* mungkin, dan tidak ada konsensus yang jelas mengenai mana yang terbaik untuk digunakan. Semua memiliki struktur dasar yang sama, tetapi berbeda dalam efisiensi dan stabilitas hasil. *Defuzzifikasi* melibatkan mengubah seperangkat nilai keanggotaan menjadi *output* tunggal nilai. Nilai *output* hampir selalu nomor. Hal ini bergantung pada memiliki satu set fungsi keanggotaan untuk nilai.Kami *output* mencoba untuk membalikkan fuzzifikasi yang Metode: untuk menemukan nilai *output* yang akan mengakibatkan nilai-nilai keanggotaan yang dimiliki. Pada Gambar selanjutnya, kita memiliki nilai keanggotaan 0,2, 0,4, dan 0,7 untuk *fuzzy* set "merayap", “berjalan," dan "lari." Fungsi keanggotaan menunjukkan bahwa tidak ada nilai yang mungkin untuk gerakan kecepatan yang akan memberikan nilai-nilai keanggotaan, jika memberi ke dalam fuzzifikasi [10].



Gambar 2.7 Defuzzyfikasi [10]

2.6 Game Engine

Game engine adalah perangkat lunak yang dirancang untuk membuat dan mengembangkan video *game*. Fungsi utama yang diberikan oleh *game engine* meliputi rendering untuk 2D atau 3D *graphic*, *collision detection*, *sound*, *scripting*, *animasi*, *artificial intelligence*, *networking*, *memory management*, *threading* dan *scene graph. Game engines* memberikan perangkat untuk visual *development* dengan tambahan komponen perangkat lunak yang dapat dipakai berulang kali. Perangkat ini pada umumnya memberikan *integrated development environment* yang dapat mempermudah, serta mempercepat pengembangan *game*. *Game engine* juga sering disebut juga “*Game Middleware*”. *Game Middleware* adalah *software* yang

didalamnya sudah terdapat elemen-elemen yang dibutuhkan oleh seorang *game* developer, kebanyakan dari *game middleware* memberikan fasilitas guna menunjang pengembangan *game* dengan mudah seperti *graphics, sound dan Artificial intelligence* [8].

Berikut beberapa *Game engine* yang beredar sekarang ini :

1. OGRE
2. Panda 3D
3. crystal space
4. blender Game engine
5. OPEN SceneGraph
6. C4 Engine
7. Game Maker
8. torque Game engine
9. 3DGamestudio
10. DXStudio
11. neoExis engine
12. esenthel engine

2.6.1 CopperCube

CopperCube 3D merupakan Software Editor untuk membuat adegan Gambar 3D yang interaktif yang hasilnya dapat dibuat dalam Format Flash swf atau WebGL dan juga dapat dibuat menjadi standalone Windows exe. CopperCube mencakup fitur baru, yaitu diantaranya sebagai berikut:

- Dukungan Android
- memungkinkan penciptaan aplikasi Android
- Partikel Sistem, cara untuk membuat api, asap, hujan, air mancur, kabut, dan banyak lagi
- Dukungan layar sentuh untuk perangkat mobile

- Dukungan layar loading Customizable
- Peningkatan render teks, termasuk multi-line text
- Tindakan baru dan ditingkatkan dan perilaku, untuk memungkinkan untuk membuat game yang lebih kompleks, seperti untuk perilaku misalnya Kendaraan.
- Kemungkinan untuk mengotomatisasi editor melalui baris perintah dan skrip
- Prefabs Baru dan contoh aplikasi
- Kinerja perbaikan untuk semua target penerbitan

Coppercube dapat mengimpor 23 format file 3D yang berbeda dan mampu membuat aplikasi 3D interaktif dengan fisika, pencahayaan, permainan AI dan banyak lagi. Tidak ada pengetahuan pemrograman diperlukan untuk menciptakan aplikasi dengan CopperCube. Editor mencakup sistem Perilaku / Tindakan sederhana dengan yang memungkinkan 'scripting' proses yang kompleks. CopperCube 5 sekarang meluas sistem ini dengan lebih Tindakan, sehingga bahkan lebih kompleks aplikasi dapat dibuat dengan mudah: Kemampuan untuk secara dinamis mengkloning benda 3D dan menghapusnya lagi, serta perilaku kendaraan properti pendukung untuk nama hanya beberapa. Berikut ini format-format file yang mendukung coppercube dapat dilihat pada tebel 2.1 :

Tabel 2.1 Format-format File Coppercube.

Format	Import	Export
AutoCAD exchange files (.dxf)	yes	-
3D Studio ASE files (.ase)	yes	-
3D Studio meshes (.3ds)	yes	-
Alias Wavefront Maya (.obj)	yes	yes
B3D files (.b3d)	yes	-
Cartography shop 4 (.csm)	yes	-
COLLADA (.xml, .dae)	yes	yes
DeleD (.dmf)	yes	-
FSRad oct (.oct)	yes	-
Irrlicht static meshes (.irrmesh)	yes	yes
Irrlicht scenes (.irr)	-	yes
Lightwave 3D meshes (.lwo)	yes	-
Microsoft DirectX (.x)	yes	-
Milkshape (.ms3d)	yes	-
My3DTools 3 (.my3D)	yes	-

OGRE meshes (.mesh)	yes	-
Pulsar LMTools (.lmts)	yes	-
Quake 3 levels (.bsp)	yes	-
Quake 2 models (.md2)	yes	-
STL meshes (.stl)	yes	Yes
Stanford Polygon Lib files (.ply)	yes	-
Truespace files (.cob and .scn)	yes	-

Dengan satu klik, adalah mungkin untuk membuat Flash swf, Windows exe, html WebGL, Mac OS X App atau sekarang bahkan file APK Android dengan CopperCube. Dengan Sistem Partikel baru, aplikasi ini sekarang dapat menjadi lebih interaktif dan tampil lebih hidup. Seperti segala sesuatu dalam editor, sistem partikel dapat benar-benar disesuaikan dengan menggunakan user interface yang sangat sederhana. Semua properti seperti jumlah partikel, arah, kecepatan, gravitasi, daerah emitor, warna, dan sejenisnya, dapat dengan mudah disesuaikan. [8]

2.7 *Object Oriented Programming* (OOP)

Object Oriented Programming (OOP) adalah teknik memodelkan suatu sistem dunia nyata dalam perangkat lunak berdasarkan objek. Objek tersebut adalah konsep inti. Suatu objek adalah sebuah perangkat lunak entitas atau konsep model dunia nyata.

Ketika sebuah program berjalan, objek individu biasanya tidak dapat berdiri sendiri. Mereka termasuk ke sebuah koleksi dari objek lain yang serupa yang merupakan anggota dari grup yang sama, atau *class*. Sebuah program akan terbentuk dari banyak *class* yang berbeda, setiap *class* terbentuk dari objek-objek yang serupa. Beberapa sistem perangkat lunak orientasi objek akan memiliki sifat berikut : [3]

1. Abtraksi dengan objek

   Abtraksi adalah mekanisme yang memungkinkan kompleks, situasi dunia nyata dapat diwakili menggunakan model yang disederhanakan. Orientasi objek abstrak dunia nyata didasarkan pada objek dan interkasi antar objek lainnya.

2. Enkapsulasi *class*

   Enkapsulasi adalah proses menyembunyikan semua bagian rinci sebuah objek dari dunia luar.

3. Interaksi lewat pesan

   Interaksi lewat pesan digunakan untuk memenuhi suatu perintah, objek butuh berinteraksi dengan objek lain. Interaksi dapat antara objek di *class* yang sama, atau objek di *class* lain. Interaksi ini ditangani dengan cara mengirimkan pesan ke objek lain untuk melewati informasi atau meminta aksi.

4. Masa hidup objek

   Masa hidup objek adalah semua objek yang memiliki masa hidup. Objek – objek tersebut dibuat dan diinisialisasi sebagaimana objek – objek tersebut dibutuhkan pada saat program dijalankan, hidup dan membawa keluar fungsi mereka, dan akhirnya dihancurkan. Sementara mereka ada, mereka mempertahankan identitas mereka sendiri dan kondisi. Banyak objek yaitu contoh dari *class* yang sama bisa hidup pada waktu tertentu. Setiap objek memiliki atribut yang berbeda dari yang lain yaitu contoh objek dari *class* yang sama.

5. Hirarki *class*

   Hirarki *class* dalam desain orientasi objek, *class* objek tersebut diatur ke dalam hirarki yang memodelkan dan menggambarkan hubungan antar *class*.Hubungan yang paling sederhana adalah sebuah asosiasi.

6. Polimorfisme

   Polimorfisme adalah karakteristik sistem orientasi objek. Ketika warisan digunakan untuk memperpanjang *class* umum ke *class* yang lebih khusus, biasanya akan mencakup memperluas beberapa perilaku dari *class* umum. *Class* khusus sering akan menerapkan perilaku yang agak berbeda dengan *class* umum, tetapi nama yang digunakan mendefinisikan perilaku akan menjadi sama. Adalah penting bahwa contoh yang diberikan dari sebuah objek menggunakan perilaku yang benar, dan *property* polimorfisme memungkinkan ini terjadi secara otomatis

dan mulus. Polimorfisme sebenarnya lebih mudah digunakan daripada dijelaskan [3].

2.8 Unified Modeling Language (UML)

UML adalah penerus gelombang berorientasi objek alnalisis dan desain (OOA & D) metode yang muncul di akhir ‘80-an dan ’90-an awal. Hal yang paling langsung menyatukan metode *Booch*, *Rumbaugh* (*OMT*), dan *Jacobson*, tapi jangkauan lebih luas dari itu. *UML* pergi melalui proses standarisasi dengan *OMG* (*Obyek Manajement Group*) dan sekarang menjadi standar *OMG*.

UML disebut bahasa permodelan, bukan metode. Sebagian besar metode terdiri, setidaknya pada prinsipnya, kedua model sebuah bahasa dan proses. Bahasa pemodelan adalah notasi (terutama grafis) bahwa metode digunakan untuk mengekspresikan desain. Proses ini menyarankan mereka apa langkah yang harus diambil dalam melakukan desain.

Bagian proses dalam buku banyak metode yang agak samar. Selain itu, kebanyakan orang, ketika mereka mengatakan bahwa mereka menggunakan metode, menggunakan bahasa pemodelan, tapi jarang mengikuti proses. Jadi, dalam banyak hal pemodelan bahasa adalah bagian paling penting dari meode ini. Hal ini tentu bagian penting untuk komunikasi. Jika anda ingin mendiskusikan desain anda dengan seseorang, itu adalah bahasa pemodelan yang anda berdua perlu pahami, bukan proses yang digunakan untuk sampai ke desain tersebut [3].

2.8.1 Use Case Diagram

Use-case diagram adalah gambaran *graphical* dari beberapa atau semua aktor, *use-case,* dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use-case* diagram menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

Use-case diagram dapat digunakan selama proses analisis untuk menangkap *requirement* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use-case* diagram berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use-case* diagram. Kebutuhan atau *requirements system* adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model *use-case* yang menggambarkan fungsi sistem yang diharapkan (*use-case*), dan yang mengelilinginya (actor), serta hubungan antara actor dengan *use-case* (*use-case diagram*) itu sendiri [3].

2.8.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business *modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Sangat bermanfaat apabila membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity* diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram* [3].

2.8.3 *Sequence Diagram*

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message*. *Sequence* Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk menghasilkan output tertentu. *Sequence* Diagram diawali dari apa yang me-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan *use case* diagram. *Sequence* diagram juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*. *Sequence* diagram juga dapat merubah atribut atau *method* pada *class* yang telah dibentuk oleh *class diagram*, bahkan menciptakan sebuah *class* baru. *Sequence* diagram memodelkan aliran logika dalam sebuah system dalam cara yang visual. *Sequence* diagram biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi *method* didalam sebuah system. *Sequence* diagram biasanya dipakai untuk memodelkan :[3]

1. Deskripsi tentang *system* yang ada pada sebuah / beberapa *use case* pada *use case* diagram, yang menggambarkan hubungan antara *actor* dan *use case* diagram.
2. Logika dari *method* (*operation*, *function* atau *procedure*).
3. Logika dari *service* (*high level method*).

2.8.4 Class Diagram

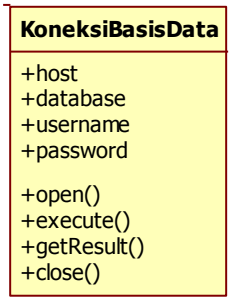
Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.

2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Kelas-Kelas yang ada pada struktur *system* harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan *system* sehingga pembuat perangkat lunak atau programmer dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Berikut adalah contoh dari diagram kelas.

Class Diagram dalam mendefinisikan metode yang ada didalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi didalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antar metode yang satu dengan metode yang lain didalam sebuah kelas.



Gambar 2.8 Contoh *class diagram* [3]

2.9 Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean[5].

2.9.1 Pengujian *Black Box*

Pengujian *black box* berfokus pada persyaratan *fungsional* perangkat lunak. Dengan demikian, pengujian *black box* menungkinkan perekayasa perangkat lunak

mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* bukan merupakan alternatif dari teknik *white box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white box*. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

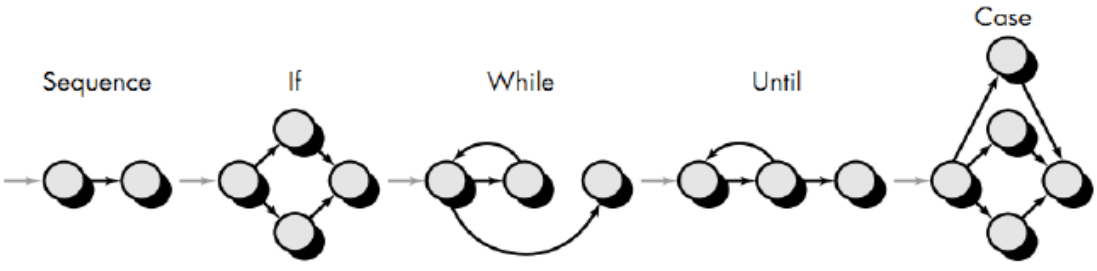
1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan dalam *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

2.9.2 Pengujian *White Box*

Pengujian *white box*, yang kadang-kadang disebut pengujian *glass box*, adalah metode desain *test case* yang menggunakan struktur kontrol desain *procedural* untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white box*, perekayasa sistem dapat melakukan *test case* sebagai berikut :

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali
2. Menggunakan semua keputusan logis pada sisi *true* dan *false*
3. Mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka
4. Menggunakan struktur data internal untuk menjamin validitasnya.

Pengujian *white box* yang berupa notasi diagram alir dapat dilihat pada gambar 2.9.



Gambar 2.9 Notasi Diagram Alir