BAB 2
LANDASAN TEORI

2.1 Awal Terjadinya Perang Medan Area

Tanggal 27 Agustus 1945 rakyat Medan baru mendengar berita proklamasi yang dibawa oleh Mr. Teuku Moh Hassan sebagai Gubernur Sumatera. Menanggapi berita proklamasi para pemuda dibawah pimpinan Achmad Tahir membentuk barisan Pemuda Indonesia.

2.1.1 Latar Belakang Pertempuran Medan Area

Pada tanggal 9 november 1945, pasukan Sekutu dibawah pimpinan Brigadir Jenderal T.E.D. Kelly mendarat di Sumatera Utara yang dikuti oleh pasukan NICA. Brigadir ini menyatakan kepada pemerintah RI akan melaksanakan tugas kemanusiaan, mengevakuasi tawanan dari beberapa tempat di luar Kota Medan. Dengah dalih menjaga keamanan, para bekas tawanan diaktifkan kembali dan dipersenjatai. Adapun latar belakang pertempuran Medan Area, antara lain:

1. Bekas tawanan yang menjadi arogan dan sewenang-wenang.
2. Ulah seorang penghuni hotel (pasukan NICA) yang merampas dan menginjak-injak lencana merah putih.
3. Ultimatum agar pemuda Medan menyerahkan senjata kepada Sekutu.
4. Pemberian batas daerah Medan secara sepihak oleh Sekutu dengan memasang papan pembatas yang bertuliskan “Fixed Boundaries Medan Area (Batas Resmi Medan Area)” di sudut-sudut pinggiran Kota Medan.

2.1.2 Proses Terjadinya Pertempuran Medan Area

Karena sulitnya komunikasi, proklamasi kemerdekaan baru diumumkan secara resmi di Medan pada tanggal 27 Agustus 1945 oleh Mr. Teuku Muhammad Hasan selaku Gubernur Sumatera. Pada tanggal 9 Oktober 1945, pasukan AFNEI dibawah pimpinan Brigjen T.E.D. Kelly mendarat di Belawan. Kedatangan

pasukan AFNEI ini diboncengi oleh pasukan NICA yang dipersiapkan untuk mengambil alih pemerintahan[4].

Kedatangan pasukan AFNEI disambut baik oleh pemerintah RI karena pemerintah RI menghormati tugas AFNEI di Indonesia. Namun dibalik itu, sehari setelah AFNEI mendarat di Belawan, pasukan AFNEI mendatangi tempat-tempat tawanan untuk membebaskan tawanan perang yang kebanyakan orang Belanda. Tawanan yang dibebaskan itu, kemudian dipersenjatai dan dibentuk menjadi Batalyon KNIL di Medan.

Operasi-operasi militer Inggris semakin intensif dilaksanakan dan kantor gubernur terpaksa dipindahkan ke kantor walikota. Markas Divisi II TKR dipindahkan ke Pematang Siantar. Demikian pula laskar-laskar pemuda memindahkan markasnya masing-masing ke luar Kota Medan untuk mengadakan konsolidasi. Pasukan laskar masih bertempur tanpa adanya kesatuan komando, maupun koordinasi. Lambat laun mereka menyadari kelemahan ini setelah beberapa kali menderita kerugian.

Atas perakasa dewan pertahanan daerah, maka diundang para komandan laskar untuk berunding di Tebing Tinggi selama 2 hari pada tanggal 8-10 Agustus 1946 untuk membahas masalah perjuangan. Akhirnya mereka sepakat membentuk Komando Resimen Laskar Rakyat Medan Area (KRLMA). Konsekuensinya dari pembentukan komando ini, laskar-laskar dibebaskan dari organisasi induknya masing-masing. Kapten Nip Karim dipilih sebagai Komandan dan Marzuki Lubis sebagai Kepala Staf. Markas Komando berada di Two Rivers. KRLMA terdiri dari 5 batalyon dan 1 kompi istimewa dengan pembagian wilayah dan tanggung jawab yang pasti.

Dengan adanya prakarsa pimpinan Divisi Gajah dan KRLRMA pada 10 Oktober 1941 disetujui untuk mengadakan serangan bersama. Sasaran yang akan direbut di Medan Timur adalah Kampung Sukarame, Sungai Kerah. Di Medan barat ialah Padang Bulan, Petisah, Jalan Pringgan, sedangkan di Medan selatan adalah kota Matsum yang akan jadi sasarannya. Rencana gerakan ditentukan, pasukan akan bergerak sepanjang jalan Medan-Belawan.

Hari "H" ditentukan tanggal 27 Oktober 1946 pada jam 20.00 WIB, sasaran pertama Medan Timur dan Medan Selatan. Tepat pada hari "H", batalyon A resimen laskar rakyat di bawah Bahar bergerak menduduki Pasar Tiga bagian Kampung Sukarame, sedangkan batalyon B menuju ke Kota Matsum dan menduduki Jalan Mahkamah dan Jalan Utama. Di Medan Barat batalyon 2 resimen laskar rakyat dan pasukan Ilyas Malik bergerak menduduki Jalan Pringgan, kuburan China dan Jalan Binjei.

Gerakan-gerakan batalyon-batalyon resimen Laskar Rakyat Medan Area rupanya tercium oleh pihak Inggris dan Belanda. Daerah Medan Selatan dihujani dengan tembakan mortir. Pasukan kita membalas tembakan dan berhasil menghentikannya.

Sementara itu Inggris menyerang seluruh Medan Selatan. Pertempuran jarak dekat berkobar di dalam kota. Pada keesokan harinya Kota Matsum bagian timur diserang kembali. Pasukan Inggris yang berada di Jalan Ismailiah berhasil dipukul mundur. Pada 3 November 1946, gencatan senjata diadakan dalam rangka penarikan pasukan Inggris dan pada gencatan senjata itu dilakukan, perundingan untuk menentukan garis kekuasaan. Pendudukan Inggris secara resmi diserahkan kepada Belanda pada tanggal 15 November 1946. Tiga hari setelah Inggris meninggalkan Kota Medan, Belanda mulai melanggar gencatan senjata.

Di Pulau Brayan pada tanggal 21 November, Belanda merampas harta benda penduduk dan pada hari berikutnya Belanda membuat persoalan lagi dengan menembaki pos-pos pasukan laskar di Stasiun Mabar, juga Padang Bulan ditembaki. Pihak laskar membalas. Kolonel Schalten ditembak ketika lewat di depan pos Laskar. Belanda membalas dengan serangan besar-besaran di pelosok kota. Angkatan Udara Belanda melakukan pengeboman, sementara itu di front Medan Selatan di Jalan Mahkamah mendapat tekanan berat, tapi di Sukarame gerakan pasukan Belanda dapat dihentikan.

Pada tanggal 1 Desember 1946, pasukan lascar rakyat medan mulai menembakkan mortir ke sasaran pangkalan Udara Polonia dan Sungai Mati. Keesokan harinya Belanda menyerang kembali daerah belakang kota. Kampung

Besar, Mabar, Deli Tua, Pancur Bata dan Padang Bulan ditembaki dan dibom. Tentu tujuannya adalah memotong bantuan logistik bagi pasukan yang berada di kota. Tapi walaupun demikian, moral pasukan kita makin tinggi berkat kemenangan yang dicapai. Karena merasa terdesak, Belanda meminta kepada pimpinan RI agar tembak-menembak dihentikan dengan dalih untuk memastikan garis demarkasi yang membatasi wilayah kekuasaan masing-masing. Dengan adanya demarkasi baru, pasukan-pasukan yang berhasil merebut tempat-tempat di dalam kota, terpaksa ditarik mundur. Sewaktu kita akan mengadakan konsolidasi di Two Rivers, Tanjung Morawa, Binjai dan Tembung, mereka diserang oleh Belanda. Pertempuran berjalan sepanjang malam. Serangan Belanda pada tanggal 30 Desember 1946 ini benar-benar melumpuhkan kekuatan laskar medan area. Daerah kedudukan laskar satu demi satu jatuh ke tangan Belanda. Dalam serangan Belanda berhasil menguasai Sungai Sikambing, sehingga dapat menerobos ke segala arah.

Perkembangan perjuangan di Medan menarik perhatian Panglima Komandemen Sumatera. Dia menilai bahwa perjuangan yang dilakukan oleh Resimen Laskar Rakyat Medan Area ialah karena kebijakan sendiri. Komandemen memutuskan membentuk komando baru, yang dipimpin oleh Letkol Sucipto. Serah terima komando dilakukan pada tanggal 24 Januari 1947 di Tanjung Morawa. Sejak itu pasukan-pasukan TRI memasuki Front Medan Area, termasuk bantuan dari Aceh yang bergabung dalam Resimen Istimewa Medan Area. Dalam waktu 3 minggu Komando Medan Area (KMA) mengadakan konsolidasi, disusun rencana serangan baru terhadap Kota Medan. Kekuatannya sekitar 5 batalyon dengan pembagian sasaran yang tepat. Hari "H" ditentukan 15 Februari 1947 pukul 06.00 WIB. Sayang karena kesalahan komunikasi serangan ini tidak dilakukan secara serentak, tapi walaupun demikian serangan umum ini berhasil membuat Belanda kalang kabut sepanjang malam. Karena tidak memiliki senjata berat, jalannya pertempuran tidak berubah. Menjelang Subuh, pasukan kita mundur ke Mariendal. Serangan umum 15 Februari 1947 ini adalah serangan besar terakhir yang dilancarkan oleh pejuang-pejuang di Medan Area. Sampai

menjelang Agresi Militer ke I Belanda, yang mana pasukan RI di Medan Area berjumlah 7 batalyon dan tetap pada kedudukan semula yang membagi Front Medan Area atas beberapa sektor, adalah Medan Timur, Medan Selatan, Medan Barat dan Medan Utara. Begitu juga membagi Medan atas 4 sektor yang sama, dan dengan demikian mereka langsung berhadapan dengan pasukan kita.

Pada saat terjadi Agresi Militer Belanda ke I, Belanda melancarkan serangannya terhadap pasukan RI ke semua sektor. Perlawanan terhadap Belanda hampir 1 minggu dan setelah itu pasukan-pasukan RI mengundurkan diri dari Medan Area.

2.1.3 Senjata Peninggalan Perang Medan Area

Ada beberapa senjata yang berhasil di simpan dalam museum TNI Perjuangan di Medan yang merupakan bekas peninggalan perang Medan Area, senjata ini di pakai untuk melawan klonial penjajah, senjata di peroleh dengan merampas dari pasukan penjajah dan pemberontak, baik itu dari pihak Belanda maupun Jepang.

Berikut ini beberapa hasil dari gambar dari senjata-senjata yang di gunakan di dalam perang Medan Area :



Gambar 2.1 Senjata yang digunakan dalam perang Medan Area



Gambar 2.2 Senjata pistol yang digunakan saat perang Medan Area



Gambar 2.3 Informasi senjata



Gambar 2.4 Keterangan senjata



Gambar 2.5 Museum Perjuangan TNI

2.2 *Game*

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

2.2.1 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. Pada awalnya, *game* identik dengan permainan anak-anak. Kita selalu berpikir game merupakan suatu kegiatan yang dilakukan oleh anak-anak yang dapat menyenangkan hati mereka.

Teori permainan (*game*) pertama kali ditemukan oleh sekelompok ahli matematika pada tahun 1994. Teori ini di kembangkan oleh john Von Neumann dan Oskar Morgenstern yg berisi "permainan terdiri atas sekumpulan peraturan-peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih starategi yang dibangun untuk memaksimalkan kemenangan sendiri ataupun meminimalkan kemenangan lawan. Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan di terima setiap pemain sebagai kemajuan bermain.

Menurut Agustinus Nilwan dalam bukunya "Pemrograman Animasi dan *Game* Profesional" terbitan Elex Media Komputindo, *game* di artikan sebagai suatu aktivitas tersetruktur atau juga digunakan sebagai alat pembelajaran. Sebuah *game* bisa dikarakteristikan dari apa pemain lakukan misalnya :

Peralatan Misal : bola, kartu, papan, atau sebuah Komputer.

1. Peraturan

   Peraturan digunakan untuk menentukan giliran pemain, hak dan keharusan masing-masing pemain, dan tujuan permainan.
2. *Skill*, Strategi dan Keberuntungan.

   Game dengan *skill*, contohnya dengan kekuatan fisik, misal gulat, menembak dan kekuatan mental seperti catur.

3. *Single Player Game* (pemain satu orang) dan *Double Player* (lebih dari satu pemain).

Jika pemain tunggal, pemain harus bermain dengan keahlian, berpacu dengan waktu dan keberuntungan sedangkan pemain *double*, pemain diharuskan untuk menggunakan suatu strategi dan kekompakan sesama pemain, untuk mencapai tujuan tertentu atau sebaliknya pemain harus berlomba dengan pemain lainnnya untuk mencapai tujuan[5].

2.2.2 Sejarah *Game*

Game pertama kali dibuat pada tahun 1966 oleh Ralph Baer bersama dengan timnya yang berjumlah 500 orang yang terdiri dari para insinyur dan teknisi. Setelah sebulan bekerja keras, Baer berhasil menampilkan dua titik putih yang berkejar-kejaran dilayar. Pada tahun 1972 muncul nama baru dalam dunia *game*. Nolan Bushel mendirikan perusahaan Atari dan membuat *game Arcade Pong.* Pada era baru dalam perkembagan dunia *game* terjadi pada tahun 1988 yang didominasi oleh perusahaan Jepang. Nintendo yang awalnya hanya memproduksi mesin *fotocopy*, dengan *video game* sistemnya telah mencapai omset miliaran dolar. Setalah masa awal perkembagan *game* ini, dunia *game* telah melalui beberapa *fase* yaitu *fase game* PC, fase *game console* dan fase *game online*. Sekarang ketiga fase *game* tersebut sudah semakin banyak variannya dan masing-masing memiliki kelompok penggemarnya sendiri [4].

2.2.3 Sejarah *Game* Pertama di Indonesia

Divinekids adalah *game* yang diakui dan mendapat *award* resmi dari MURI (Museum Rekor Indonesia) sebagai *game* pertama Indonesia pada tahun 2004 [6]. Akan tetapi sebenarnya sudah banyak *game-game* buatan orang-orang Indonesia yang lain tetapi karena kekurangan syarat-syarat minimal seperti karakter yang menggunakan karakter asing, penyebaran yang kurang luas, bahasa yang digunakan bahasa asing dan lain sebagainya. Oleh karena itu tidak terpilih sebagai *game* pertama Indonesia secara resmi.

Menurut sebagian besar pengamat *game* di Indonesia, ini adalah kurangnya investigasi MURI dalam melihat adanya *game-game* tersebut sebelum memberikan gelar *game* pertama Indonesia. Pengamat yang berpendapat berbeda dengan MURI memposisikan bahwa sejarah *game* pertama Indonesia adalah *Game* Devid. Pada tahun 2006, *game-game* Divinekids berjumlah 38 lebih dan terus bertambah yang pembuatnya adalah David Setiabudi. Pada tahun 1992, David Setiabudi juga membuat *game* namun tidak diakui sebagai *game* pertama Indonesia karena kurang memenuhi syarat. Ada juga *game-game* buatan Korea yang terkenal di terjemahkan ke bahasa Indonesia dan *game-game* yang penyebarannya kurang luas serta berkarakter asing atau berbahasa asing secara resmi tidak terpilih menjadi *game* pertama Indonesia [5].

2.2.4 *Game* Edukasi

Menurut Randel (1991), *game* sangat berpotensi untuk menumbuhkan kembali motivasi belajar anak yang mengalami penurunan. Berdasarkan penelitian yang dilakukan oleh Randel pada tahun 1991 tercatat bahwa pemakain *game* sangat bermanfaat pada materi-materi yang berhubungan dengan matematika, fisika dan kemampuan berbahasa (seperti studi sosial, biologi dan logika)[6].

2.2.5 Jenis-jenis Game

Berikut ini beberapa jenis *game* berdasarkan cara pembuatannya, cara pemasarannya dan mesin yang menjalankannya [7]. Jenis-jenis *game* tersebut adalah :

1. ***Game* PC**

   Game PC adalah *game* yang dimainkan pada PC (*Personal Computer*) yang memiliki kelebihan yaitu tampilan antarmuka yang baik untuk *input* maupun *output*. *Output* visual berkualitas tinggi karena layar komputer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televisi biasa. Kekurangannya adalah spesifikasi komputer yang sangat bervariasi

antar satu komputer dengan komputer yang lainnya menyebabkan beberapa *game* dapat ditampilkan dengan baik pada satu komputer tetapi tidak berjalan dengan baik pada komputer yang lainnya.

2. ***Game Console***

   Game console adalah *game* yang dijalankan pada suatu mesin spesifik yang biasanya tersedia di rumah seperti Xbox, Nintendo Wii dan lain-lain.

3. ***Game Arcade***

   Game arcade adalah *game* yang dijalankan pada mesin dengan *input* dan *output* audio visual yang telah terintegrasi dan tersedia ditempat-tempat umum.

4. ***Game Online***

   Game online adalah *game* yang hanya dapat dumainkan secara *online* melaui LAN atau internet.

2.2.6 *Genre Game*

Berdasarkan *genre*, *game* dapat dibagi menjadi beberapa *genre* yaitu [7] :

A. *Action Game*

Action game dikategorikan sebagai *gameplay* dengan model pertarungan.Berikut beberapa macam *game* yang termasuk dalam *genre action game* yaitu :

a. *Action Adventure Game*

Genre game yang berfokus pada eksplorasi dan biasanya mempunyai unsur *item gathering*, penyelesaiaan *puzzle simple* dan pertarungan. Contoh *game* dari *genre* ini adalah The Legend Of Zelda series dan Metroid series.

b. *Stealth Game*

Termasuk dalam *genre* terbaru, biasanya digolongkan dalam mata-mata yang biasa melakukan aksinya secara rahasia. Contoh *game* dari *genre* ini adalah Metal Gear series.

c. *Survival Horror Game*

Genre game yang berusaha membuat pemain menjadi tegang dan takut dengan elemen-elemen horror. Contoh *game* dari *genre* ini adalah Resident Evil series dan Alone in The Dark.

d. *Beat'em Up Game*

Genre game combat dimana satu orang melawan banyak musuh yang telah disediakan. Contoh *game* dari *genre* ini adalah Dynasty Warrior series dan Final Fight.

e. *Fighting Game*

Game pertarungan dua pemain dengan jurus-jurus yang biasa dikeluarkan dengan menekan beberapa tombol pada *keyboard* dengan urutan tertentu.Contoh *game* dari *genre* ini adalah Street Fighter dan Tekken series.

f. *Maze Game*

Genre game yang membutuhkan kecepatan berpikir dan bereaksi serta berunsur ketepatan menavigasi. Contoh *game* dari *genre* ini adalah Pac-Man.

g. *Platfrom Game*

Genre game dengan *game play* berlari,melompat,mengayun dan sebagainya. Contoh *game* dari *genre* ini adalah Donkey Kong dan Ray Man.

h. *Shooter*

Genre game dengan tembak menembak mempunyai banyak macam jenis, antara lain :

1. *First Person Shooter Game*

Genre game yang mengutamakan *shooting* dan *combat* dari perspektif langsung mata karakter yang bertujuan untuk memberikan pemain perasaan berada ditempat itu dan bisa fokus menembak.

2. *Massively Multiplayer Online First Person Shooter Game*

Genre game yang mengkombinasikan *game play first person shooter* dengan dunia virtual dimana banyak *player* juga ikut bermain melaui internet. Contoh *game* dari *genre* ini adalah Counter Strike *Online*.

3. *Third Person Shooter Game*

Genre game yang sama seperti *first person shooter game* yaitu mengutamakan *shooting* dan *combat* dari perspektif karakter yang bertujuan untuk memberikan pemandangan yang lebih luas dan gerakan yang lebih banyak.

4. *Tactical Shooter Game*

Genre yang mengutamakan perencanaan dan kerja sama tim untuk memenangkan *game*. Contoh *game* dari *genre* ini adalah Tom Clancy's Ghost Recon series.

5. *Light Gun Game*

Genre dengan lebih banyak pada *arcade* dengan peralatan tertentu seperti senjata mainan yang mempunyai sensor khusus terhadap layar.Contoh *game* dari *genre* ini adalah Time Crisis dan Duck Hunt.

6. *Shoot'em Up Game*

Genre dengan ciri khas gambar 2D dan *scrolling playing area*.Contoh *game* dari *genre* ini adalah Star Fox series.

B. *Adventure Game*

Adventure game di kategorikan sebagai *gameplay* yang mengharuskan pemain memecahkan bermacam-macam teka-teki melalui interaksi dengan orang lingkungan dalam *game* tersebut.

a. *Text Adventure*

Pemain akan menggunakan *keyboard* untuk mengetikkan berupa perintah dan komputer akan menganalisa perintah tersebut lalu menjalankan karakter sesuai perintah tersebut.

b. *Graphical Adventure Game*

Genre yang merupakan perkembangan dari *text adventure*. Pemain dapat menggunakan *mouse* untuk menggerakkan karakter.

c. *Visual Novel Game*

Genre yang memberikan keleluasaan untuk memilih jalan ceritanya sendiri.

d. *Interactive Movie Game*

Genre game dengan rangkaian *live action* dari karakter yang dimainkan pemain. Contoh *game* dari *genre* ini adalah *Space Ace*.

e. *Dialog Game*

Pada *genre* ini, pemain akan mengalami kemajuan tergantung pada apa yang mereka katakan.Contoh *game* dari *genre* ini adalah *Law And Order* : *The Vangeful Heart*.

C. *Role Playing Game*

Role playing game adalah *game* yang memiliki *gameplay* dimana karakter milik *player* akan berpetualang dengan *skill combat* dalam cerita *game*.

a. *Action Role Playing Game*

Genre game yang memasukkan unsur *action game* dan *action adventure game*.Contoh *game* dari *genre* ini adalah Diablo 1 & 2.

b. *Massively Multyplayer Online Role Playing Game*

Konsep dari *genre* ini terkombinasi dengan *genre-genre* lainnya yang berupa fantasi. Contoh *game* dari *genre* ini adalah Rising Force *Online*.

c. *Tactical Role Playing Game*

Dalam *genre* ini,pemain akan diberikan giliran masing-masing untuk menentukan langkah-langkah yang akan dilakukan oleh karakter.Contoh *game* dari *genre* ini adalah Final Fantasy Tactics.

D. *Simulation Game*

Genre ini bertujuan untuk memberikan pengalaman simulasi kepada pemain.

a. Construction and Management Simulation Game

Genre ini merupakan bagian dari *economic simulation game*. Contoh *game* dari *genre* ini adalah Sims City series.

b. *Economic Simulation Game*

Genre ini berupa simulasi keadaan ekonomi dimana pemain mengontrol keadaan ekonomi dari *game* tersebut. Contoh dari *genre* ini adalah Monopoly Tycoon.

c. *God Game*

Dalam *genre* ini tidak ada tujuan akhir yang membuat pemain memenangakan *game*. Contoh *game* dari *genre* ini adalah The Sims series.

d. *Government Simulation Game*

Genre game yang memasukkan unsur kepolisian,pemerintahan atau politik sebuah negara.

E. *Strategy Game*

Genre strategy game berfokus pada *game play* dimana dibutuhkan pemikiran yang tepat agar dapat meraih kemenangan.

a. *Real Time* Strategi

Dalam *real time* strategi,*action* dilakukan dalam waktu yang bersamaan oleh masing-masing pihak dimana *action* dimainkan per *ronde* atau bergiliran.Contoh *game* dari *genre* ini adalah Warcraft series.

b. *Tactical Game*

Dalam *genre* ini pemain harus menggunakan bermacam-macam taktik dan strategi untuk mencapai kemenangan. Contoh *game* dari *genre* ini adalah Dark Omen.

c. 4X *Game*

Genre ini berarti penjelajahan,menjajah dan memusnahkan. Contoh *game* dari *genre* ini adalah Galactic Civizations.

d. *Artillery Game*

Genre game ini biasanya mengikutkan *combat* dengan tank atau tentara militer. Contoh *game* dari *genre* ini adalah Tanarus.

F. *Vehicle Simulation Game*

Genre ini merupakan simulasi yang memberikan pemain sebuah pengalaman realistik dalam mengendarai kendaraan tertentu.

a. *Flight Game*

Dalam *genre* ini,pemain tidak hanya bersimulasi mengontrol pesawat terbang tetapi juga bisa *combat* di udara. Contoh *game* dari *genre* ini adalah Falcon 4.0.

b. *Racing Game*

Genre yang menempatkan pemain sebagai *driver* dengan kendaraan seperti mobil. Contoh *game* dari *genre* ini adalah Need For Speed series.

c. *Space Game*

Genre ini bersifat pertarungan di angkasa luar. Contoh *game* dari genre ini adalah Star Wars dan Homeworld.

d. *Train Game*

Genre ini mensimulasikan yang berhubungan dengan trasnportasi kereta. Contoh dari *genre* ini adalah Rail Simulator.

2.2.7 Game First Person Shooter (FPS)

Tembak - menembak orang pertama (*first-person shooter atau* FPS) adalah genre permainan video yang ciri utamanya adalah pengguna sudut pandang orang pertama dengan tampilan layar yang mensimulasikan apa yang dilihat melalui mata karakter yang dimainkan. Ciri utama lain adalah penggunaan senjata genggam jarak jauh. FPS modern muncul ketika komputer pribadi sudah mampu menggambar grafik 3D secara waktu nyata. *Wolfenstein 3D* dan *Doom* buatan id Software secara luas dianggap sebagai pelopor permainan video genre ini. Contoh populer lain genre ini antara lain *Duke Nukem 3D*, *Quake*, *Blood*, *Unreal*, *Unreal Tournament*, seri *Half-Life*, *Counter-Strike*, seri *Halo*, *Perfect Dark*, *TimeSplitters*, *Call of Duty*, *System Shock*, dan *GoldenEye 007* termasuk berapa tembak-menembak orang pertama yang di campuran dengan genre lainnya seperti seri operation flashpoint (campuran FPS dengan tatical), seri Battle field (campuran FPS dengan RPG), Rage (Video Game campuran FPS dengan sandbox) dan lain-lain.

2.2.8 Unsur Game

Dalam sebuah *game* terdapat unsur-unsur yang melengkapinya. Berikut beberapa unsur dalam suatu *game* yaitu [7]:

a. Warna

Warna mempunyai kemampuan untuk membuat orang tanggap terhadap semua yang dilihat karena tidak ada sesuatu hal yang bermakna tanpa warna. Warna terlihat sebelum penampakan gambar. Mata manusia tertarik oleh warna pada suatu *level* karena warna dari objek diterima sebelum detail-detail dipisahkan oleh bentuk-bentuk dan garisnya. Warna merah memiliki panjang gelombang yang terpanjang, biru memiliki panjang gelombang yang pendek sedangkan hijau memiliki panjang gelombang menengah. Pada anak-anak cenderung tertarik pada warna-warna yang cerah dan mencolok. Warna-warna yang cerah terutama warna *primer* (merah, kuning, biru) dan warna *sekunder* (orange, ungu, hijau).



Gambar 2.6 Warna *Primer* dan *Sekunder* [7]

b. Komposisi

Komposisi adalah pengaturan segala elemen didalam sebuah karya desain yang sedemikian rupa dengan tujuan tertentu.Komposisi yang baik adalah komposisi yang mampu memenuhi kebutuhan dan tujuan desain, mudah dipahami dan membentuk kesatuan yang serasi dan harmonis. Kemudian *layout* yaitu perencanaan, penempatan semua unsur mulai dari tulisan, gambar, ilustrasi, teks, nama dan sebagainya dengan pengukuran secara seksama. Komposisi yang sesuai

dengan anak-anak adalah komposisi yang sederhana dan tidak menggunakan petunjuk terlalu rumit. Kesederhanaan diwujudkan dengan penggunaan visual 2D dan penerapan warna-warna dalam seluruh aspek desain.

c. Bentuk dasar

Bentuk dasar adalah bentuk-bentuk yang mudah ditemui dalam kehidupan sehari-hari seperti kotak, lingkaran, segitiga dan lain sebagainya.



Gambar 2.7 Bentuk Dasar [7]

d. Tipografi

Tipografi merupakan representasi visual dari sebuah bentuk komunikasi *verbal* dan merupakan *property visual* yang pokok dan efektif. Huruf memainkan peranan penting dalam keberhasilan suatu bentuk komunikasi grafis. Dalam media pembelajaran untuk anak-anak, sebuah huruf harus *legible* yaitu jelas dan memiliki tingkat kemudahan untuk dibaca.

b. Audio

Audio adalah sinyal elektrik yang digunakan untuk membawa suara dalam batas pendengaran manusia. Audio merupakan komponen sistem yang sudah termasuk didalamnya atau dapat ditambahkan pada komputer.

2.2.9 Elemen dasar game

Menurut Teresa Dillon (futurelab.com, 2005) elemen-elemen dasar sebuah game adalah ;

a. *Game rules*

Merupakan aturan perintah, cara menjalankan, fungsi objek dan karakter di dunia permainan dunia game. Dunia game bisa berupa pulau, dunia khayalan,

dan tempat – tempat lain sejenis yang di pakai sebagai setting tempat dalam permainan game.

b. *Plot*

Plot biasanya berisi informasi tentang hal-hal yang akan dilakukan oleh player dalam *game* dan secara detail, perintah tentang hal yang harus di capai dalam game.

c. *Thema*

Didalam *game* biasanya ada pesan moral yang akan di sampaikan.

d. *Character*

Pemain sebagai karakter utama maupun karakter yang lain yang memiliki ciri dan sifat tertentu.

e. *Object*

Merupakan sebuah hal yang penting dan biasanya digunakan pemain untuk memecahkan masalah, adakalanya pemain harus punya keahlian dan pengetahuan untuk bisa memainkannya.

f. *Text,grafik*, dan *sound*

Game biasanya merupakan kombinasi dari media teks grafik maupun suara, walaupun tidak harus ada dalam permainan game.

g. *Animation*

Animasi ini selalu melekat pada dunia game, khususnya untuk gerakan karakter-karakter yang ada dalam game, property dari objek.

h. *User interface*

Merupakan fitur-fitur yang mengkonsumsi user dengan game.

2.3 Kecerdasan Buatan

Kecerdasan buatan atau *Artificial Intelligence* (AI) merupakan cabang dari ilmu komputer yang berhubungan dengan pengautomatisan tingkah laku cerdas. Kecerdasan buatan didasarkan pada teori suara (*sound theoretical*) dan prinsip-prinsip aplikasi dari bidangnya [8]. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritma yang diperlukan untuk

mengaplikasikan pengetahuan tersebut serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya.

Berdasarkan sudut pandang, AI dapat dipandang sebagai berikut :

a) Sudut pandang kecerdasan, AI adalah bagaimana membuat mesin yang cerdas dan dapat melakukan hal-hal yang sebelumnya hanya dapat dilakukan manusia.
b) Sudut pandang bisnis, AI adalah sekelompok alat bantu yang berdayaguna dan metodologi yang menggunakan alat-alat bantu tersebut untuk menyelesaikan masalah-masalah bisnis.
c) Sudut pandang pemrograman, AI meliputi studi tentang pemrograman simbolik, pemecahan masalah dan proses pencarian.
d) Sudut pandang penelitian.

Riset tentang AI dimulai pada awal tahun 1960-an, percobaan pertama adalah membuat program permainan catur, membuktikan teori dan *general problem solving*.

2.3.1 Perbandingan Kecerdasan Buatan dengan Kecerdasan Alamiah

Ada beberapa keuntungan dari kecerdasan buatan dibandingkan dengan kecerdasan alamiah yaitu [8] :

1. Kecerdasan buatan lebih permanen dibandingkan kecerdasan alami karena kecerdasan alami bisa berubah karena sifat manusia pelupa. Kecerdasan buatan tidak berubah selama sistem komputer & program tidak mengubahnya.
2. Memberikan kemudahan dalam duplikasi dan penyebaran.
3. Relatif lebih murah dari kecerdasan alamiah.
4. Konsisten dan teliti.
5. Dapat di dokumentasikan.
6. Dapat mengerjakan beberapa *task* dengan lebih cepat dan lebih baik dibanding manusia.

2.4 Teknik Dasar Pencarian

Pencarian atau pelacakan merupakan salah satu teknik untuk menyelesaikan permasalahan AI. Keberhasilan suatu sistem salah satunya ditentukan oleh kesuksesan dalam pencarian dan pencocokan. Teknik dasar pencarian memberikan suatu kunci bagi banyak sejarah penyelesaian yang penting dalam bidang AI.

Pencarian adalah proses mencari solusi dari suatu permasalahan melalui sekumpulan kemungkinan ruang keadaan *(state space)*. Ruang keadaan merupakan suatu ruang yang berisi semua keadaan yang mungkin. Kondisi suatu pencarian meliputi :

1) Keadaan sekarang atau awal.
2) Keadaan tujuan-solusi yang dijangkau dan perlu diperiksa apakah telah mencapai sasaran.
3) Biaya atau nilai yang diperoleh dari solusi.

Solusi merupakan suatu lintasan dari keadaan awal sampai keadaan tujuan. Secara umum, proses pencarian dapat dilakukan seperti berikut :

1) Memeriksa keadaan sekarang atau awal.
2) Mengeksekusi aksi yang dibolehkan untuk memindahkan ke keadaan berikutnya.
3) Memeriksa jika keadaan baru merupakan solusinya. Jika tidak, keadaan baru tersebut menjadi keadaan sekarang dan proses ini diulangi sampai solusi ditemukan atau ruang keadaan habis terpakai.

2.4.1 Masalah Pencarian

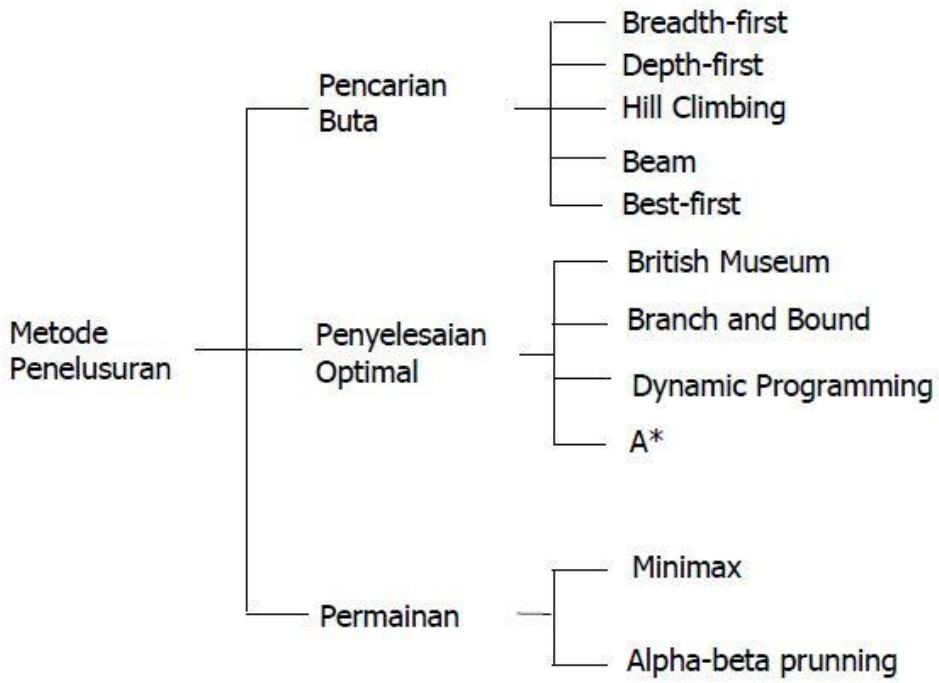
Masalah pencarian merupakan proses pencarian solusi yang direncanakan, yang mencari lintasan dari keadaan sekarang sampai keadaan tujuan. Suatu masalah pencarian direpresentasikan sebagai *graf* berarah. Keadaan di representasikan sebagai simpul *(node)*, sedangkan langkah yang dibolehkan atau

aksi di representasikan dengan busur *(arc).* Dengan demikian, secara khusus masalah pencarian didefinisikan sebagai :

1) *State space* (ruang keadaan).
2) *Start node* (permukaan simpul).
3) Kondisi tujuan dan uji untuk mengecek apakah kondisi tujuan ditemukan atau tidak.
4) Kaidah yang memberikan bagaimana mengubah keadaan.

2.4.2 Algoritma Pencarian

Permasalahan pencarian adalah merupakan yang sering dijumpai oleh peneliti di bidang kecerdasan buatan. Permasalahan ini merupakan hal penting dalam menentukan keberhasilan sistem kecerdasan buatan, dalam metode pencarian yang pertama adalah metode yang sederhana yang hanya berusaha mencari kemungkinan penyelesaian. Metode yang termasuk pada bagian ini adalah deft-first search, hill climbing, breadth-first search, beam search dan best first search. Yang kedua adalah mempelajari metode yang lebih kompleks yang akan mencari jarak terpendek. Metode ini adalah British Museum Procedure, Branch and Bound, Dynamic programming dan A*. metode ini digunakan pada saat harga perjalanan untuk mencari kemungkinan menjadi perhitungan. Yang ketiga, adalah metode yang diterapkan ketika berhadapan dengan musuh. Prosedur ini adalah minimax search, alpha-beta pruning. Dapat terlihat di bawah ini pembagian dari algoritma pencarian :



Gambar 2.8 Bagan Algoritma pencarian

Metode pencarian dikatakan penting untuk menyelesaikan permasalahan karena setiap *state* (keadaan) menggambarkan langkah-langkah untuk menyelesaikan permasalahan.

Metode pencarian dikatakan penting untuk perencanaan karena di dalam sebuah permainan akan menentukan apa yang harus di lakukan, dimana setiap state menggambarkan kemungkinan posisi pada suatu saat.

Metode pencarian adalah bagian dari kesimpulan, dimana setiap state menggambarkan hipotesis dalam sebuah rangkaian deduktif.

2.5 AI yang Digunakan

Artificial Intelligence atau yang bahasa Indonesia merupakan kecerdasan buatan Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Adapun AI yang digunakan dalam pembuatan game FPS Sejarah Medan Area adalah algoritma A* dan Fuzzy logic, yang akan di jelaskan dibawah ini.

2.5.1 Algoritma A* *(*A Star)

Algoritma A* menyelesaikan masalah yang menggunakan graf untuk perluasan ruang statusnya. Dengan menerapkan suatu heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan. Algoritma A* membangkitkan simpul yang paling mendekati solusi. Simpul ini kemudian disimpan suksesornya ke dalam list sesuai dengan urutan yang paling mendekati solusi terbaik. Kemudian, simpul pertama pada list diambil, dibangkitkan suksesornya dan kemudian suksesor ini disimpan ke dalam list sesuai dengan urutan yang terbaik untuk solusi.

Algoritma ini akan mengunjungi secara mendalam (mirip DFS) selama simpul tersebut merupakan simpul yang terbaik. Jika simpul yang sedang dikunjungi ternyata tidak mengarah kepada solusi yang diinginkan, maka akan melakukan runut balik ke arah simpul akar untuk mencari simpul anak lainnya yang lebih menjanjikan dari pada simpul yang terakhir dikunjungi. Bila tidak ada juga, maka akan terus mengulang mencari ke arah simpul akar sampai ditemukan simpul yang lebih baik untuk dibangkitkan suksesornya. Strategi ini berkebalikan dengan algoritma DFS yang mencari sampai kedalaman yang terdalam sampai tidak ada lagi suksesor yang bisa dibangkitkan sebelum melakukan runut balik, dan BFS yang tidak akan melakukan pencarian secara mendalam sebelum pencarian secara melebar selesai. A* baru berhenti ketika mendapatkan solusi yang dianggap solusi terbaik.

Algoritma A* menggabungkan jarak estimasi/heuristik [h(n)] dan jarak sesungguhnya/cost [g(n)] dalam membantu penyelesaian persoalan. Heuristik adalah nilai yang memberi harga pada tiap simpul yang memandu A* mendapatkan solusi yang diinginkan. Dengan heuristik, maka A* pasti akan mendapatkan solusi (jika memang ada solusinya). Dengan kata lain, heuristik adalah fungsi optimasi yang menjadikan algoritma A* lebih baik dari pada algoritma lainnya. Namun heuristik masih merupakan estimasi/perkiraan biasa

saja. Setiap sisi mempunyai *cost* yang berbeda-beda, seberapa *cost* untuk pergi dari satu simpul ke simpul lain. *Cost* dari setiap simpul ke simpul tujuan bisa diperkirakan. Ini membantu pencarian sehingga lebih kecil kemungkinan kita mencari ke arah yang salah. *Cost* untuk setiap simpul tidak harus berupa jarak. *Cost* bisa saja berupa waktu bila ingin mencari jalan dengan waktu tercepat untuk dilalui. Algoritma A* bekerja dengan prinsip yang hampir sama kecuali dengan dua perbedaan yaitu :

1. Simpul-simpul di *list* terbuka diurutkan oleh *cost* keseluruhan dari simpul awal ke simpul tujuan, dari *cost* terkecil sampai *cost* terbesar. *Cost* keseluruhan dihitung dari *cost* dari simpul awal ke simpul sekarang (*current node*) ditambah *cost* perkiraan menuju simpul tujuan.
2. Simpul di *list* tertutup bisa dimasukkan ke *list* terbuka bila jalan terpendek menuju simpul tersebut ditemukan.
3. *Cost* antara simpul adalah jaraknya dan perkiraan *cost* dari suatu simpul ke simpul tujuan adalah penjumlahan jarak dari simpul tersebut ke simpul tujuan. Untuk lebih mudah dimengerti, berikut rumusnya.

$$f(n) = g(n) + h(n) \quad (2.1)$$

keterangan :

f(n) = fungsi evaluasi

g(n) = biaya yang sudah dikeluarkan dari keadaan awal sampai keadaan

h(n) = estimasi biaya untuk sampai pada suatu tujuan mulai dari n

Node dengan nilai terendah merupakan solusi terbaik untuk diperiksa pertama kali pada g(n) + h(n). Dengan fungsi *heuristic* yang memenuhi kondisi tersebut maka pencarian dengan algoritma A* dapat optimal.Keoptimalan dari A* cukup langsung dinilai optimal jika h(n) adalah *admissible heuristic* yaitu nilai h(n) tidak akan memberikan penilaian lebih pada *cost* untuk mencapai tujuan.

Salah satu contoh dari *admissible heuristic* adalah jarak dengan menarik garis lurus karena jarak terdekat dari dua titik adalah dengan menarik garis lurus [9].

Beberapa terminologi dasar yang terdapat pada algoritma ini adalah *starting point*, *current node*, simpul, *neighbor node*, *open set*, *closed set*, *came from*, harga (*cost*), *walkability*, *target point*.

Istilah-istilah yang terdapat pada A* :

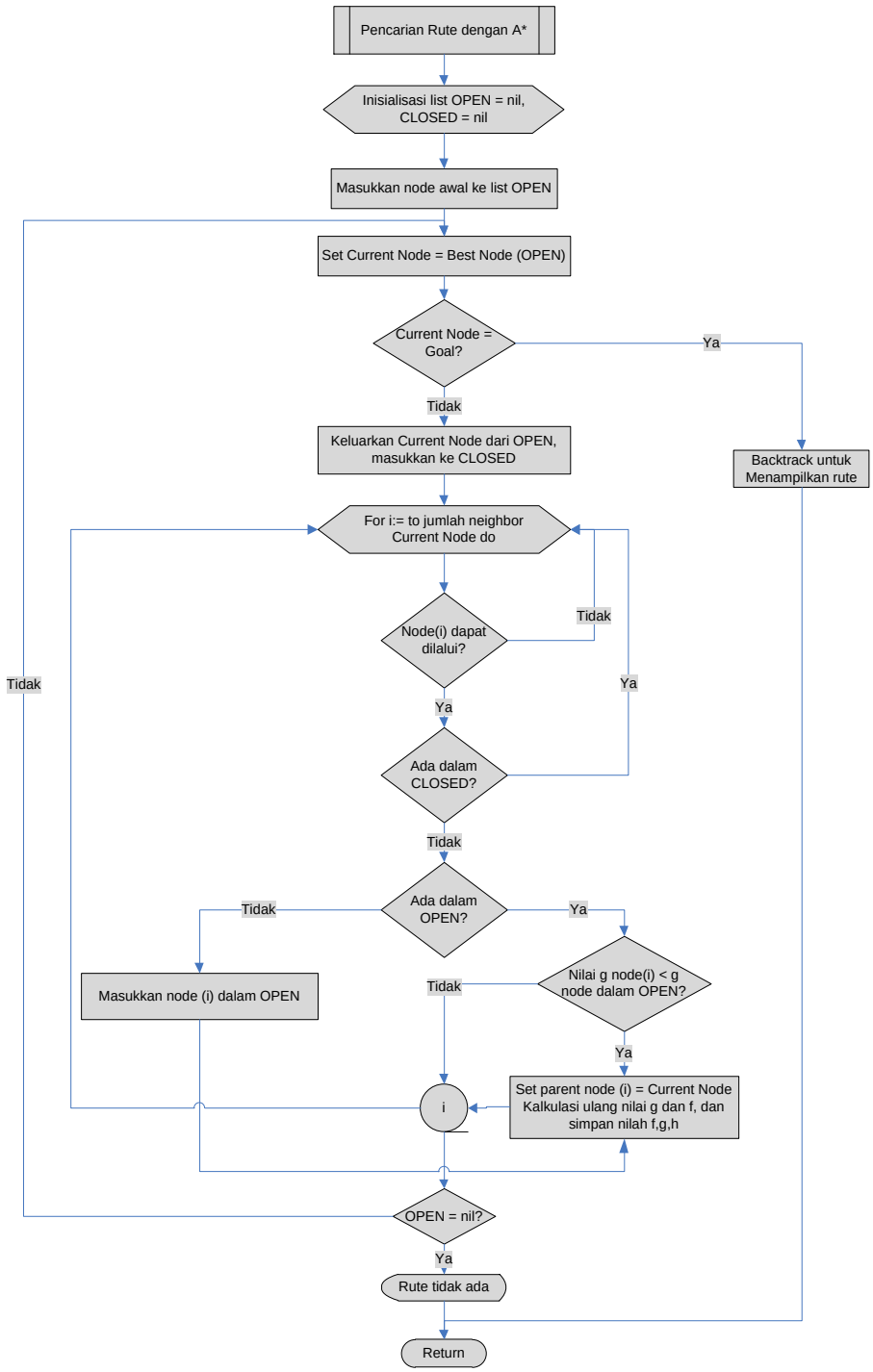
1. Simpul awal merupakan sebuah terminologi untuk posisi awal sebuah benda.
2. *Current node* adalah simpul yang sedang dijalankan dalam algoritma pencarian jalan terpendek.
3. Simpul adalah petak-petak kecil sebagai representasi dari area *pathfinding*. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.
4. *Neighbor node* adalah simpul-simpul yang bertetangga dengan *current node*.
5. *Open List* adalah tempat menyimpan data simpul yang mungkin diakses dari *starting point* maupun simpul yang sedang dijalankan.
6. *Closed List* adalah tempat menyimpan data simpul sebelum *current node* yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan.
7. *Parent* adalah tempat menyimpan data ketetanggaan dari suatu simpul, misalnya y *parent* x artinya *neighbor node* y dari *current node* x.
8. Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke *current node*, dan H, jumlah nilai perkiraan dari sebuah simpul ke *target point*.
9. *Target point* yaitu simpul yang dituju.
10. *Walkability* adalah sebuah atribut yang menyatakan apakah sebuah simpul dapat atau tidak dapat dilalui oleh *current node*.

Algoritma A* secara ringkas langkah demi langkahnya adalah sebagai berikut:

1. Tambahkan simpul awal ke dalam *open list*
2. Ulangi langkah berikut sampai pencarian berakhir:

3. Carilah simpul n dengan biaya F(n) paling rendah, dalam *open list*. Simpul dengan biaya F terendah kemudian disebut *current node*.
4. Keluarkan *current node* dari *open list* dan masukkan ke dalam *closed list*.
5. Untuk setiap 8 simpul (*neighbor node*) dari *current node* lakukan langkah berikut.
6. Jika sudah terdapat dalam *closed list* atau tidak *walkable*, maka abaikan, jika tidak lanjutkan.
7. Jika belum ada pada *open list*, tambahkan ke *open list*. Simpan *current node* sebagai *parent* dari *neighbor node* ini. Simpan harga F masing-masing simpul.
8. Jika sudah ada dalam *open list*, periksa apakah ini jalan dari simpul ini ke *current node* yang lebih baik dengan menggunakan biaya G sebagai ukurannya. Simpul dengan biaya G yang lebih rendah berarti bahwa ini adalah jalan yang lebih baik. Jika demikian, buatlah simpul ini (*neighbor node*) sebagai *parent* dari *current node*, dan menghitung ulang nilai G dan F dari simpul ini.
9. Berhenti ketika: Menambahkan *target point* ke dalam *closed list*, dalam hal ini jalan telah ditemukan, atau, gagal untuk menemukan *target point*, dan *open list* kosong. Dalam kasus ini, tidak ada jalan. walaupun telah mencapai *target point*, jika masih ada *neighbor node* yang memiliki nilai yang lebih kecil, maka simpul tersebut akan terus dipilih sampai bobotnya jauh lebih besar atau mencapai *target point* dengan bobot yang lebih kecil dibanding dengan simpul sebelumnya yang telah mencapai *target point*. Pada saat pemilihan simpul berikutnya, nilai F(n) akan dievaluasi, dan jika terdapat nilai F(n) yang sama maka akan dipilih berdasarkan nilai G(n) terbesar. Simpan jalan. Bekerja mundur dari *target point*, pergi dari masing-masing simpul ke simpul *parent* sampai mencapai *starting point*.

Secara umum dapat digambarkan dengan *flowchart* sebagai berikut:



Gambar 2.9 *Flowchart* Algoritma A* [9]

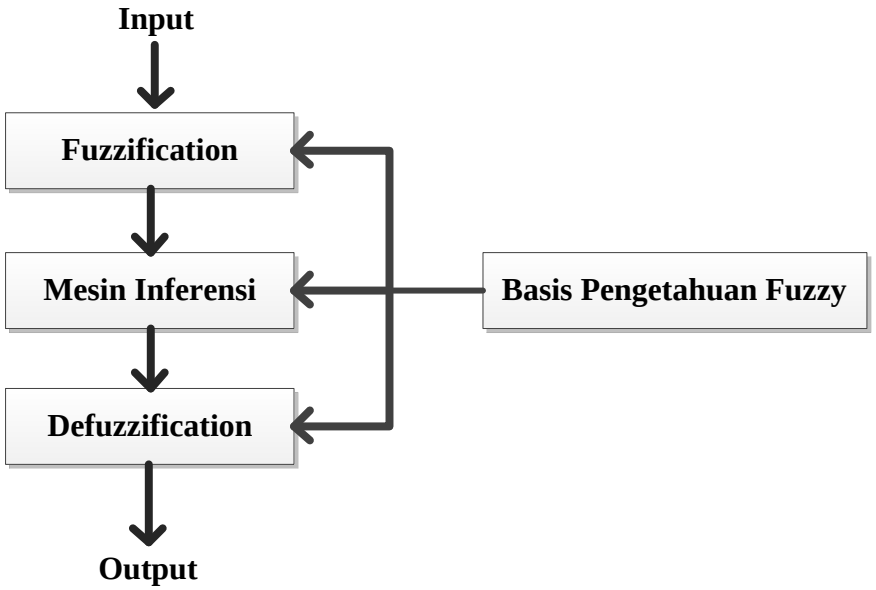
2.5.2 *Fuzzy Logic*

Konsep tentang logika *fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok diimplementasikan pada sistem, mulai dari sistem yang sederahana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akurasi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, "Ya atau Tidak", "Benar atau Salah", "Baik atau Buruk" dan lain-lain. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 dan 1. Akan tetapi, dalam logika *fuzzy* memungkinkan nilai keanggotaan berada diantara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai "Ya dan Tidak", "Benar dan Salah", "Baik dan Buruk" secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan diberbagai bidang, seperti pada sistem diagnosis penyakit (dalam bidang kedokteran), pemodelan sistem pemasaran, riset operasi (dalam bidang ekonomi), kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola (dalam bidang teknik) [10].

Konsep himpunan fuzzy memiliki 2 atribut, yaitu [10] :

1. *Linguistik*, yaitu nama suatu kelompok yang mewakili 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.

Cara kerja logika fuzzy dapat digambarkan sebagai struktur elemen dasar sistem inferensi *fuzzy* sebagai berikut:



Gambar 2.10 Struktur sistem *inferensi fuzzy* [10].

Keterangan :

Basis Pengetahuan *Fuzzy* merupakan kumpulan *rule-rule fuzzy* dalam bentuk pernyataan IF…THEN.

Fuzzyfikasi adalah proses untuk mengubah input sistem yang mempunyai nilai tegas menjadi variabel linguistic menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.

Mesin Inferensi merupakan proses untuk mengubah input *fuzzy* dengan cara mengikuti aturan-aturan (*IF-THEN Rules*) yang telah ditetapkan pada basis pengetahuan *fuzzy*.

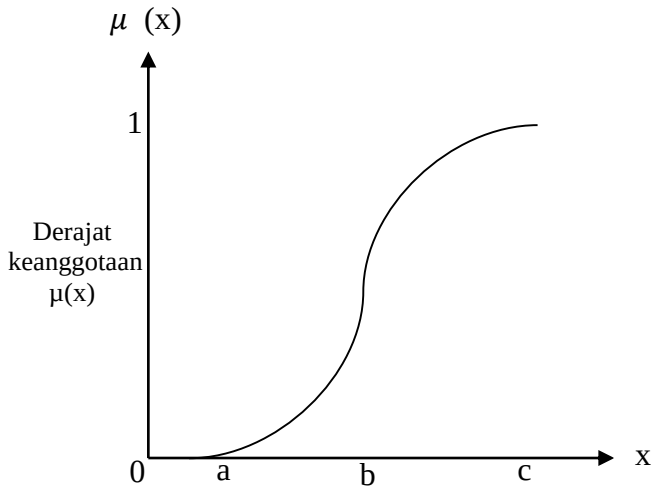
DeFuzzyfikasi merupakan proses mengubah output *fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan *fuzzyfikasi*.

2.5.2.1 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] :

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 :



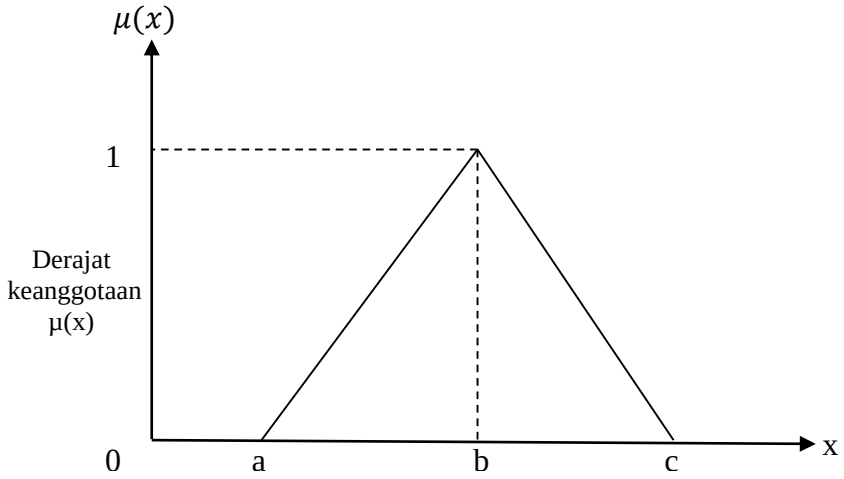
Gambar 2.11 Grafik fungsi sigmoid [10].

Fungsi keanggotaan :

$$Sigmoid(x,a,b,c) = \begin{cases} 0;\ x \le a \\ 2(x-a)/(c-a)^2; a \le x \le b \\ 1-2((c-x)/(c-a))^2; b \le x \le c \\ 1; x \ge c \end{cases} \quad (2.2)$$

Fungsi Segitiga

Fungsi segitiga hanya terdapat satu nilai x yang memiliki derajat kenanggotaan 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 sebagai berikut :



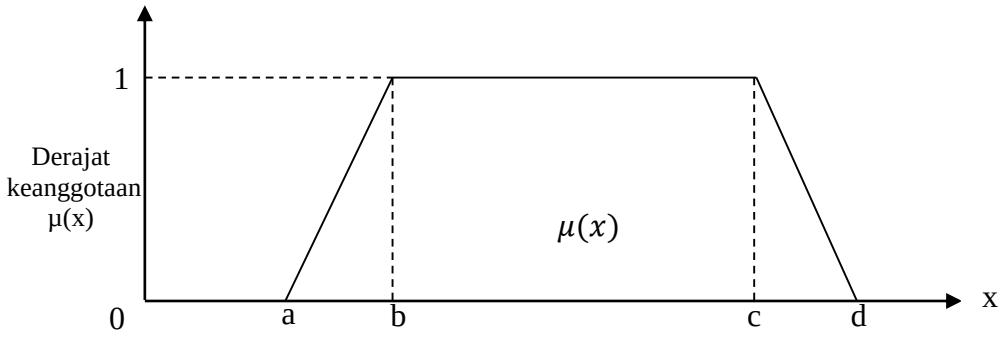
Gambar 2.12 Grafik fungsi segitiga [10].

Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0;\ x \leq a\ atau\ x\ \geq c \\ (x-a)/(b-a)\,;\, a \leq x \leq b \\ (b-x)/(c-b)\,;\, b \leq x \leq c \end{cases} \tag{2.3}$$

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 sebagai berikut :



Gambar 2.13 Grafik fungsi trapesium [10].

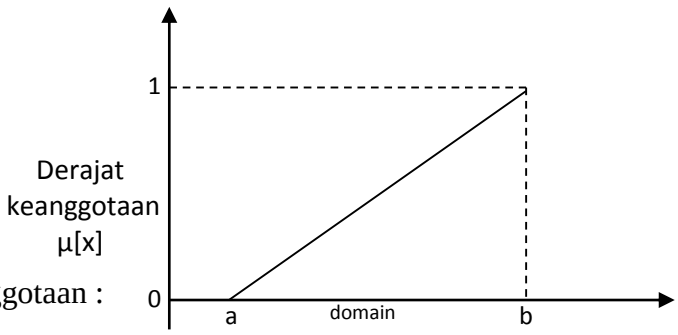
Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \le a \text{ atau } x \ge d \\ (x-a)/(b-a); & a \le x \le b \\ 1; & b \le x \le c \\ (d-x)/(d-c); & c \le x \le d \end{cases} \quad (2.4)$$

Fungsi Linear

Pada fungsi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk kurva ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan 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

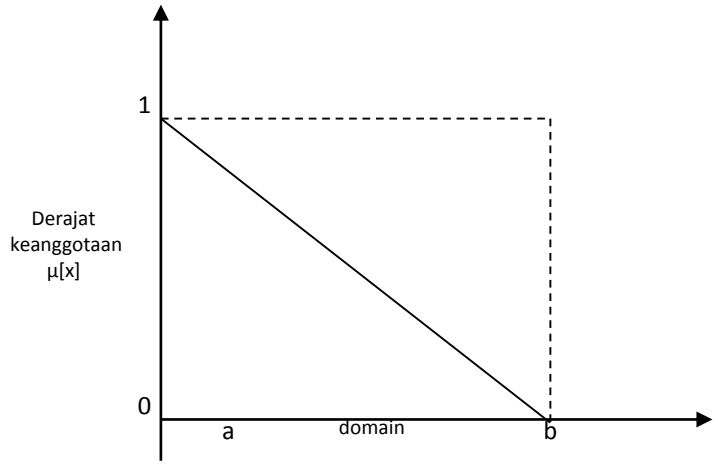
Fungsi keanggotaan :



Gambar 2.14 Representasi Linear Naik [10].

$$\mu[x] = \begin{cases} 0; & x < a \\ (x-a)/(b-a); & a < x \leq b \\ 1; & x \geq b \end{cases} \quad (2.5)$$

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 (gambar 2.10).



Gambar 2.15 Representasi Linear Turun [10]

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \leq x < b \\ 0; & x \geq b \end{cases} \quad (2.6)$$

2.5.3 Metode *Waypoint*

Waypoint merupakan kumpulan dari beberapa titik kordinat yang kemudian dijadikan sebagai navigasi pergerakan. Dalam sebuah *game*, pergerakan NPC (*non playable character*) umumnya menggunakan titik arah antara titik kordinat yang satu dengan yang lain ditentukan oleh pencipta *game*. *Waypoint* dapat digunakan pada *non playable character* untuk melakukan patroli di dalam game dengan melintasi titik-titik kordinat yang ditentukan. Gambar 2.16 adalah gambar lingkungan game yang belum diberikan waypoint, sedangkan di gambar

2.17 adalah gambar game yang sudah dimasukkan waypoint. Dapat dilihat seperti berikut



Gambar 2.16 Salah satu tempat dalam permainan.

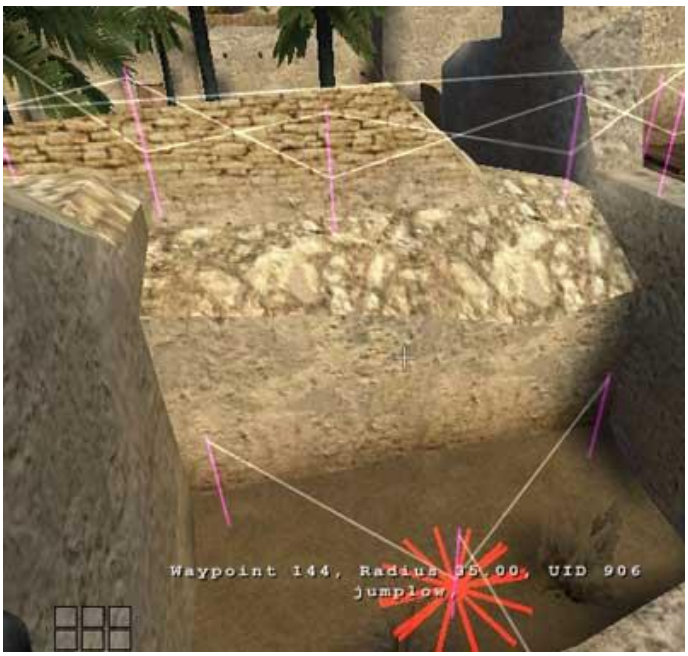


Gambar 2.17 Salah satu permainan dengan *waypoint*.

Dari gambar 2.17 dapat terlihat 29 titik kordinat dari satu ke yang lain saling berhubungan, yang merupakan rute dari NPC untuk melakukan patroli untuk menemukan dan menyerang karakter *player*, dimana NPC akan terus melintas dari satu titik ke titik selanjutnya.

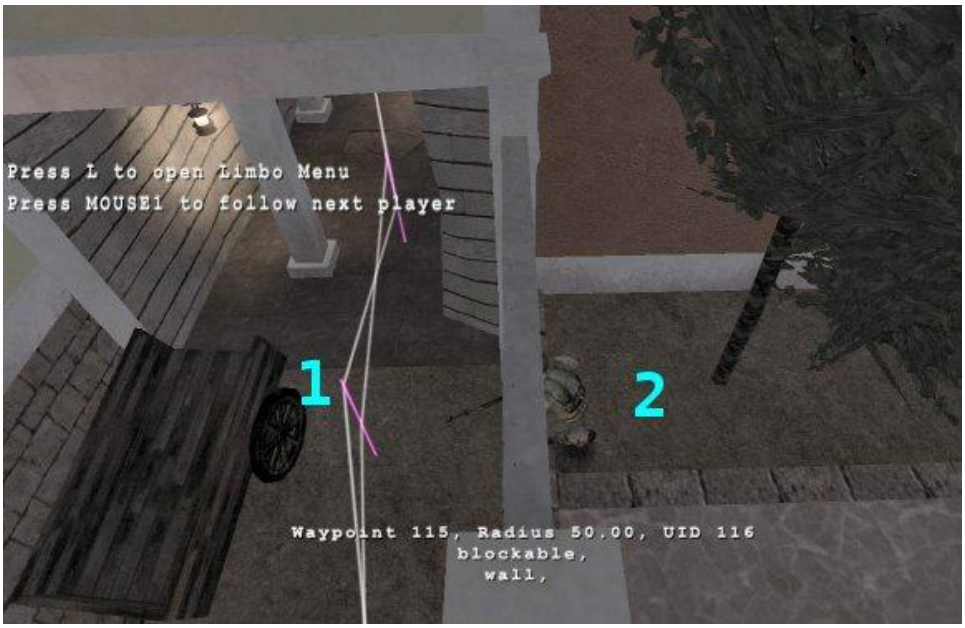
Cara NPC menentukan titik arah dapat dilakukan dengan menggunakan *graph* waypoint untuk menentukan keputusanyang baik tentang bagaimana menempatkan titik arah, cara mengatur radius, bagaimana memahami ketika hal-hal yang mungkin tidak berperilaku sesuai dengan yang di harapkan. Ketika NPC akan memutuskan tujuan yang mana akan dituju dalam peta permainan, maka NPC akan terlebih dahulu melakukan pemeriksaan mana *waypoint* terdekat untuk NPC, dan kemudian NPC akan berjalan menuju *waypoint* tersebut, dengan menghindari jalur penghalang.

NPC tidak dapat berjalan dengan baik atau sempurna jika letak *waypoint* terdapat di tempat yang tidak memungkinkan dalam fisik permainan seperti penghalang, berikut ini adalah contoh



Gambar 2.18 *Waypoint* yang menggangu pergerakan NPC

Pada gambar diatas terlihat 3 *waypoint* yang tidak perlu untuk dilewati oleh NPC karena jika dilewati NPC bisa jatuh kebawah sehingga terjebak dan tidak bisa melintas untuk menuju *waypoint* selanjutnya. Berikut ini juga merupakan penempatan waypoint yang kurang ektif sehingga dapat membuat NPC tidak bias melintas dengan sempurna



Gambar 2.19 *Waypoint* yang terpisah penghalang

Waypoint diatas mengakibatkan NPC mengabaikan titik yang ke dua sehingga menimbulkan ke tidak sempurnaan dalam pergerakan NPC untuk melakukan patrol dalam permainan.

2.6 OOP (Object Oriented Programming)

OOP atau *Object Oriented* Programming adalah teknik pemrograman berbasis *object*. Pemrograman dibagi dalam object-object kemudian di dalam object-object ini ada *property* dan *method*. Property ini lebih dikenal dengan *variable* yang bersifat public dan *method* ini merupakan function. Object-object ini bisa diperluas lagi dengan *inheritance* atau turunan. Object-object turunan ini biasa dipakai untuk membuat object yang merupakan modifikasi dari object lain.

Berikut ini beberapa bahasa pemrograman yang sudah menggunakan konsep OOP adalah :

1. *C++*.

2. *Visual C++*.

3. *Visual Basic*.

4. *Java*.

2.7 UML (*Unified Modeling Language*)

UML adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sebuah sistem. *UML* adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya[12]. *UML* adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut.

UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM. UML menyediakan 10 macam diagram yang dapat digunakan untuk memodelkan aplikasi berorientasi objek yang dibuat.

2.7.1 Use Case Diagram

Menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Menggambarkan kebutuhan system dari sudut pandang user, Memfokuskan pada proses komputerisasi (automated processes), Menggambarkan hubungan antara use case dan actor.

Use case menggambarkan proses sistem (kebutuhan sistem dari sudut pandang user).

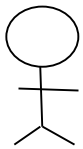
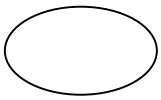
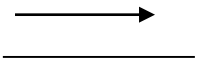
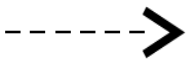
Secara umum use case adalah:

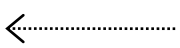
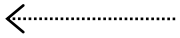
a. Pola perilaku sistem
b. Urutan transaksi yang berhubungan yang dilakukan oleh satu actor

Use case diagram terdiri dari

1. Use case
2. Actors
3. Relationship
4. System boundary boxes (optional)
5. Packages (optional)

Tabel 2.1 Simbol *use case diagram*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case
	*Use case* : Abstraksi dari interaksi antara sistem dan aktor
	*Association* adalah abstraksi dari penghubung antara aktor dan use case
	*Generalisasi* : Menunjukan spesialisai aktor untuk dapat berpartisipasi dalam *Use case*

<<include>> 	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lain nya
<<extend>> 	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lain nya jika suatu kondisi terpenuhi.

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain [12].

2.7.2 *Activity Diagram*

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur 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, dimana 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 [12].

2.7.3 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram mempunyai beberapa jenis seperti :

1. Entity Class
2. Boundary Class
3. Control Class

2.7.4 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan [12].

2.8 *C#*

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET *Framework* [12]. C# adalah Java versi Microsoft, sebuah bahasa multi flatform yang didesain untuk bisa berjalan di berbagai mesin. C# adalah pemrograman berorientasi Object (OOP). C# memiliki kekuatan bahasa C++ dan portabilitas seperti *Java*. Fitur-fitur yang diambilnya dari bahasa C++ dan *Java* adalah desain

berorientasi objek, seperti *garbage collection, reflection*, akar kelas (*root class*), dan juga penyederhanaan terhadap pewarisan jamak (*multiple inheritance*).

Bahasa pemrograman C# dibuat sebagai bahasa pemrograman yang bersifat *general-purpose* (untuk tujuan jamak), berorientasi objek, modern, dan sederhana. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem *clien-server (hosted system)* maupun sistem *embedded (embedded system),* mulai dari program aplikasi yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada program aplikasi yang sangat kecil.

Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran program aplikasi yang dibuat dengan menggunakan bahasa pemrograman C [13].

2.9 *Javascript*

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan *"LiveScript"* yang berfungsi sebagai bahasa sederhana untuk browser Netscape Navigator 2. Pada masa itu bahasa ini banyak di kritik karena kurang aman, pengembangannya yang terkesan buru-buru dan tidak ada pesan kesalahan yang di tampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan *Sun* (pengembang bahasa pemrograman *"Java"*) pada masa itu, maka Netscape memberikan nama *"JavaScript"* kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai *"Jscript"* di browser Internet Explorer 3.

2.10 Multimedia

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi dan video dengan alat bantu (*tool*)

dan koneksi (*link*) sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi (Hofstetter 2001). Multimedia sering digunakan dalam dunia hiburan. Selain dari dunia hiburan, Multimedia juga diadopsi oleh dunia Game. Multimedia juga dapat diartikan sebagai penggunaan beberapa media yang berbeda dalam menyampaikan informasi berbentuk text, audio, grafik, animasi, dan video.

2.10.1 Katagori Multimedia

Multimedia dapat di definisikan menjadi 2 kategori, yaitu Multimedia Content Production dan Multimedia Communication.

1. *Multimedia Content Production* adalah penggunaan beberapa media (teks, audio, graphics, animation, video dan interactivity) yang berbeda dalam menyampaikan suatu informasi atau menghasilkan produk multimedia seperti video, audio, musik, film, game, entertaintment, dll. Bisa juga dikatakan sebagai penggunaan beberapa teknologi yang berbeda yang memungkinkan untuk menggabungkan media (teks, audio, graphics, animation, video, dan interactivity) dengan cara yang baru untuk tujuan komunikasi. Dalam kategori ini media yang digunakan adalah :
   a. Media teks/tulisan
   b. Media audio/suara
   c. Media video
   d. Media animasi
   e. Media gambar
   f. Media Interaktif
   g. Media spesial effect
2. *Multimedia Communication* adalah penggunaan media (massa), seperti televisi, radio, media cetak dan internet untuk mempublikasikan, menyiarkan ,mengkomunikasikan material periklanan, publikasi, entertaintment, berita, pendidikan, dll. Dalam kategori ini media yang digunakan adalah :
   a. TV

b. Radio
c. Film
d. Media Cetak
e. Musik
f. Game
g. Entertainment
h. Tutorial
i. Internet

Dengan penggunaan multimedia, penyampaian informasi akan menjadi lebih menarik dan mempermudah pengguna dalam mendapatkan informasi tersebut. Seperti yang disebutkan dalam laporan hasil penelitian yang dikeluarkan oleh *Computer Technology Research* (Hofstetter) bahwa seseorang hanya akan mendapatkan 20% dari apa yang mereka lihat dan 30% dari yang mereka dengar. Sedangkan melalui multimedia akan mendapatkan 50% dari apa yang mereka lihat dan dengar, sampai 80% dari apa yang mereka lihat, dengar dan berinteraksi dengan pada waktu yang sama.

2.10.2 Elemen Multimedia

Menurut Hofstetter (2001) komponen multimedia terbagi atas lima jenis yaitu:

1. Teks

   Teks merupakan elemen multimedia yang menjadi dasar untuk menyampaikan informasi, karena teks adalah jenis data yang paling sederhana dan membutuhkan tempat penyimpanan yang paling kecil. Teks merupakan cara yang paling efektif dalam mengemukakan ide-ide kepada pengguna, sehingga penyampaian informasi akan lebih mudah dimengerti oleh masyarakat. Jenis-jenis teks seperti Printed Text, yaitu teks yang dihasilkan oleh word processor atau word editor dengan cara diketik yang nantinya dapat dicetak. Scanned Text yaitu teks yang dihasilkan melalui proses scanning tanpa pengetikan.

Dan Hypertext yaitu jenis teks yang memberikan link ke suatu tempat / meloncat ke topik tertentu.

2. Grafik (image)

   Sangat bermanfaat untuk mengilustrasi informasi yang akan disampaikan terutama informasi yang tidak dapat dijelaskan dengan kata-kata. Jenis-jenis grafik seperti bitmap yaitu gambar yang disimpan dalam bentuk kumpulan pixel, yang berkaitan dengan titik-titik pada layar monitor. Digitized picture adalah gambar hasil rekaman video atau kamera yang dipindahkan ke komputer dan diubah ke dalam bentuk bitmaps. Hyperpictures, sama seperti hypertext hanya saja dalam bentuk gambar.

3. Audio

   Multimedia tidak akan lengkap jika tanpa audio (suara). Audio bisa berupa percakapan, musik atau efek suara. Format dasar audio terdiri dari beberapa jenis :

   a. WAVE

      Merupakan format file digital audio yang disimpan dalam bentuk digital dengan eksistensi WAV.

   b. MIDI (Musical Instrument Digital Interface) MIDI memberikan cara yang lebih efisien dalam merekam music dibandingkan wave, kapasitas data yang dihasilkan juga jauh lebih kecil. MIDI disimpan dalam bentuk MID.

4. Video

   Video menyediakan sumber yang kaya dan hidup untuk aplikasi multimedia. Dengan video dapat menerangkan hal-hal yang sulit digambarkan lewat kata-kata atau gambar diam dan dapat menggambarkan emosi dan psikologi manusia secara lebih jelas.

5. Animasi

   Animasi adalah simulasi gerakan yang dihasilkan dengan menayangkan rentetan frame ke layer. Frame adalah satu gambar tunggal pada rentetan

gambar yang membentuk animasi. Menurut Foley, Van Dam, Feiner dan Hughes Animate adalah untuk membuat sesuatu hidup, sebagian orang mengira bahwa animasi itu sama dengan motion (gerakan), tetapi animasi mencakup semua yang mengandung efek visual sehingga animasi mencakup perubahan posisi terhadap waktu, bentuk, warna, struktur, tekstur dari sebuah objek, posisi kamera, pencahayaan, orientasi dan focus dan perubahan dalam teknik rendering.

2.11 Teknik Pembuatan Kuesioner

Tata cara dalam pembuatan kuesioner untuk responden sangatlah penting untuk diperhatikan, baik dari segi tujuan, jenis pertanyaan, dan petunjuk membuat pertanyaan.

2.11.1 Tujuan pembuatan kuesioner

Adapun tujuan pembuatan kuesioner adalah sebagai berikut :

1. Memperoleh informasi yang relevan dengan tujuan survai.
2. Memperoleh informasi dengan tingkat keandalan (*reliability*) dan keabsahan atau validitas (*validity*) setinggi mungkin.

Validitas atau keabsahan adalah menyangkut pemahaman mengenai kesesuaian antara konsep dengan kenyataan empiris. *Reabilitas* memberikan kesesuaian antara hasil-hasil pengukuran atau konsistensi pengukuran sedangkan validitas merupakan kesesuaian konsep pengukuran tersebut dengan fakta dilapangan. Suatu alat ukur (pengukuran) yang validitasnya atau tingkat keabsahannya tinggi secara otomatis biasanya dapat diandalkan (*reliable*). Namun sebaliknya, suatu pengukuran yang andal, belum tentu memiliki keabsahan yang tinggi.

Contoh:

Apabila kita sakit demam, biasanya suhu meningkat, misalnya $40.^0$ C. Tetapi bisa saja termometer (atau suhu tubuh anda) menunjukkan angka $40.^0$ C, padahal anda tidak sakit demam.

Dengan demikian, konsep suhu tubuh dan demam tidak konsisten. Untuk itu perlu dipahami lebih dalam pengertian antara konsep dengan kenyataan empiris.

Kesimpulannya, penelitian yang baik harus memiliki reabilitas yang tinggi sekaligus memiliki validitas yang tinggi.

2.11.2 Jenis Pertanyaan

Ada 3 jenis pertanyaan, yaitu terbuka *(open-ended question),* tertutup *(close-ended question),* dan kombinasi terbuka dan tertutup[18].

1. Pertanyaan Terbuka

Pertanyaan terbuka adalah pertanyaan yang tidak menggiring ke jawaban yang sudah ditentukan dan tinggal dipilih dari alternatif yang ditawarkan.

Contoh: Menurut pendapat anda merek mie instan apa yang paling disukai anak-anak?

2. Pertanyaan Tertutup

Pertanyaan tertutup adalah pertanyaan yang sudah menggiring ke jawaban yang alternatifnya sudah ditetapkan (ya atau tidak).

Contoh: Apakah anda sudah mendengar bahwa Dayun Kancil meloloskan diri?

(1) Ya (2) Tidak

3. Kombinasi Tertutup dan Terbuka

Contoh pertanyaan :

1. Ada sementara orang yang beranggapan bahwa kebebasan berpolitik di negeri ini tidaklah seperti yang diharapkan. Menurut anda sampai seberapa jauhkah kebebasan yang anda rasakan untuk hidup di negeri ini?

   a. Sangat bebas → lanjutkan ke pertanyaan no. 3.

   b. Cukup bebas → lanjutkan ke pertanyaan no. 3.

   c. Tidak begitu bebas → lanjutkan ke pertanyaan no. 2.

   d. Tidak bebas sama sekali → lanjutkan ke pertanyaan no. 2.

2. Dalam hal apakah anda merasa tidak begitu bebas?
3. Dengan mempertimbangkan berbagai segi, seberapa jauh kepuasan anda mengenai kehidupan berpolitik di negeri ini?
   Angka manakah yang paling dekat mencerminkan tingkat kepuasan yang anda rasakan?

1 ——— 2 ——— 3 ——— 4 ——— 5 ——— 6 ——— 7

sangat puas netral sangat tidak puas

2.11.3 Petunjuk Membuat Pertanyaan

Dalam membuat suatu pertanyaan dalam kuesioner, diperlukan beberapa hal sebagai petunjuk pembuatan pertanyaan seperti dibawah ini :

1. **Gunakan Kata-kata Sederhana**

Gunakan kata-kata yang sederhana yang diketahui oleh semua responden. Hindari istilah yang hebat tetapi responden kurang atau tidak mengerti.

Contoh: Bagaimana status perkawinan Bapak?

Pertanyaan yang lebih baik : Apakah Bapak beristri?

2. **Pertanyaan Jelas dan Khusus**

Usahakan pertanyaan jelas dan khusus**.**

Contoh: Berapa orang berdagang di sini?

(di sini, maksudnya di dalam toko, dipasar)

3. **Pertanyaan Berlaku Bagi Semua Responden**

Usahakan agar pertanyaan berlaku untuk semua responden**.**

Contoh: Apa pekerjaan Saudara sekarang?

(Ternyata dia menganggur! Seharusnya ditanyakan dahulu “Saudara bekerja?” Kalau jawabannya “ya”, lalu ditanyakan apa pekerjaannya.

4. **Berkaitan dengan Masalah dan Sasaran Penelitian**

Pertanyaan harus berkaitan dengan masalah penelitian dan sasaran-sasaran penelitian. (Setiap pertanyaan dimaksudkan untuk memancing informasi yang dapat digunakan untuk menguji hipotesis penelitian).

5. **Tidak Ambigu**

Pertanyaan harus jelas dan tidak mengandung tafsir majemuk.

Contoh: Bagaimana kondisi anda hari ini?

Kondisi keuangan, kesehatan, perkawinan, dsb?).

6. **Tidak Menggiring**

Pertanyaan tidak boleh menggiring responden untuk memberikan alternatif jawaban tertentu.

Contoh: Apakah anda telah membaca tulisan-tulisan tentang kredit macet konglomerat? (Jawaban "Ya")

7. **Tidak Memuat Informasi yang Tidak Dimiliki oleh Responden**

Pertanyaan tidak boleh memuat informasi (pengetahuan) yang tidak dimiliki oleh responden.

Contoh: Bagaimana pendapat anda mengenai nilai BCR Proyek Kanindotex?

8. **Tidak Memuat Hal yang Bersifat Pribadi dan Peka**

Pertanyaan tidak boleh memuat hal-hal yang bersifat pribadi dan peka sehingga responden mungkin menolak menjawabnya.

Contoh: Pertanyaan mengenai penghasilan, atau menanyakan uang secara umum tidak disetujui. Misalnya, apakah Bapak pernah terlibat G30S?

9. **Tidak Bersifat Klise**

Pertanyaan tidak boleh bersifat klise, sehingga jawabannya juga cenderung klise (stereotip).

Contoh: Apakah Bapak senang berpenghasilan tinggi?

2.11.4 Metode Linkert

Metode Skala likert adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Pilihan jawaban pertanyaan kuesioner lebih dari 2 dan minimal 3[19]. Skala Likert digunakan untuk mengukur sikap, pendapat, dan presepsi seseorang mulai dari yang bersifat sangat positif sampai ke sangat negative. Biasanya cara pengisian kuisioner jenis ini dengan menggunakan ceklist atau pilihan ganda[3]. Kemudian untuk masing-masing sikap kemudian di beri bobot berikut dapat dilihat pada tabel 2.2

Contoh Skala Likert adalah :

Tabel 2.2 Contoh Skala likert

Presepsi responden	Nilai Sikap
Sangat setuju (SS)	5
Setuju (S)	4
Ragu-Ragu (R)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju	1

2.12 Pengujian Perangkat Lunak

Pengujian adalah proses untuk menemukan *error* pada perangkat lunak sebelum di-delivery kepada pengguna. Pengujian adalah proses menjalankan program dengan maksud mencari kesalahan (*error*).

Kasus uji yang baik adalah kasus yang memiliki peluang untuk mendapatkan kesalahan yang belum diketahui. Pengujian dikatakan berhasil bila dapat memunculkan kesalahan yang belum diketahui. Jadi pengujian yang baik

bukan untuk memastikan tidak ada kesalahan tetapi untuk mencari sebanyak mungkin kesalahan yang ada pada program.

2.12.1 Metode Perancangan Kasus Uji

Black Box adalah pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinsikan. Metode Black Box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Black Box dapat menemukan kesalahan dalam kategori berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam strutur data atau akses basisdata eksternal
4. Inisialisasi dan kesalahan terminasi
5. Validitas fungsional
6. Kesensitifan sistem terhadap nilai input tertentu
7. Batasan dari suatu data.

Tipe dari *Black Box Testin*g :

1. Equivalence class partitioning
2. Sample testing.
3. Limit testing.
4. Robustness testing.
5. Behavior testing.
6. Requirement testing.

White Box : pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya.

Metode pengujian dengan menggunakan struktur kontrol program untuk memperoleh kasus uji.

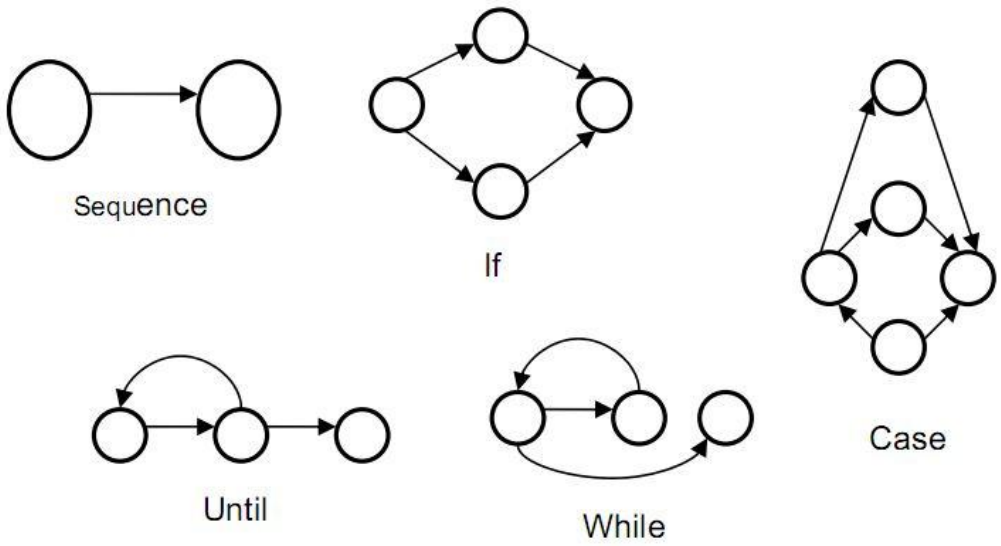
Dengan menggunakan white box akan didapatkan kasus uji yang :

a) Menjamin seluruh jalur independen di dalam modul yang dieksekusi sekurang-kurangnya sekali.
b) Menguji semua keputusan logical.
c) Menguji seluruh Loop yang sesuai dengan batasannya.
d) Menguji seluruh struktur data internal yang menjamin validitas.

Basis Path adalah teknik uji coba white box (Tom Mc Cabe).

Basis Path : untuk mendapatkan kompleksitas lojik dari suatu prosedur dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan himpunan jalur yang akan diuji.

Basis Path menggunakan notasi graph untuk menggambarkan aliran kontrolnya. Gambar di bawah ini menunjukkan notasi *graph* untuk menggambarkan skema dasar pemrograman[20].



Gambar 2.20 Notasi Graph untuk Skema Dasar Pemrograman

Cyclomatic Complexity adalah ukuran yang menunjukkan kompleksitas lojik suatu program. Cyclomatic Complexity dapat diperoleh dengan menghitung

daerah yang dapat dibenuk oleh graph. Cyclomatic Complexity V(G) dapat juga dihitung dengan :

V (G) = E – N + 2

Dimana :

E = jumlah edge pada flowgraph

N = Jumlah Node pada flowgraph

Atau Cyclomatic Complexity juga dapat dihitung dengan rumus :

V (G) = P + 1

dimana P = jumlah predikat Node pada flow graph.

Jalur independen adalah jalur yang melintasi atau melalui program dimana sekurang – kurangnya dieksekusi satu kali. Jalur independen sama dengan jumlah Cyclomatic Complexitynya.

2.13 *Unity*

Unity Game Engine adalah *software* atau *game engine* yang digunakan untuk membuat *video game* berbasis dua atau tiga dimensi dan dapat digunakan secara gratis. Selain untuk membuat *game*, *unity* juga dapat digunakan untuk membuat konten yang interaktif lainnya seperti, visual arsitektur dan *real-time* 3D animasi.

Unity adalah sebuah *tool* yang terintegrasi untuk membuat *game*, arsitektur bangunan dan simulasi. *Unity* bisa digunakan untuk *game*s PC dan *games online*. Untuk *games online* diperlukan sebuah plugin, yaitu *Unity Web Player*, yang sama halnya dengan *flash player* pada *browser*. Bahasa pemrograman yang digunakan bermacam-macam, mulai dari *javascript*, C#, dan *boo* [14]. *Unity* tidak bisa melakukan desain atau *modelling*, dikarenakan *unity* bukan merupakan *tools* untuk mendesain. Banyak hal yang bisa di lakukan di unity, ada fitur *audio reverb zone* , *particle effect* , *sky box* untuk menambahkan langit, dan masih banyak lagi, dan juga bisa langsung *edit texture* dari *editor* seperti *photoshop* dll.

Features (Scripting) di dalam *unity* adalah sebagai berikut:

1. Mendukung 3 bahasa pemrograman, *JavaScript*, C#, dan *Boo*. *Flexible and EasyMoving*, *rotating*, dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *duplicating, removing*, dan *changing properties.*
2. *Multi Platform Game* bisa di deploy di PC, Mac, Wii, iPhone, iPad dan *browser, android*.
3. *Visual Properties Variables* yang di definisikan dengan *scripts* ditampilkan pada editor. Bisa digeser, di *drag* and *drop*, bisa memilih warna dengan *color picker.*
4. Berbasis .NET, penjalanan program dilakukan dengan *Open Source* .NET *platform, Mono.*

2.14 *3DS Max*

3ds Max atau *3D Studio Max* adalah salah *satu software* atau perangkat lunak yang sering digunakan oleh perancang produk untuk membuat animasi atau pemodelan dalam bentuk 3 dimensi. Aplikasi canggih ini dirilis oleh salah satu perusahaan *Autodesk Media & Entertainment* yang pada mulanya dikenal sebagai *Discreet and Kinetix*. *3D Max* merupakan salah satu dari sekian banyak aplikasi *modeling* untuk membuat model 3D dan paling banyak digunakan oleh perancang yang tersebar di seluruh dunia [15]. Sejalan dengan berkembangnya teknologi termasuk juga dalam bidang komputerisasi. 3D*Max* pun mengalami perubahan-perubahan untuk menyesuaikan dengan kemampuan komputer yang semakin tinggi dalam hal grafis. *3D Max* dikembangkan dari aplikasi sebelumnya yang bernama *3D Studio for Dos*, tetapi aplikasi ini hanya diperuntukkan untuk *platform Win32*.

Dengan semakin canggihnya kemampuan *software* ini, maka tidak aneh *3D Max* menjadi program animasi komputer 3D dengan penjualan terbesar di dunia. *Software* ini memiliki kemampuan *modeling* yang kuat dan merupakan *plugin architecture* yang fleksibel dan bekerja dengan *platform Microsoft Windows*. *3D Studio Max* banyak digunakan oleh para pembuat dan perancang

video game, visual architecture, design product dan juga studio TV untuk pembuatan animasi [15].

2.15 *Google Sketchup*

Google SketchUp merupakan program modeling yang diperuntukkan bagi para profesional di bidang arsitektur, teknik sipil, pembuat film, pengembang *game*, dan profesi terkait. Program ini disediakan *Google* untuk dapat didownload secara gratis. *Google SketchUp* memiliki kelebihan pada kemudahan penggunaan dan kecepatan dalam melakukan desain, berbeda dengan program 3D CAD lainnya.

Google SketchUp adalah sebuah perangkat lunak desain grafis yang dikembangkan oleh *Google*. Pendesain grafis ini dapat digunakan untuk membuat berbagai jenis model, dan model yang dibuat dapat diletakkan di *Google Earth* atau dipamerkan di *Google 3D Warehouse.* SketchUp di kembangkan oleh perusahaan startup @Last Software, Boulder, Coloroda yang di bentuk pada tahun 1999. SketchUp pertama kali dirilis pada tahun 2000 bulan Agustus sebagai tujuan umum alat pembuatan konten 3D. pada tanggal 14 maret 2006 google mengakuisisi @ Last Software, karena google tertarik buat plugin untuk google Earth.

Pada tanggal 27 April 2006, Google mengumumkan Google SketchUp, yang bebas-download versi SketchUp. Versi gratis ini beda dengan versi SketchUp Pro, tetapi terpadu mencakup alat untuk meng-upload konten ke Google Earth dan Google 3D Warehouse, repositori model dibuat dalam SketchUp. Mereka juga menambahkan kotak peralatan baru di mana Anda dapat berjalan, melihat segala sesuatu dari sudut pandang seseorang, label untuk model, melihat-lihat alat, dan "Setiap poligon" bentuk alat. Sedangkan versi gratis dari Google Sketchup 3D hanya dapat mengekspor ke SKP dan Google Earth. Kmz format file, versi Pro bisa mengekspor dan memasukkan .3ds, .Dae, .DWG, .DXF, .Fbx, .Obj, .XSI, dan .wrl format file. Google SketchUp juga dapat menyimpan

"screenshot" dari model .Bmp, .Png, .Jpg, .Tif dengan versi Pro juga mendukung. Pdf, .Eps, .Epx, .DWG, dan .DXF.

2.16 *Adobe Photoshop*

Adobe Photoshop, atau biasa disebut *Photoshop*, adalah perangkat lunak editor citra buatan *Adobe Systems* yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar/foto, dan, bersama *Adobe Acrobat*, dianggap sebagai produk terbaik yang pernah diproduksi oleh *Adobe Systems*. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (*Creative Suite*), versi sembilan disebut *Adobe Photoshop* CS2, versi sepuluh disebut *Adobe Photoshop* CS3 , versi kesebelas adalah *Adobe Photoshop* CS4 , versi keduabelas adalah *Adobe Photoshop* CS5 , dan versi yang terakhir (ketigabelas) adalah *Adobe Photoshop* CS6.

Photoshop tersedia untuk *Microsoft Windows*, *Mac* OS X, dan Mac OS versi 9 ke atas, juga dapat digunakan oleh sistem operasi lain seperti Linux dengan bantuan perangkat lunak tertentu seperti *CrossOver*.

2.14.1 Sejarah Photoshop

Seorang profesor dari Michigan (USA) bernama Glenn Knoll membuat sebuah eksperimen untuk mengolah foto secara digital pada tahun 1987. Dengan alat seadanya, sang professor bekerja keras di ruang gelap milik pribadi. Beliau memiliki dua orang anak yang bernama John Knoll dan Thomas Knoll. Kedua anak tersebut meneruskan cita-cita Glenn Knoll untuk membuat sebuah program pengolah gambar secara digital tersebut. Versi pertama diliris oleh Adobe pada tahun 1990. Versi awal Photoshop yang diberi nama “Knoll Software” diliris sebelum kerjasama dengan Adobe resmi dibuat. Photoshop versi pertama ini berukuran 1.4MB. Lebih jauh, aplikasi Photoshop 0.63 lengkap dengan manual online dapat disimpan dalam disket dan bahkan masih menyisakan 200kb di dalam “disk space” bebas. Meskipun pada awalnya Photoshop dirancang untuk

menyunting gambar untuk cetakan berbasis kertas Photoshop yang ada saat ini juga dapat digunakan untuk memproduksi gambar untuk (World Wide Web). Beberapa versi terakhir juga menyertakan aplikasi tambahan, (Adobe Image Ready), untuk keperluan tersebut. Photoshop juga memiliki hubungan yang erat dengan beberapa perangkat lunak penyunting media, animasi, dan *authoring* buatan *adobe* lainnya. File format asli Photoshop, .PSD, dapat diekspor ke dan dari *Adobe ImageReady*, *Adobe illustrator*, *Adobe Premiere Pro*, *Adobe After Effect dan Adobe Ecore* DVD untuk membuat DVD professional, menyediakan penyuntingan gambar *non-linear* dan layanan dan *special effect* seperti *background, tekstur*, dan lain-lain untuk keperluan televisi, film, dan situs web. Photoshop dapat menerima penggunaan beberapa model warna yaitu :

1. RGB color model,
2. Lab color model,
3. CMYK color model,
4. Grayscale,
5. Binary imageBitmap,
6. Duotone.

Photoshop memiliki kemampuan untuk membaca dan menulis gambar berformat *raster* dan *vector* seperti, .png, .gif, .jpg, dan lain-lain. Photoshop juga memiliki beberapa format file khas .PSD(“Photoshop Document”) format yang menyimpan gambar dalam bentuk “*layer*”, termasuk *teks*, “*mask”, ‘‘opacity*”, “*blend mode*”, “*channel warna*”, “*channel alpha*”, “*clipping paths*”, dan *setting duotone*. Kepopuleran *photoshop* membuat format ini digunakan secara luas, sehingga memaksa programmer program penyunting gambar lainnya menambahkan kemampuan untuk membaca format PSD dalam perangkat lunak mereka. “.PSB” adalah versi terbaru dari PSD yang didesain untuk file yang berukuran lebih dari 2 (dua) GB. “.PDD” adalah versi lain dari PSD yang hanya dapat mendukung fitur perangkat lunak *PhotoshopDeluxe*.