BAB 2
LANDASAN TEORI

2.1 Keanekaragaman Budaya Indonesia

Indonesia merupakan negara kepulauan dengan kondisi geografisnya yang luas yang memiliki 17.504 pulau besar dan kecil, sekitar 6000 diantaranya tidak berpenghuni yang menyebar di khatulistiwa. Berdasarkan luas daratan wilayah Indonesia terdiri dari 5 pulau besar, diantaranya pulau Jawa seluas 132.107 km2, Sumatera seluas 473.606 km2, Kalimantan seluas 539.460 km2, Sulawesi seluas 189.216 km2, dan Papua seluas 421.981 km2 [1]. Dari 5 pulau besar tersebut tersebar keanekaragaman suku dan seni budaya yang berbeda-beda, yang menjadi kekayaan budaya Indonesia. Keberagaman budaya tersebut dapat terlihat dari suku adat, rumah adat dan hewan khas dari pulau-pulau tersebut, diantaranya yaitu sebagai berikut [9].

1. **Pulau Kalimantan**

   Pulau Kalimantan atau Borneo terkenal dengan julukan "Pulau Seribu Sungai" karena banyaknya sungai yang mengalir di pulau ini. Kalimantan merupakan pulau terbesar di Indonesia dengan luas 539.460 km2. Suku adat yang tersebar di wilayah ini salah satunya adalah suku Dayak, suku mayoritas di Pulau Kalimantan, nama suku yang oleh penduduk pesisir pulau Borneo diberi kepada penghuni pedalaman. Rumah adat dari suku dayak disebut rumah Banjar, yaitu rumah dengan sayap bangunan yang menjorok dari samping kanan dan kiri bangunan utama. Senjata tradisionalnya adalah mandau dan talawang. Hewan khas dari Pulau Kalimantan salah satunya adalah buaya.

2. **Pulau Sumatera**

   Pulau Sumatera merupakan pulau terbesar kedua di Indonesia dengan luas 473.606 km2. Suku adat yang mayoritas tersebar di wilayah ini adalah suku adat Minangkabau atau Minang. Rumah adat dari suku Minang yaitu

rumah Gadang dengan bentuk atap yang runcing menjulang. Senjata tradisionalnya adalah kerambit. Hewan khas dari pulau Sumatera salah satunya adalah harimau Sumatera dan burung Endemik.

3. Pulau Papua

Pulau Papua merupakan pulau terbesar ketiga di Indonesia dengan luas 421.981 km2. Suku adat yang tersebar di wilayah ini salah satunya adalah suku adat Dani, suku dengan populasi terbesar di pulau ini. Suku Dani bermukim atau mendiami wilayah Pegunungan Tengah Papua. Rumah adat dari suku Dani yaitu rumah Honai yang terbuat dari kayu dengan atap berbentuk kerucut yang terbuat dari jerami atau ilalang. Honai sengaja dibangun sempit atau kecil dan tidak berjendela yang bertujuan untuk menahan hawa dingin pegunungan Papua. Senjata tradisionalnya adalah panah dan busur. Hewan khas dari pulau Papua diantaranya adalah kelelawar dan sejenis burung-burungan.

4. Pulau Sulawesi

Pulau Sulawesi adalah surga perairan di Indonesia. Sulawesi merupakan pulau terbesar keempat di Indonesia dengan luas 189.216 km2. Suku adat yang mayoritas tersebar di wilayah ini salah satunya adalah suku adat Bugis yang berdomisili di Sulawesi Selatan. Rumah adat dari suku Bugis yaitu rumah Saoraja (Sallasa) yang berarti rumah besar yang di tempati oleh keturunan raja (kaum bangsawan). Senjata tradisionalnya adalah kawali atau badik bugis. Hewan khas dari pulau Sulawesi diantaranya adalah terumbu karang dan berbagai macam ikan karena pulau Sulawesi terkenal dengan keindahan wisata baharinya.

5. Pulau Jawa

Pulau Jawa merupakan pulau terbesar kelima di Indonesia dengan luas 189.216 km2. Suku adat yang tersebar di wilayah ini salah satunya adalah suku Sunda di Jawa Barat. Rumah adat dari suku Sunda yaitu rumah kasepuhan, yaitu rumah panggung menggunakan atap daun (kiray dan daun

tepus) dengan bilik bambu dan tiang kayu, atau juga bisa berarti harus menggunakan bahan-bahan alami. Senjata tradisionalnya adalah kujang. Hewan khas dari pulau Jawa salah satunya adalah harimau.

2.2 Sangkuriang

Cerita Sangkuriang merupakan cerita rakyat (folklore) Tatar Sunda bergenre dongeng. Menurut golongannya termasuk ke dalam jenis legenda, yaitu cerita yang mengisahkan asal-usul terjadinya suatu tempat, binatang, tumbuh-tumbuhan, dan lain-lain. Cerita Sangkuriang adalah legenda yang mengisahkan asal-usul terciptanya Gunung Tangkuban Perahu yang menceritakan seorang anak bernama Sangkuriang yang melakukan segala cara untuk menikahi ibunya sendiri.

Menurut tokoh pendidik dan pemerhati anak, DR. Seto Mulyadi, mengatakan, cerita rakyat merupakan media yang bermanfaat untuk memperkenalkan nilai-nilai luhur kepada anak. Pengkontrasan tokoh baik dan jahat perlu dilakukan agar terlihat perbedaan keduanya. Sehingga cerita rakyat tidak diartikan berbeda oleh anak seperti Sangkuriang yang melakukan segala cara untuk menikahi ibunya sendiri.

Legenda ini berawal dari kisah seorang puteri raja di Jawa Barat bernama Dayang Sumbi. Ia mempunyai seorang anak laki-laki yang diberi nama Sangkuriang. Anak tersebut sangat gemar berburu. Ia berburu dengan ditemani oleh Tumang, anjing kesayangan istana. Sangkuriang tidak tahu, bahwa anjing itu adalah titisan dewa dan juga bapaknya. Pada suatu hari Tumang tidak mau mengikuti perintahnya untuk mengejar hewan buruan. Maka anjing tersebut disembelih dan diambil hatinya. Ketika kembali ke rumah, Sangkuriang menceritakan kejadian itu pada ibunya. Bukan main marahnya Dayang Sumbi begitu mendengar cerita itu. Tanpa sengaja ia memukul kepala Sangkuriang dengan sendok nasi yang dipegangnya. Sangkuriang terluka. Ia sangat kecewa dan pergi mengembara. Setelah kejadian itu, Dayang Sumbi sangat menyesali dirinya. Ia selalu berdoa dan sangat tekun bertapa. Pada suatu ketika, para dewa memberinya sebuah hadiah. Ia akan selamanya muda dan memiliki kecantikan abadi.

Setelah bertahun-tahun mengembara, Sangkuriang akhirnya berniat untuk kembali ke tanah airnya. Sesampainya disana, tanah kelahirannya sudah berubah total. Disana dijumpainya seorang gadis jelita, yang tak lain adalah Dayang Sumbi. Terpesona oleh kecantikan wanita tersebut maka, Sangkuriang melamarnya. Oleh karena pemuda itu sangat tampan, Dayang Sumbi pun sangat terpesona padanya.

Pada suatu hari Sangkuriang minta pamit untuk berburu. Ia minta tolong Dayang Sumbi untuk merapikan ikat kepalanya. Alangkah terkejutnya Dayang Sumbi demi melihat bekas luka di kepala calon suaminya. Luka itu persis seperti luka anaknya yang telah pergi merantau. Setelah lama diperhatikannya, ternyata wajah pemuda itu sangat mirip dengan wajah anaknya. Ia menjadi sangat ketakutan.

Maka kemudian ia mencari daya upaya untuk menggagalkan proses peminangan itu. Ia mengajukan dua buah syarat. Pertama, ia meminta pemuda itu untuk membendung sungai Citarum. Dan kedua, ia minta Sangkuriang untuk membuat sebuah perahu besar untuk menyeberang sungai itu. Kedua syarat itu harus sudah dipenuhi sebelum fajar menyingsing.

Malam itu Sangkuriang melakukan tapa. Dengan kesaktiannya ia menyelesaikan pekerjaan itu. Dayang Sumbi pun diam-diam mengintip pekerjaan tersebut. Begitu pekerjaan itu hampir selesai, Dayang Sumbi memerintahkan pasukannya untuk menggelar kain sutra merah di sebelah timur kota.

Ketika menyaksikan warna memerah di timur kota, Sangkuriang mengira hari sudah menjelang pagi. Ia pun menghentikan pekerjaannya. Ia sangat marah oleh karena itu berarti ia tidak dapat memenuhi syarat yang diminta Dayang Sumbi. Dengan kekuatannya, ia menjebol bendungan yang dibuatnya. Terjadilah banjir besar melanda seluruh kota. Ia pun kemudian menendang perahu besar yang dibuatnya. Perahu itu melayang dan jatuh menjadi sebuah gunung yang bernama "Tangkuban Perahu."

Pada awalnya, cerita yang tidak pernah diketahui pengarangnya ini, hidup di tengah-tengah masyarakat Sunda sebagai tradisi lisan yang diwariskan turun temurun dari generasi ke generasi. Hal inilah yang mengundang perdebatan tentang keberadaan cerita Sangkuriang dalam khasanah sastra. Seperti yang kita ketahui beberapa kalangan masih terjebak pada pengertian sastra sebagai karya imajiner yang tercetak dalam bentuk tulisan.

Akan tetapi, setelah cerita Sangkuriang ditransformasi ke dalam bentuk tulisan kiranya keberadaan cerita Sangkuriang dalam dunia sastra sudah tidak perlu dipersoalkan lagi. Berbagai bentuk dan versi cerita yang hadir di tengah-tengah masyarakat merupakan bukti diterimanya cerita legenda Sangkuriang dalam kesusastraan. Tidak hanya terbatas dalam bingkai kedaerahan (Sunda) tetapi cerita legenda Sangkuriang kini sudah menjadi milik Indonesia **[10]**.

2.3 *Game*

Dalam bagian ini akan dibahas pengertian *game*, klasifikasi *game* berdasarkan *platform*, berdasarkan *genre*, berdasarkan sudut pandang permainan (*game view point*) dan berdasarkan pembatasan usia dari pemainnya.

2.3.1 Pengertian *Game*

Game merupakan suatu bentuk hiburan yang seringkali dijadikan sebagai penyegar pikiran dari rasa penat yang disebabkan oleh aktivitas dan rutinitas kita, bertujuan bertujuan untuk menghibur yang banyak disukai oleh anak-anak hingga orang dewasa.

Istilah *"video game"* awalnya mengacu pada jenis spesifik perangkat, yaitu sebuah sistem komputer yang menciptakan sinyal tampilan *video* untuk televisi, tapi kini istilah tersebut telah menjadi istilah yang mencakup *sega*la jenis perangkat.

Video game dikendalikan oleh komputer melalui interaksi pengguna dan imersi audio-visual, *video game* memungkinkan pemain untuk menanggapi situasi yang menantang dalam dunia fantasi. Sangat penting untuk dicatat bahwa "fantasi" merujuk pada fakta bahwa dunia yang dimaksud adalah maya, dan tidak

nyata, misalnya pemain dapat menjelajahi fantasi bermain di liga utama bisbol **[11]**.

2.3.2 Sejarah Singkat *Game*

Di antara jajaran nama tokoh penemuan penting, nama besar seorang penemu permainan komputer, atau lebih lazim kita menyebutnya dengan *game*, jarang sekali disebutkan. Namun bila penemuannya sampai sekarang masih digunakan dan bahkan sangat diminati dan dinikmati. Sebagian orang bahkan saat ini mengangap *game* sebagai sebuah kebutuhan yang harus terpenuhi.

Game elektronik atau selanjutnya dapat disebut *video game* pertama sekali ditemukan oleh **Thomas T. Goldsmith** dan **Estle Ray Mann**. Penemuan ini dipatenkan pada Januari 1947, yang mendasari perkembangannya saat itu adalah ketika mereka menemukan *Cathode-Ray Tube* sebuah tabung digunakan sebagai media untuk membuat simulasi kecepatan tembakan dan arah tembakan sebuah roket. Pada Februari 1951, **Christopher Strachey** memulai perkembangannya kearah pemrograman yang mulai menggunakan memori dimana aplikasinya diterapkan untuk kebutuhan para pilot. Dan penemuan baru terus berkembang hingga tahun 1959.

Namun era perkembangan konsol *game* di mulai pada generasi pertama tahun 1972 sampai tahun 1977. Generasi kedua (1976-1982) sering disebut generasi konsol 8 bit. Di mana pemrograman *video game* dibuat lebih *advance*. Generasi Ketiga (1983-1992), *Nintendo Entertainment System* (NES) mendominasi pasarnya hingga Amerika Utara, *Sega Master System* juga berkembang pesat di Eropa dan Brasil. Generasi Keempat (1987-1996) lebih dikenal dengan generasi 16 Bit dimana pengembangannya dimulai sekitar Oktober 1987. **Generasi Kelima (1993-2002),** konsol 64 bit mulai muncul pada era ini, grafis 3D mulai menjadi alasan untuk perang konsol ini. **Generasi Keenam (1998-2006)** merupakan generasi 128 bit, teknologi semakin maju membuat beberapa konsol baru mulai muncul dan semuanya memiliki pasar tersendiri. **Generasi Ketujuh (2004)** beberapa pengembang konsol mulai focus dengan pengembangan saja. Mulai tahun 2004 hanya beberapa perusahaan konsol

yang masih bertahan dengan pengembangannya *Nintendo*, *Sony* dan *Microsoft* **[12]**.

2.3.3 Sejarah *Game* di Indonesia

Divinekids adalah *game* yang diakui dan mendapat award resmi dari MURI (M*use*um Rekor Indonesia) sebagai *Game* Pertama Indonesia pada tahun 2004 dan banyak majalah-majalah *Game* dan Komputer (*Game*station, PC*GAME*, Bobo, INDOPOS, *Game*21 dan lain sebagainya). Sebenarnya sudah banyak *game-game* buatan orang-orang indonesia yang lain tetapi karena kekurangan syarat-syarat minimal tertentu seperti: Karakter menggunakan karakter asing, penyebaran yang kurang luas, bahasa yang digunakan bukan bahasa Indonesia, tidak terdaftar resmi Copyright ataupun Trademarknya dan lain sebagainya, karena itu tidak dipilih sebagai *Game* Pertama Indonesia Resmi.

Menurut sebagian pengamat *game* ini adalah kurangnya investigasi MURI dalam hal melihat adanya *game-game* tersebut sebelum memberikan gelar *game* pertama ke David Setiabudi. Terlebih mereka tidak meminta pendapat dari pihak yang memahami dunia *game* Indonesia itu sendiri.

Pengamat-pengamat yang berpendapat berbeda dengan MURI dan majalah-majalah PC dan *Game*, memposisikan sejarah *game* Indonesia sebagai berikut: Sejarah *Game* Indonesia menurut *Game* Devid. Selain itu juga masih ada banyak *game-game* lain yang belum terdokumentasi dengan baik. Beberapa *game* buatan Elex Media Komputindo dengan copyright resmi dan penyebaran yang luas ada juga yang dibuat sebelum sebelum dirilisnya Divine Kids. *Game* ini dibuat dan dibagikan secara gratis melalui CD majalah2 *game*/ komputer dan melalui Internet.

Game-game Divinekids berjumlah 38 lebih dan terus bertambah (2006). Pembuatnya adalah David Setiabudi dan mahasiswa-mahasiswi bimbingannya. Pembuatan musik *game*nya dibuat oleh Rama Aurora, tunanetra berbakat membuat musik digital.

2.3.4 Klasifikasi *Game*

Game bisa diklasifikasikan kedalam beberapa hal, diantaranya klasifikasi berdasarkan *platform*, berdasarkan *genre*, berdasarkan sudut pandang permainan (*game view point*) dan berdasarkan pembatasan usia dari pemainnya.

2.3.4.1 *Berdasarkan Platform*

Tiga kategori utama *platform* pada *game* **[11]** adalah personal computer (PC), console, dan mobile.

a. *Personal Computer (PC)*

   PC game adalah *game* yang dibuat untuk computer baik berbasis Windows, Mac, ataupun Linux. *PC* menyediakan kekuatan grafis dan pemrosesan yang kuat yang memungkinkan pengembang untuk membuat *game* yang mutakhir. Tetapi kelemahan *game PC* adalah mahal karena pengguna harus menggunakan *hardware* yang *up-to-date* untuk memainkan *game PC* dengan baik. Selain itu, kelemahan *game PC* bagi pengembang adalah banyaknya varian dari konfigurasi *PC* membuat sulit bagi pengembang untuk memastikan *game* tersebut berjalan dengan benar pada semua *setup PC*.

b. *Console*

   Console adalah *hardware* yang dibuat oleh pihak ketiga seperti *Sony*, *Microsoft*, dan *Nintendo*. *Console* terhubung ke televisi dan tujuan utamanya adalah untuk bermain *game*. *Game Console* sangat menarik bagi *game* pengembang karena mereka hanya perlu memikirkan satu konfigurasi *hardware* ketika membuat *software* untuk konsol. Sangat kontras dengan *PC* yang memiliki opsi konfigurasi yang tak terbatas.

c. *Mobile*

   Mobile platform terdiri atas sesuatu yang *portable* dan bisa digenggam, termasuk ponsel, PDA, iPods, dan *handheld game* seperti *Nintendo* DSi atau *Sony* PSP. *Game mobile* memiliki kontrol yang sederhana (terutama jika dibandingkan dengan *PC*).

2.3.4.2 *Berdasarkan Genre*

Menurut buku "Fundamentals of *Game* Design", *genre game* bisa dibagi menjadi 9 **[13]**, yaitu:

1. *Action*

   Action game adalah *game* dimana kebanyakan dari tantangan yang disajikan merupakan dari tes *physical skill* dan koordinasi pemain. Salah satu sub-*genre action game* adalah *shooters game*, baik yang 2D maupun 3D seperti *First Person Shooters* (FPS).

2. *Strategy*

   Strategy *game* menantang pemain untuk mencapai kemenangan dengan *perencanaan*, khususnya melalui perencanaan serangkaian tindakan yang dilakukan melawan satu lawan atau lebih. Kemenangan diraih dengan perencanaan matang dan pengambilan keputusan yang optimal.

3. *Role Playing Game (RPG)*

   RPG adalah *game* dimana pemain mengontrol satu atau lebih karakter yang biasanya di desain oleh pemain itu sendiri, dan memandu mereka melewati berbagai rintangan yang diatur oleh komputer. Perkembangan karakter dalam hal kekuatan dan kemampuannya adalah kunci dari *game* jenis ini.

4. *Sports*

   Sports *game* mensimulasikan berbagai aspek dari olahraga atletik nyata maupun imajiner, apakah itu memainkan pertandingan, me-manage tim dan karir, atau keduanya. Salah satu contoh *game* jenis ini adalah Pro Evolution Soccer 2012 (PES 2012), dimana pemain bisa memainkan pertandingan, menjadi manajer tim, maupun menjadi pemain dan mengembangkan karirnya sendiri.

5. *Vechicle Simulation*

   Vechicle *simulation* membuat feeling mengendarai kendaraan, baik kendaraan nyata maupun kendaraan imajiner. Performa dan karakteristik

kasar mesin harus menyerupai kenyataan, kecuali jika yang didesain adalah kendaraan imajiner.

6. *Construction and Management Simulation*

   CMS *game* adalah *game* tentang proses. Tujuan pemain bukan untuk mengalahkan musuh, tetapi membangun sesuatu dengan konteks proses yang sedang berjalan. Semakin pemain mengerti dan mengontrol proses, semakin sukses sesuatu yang ia bangun. *Game* seperti ini biasanya menyediakan dua jenis permainan, yaitu mode bebas dimana pemain bebas membangun sesuatu, dan mode misi dimana terdapat skenario hal apa yang harus dibangun oleh pemain.

7. *Adventure/ Platform*

   Adventure atau *platform game* adalah cerita interaktif tentang karakter protagonis yang dimainkan oleh pemain. Penyampaian cerita dan eksplorasi adalah elemen inti dari *game* ini. Penyelesaian teka-teki dan tantangan konseptual adalah bagian besar dari permainan.

8. *Artificial Life and Puzzle Game*

   Artificial Life *game* adalah *game* yang membuat tiruan dari kehidupan sebenarnya. Biasanya ada dua jenis *game* ini, tiruan kehidupan manusia, contohnya The SIMS, dan tiruan kehidupan binatang, contohnya Tamagochi.

9. *Online Game*

   Istilah *online game* disini mengacu kepada *multiplayer game* dimana mesin dari para pemain terhubung dengan jaringan.

2.3.4.3 *Berdasarkan Sudut Pandang Permainan (Game View Point)*

Sebuah *game* biasanya mempunyai sudut pandang permainan tersendiri disesuaikan berdasarkan *genre game* yang diambil. Berikut beberapa macam sudut pandang permainan yang biasa digunakan **[13]**:

1. ***Side Scrolling***

   Sudut pandang permainan yang terlihat dari samping dan memungkinkan karakter utama untuk bergerak dari kiri ke kanan serta memungkinkan *background* pada *game* seolah-olah bergeser mengikuti pergerakan karakter utama.

2. ***Top Down***

   Sudut pandang permainan yang memungkinkan karakter utama bermanuver ke empat arah namun cara permainannya sendiri bergeser dari bawah ke atas, dan biasanya *game* yang menggunakan sudut pandang permainan jenis ini adalah shooter *game*.

3. ***Isometric***

   Sudut pandang permainan yang memungkinkan permainan terlihat diantara sisi Side Scrolling dan juga Top Down, dan biasanya diterapkan pada *game* dengan *genre* RTS (Real Time Strategy).

2.3.4.4 *Berdasarkan pembatasan usia dari pemainnya*

Menurut ESRB (Entertainment *Software* Rating Board) ada 7 pembatasan usia dari pemainnya **[14]** adalah sebagai berikut:

1. ***Early Childhood (3+)*** memiliki isi yang mungkin cocok untuk usia 3 keatas. Tidak mengandung materi yang tidak pantas seperti adegan kekerasan.
2. ***Everyone (6+)*** memiliki konten yang mungkin cocok untuk usia 6 keatas. Judul-judul dalam kategori ini mungkin berisi sedikit kartun, fantasi atau kekerasan ringan dan tidak jarang menggunakan bahasa yang ringan.
3. ***Everyone (10+)*** memiliki konten yang mungkin cocok untuk usia 10 keatas. Judul-judul dalam kategori ini mungkin berisi lebih banyak kartun, fantasi atau kekerasan ringan, bahasa ringan dan juga minimal tema sugestif.
4. ***Teen (13+)*** memiliki konten yang mungkin cocok untuk usia 13 keatas. Judul-judul dalam kategori ini mungkin mengandung

kekerasan, tema sugestif, humor kasar, sedikit darah, simulasi perjudian, dan tidak jarang menggunakan bahasa yang kuat.

5. ***Mature (17+)*** memiliki konten yang mungkin cocok untuk orang usia 17 dan lebih tua. Judul-judul dalam kategori ini mungkin berisi intens kekerasan, darah, konten seksual dan bahasa yang kuat. Mature (17+) memiliki konten yang mungkin cocok untuk orang usia 17 dan lebih tua. Judul judul dalam kategori ini mungkin berisi intens kekerasan, darah, konten seksual dan bahasa yang kuat.
6. ***Adults Only (18+)*** memiliki konten yang hanya boleh dimainkan oleh orang-orang 18 tahun keatas. Judul-judul dalam kategori ini mungkin termasuk adegan yang intens, kekerasan dan juga mengandung konten seksual.
7. ***RP (Rating Pending)*** telah disampaikan kepada ESRB dan sedang menunggu penilaian akhir. (Simbol ini hanya muncul dalam iklan sebelum permainan dibebaskan).

2.4 Kecerdasan Buatan / AI (*Artificial Intelligence*)

Kecerdasan buatan atau dalam bahasa Inggrisnya *Artificial Intelligence* sering disingkat AI merupakan cabang terpenting dalam dunia komputer. Akhir-akhir ini teknologi Ai telah begitu banyak mempengaruhi kehidupan manusia.

AI ialah ilmu dan rekayasa, yang membuat mesin mempunyai intelegensi tertentu khususnya program komputer yang ‘cerdas’. Intelegensi merupakan bagian kemampuan komputasi untuk mencapai tujuan di dalam dunia. Ada beberapa macam jenis dan derajat intelegensi untuk manusia, binatang dan beberapa mesin.

AI berhubungan dengan penggunaan komputer untuk melaksanakan tugas atau menyelesaikan masalah yang memerluakan kecerdasan atau kepintaran. Pada umumnya komputer memerlukan aplikasi atau *software* untuk melaksanakan tugas atau menyelesaikan masalah, maka aplikasi atau *software* yang berdasarkan teknologi AI diperlukan. Intelegensi perlu dilihat dari berbagai sudut pandang,

termasuk pemahaman, inferensi, daya penjelasan, pembelajaran, komunikasi dan sebagainya **[15]**.

2.4.1 Tujuan akhir AI

Terdapat sembilan tujuan AI **[16]** yaitu:

1. Memahami kognisi manusia. Bagaimana manusia menyelesaikan masalah? Coba untuk mendapatkan pengetahuan ingatan manusia yang mendalam, kemampuan *problem-solving*, belajar, membuat keputusan, dan lain-lain.
2. Otomatisasi biaya-efektif. Menggantikan manusia dalam tugas-tugas intelegensi.
3. Penguatan intelegensi biaya-efektif. Membangun sistem untuk membantu manusia berpikir lebih baik, lebih cepat, lebih dalam.
4. Intelegensi manusia *super*. Membangun program yang mempunyai kemampuan untuk melebihi intelegensi manusia.
5. Problem-*solving* umum. Sistem penyelesaian berbagai masalah yang luas.
6. Wacana koheren. Komunikasi dengan manusia mengguanakan bahasa, alami.
7. Otonomi. Mempunyai sistem intelegensi yang beraksi atas inisiatif sendiri.
8. Belajar (induksi). Sistem sebaiknya dapat untuk memperoleh data sendiri, dan tau bagaimana memperolehnya.
9. Informasi. Simpan informasi dan mengetahui bagaimana untuk mengambil informasi.

2.4.2 Karakteristik AI

Karakteristik ini didasarkan pada pandangan bahwa AI ada 4 kategori **[15]** yaitu:

a. Sistem yang dapat berpikir seperti manusia.
b. Sistem yang dapat beraksi seperti manusia.
c. Sistem yang dapat berpikir secara rasional.

d. Sistem yang dapat beraksi secara rasional.

2.4.3 Perkembangan AI dan Aplikasinya

AI merujuk pada satu bidang komputer yang berupaya melakukan tugas sama seperti manusia. Dalam keadaan ini, komputer tersebut boleh dikatakan cerdas. Banyak bidang yang melibatkan AI antara lain. Penyelesaian masalah, logika kabur (*fuzzy*), pengenalan bahasa alamiah, robotik, dan sistem pakar. Ai boleh di identifikasikan sebagai kajian yang menyeluruh bagaimana sebuah komputer berupaya melakukan tugas dengan terapan yang lebih baik jika di bandingkan manusia. Kajian tersebut meliputi kebolehan komouter belajar, memahami, membuat keputusan, dan meningkatkan prestasi atas hasil kesalahan yang telah dilakukan sebelumnya **[15].**

2.4.4 Metode Pencarian (*Searching*)

Pada dasarnya teknik pencarian (*searching*) dapat dibagi menjadi 2 kelompok besar, yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*). Untuk mengukur performa metode pencarian, terdapat empat kriteria yang dapat dilakukan **[16]** yaitu :

a. *Completeness* : apakah metode tersebut menjamin adanya solusi jika solusinya ada?
b. *Time Complexity* : berapa lama waktu yang diberikan untuk menemukan solusi tersebut.
c. *Space complexity* : berapa banyak memori yang dibutuhkan untuk menemukan solusi tersebut.
d. *Optimaly* : apakah metode tersebut menjamin menemukan solusi terbaik jika terdapat beberapa solusi yang berbeda.

2.4.5 AI yang digunakan

Algoritma A* (A *Star*) merupakan algoritma *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best First Search*. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. Dalam notasi matematika dituliskan sebagai berikut :

$$f(n) = g(n) + h(n) \ldots\ldots \textbf{(2.1)}$$

Dengan perhitungan biaya seperti ini, algoritma A* (A *Star*) adalah *complete* dan optimal.

Sama dengan algoritma dasar *Best First Search*, algoritma A* (A *Star*) ini juga menggunakan *OPEN* dan *CLOSED*. Terdapat tiga kondisi bagi setiap suksesor yang dibangkitkan, yaitu : sudah berada di-*OPEN*, sudah berada di-*CLOSED*, dan tidak berada di-*OPEN* maupun *CLOSED*.

Pada ketiga kondisi tersebut diberikan penanganan yang berbeda-beda. Jika suksesor sudah berada di-*OPEN*, maka dilakukan pengecekan apakah perlu pengubahan *parent* atau tidak tergantung pada nilai g nya melalui *parent* lama atau *parent* baru. Jika melalui *parent* baru memberikan nilai g yang lebih kecil, maka dilakukan pengubahan *parent*. Jika pengubahan *parent* dilakukan, maka dilakukan pula perbaruan (*update*) nilai g dan f pada suksesor tersebut. Dengan perbaruan ini, suksesor tersebut memiliki kesempatan yang lebih besar untuk terpilih sebagai simpul terbaik (*best node*).

Jika suksesor sudah pernah berada di-CLOSED, maka akan dilakukan pengecekan apakah perlu pengubahan parent atau tidak. Jika ya, maka dilakukan perbaruan nilai g dan f pada suksesor tersebut serta pada semua "anak cucunya" yang sudah pernah berada di-OPEN. Dengan perbaruan ini, maka anak cucunya tersebut memiliki kesempatan lebih besar untuk terpilih sebagai simpul terbaik (best *node*).

Jika suksesor tidak berada di-OPEN maupun CLOSED, maka suksesor tersebut dimasukkan ke dalam OPEN. Tambahkan suksesor tersebut sebagai suksesornya best *node*. Hitung biaya suksesor tersebut dengan rumus f = g + h.

Beberapa fungsi *heuristic* yang di yang umumnya di gunakan dalam algoritma A* **[16]** adalah sebagai berikut:

1. *Manhattan Dinstance*

   Manhattan Distance adalah fungsi standar untuk algoritma A*. Digunakan untuk aplikasi yang memiliki 4 arah gerakan (tidak dapat bergerak diagonal)

$$h(n)= abs(Xn-Xgoal) + abs (Yn-Ygoal) \ldots\ldots \textbf{(2.2)}$$

dimana:

a. Xn adalah koordinat x dari node pertama pada grid.
b. Xgoal adalah koordinat X pada final node.
c. Yn adalah koordinat dari node pertama pada grid.
d. Ygoal adalah koordinat Y pada final node.

2. Straight Line *Distance*

Straight Line *Distance* adalah fungsi heuristik yang digunakan pada aplikasi yang dapat bergerak ke segala arah/sudut.

$$h(n)= sqrt ((Xn-Xgoal)^2 + abs (Yn-Ygoal)^2) \ldots\ldots \textbf{(2.3)}$$

dimana:

a. Xn adalah koordinat x dari node pertama pada grid.
b. Xgoal adalah koordinat X pada final node
c. Yn adalah koordinat dari node pertama pada grid.
d. Ygoal adalah koordinat Y pada final node.

3. Diagonal *Distance*

Diagonal *Distance* adalah fungsi heuristik yang digunakan pada aplikasi yang yang memiliki delapan arah gerakan (dapat bergerak diagonal).

$$h(n)= d* maxabs (Xn-Xgoal), abs (Yn-Ygoal)) \ldots\ldots \textbf{(2.4)}$$

dimana:

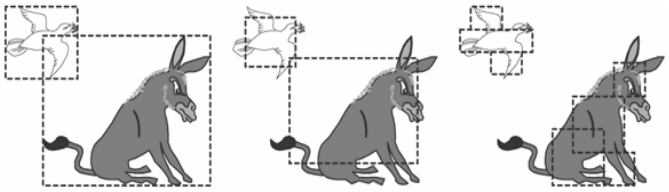
a. d adalah nilai biaya. Nilai d di dapat antara nilai minimum cost di pindah antar node
b. Xn adalah koordinat x dari node pertama pada grid.
c. Xgoal adalah koordinat X pada final node
d. Yn adalah koordinat dari node pertama pada grid.
e. Ygoal adalah koordinat Y pada final node.

2.5 Metode *Collision detection*

Hampir setiap permainan *video* yang ada sudah menerapkan *collision detection* (deteksi tabrakan), baik itu dalam hal tabrakan antara sprite dengan

sprite maupun antara sprite dengan peluru dan lain-lain. Dalam aplikasi permainan ada beberapa reaksi yang ditunjukkan dari *collision detection*, yaitu reaksi dari spirte yang berupa berkurangnya nyawa, berkurangnya darah, meledak, dan lain-lain.

Proses collision dapat dibagi menjadi dua kategori dasar, yaitu *collision detection* dan collision response, dengan jarak respon yang telah diaplikasikan secara spesifik. Dalam *collision detection*, terdapat banyak sekali jenis dari *collision detection* itu sendiri, pada gambar II.1 merupakan tiga tipe dari *collision detection*.



Gambar 2.1. Tiga jenis *collision detection*

Setiap objek diberi sebuah kotak sebagai acuan jika collision itu terjadi. Untuk gambar yang pertama, *collision detection* terjadi ketika objek burung bertabrakan dengan kotak yang memuat objek keledai, pada jenis *collision detection* ini rentan terhadap ketidaktelitian, untuk gambar yang kedua, *collision detection* terjadi ketika kotak dari kedua objek tersebut bersinggungan, pada jenis *collision detection* ini sudah lebih akurat dibanding jenis yang pertama, namun untuk memilih factor reduksi yang cocok itu sulit, dan untuk gambar yang ketiga, *collision detection* akan terjadi ketika objek saling bersinggungan tanpa memperdulikan kotak secara keseluruhan, pada jenis *collision detection* ini adalah yang paling akurat dibandingkan dengan jenis yang lainnya.

Selain itu pada sumber lain disebutkan bahwa *collision detection* terbagi menjadi dua jenis, yaitu *collision detection* object to object dan *collision detection object to world*. Pada *object to object* yang terjadi adalah adanya tubrukan antara satu objek dengan objek lainnya yang berada di dalam satu *frame*, baik itu objek

tersebut bertubrukan secara langsung ataupun bertubrukan antara kotak maupun lingkaran yang mengandung objek tersebut, sedangkan pada object to world yang terjadi adalah adanya tubrukan antara objek dengan *frame* itu sendiri, contohnya adalah tubrukan antara sprite dengan dinding pada aplikasi permainan itu sendiri sehingga apabila terjadi, maka sprite tersebut tidak dapat melewati dinding tersebut **[17].**

2.6 Object Oriented Paradigm

Konsep dasar dari paradigma berorientasi objek (cara melakukan sesuatu) relatif mudah untuk dipahami dan diterapkan. Alan Kay, penemu Smalltalk, telah mengerjakan *'Personal Computer for Children of all Ages'* sejak 1968, sebagai sasarannya adalah anak-anak. Pendekatan berorientasi objek diciptakan (karena kesulitan orang-orang yang telah berusaha untuk mendapatkan sistem berkualitas baik, dihasilkan tepat waktu, dan sesuai anggaran, terutama untuk sistem yang besar dengan banyak orang yang terlibat.

Beberapa justifikasi yang biasanya diberikan kepada orientasi objek **[18]**, yaitu sebagai berikut:

1. Objek lebih mudah dipahami masyarakat
2. Spesialis dapat berkomunikasi lebih baik.
3. Data dan proses tidak dipisahkan secara artifisial.
4. Kode dapat digunakan kembali lebih mudah.
5. Orientasi Obyek matang dan terbukti dengan baik.

2.7 Tools yang digunakan

Tools yang digunakan dalam pembuatan *Game* Adventure “Sangkuriang and Indonesian Five Largest Islands” diantaranya adalah UML (Unified Modelling Language), Java, Android, *Framework Kilobolt* dan Adobe Photoshop.

2.7.1 UML (Unified Modeling Language)

Unified Modelling Language (UML) yang berarti bahasa pemodelan standar. UML memiliki sintak dan semantic yang diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

a. Merancang perangkat lunak.
b. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
c. Menjabarkan system secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
d. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Blok pembangun utama UML adalah diagram. Beberapa diagram ada yang rinci, dan ada yang bersifat umum **[19]**.

2.7.2 Diagram-diagram UML

Jenis-jenis Diagram UML **[19]** yaitu :

a. Diagram kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antar muka, kolaborasi, serta relasi-relasi. Pemodelan ini umum dijumpai pada pemodelan sistem berorientasi objek.
b. Diagram Paket. Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.
c. Diagram *Use-case*. Bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan actor-aktor. Diagram *use case* adalah abstraksi dari interaksi antara sistem dan actor, mendeskripsikan bagaimana sistem akan terlihat dimata user.
d. Diagram interaksi dan *Sequence*. Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menampilkan urutan kegiatan dan hubungan kelas.
e. Diagram komunikasi. Bersifat dinamis. Diagram yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
f. Diagram *Statechart*. Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian sera aktifitas.
g. Diagram aktivitas. Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainya dalam suatu sistem.

h. Diagram komponen. Bersifat statis. Diagram komponen ini memperlihatkan organisani serata kebergantungan system / perangkat lunak pada komponen-komponen yang telah ada sebelumnya.
i. Diagram *Deployment*. Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi di jalankan.

2.7.3 Java

Java adalah bahasa pemrograman yang sederhana, berorientasi-objek, bertipe statik, terkompilasi, *multi-threaded, garbage collected, netral arsitektur*, *robust*, aman, dan dapat di perluas. Berikut ini yang terdapat dalam bahasa pemograman Java **[20]**:

a. Sederhana

   Pengembangan java banyak membuang fitur-fitur yang tidak di perlukan seperti yang dimiliki oleh bahasa pemrograman tingkat tinggi lainnya.
b. Berorientasi-objek

   Seperti halnya C++, java menggunakan kelas-kelas untuk mengelola kode ke dalam modul-modul logika.
c. Bertipe statik.

   Semua objek yang digunakan dalam program harus di deklarasikan sebalum digunakan.
d. Terkompilasi

   Sebelum dapat menjalankan program yang di tulis dengan bahasa pemrograman java, program harus dikompilasi oleh kompilator java.
e. *Multi*-treaded.

   Program java dapat berisi eksekusi *multiple treadeds*, yang memungkinkan program mengulas citra pada layar pada satu *thread*, sementara tetap menerima masukan dari *keyboard* pada *thread* utamanya.
f. Garbage *collected* (pembuang sampah)

   Program java menyingkirkan sendiri ‘sampah-sampah’ yang tak berguna.
g. Bebas arsitektur

Pada dasarnya, java tidak dirancang untuk prosesor ataupun sistem operasi tertentu.

h. Tangguh (*robust*)

Karena interpreter java memeriksa seluruh akses sistem yang dilakukan program, maka program java tidak akan membuat sistem menjadi crash.

i. *Aman*

Sistem java hanya memverifikasi seluruh akses memori tetapi juga menjamin bahwa tidak ada virus yang ikut pada applet yang sedang berjalan.

j. Dapat *diperluas*

Program java mendukung metode native yakni fungsi-fungsi yang dituliskan dalam bahasa lain, biasanya C++.

2.7.4 Android

Android adalah sistem operasi yang digunakan di *smartphone* dan juga tablet PC. Android pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 di akuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. Keunggulan utama Android adalah gratis dan *open source*. Keuntungan *open source*, banyak pengembang *software* yang bisa melihat dan memanfaatkan kode itu serta bisa membuat aplikasi baru di dalamnya. Berbagai aplikasi android diwadahi dalam sebuah portal, yaitu Android *Market*, sehingga pengguna tinggal meng-*install* aplikasi pilihannya **[5]**.

2.7.4.1 *Fitur-fitur yang ada di* android

Fitur-fitur yang ada di android **[5]** diantaranya sebagai berikut.

a) ***Framework* Aplikasi** , pengembang memiliki akses penuh *framework* API (*Application Programming Interface*) yang sama yang digunakan oleh aplikasi inti.

b) **Mesin virtual Dalvik** dioptimalkan untuk perangkat mobile.

c) **Integrated browser** berdasarkan *engine open source* WebKit
d) **Grafis** yang dioptimalkan dan didukung oleh perpustakaan grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0
e) **SQLite** untuk penyimpanan data
f) **Media Support** yang mendukung audio, *video*, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
g) **GSM Telephony** (tergantung *hardware*)
h) *Bluetooth*, EDGE, 3G, dan WiFi
i) Kamera, GPS, kompas, dan *accelerometer*
j) **Lingkungan Development** yang lengkap: perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk *Eclipse* IDE.

2.7.5 Tools Pendukung Pembuatan Aplikasi Android

Tools pendukung pembuatan aplikasi android yang digunakan dalam pembuatan *Game* Adventure "Sangkuriang and Indonesian Five Largest Islands" diantaranya adalah Eclipse, Android SDK dan ADT (Android Development Tools).

2.7.5.1 *Eclipse*

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*). Berikut ini adalah sifat dari *Eclipse*:

a) **Multi-*platform***
b) **Mulit-language**: *Eclipse* dikembangkan dengan bahasa pemrograman Java, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
c) **Multi-role**: Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Secara standar *Eclipse* selalu dilengkapi dengan JDT (Java *Development Tools*), *plug-in* yang membuat *Eclipse* kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. *Eclipse* beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java. Konsep *Eclipse* adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*) **[5]**.

2.7.5.2 Android *SDK (Software Development Kit)*

Android SDK merupakan *library* yang berisi kumpulan *tools* / alat bantu yang dibutuhkan dalam membangun / mengembangkan sebuah aplikasi di android dengan menggunakan bahasa pemrograman Java **[5]**.

2.7.5.3 *ADT (*Android *Development Tools)*

ADT merupakan sebuah *plugin* untuk *eclipse*. Sebelum menggunakan atau menginstal ADT, pastikan bahwa *eclipse* nya sudah terinstal dan kompatibel dengan ADT dan android SDK **[5]**.

2.7.6 Kilobolt

Kilobolt merupakan *framework* untuk membangun dan mengembangkan *game* yang dinamis pada sistem operasi android khususnya. Tutorial untuk mengunakan *framework* ini telah disediakan, seperti tutorial Android *Game Development*, Java *Game Development*, *General Game Development Concepts*, dll. Adapun kelebihan dalam menggunakan *framework* ini yaitu sebagai berikut.

1. **Dokumentasi Lengkap**

   Dokumentasi kilobolt sangat lengkap sangat mudah dipahami, tidak hanya berupa *list* dari *method-method* yang ada di kilobolt, tapi lebih seperti sebuah ensiklopedia ditambah dengan contoh penggunaannya.

2. **Plug & *Play***

   Tentu tidak sedikit *framework* yang pada tahap instalasi saja sudah sulit, tidak dengan Kilobolt yang sangat memudahkan programmer.

3. **Small Simple Yet Fast**

   Kilobolt *framework* yang didesain untuk pengembangan aplikasi kelas menengah, tetapi jika ingin melakukan *extend* kilobolt lebih jauh lagi telah tersedia plugin sesuai kebutuhan.

4. **Small Learning Curve**

   Dengan dokumentasi yang bisa dikatakan sangat bagus dan jelas, proses belajar *framework* ini menjadi sangat mudah dan menyenangkan.

2.7.7 Adobe Photoshop

Adobe Photoshop adalah *software* yang dibuat oleh perusahaan *Adobe System*, yang di khususkan untuk pengeditan foto atau gambar dan pembuatan effect. Perangkat lunak ini banyak digunakan oleh Fotografer Digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar. 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 ImageReady*, untuk keperluan tersebut.

Photoshop juga memiliki hubungan 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, After Effects dan Adobe Encore DVD* untuk membuat DVD profesional, menyediakan penyuntingan gambar non-linear dan layanan *special effect* seperti *background*, tekstur, dan lain-lain untuk keperluan televisi, film, dan situs web.

2.7.8 Extensible Markup Language (XML)

XML (*Extensible Markup Language*) adalah bahasa markup yang mendifinisikan seperangkat aturan untuk *encoding* dalam dokumen yang bersifat *human-readable* dan *machine-readable*. Pada android sendiri berkas berjenis XML memiliki banyak fungsi **[21]**, diantaranya :

1. Berkas Android *Manifest*

   Setiap aplikasi android harus memiliki berkas ini (dengan nama Android Manifest.xml) pada direktori root. Pada berkas ini berisi informasi penting tentang aplikasi untuk sistem android.

2. *Layout*

   Android menyediakan kosakata XML sederhana yang sesuai dengan kelas View dan subclass-nya, seperti untuk widget dan layout.

3. Database

   Database XML adalah persistensi data perangkat lunak sistem yang memungkinkan data yang akan disimpan dalam format XML. Data ini kemudian dapat dilihat, diekspor dan serialisasi ke dalam format yang diinginkan. XML database biasanya berhubungan dengan dokumen berorientasi database.

2.8 Pengujian Sistem

Pengujian menyajikan anomali yang menarik bagi perekayasa perangkat lunak. Pada proses perangkat lunak, perekayasa berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

2.8.1 Pengujian *Black-Box*

Menurut [Rog07] Pengujian black-box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian black-box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian black-box berusaha menemukan kesalahan dalam kategori sebagai berikut :

a. Fungsi-fungsi yang tidak benar atau hilang,
b. Kesalahan interface.
c. Kesalahan dalam struktur data atau akses eksternal.
d. Kesalahan kinerja.

e. Inisialisasi dan kesalahan terminasi

2.8.2 Pengujian White – Box

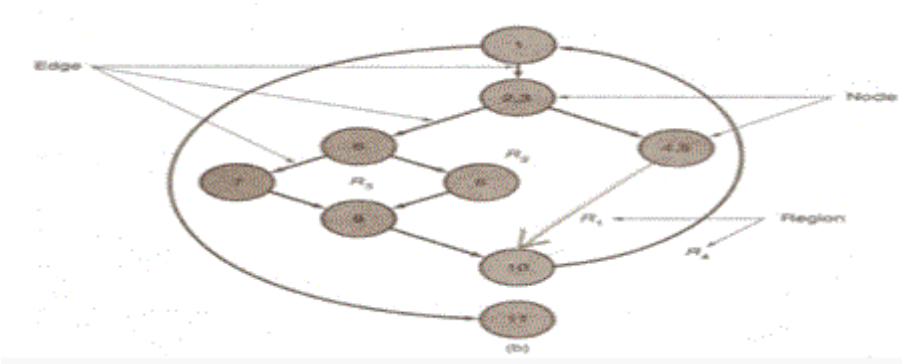
White Box Testing adalah meramalkan cara kerja perangkat lunak secara rinci, karenanya logikal path (jalur logika) perangkat lunak akan ditest dengan menyediakan test case yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara spesifik.

2.8.3 Pengujian Path Basis

Pada pengujian white box terdapat 2 metode yaitu Basis Path dan Struktur kendali. Pengujian basis path adalah pengujian yang memungkinkan penguji dapat mengukur kompleksitas logis dari desain procedural dan menggunakannya sebagai pedoman untuk menetapkan himpunan basis dari semua jalur eksekusi. Kasus uji dihasilkan untuk melakukan sekumpulan basis yang dijamin untuk mengeksekusi setiap perintah dalam program, sedikitnya satu kali selama uji coba.

1. Notasi Diagram Alir (Path Graph Notation)

   Notasi yang digunakan untuk menggambarkan jalur eksekusi adalah notasi diagram alir (atau grafik program), yang menggunakan notasi lingkaran (simpul atau node) dan anak panah (link atau edge). Notasi ini menggambarkan aliran control logika yang digunakan dalam suatu bahasa pemrograman.



Gambar 2.2. Grafik alir (Flow graph)

2. Kompleksitas Siklomatis (Cyclomatic Complexity)

   Kompleksitas Siklomatis adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Ketika digunakan dalam konteks metode uji coba berbasis alur, nilai yang didapat akan menentukan jumlah jalur independen dalam himpunan path, serta akan memberi nilai batas atas bagi jumlah pengujian yang harus dilakukan, untuk memastikan bahwa semua pernyataan telah dieksekusi sedikitnya satu kali. Jalur independent adalah jalur yang terdapat dalam program yang mengintroduksi sedikitnya satu rangkaian pernyataan proses atau kondisi baru.