BAB II
LANDASAN TEORI

II.1 Kedatangan Tentara Inggris

Kekalahan Jepang tanpa syarat kepada Sekutu mengakibatkan Jepang kehilangan semua jajahannya termasuk Indonesia. Jepang harus menyerahkannya kepada Sekutu. Tugas pengalihan kekuasaan dari tangan Jepang dilakukan oleh Komando Asia Tenggara (*South East Asia Command*). Pasukan ini dipimpin Laksamana Lord Lousi Mountbatten. Untuk melaksanakan tugas itu, Mountbatten membentuk komando khusus yang diberi nama *Alied Force Netherland East Indies* (AFNEI).

Pasukan Sekutu dan AFNEI mendarat di Jakarta pada tanggal 29 September 1945. Kedatangan Sekutu semula disambut dengan sikap terbuka oleh pihak Indonesia. Namun setelah diketahui bahwa pasukan Sekutu datang bersama orang-orang NICA, sikap Indonesia berubah menjadi curiga dan kemudian bermusuhan. Situasi semakin memanas karena orang-orang NICA mempersenjatai bekas tentara KNIL yang baru dibebaskan dari tahanan Jepang. Orang-orang NICA dan KNIL di Jakarta, Surabaya dan Bandung mulai memancing kerusuhan dengan cara melakukan provokasi [2].

II.1.1 Insiden bendera di Surabaya

Di beberapa daerah di Indonesia terjadi perebutan kekuasaan, dengan cara kekerasan maupun dengan jalan perundingan. Di beberapa keresidenan di Jawa, pada bulan September 1945 pemimpin masing-masing menyambut Proklamasi Kemerdekaan dengan menyatakan diri sebagai pemerintah Republik Indonesia dan mengancam bahwa terhadap segala tindakan yang menentang pemerintah RI akan diambil tindakan keras. Pegawai-pegawai Jepang dirumahkan, dilarang memasuki kantor-kantor mereka. Pada tahap selanjutnya para pemuda berusaha merebut kantor-kantor mereka.

Pada tanggal 19 September 1945, ketika perebutan senjata Jepang masih berlangsung, terjadi insiden bendera di Hotel Yamato, Insiden ini pecah ketika

orang-orang Belanda bekas tawanan Jepang menduduki hotel tersebut dibantu oleh serombongan pasukan Inggris yang diterjunkan di Gunungsari, untuk mendirikan markas RAPWI. Orang-orang Belanda tersebut mengibarkan bendera mereka di puncak hotel Yamato. Hal ini memancing kemarahan para pemuda. Hotel tersebut diserbu oleh pemuda, setelah permintaan Residen Sudirman dengan cara baik-baik untuk menurunkan bendera Belanda ditolak oleh penghuni hotel. Bentrokan tidak dapat dihindari. Beberapa orang pemuda berhasil memanjat atap hotel serta menurunkan bendera Belanda yang berkibar di atasnya. Mereka merobek warna birunya dan mengibarkannya kembali sebagai Merah Putih [4].

II.1.2 Pertempuran di Surabaya

Kontak senjata yang terjadi di Surabaya antara pasukan Indonesia dan pasukan Sekutu berkaitan dengan usaha perebutan kekuasaan dan senjata dari tangan Jepang yang dimulai sejak tanggal 2 September 1945. Perebutan tersebut membangkitkan pergolakan, yang kemudian berubah menjadi revolusi yang konfrontatif.

Pada tanggal 25 Oktober 1945, Brigade 49 yang dipimpinan Brigjen A.W.S Mallaby mendarat di Surabaya. Mereka bertugas untuk melucuti serdadu Jepang dan membebaskan para interminan. Kedatangan Mallaby disambut oleh R.M.T.A. Suryo (Gubernur Jawa Timur). Dalam pertemuan itu dihasilkan beberapa kesepakatan sebagai berikut [2].

1. Inggris berjanji bahwa di antara tentara Inggris tidak terdapat angkatan perang Belanda.
2. Disetujui kerja sama antara kedua belah pihak untuk menjamin keamanan dan ketentraman.
3. Akan segera dibentuk *Contact Bureau* (Biro Kontak) agar kerja sama dapat terlaksana sebaik-baiknya.
4. Inggris hanya akan melucuti senjata Jepang.
5. Atas kesepakatan tersebut, maka Inggris diizinkan masuk kota Surabaya. Ternyata dalam praktiknya, Inggris tidak menepati janjinya. Pasukan Inggris justru berusaha mengusasi Surabaya.

Secara kronolgis serangan Inggris terhadap Indonesia, antara lain sebagai berikut [2].

1. Tanggal 26 Oktober 1945, satu peleton *Field Security Section* di bawah pimpinan Kapten Shaw menyerang sebuah penjara di Kalisosok dan juga pusat-pusat penting lainnya, seperti pangkalan udara, kantor pos dan gedung pemerintahan.
2. Tanggal 27 Oktober 1945, pasukan Inggris menyebarkan pamflet-pamflet yang berisi perintah agar rakyat Surabaya dan Jawa Timur menyerahkan senjata hasil rampasan dari Jepang.
3. Tanggal 28 Oktober 1945, pemuda Indonesia menyerang pos-pos Sekutu di seluruh Surbaya.
4. Tanggal 29 Oktober 1945, Komando Sekutu menghubungi Presiden Soekarno untuk menyelematakan pasukan Inggris agar tidak mengalami kehancuran total. Presiden Soekarno dan Jenderal Mallaby mengadakan perundingan. Pertemuan itu menghasilkan dua kesepakatan, yaitu penghentian kontak senjata dan keberadaan RI diakui oleh Inggris.

II.1.3 Kematian Jenderal Mallaby

Presiden Soekarno didampingi oleh Wakil Presiden Drs. Moh. Hatta dan Menteri Penerangan Amir Syarifudin segera berunding dengan Mallaby. Perundingan menghasilkan keputusan yaitu menghentikan kontak senjata. Perundingan dilanjutkan pada malam hari antara Presiden Soekarno, wakil pemerintah RI di Surabaya, wakil pemuda dan pihak Inggris yang didampingi oleh Jenderal Hawthron. Dalam perundingan tersebut didapat kesepakatan yaitu eksistensi RI harus diakui oleh Inggris.

Sementara itu, di beberapa tempat masih terjadi pertempuran, sekalipun sudah diumukan gencatan senjata. Oleh karena itu, para anggota Kontak Biro dari kedua belah pihak mendatangi tempat-tempat tersebut dengan maksud menghentikan pertempuran setempat-setempat itu. Pada pukul 17.00, tanggal 30 Oktober 1945, seluruh anggota Kontak Biro pergi bersama-sama menuju beberapa tempat.

Tempat yang terakhir adalah Gedung Bank Internatio di Jembatan Merah. Gedung ini masih diduduki oleh pasukan Inggris, dan pemuda-pemuda masih mengepungnya. Setibanya di tempat ini terjadi insiden. Pemuda-pemuda menuntut agar pasukan Mallaby menyerah akan tetapi Mallaby tidak dapat menerima tuntutan itu. Tiba-tiba terdengar tembakan gencar dari dalam gedung yang dilakukan oleh pasukan Inggris. Pemuda-pemuda membalas. Ditengah-tengah keributan dan kekacauan itu para anggota Kontak Biro mencari perlindungan sendiri-sendiri. Mallaby menjadi sasaran para pemuda, Dia ditusuk dengan bayonet dan bambu runcing, hingga Mallaby terbunuh.

Kontak Biro mengumumkan bahwa kematian Mallaby adalah akibat kecelakaan, tidak dapat dipastikan apakah akibat tembakan rakyat atau tembakan dari tentaranya sendiri [4].

II.1.4 Pertempuran 10 November 1945

Sesudah kematian Mallaby, pihak Inggris mendatangkan pasukan baru di bawah pimpinan Mayor Jenderal E.C. Mansergh. Kemudian, pada tanggal 7 November 1945, Mansergh menulis surat kepada Gubernur Soeryo berisi tuduhan bahwa Gubernur tidak menguasai keadaaan, seluruh kota telah dikuasai oleh para ekstremis. Mereka menghalang-halangi tugas Inggris. Akhirnya, ia mengancam akan menduduki kota Surabaya untuk melucuti “gerombolan yang tidak mengenal tertib hukum” itu. Ia juga “memanggil” Gubernur Soeryo untuk ”menghadap”.

Dalam surat jawaban tanggal 9 November 1945 Gubernur Soeryo membantah semua tuduhan Mansergh. Ia mengutus Residen Sudirman dan Roeslan Abdulgani untuk menyampaikan suratnya. Kepada utusan pihak itu pihak Inggris menyampaikan dua surat, satu di antaranya berisi ultimatum kepada bangsa Indoneisa di Surabaya. Ultimatum itu bertanggal 9 November 1945. Isi dan maknanya merupakan penghinaan terhadap martabat dan harga diri bangsa Indonesia. Isi pokoknya adalah tuntutan agar semua pemimpin Indonesia, pemimpin pemuda, kepala polisi, dan kepala pemerintahan, harus melapor pada tempat dan waktu yang ditentukan dengan meletakan tangan mereka di atas

kepala, dan kemudian menandatangani dokumen yang disediakan sebagai tanda menyerah tanpa syarat.

Pemuda-pemuda bersenjata diharuskan menyerahkan senjatanya dengan berbaris serta membawa bendera putih. Batas waktu yang ditentukan adalah pukul 06.00 tanggal 10 November 1945. Apabila tidak diindahkan, Inggris akan mengerahkan seluruh kekuatan darat, laut dan udara. Ultimatum ini ditandatangani oleh Mayor Jenderal E.C. Mansergh. Untuk menentukan sikap terhadap ultimatum ini para pemimpin di Surabaya mengadakan pertemuan. Mereka berusaha menghubungi Presiden Soekarno untuk meminta intruksi mengenai tindakan apa yang akan diambil, menerima atau menolak ultimatum. Namun, mereka hanya berhasil mengadakan hubungan dengan Menteri Luar Negeri Mr. Ahmad Subardjo. Menteri Luar Negeri menyerahkan "kita putus" kepada rakyat Surabaya. Secara resmi pada pukul 22.00 Gubernur Soeroyo melalui radio menyatakan menolak ultimatum Inggris.

Sementara itu, para pemuda yang sudah siap menjaga membuat pertahanan di dalam kota. Komandan Pertahanan Kota, Soengkono, pada tanggal 9 November pukul 17.00 mengundang semua unsur kekuatan rakyat yang terdiri dari komandan TKR, PRI, BPRI, Tentara Pelajar, Polisi Istimewa, BBI, PTKR dan TKR Laut untuk berkumpul di Markas Pregolan 4. Soengkono menyatakan bahwa siapa yang ingin meninggaklan kota dipersilahkan. Ternyata, semuanya bertekad untuk mempertahankan Surabaya. Mereka masing-masing membutuhkan tanda tangan pada secarik kertas sebagai tanda setuju, dan diteruskan dengan ikrar bersama. Soengkono dipilih sebagai komandan pertahanan.

Kota Surabaya dibagi dalam tiga sektor pertahanan. Garis pertahanan ditentukan dari Jalan Jakarta, tetapi penempatan pasukannya agar mundur ke belakang ke krembangan, Kapasan dan Kedungcowek. Gedung kedua di sekitar *Viaduct.* Garis ketiga di daerah Darmo.

Pembagian tiga sektor meliputi sektor barat, sektor tengah dan sektor timur. Sektor barat dipimpin oleh Koenkiyat. Sektor tengah oleh Kretarto dan Marhadi, sedangkan sektor timur oleh Kadim Prawirodihardjo.

Sementara itu, radio pemberontakan yang dipimpin oleh Bung Tomo membakar semanagat juang rakyat. Siaran ini dipancarkan dari Jalan Mawar No. 4 Sesudah batas waktu ultimatum habis, keadaan semakin eksplosif. Kontak senjata pertama terjadi di Perak, yang berlangsung sampai pukul 18.00. Inggris berhasil menguasai garis pertama dari pertahanan pihak Indonesia. Gerakan pasukan Inggris disertai dengan pengeboman yang ditujukan pada sasaran yang diperkirakan menjadi temapt pemusatan pemuda. Surabaya yang digempur oleh Inggris itu berhasil dipertahankan oleh para pemuda hampir tiga minggu lamanya. Sektor demi sektor dipertahankan secara gigih walaupun pihak Inggris mempergunakan senjata-senjata modern dan berat. Pertempuran terakhir terjadi di Gunungsari, pada 28 November 1945, tetapi perlawanan secara sporadis masih dilakukan. Setelah itu, Markas Pertahanan Surabya dipindahkan ke desa Lebaniwaras, yang terkenal dengan nama Markas Kali [4].

II.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 [5].

II.2.1 Pengertian Game

Game merupakan sebuah bentuk seni dimana penggunanya disebut dengan pemain (*player*), diharuskan membuat keputusan-keputusan dengan tujuan mengelola sumber daya yang diperoleh dari kesempatan-kesempatan bermain (*token*) miliknya untuk mencapai tujuan tertentu. *Video game* adalah bentuk *game* yang interaksi umunya melibatkan media *video* dan *audio*.

Menurut Andang Ismail terdapat dua pengertian *game* (permainan). Pertama, *game* (permainan) adalah sebuah aktifitas bermain yang murni mencari kesenangan tanpa mencari menang atau kalah. Kedua, permainan diartikan sebagai aktifitas bermain yang dilakukan dalam rangka mencari kesenangan dan kepuasan, namun ditandai pencarian menang – kalah.

Berdasarkan representasinya, *game* dapat dibedakan menjadi 2 jenis yaitu *game* 2 dimensi (2D) dan 3 dimensi (3D). *Game* 2D adalah *game* yang secara matamatis hanya melibatkan 2 elemen koordinat kartesius yaitu x dan y, sehingga konsep kamera pada *game* 2D hanya menentukan gambar pada *game* yang dapat dilihat oleh pemain. Sedangkan *game* 3D adalah *game* yang selain melibatkan elemen x dan y juga melibatkan elemen z pada perhitungannya sehingga konsep kamera pada *game* 3D benar-benar menyerupai konsep kamera pada kehidupan nyata [5].

II.2.2 Sejarah Game

Sejak pertama kali ditemukan sampai saat ini, *game* telah melewati beberapa fase evolusi penting dimana yang berperan besar terhadap perkembangan *game* sudah semakin pesat sekarang ini.

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 bekejar-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 [5].

II.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 [5]. 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-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 permbuatnya 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 diterjemahkan 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].

II.2.4 Jenis-jenis Game

Berikut ini beberapa jenis *game* berdasarkan cara pembuatannya, cara pemasarannya dan mesin yang menjalankannya [6]. 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.

II.2.5 Genre Game

Berdasarkan *genre, game* dapat dibagi menjadi beberapa *genre* yaitu [6] :

1. *Action Game*

Action game dikategorikan sebagai *game play* 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 bias 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*

a) *First Person Shooter Game*

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

b) *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*.

c) *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.

d) *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.*

e) *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.*

f) *Shoot'em Up Game*

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

2. *Adventure Game*

Adventure game dikategorikan sebagai *game play* yang mengharuskan pemain memecahkan bermacam-macam teka-teki melaui 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.*

3. *Role Playing Game*

Role playing game adalah *game* yang memiliki *game play* 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*.

4. *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.

5. *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.

6. *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.

II.2.6 Unsur Game

Dalam sebuah *game* terdapat unsur-unsur yang melengkapinya. Berikut beberapa unsur dalam suatu *game* yaitu [6]:

1. Warna

Warna mempunyai kemampuan untuk membuat orang tanggap terhadap semua yang dilihat karena tidak ada seseuat hal 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 II.1 Warna *Primer* dan *Sekunder* [6]

2. 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 pengkuran 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.

3. Bentuk dasar

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



Gambar II.2 Bentuk Dasar [6]

4. 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.

5. 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.

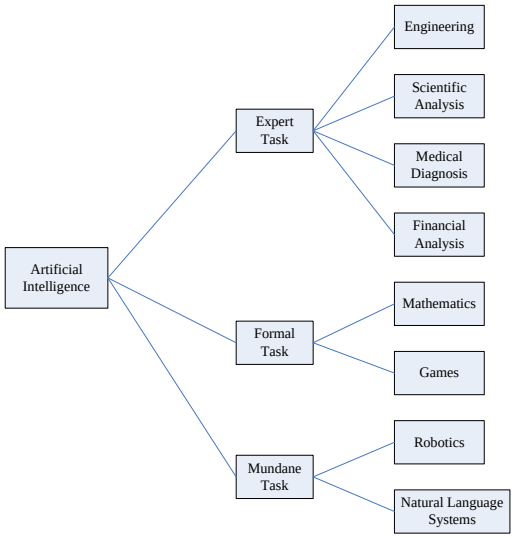
II.3 Kecerdasan Buatan

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 [7]. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritama yang diperlukan untuk mengaplikasikan pengetahuan tersebut serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya.

Berdasarkan sudut pandang, AI dapat dipandang sebagai berikut :

1. Sudut pandang kecerdasan, AI adalah bagaimana membuat mesin yang cerdas dan dapat melakukan hal-hal yang sebelumnya hanya dapat dilakukan manusia.
2. Sudut pandang bisnis, AI adalah sekelompok alat bantu yang berdayaguna dan metodologi yang menggunakan alat-alat bantu tersebut untuk menyelesaikan masalah-masalah bisnis.
3. Sudut pandang pemrograman, AI meliputi studi tentang pemrograman simbolik, pemecahan masalah dan proses pencarian.
4. Sudut pandang penelitian :
   a. Riset tentang AI dimulai pada awal tahun 1960-an, percobaan pertama adalah membuat program permainan catur, membuktikan teori dan *general problem solving*.
   b. AI adalah nama pada akar dari studi area.

Kecerdasan buatan memiliki sejumlah sub disiplin ilmu yang sering digunakan untuk pendekatan yang esensial bagi penyelesaian suatu masalah dan dengan aplikasi bidang AI yang berbeda. Berikut gambar bidang-bidang tugas dari AI yaitu :



Gambar II.3 Bidang-bidang tugas (*task domains*) dari AI [7]

Aplikasi penggunaan AI dapat dibagi ke dalam tiga kelompok yaitu :

1. *Expert Task*

   AI dibentuk berdasarkan pengalaman dan pengetahuan yang dimiliki oleh para ahli. Penggunaan ini dapat membantu para ahli untuk menyampaikan ilmu-ilmu yang dimiliki. Contohnya adalah :

   a. Analisis finansial.
   b. Analisis medikal.
   c. Analisis ilmu pengetahuan.
   d. Rekayasa (desain, pencarian, kegagalan, perencanaan, manufaktur).

2. *Formal Task*

   AI digunakan untuk melakukan tugas-tugas formal yang selama ini manusia biasa lakukan dengan lebih baik. Contohnya adalah :

   a. *Game*.
   b. Matematika (geometri, logika, kalkulus, integral).

3. *Mundane Task*

   Secara harfiah *mundane* adalah keduniaan. AI digunakan untuk melalukan hal-hal yang sifatnya duniawi atau melakukan kegiatan yang dapat membantu manusia. Contohnya adalah :

   a. Persepsi.
   b. Bahasa alami.
   c. Robot control.

Aplikasi kecerdasan buatan memiliki dua bagian utama yaitu :

1. Basis Pengetahuan (*Knowledge Base*) : berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor Inferensi (*Inference Engine*) : kemampuan menarik kesimpulan berdasarikan pengalaman.

II.3.1 Perbandingan Kecerdasan Buatan dengan Kecerdasan Alamiah

Ada beberapa keuntungan dari kecerdasan buatan dibandingkan dengan kecerdasan alamiah yaitu [7] :

1. Lebih permanen.
2. Memberikan kemudahan dalam duplikasi dan penyebaran.
3. Ralatif lebih murah dari kecerdasan alamiah.
4. Konsisten dan teliti.
5. Dapat didokumemtasikan.
6. Dapat mengerjakan beberapa task dengan lebih cepat dan lebih baik disbanding manusia.

II.4 Teknik Dasar Pencarian

Ada beberapa keuntungan dari kecerdasan buatan dibandingkan dengan kecerdasan alamiah yaitu [7] :

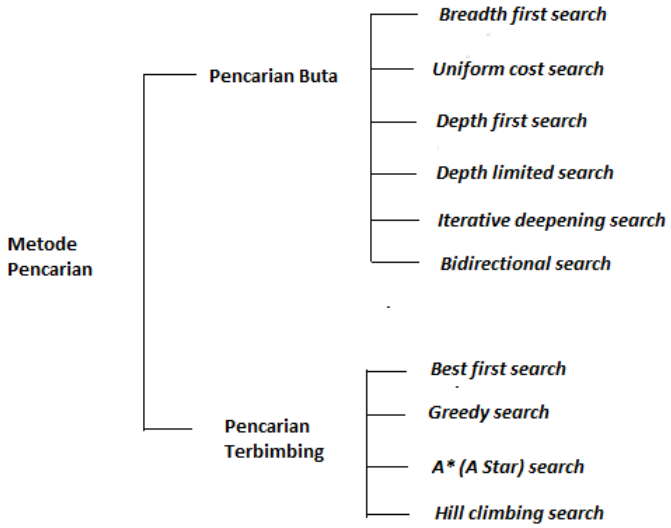
1. Lebih permanen.
2. Memberikan kemudahan dalam duplikasi dan penyebaran.
3. Ralatif lebih murah dari kecerdasan alamiah.
4. Konsisten dan teliti.
5. Dapat didokumemtasikan.
6. Dapat mengerjakan beberapa task dengan lebih cepat dan lebih baik dibanding manusia.

II.5 Algoritma Pencarian

Permasalahan pencarian dapat diselesaikan dengan beberapa metode yaitu [7] :

1. Metode pencarian yang pertama adalah metode sederhana yang hanya berusaha mencari kemungkinan penyelesaian yang disebut juga pencarian buta.
2. Metode yang lebih kompleks yang akan mencari jarak terpendek. Metode ini adalah *British Museum Procedure*, *Branch and Bound, Dynamic Programming, Best First Search, Greedy Search*, A* (A Star) dan *Hill Climbing Search*. Metode-metode ini digunakan pada saat perjalanan untuk mencari kemungkinan menjadi perhitungan.

Metode pencarian sangat penting untuk menyelesaikan permasalahan karena setiap *state* atau keadaan menggambarkan langkah-langkah untuk menyelesaikan permasalahan. Dalam sebuah permainan, metode pencarian akan menentukan apa yang harus dilakukan dimana setiap state menggambarkan kemungkinan posisi pada suatu saat. Metode pencarian adalah bagian dari kesimpulan dimana setiap state menggambarkan hipotesis dalam sebuah rangkaian deduktif.



Gambar II.4 Bagan Metode Pencarian [7]

Pada gambar di atas, metode penulusuran dibagi menjadi dua golongan, yakni pencarian buta *(blind search)* dan pencarian terbimbing *(heuristic search)*.

Beberapa contoh algoritma pencarian yang menggunakan metode *heuristic search* adalah : *Best First Search, Greedy Search,* A* *(A Star) Search*, dan *Hill Climbing Search*.

II.5.1 Pencarian Buta

Pencarian buta (*Blind Search*) adalah pencarian solusi tanpa adanya informasi yang dapat mengarahkan pencarian untuk mencapai *goal state* dari *current state* [7]. Informasi yang ada hanyalah definisi *goal state* itu sendiri sehingga algoritma dapat mengenali *goal state*. Apabila tidak ada informasi maka pencarian buta dalam kerjanya akan memeriksa *node-node* secara tidak terarah

dan kurang efisien untuk kebanyakan kasus karena banyanknya *node* yang dikembangkan. Beberapa contoh algoritma yang termasuk *blind search* adalah *Breadth First Search, Uniform Cost Search, Depth First Search, Depth Limited Search, Interative Deepening Search* dan *Bidirectional Search* [7].

II.5.2 Pencarian Terbimbing

Kata heuristik berasal dari kata kerja bahasa Yunani, heuriskein, yang berarti mencari atau menemukan. Dalam dunia pemograman, sebagian orang menggunakan kata heuristik sebagai lawan kata dari algoritmik, dimana kata heuristik ini diartikan sebagai suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditemukan. Di dalam mempelajari metode – metode pencarian ini, kata heuristik diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi [7].

Pencarian terbimbing (*Heuristic Search*) mempunyai informasi tentang biaya untuk mencapai *goal state* dari *current state*. Pencarian terbimbing dapat melakukan pertimbangan untuk mengembangkan atau memeriksa *node-node* yang mengarah ke *goal state*. Pencarian terbimbing untuk menghitung *cost* ke *goal state* digunakan fungsi *heuristic*. Fungsi *heuristic* berbeda dari pada algoritma dimana *heuristic* lebih merupakan perkiraan untuk membantu algoritma dan tidak harus valid setiap waktu. Beberapa contoh algoritma pencarian yang mengguanakan metode *heuristic search* adalah *Best First Search, Greedy Search*, A* (A Star) dan *Hill Climbing Search* [7].

II.6 AI yang Digunakan

II.6.1 Algoritma A* *(A Star)*

Algoritma A* *(A Star)* merupakan perbaikan dari metode BFS dengan memodifikasi fungsi *heuristic*nya. A* *(A Star)* akan meminimumkan total biaya lintasan. Pada kondisi yang tepat, A* akan memberikan solusi yang terbaik dalam waktu yang optimal. Pencarian menggunakan algoritma A* mempunyai prinsip yang sama dengan algoritma BFS hanya saja dengan dua faktor tambahan yaitu :

1. Setiap sisi mempunyai *cost* yang berbeda-beda, seberapa *cost* untuk pergi dari satu simpul ke simpul lain.
2. *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.

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.

$$f(n) = g(n) + h(n) \quad \text{(II.1)}$$

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 [8].

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.*

1. **Simpul awal** adalah 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:
   a. Carilah simpul n dengan biaya F(n) paling rendah, dalam *open* list. Simpul dengan biaya F terendah kemudian disebut *current node*.
   b. Keluarkan *current node* dari *open list* dan masukkan ke dalam *closed list*.

c. Untuk setiap 8 simpul (*neighbor node*) dari *current node* lakukan langkah berikut:
    1). Jika sudah terdapat dalam *closed list* atau tidak *walkable*, maka abaikan, jika tidak lanjutkan.
    2). 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.
    3). 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.

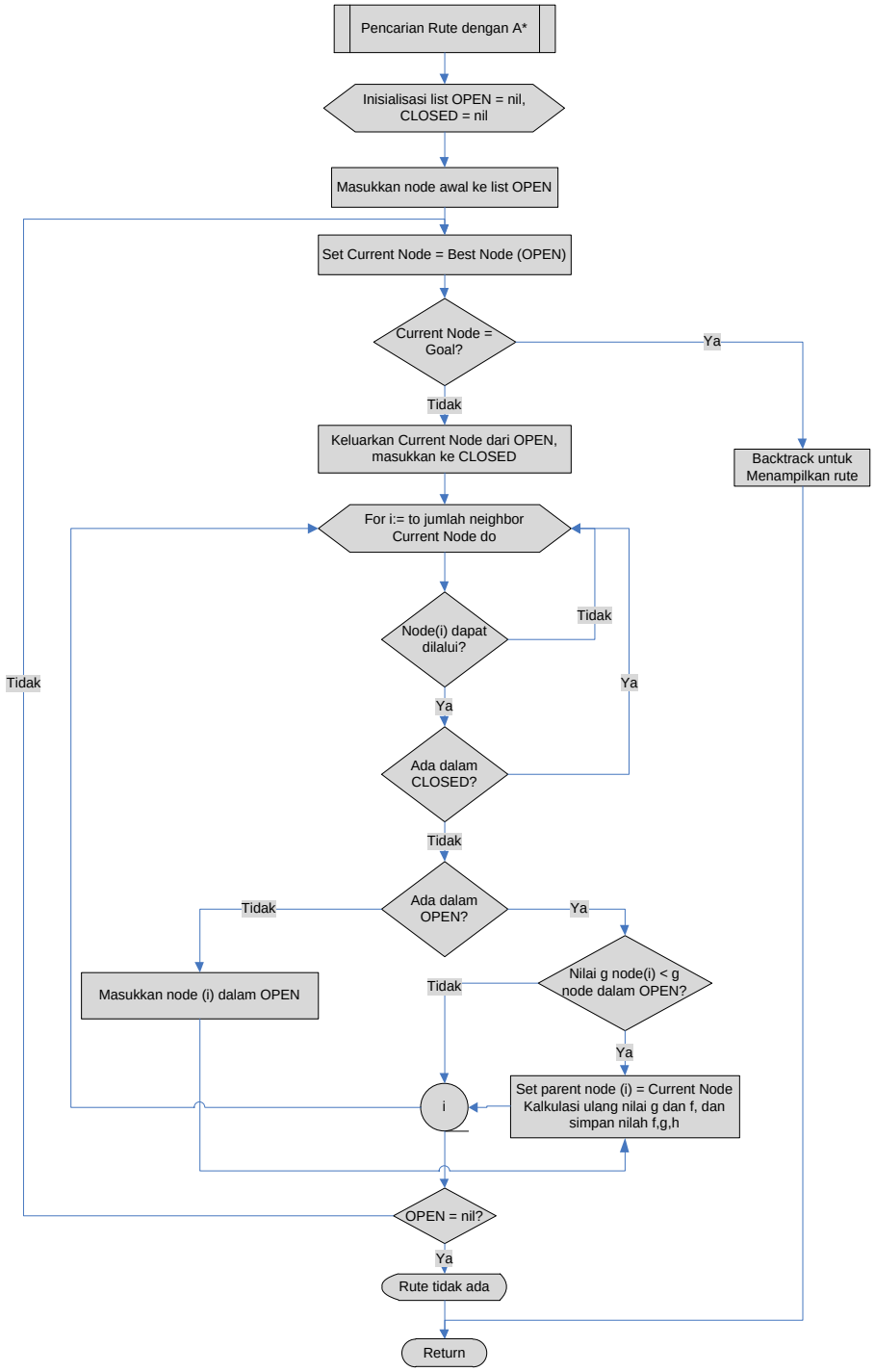
d. Berhenti ketika:
    1. Menambahkan *target point* ke dalam *closed list*, dalam hal ini jalan telah ditemukan, atau,
    2. Gagal untuk menemukan *target point*, dan *open list* kosong. Dalam kasus ini, tidak ada jalan.

e. 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*.

f. 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.

3. 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 II.5 *Flowchart* Algoritma A* [8]

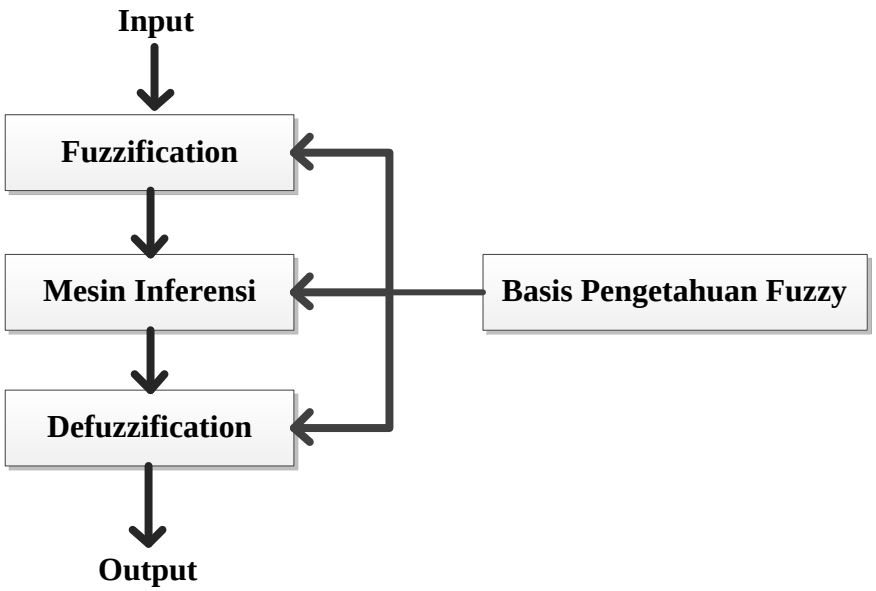
II.6.2 Fuzzy Logic

Konsep tentang logika *fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocol 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* memungkinakan 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) [9].

Konsep himpunan fuzzy memiliki 2 atribut, yaitu [9] :

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 II.6 Struktur sistem *inferensi fuzzy* [9].

Keterangan :

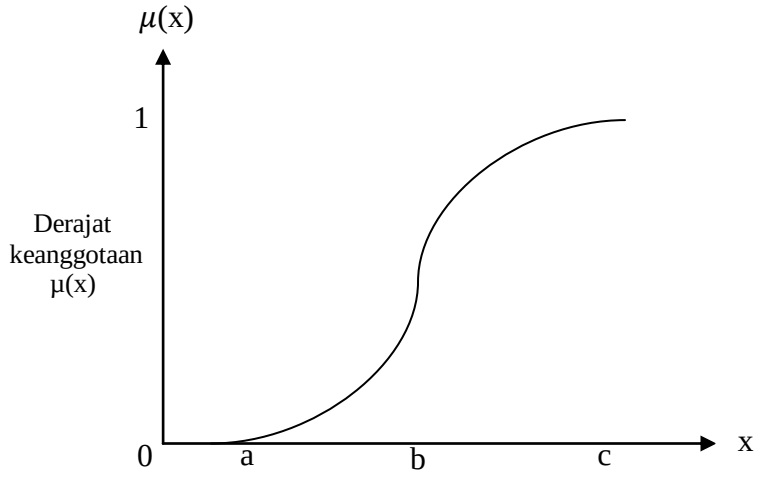
1. Basis Pengetahuan *Fuzzy* merupakan kumpulan *rule-rule fuzzy* dalam bentuk pernyataan IF…THEN.
2. *Fuzzyfikasi* adalah proses untuk mengubah input sistem yang mempunyai nilai tegas menjadi variabel linguistic menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.
3. Mesin Inferensi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

II.6.2.1 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] :

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 :



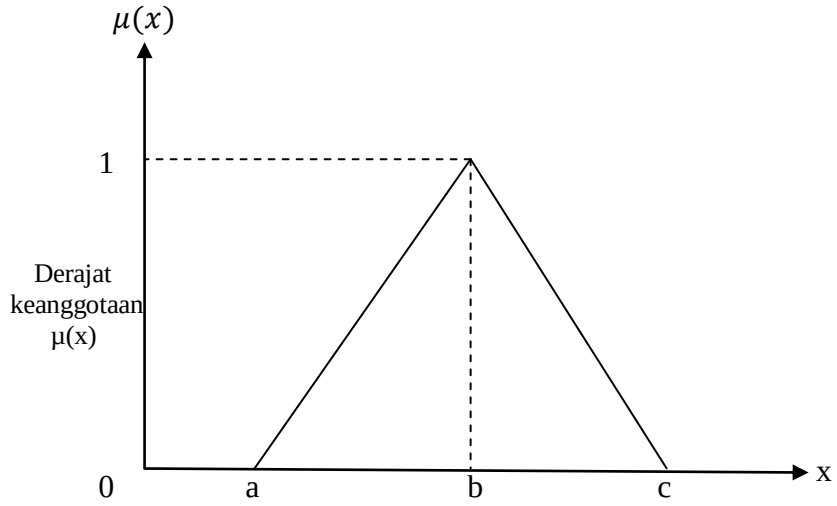
Gambar II.7 Grafik fungsi sigmoid [9].

Fungsi keanggotaan :

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

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 :



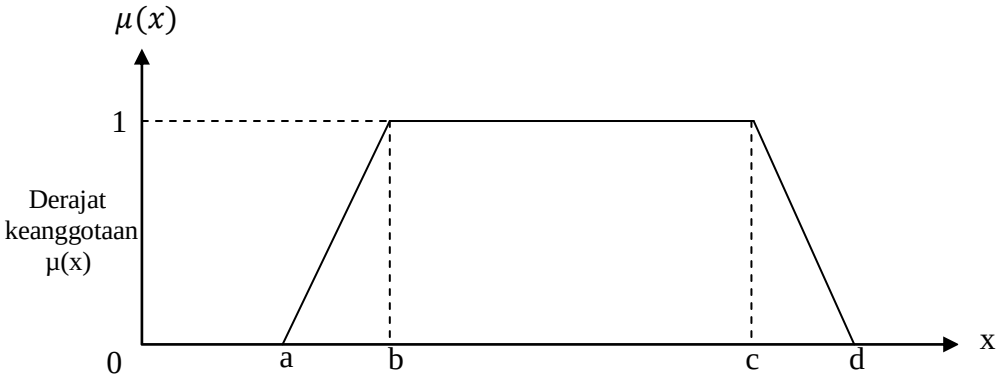
Gambar II.8 Graik fungsi segitiga [9].

Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0;\ x \leq a\ atau\ x\ \geq c \\ (x-a)/(b-a)\,;\, a \leq x \leq b \\ (b-x)/(c-b)\,;\, b \leq x \leq c \end{cases} \quad \text{(II.3)}$$

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 sebagai berikut :



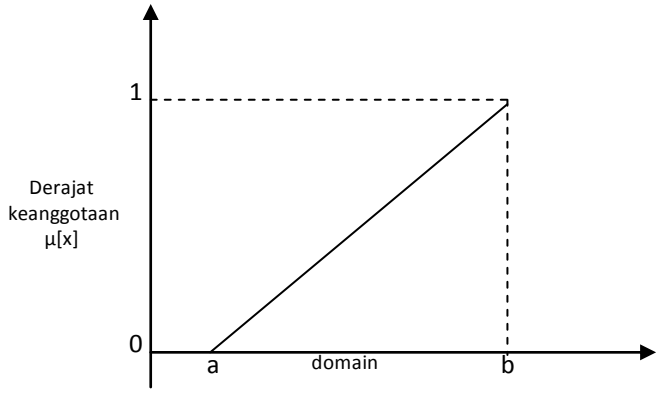
Gambar II.9 Graik fungsi trapesium [9].

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & c \leq x \leq d \end{cases} \quad \text{(II.4)}$$

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

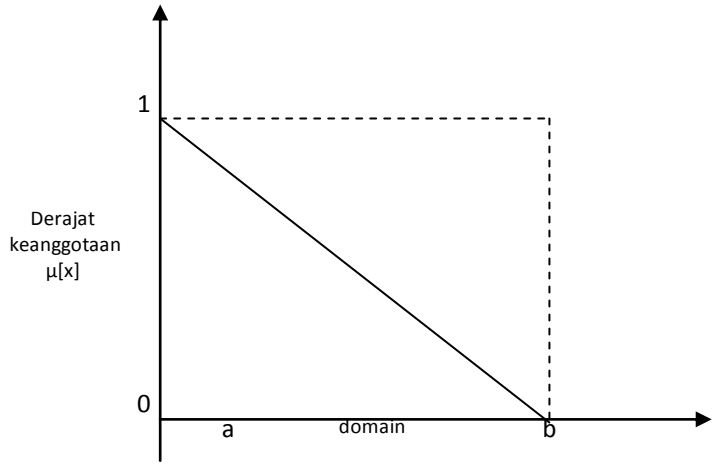


Gambar II.10 Representasi Linear Naik [10].

Fungsi keanggotaan :

$$\mu[x] = \begin{cases} 0; & x < a \\ (x-a)/(b-a); & a < x \leq b \\ 1; & x \geq b \end{cases} \quad \text{(II.5)}$$

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 II.8).



Gambar II.11 Representasi Linear Turun [10]

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \le x < b \\ 0; & x \ge b \end{cases} \quad \text{(II.6)}$$

II.7 OOP (Object Oriented Programming)

Objek adalah kesatuan entitas yang memiliki sifat dan tingkah laku. Dalam kehidupan sehari-hari, objek adalah benda, baik benda berwujud nyata seperti manusia, hewan, mobil, komputer, *handphone*, pena, ataupun benda yang tidak nyata atau konsep, seperti halnya tabungan bank, sistem antrian, sistem internet banking, dan sebagainya. Jadi pengertian OOP adalah konsep yang membagi program menjadi objek-objek yang saling berinteraksi satu sama lain. Objek adalah benda, baik benda yang berwujud nyata maupun benda yang tidak nyata (konsep). Jika kita menggunakan OOP maka akan ada enam keuntungan yang dapat diperoleh, yaitu [11]:

1. Alami (*Natural*).
2. Dapat diandalkan (*Reliable*).
3. Dapat digunakan kembali (*Reusable*).
4. Mudah untuk dalam perawatan (*Maintainable*).
5. Dapat diperluas *(Extendable*).
6. Efisiensi waktu.

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

1. *C++*.
2. *Visual C++*.
3. *Visual Basic*.
4. *Java*.

II.8 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, yaitu :

II.8.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke

sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain [12].

II.8.2 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 menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain [12].

II.8.3 Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring [12].

II.8.4 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].

II.8.5 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].

II.8.6 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki *prefiks* yang sama [12].

II.8.7 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time*, maupun *run time*.

Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain [12].

II.8.8 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal

Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini [12].

II.9 C#

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yangdikembangkan 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].

II.10 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 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*.
2. *Flexible and EasyMoving, rotating,* dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *duplicating, removing*, dan *changing properties.*
3. *Multi Platform Game* bisa di deploy di PC, Mac, Wii, iPhone, iPad dan *browser, android*.
4. *Visual Properties Variables* yang di definisikan dengan *scripts* ditampilkan pada editor. Bisa digeser, di *drag* and *drop*, bisa memilih warna dengan *color picker.*
5. Berbasis .NET, penjalanan program dilakukan dengan *Open Source* .NET *platform, Mono.*

II.11 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 diperuntukan 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].