BAB II
LANDASAN TEORI

II.1 Game

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 segala jenis perangkat [8].

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.

II.2 Klasifikasi Game

Game bisa diklasifikasikan kedalam beberapa hal diantaranya klasifikasi berdsarkan *platform* dan berdasarkan *genre*

II.3.1 Berdasarkan *Platform*

Tiga kategori utama *platform* pada game adalah *personal computer* (PC), *console* dan *mobile* [2].

1. *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*.

2. *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.

3. *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*).

II.3.2 Berdasarkan Genre

Menurut buku “Fundamentals of Game Design” [3], genre game bisa dibagi menjadi 9, 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*

   Adventure 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.

II.3 Edukasi

Edukasi atau pendidikan ialah suatu usaha yang sadar yang teratur dan sitematis, yang dilakukan oleh orang – orang yang diserahi tanggung jawab untuk mempengaruhi anak agar mempunyai sifat dan tabiat yang sesuai dengan cita – cita pendidikan. Edukasi juga bisa diartikan sebagai bantuan yang diberikan secara sengaja kepada anak dalam pertumbuhan jasmani maupun rohani.

Secara teoritis pengertian mendidik dan mengajar tidaklah sama. Mengajar berarti menyerahkan atau manyampaikan ilmu pengetahuan atau keterampilan dan lain sebagainya kepada orang lain, dengan menggunakan cara – cara tertentu sehingga ilmu – ilmu tersebut bisa menjadi milik orang lain.

Lain halnya mendidik, bahwa mendidik tidak hanya cukup dengan hanya memberikan ilmu pengetahuan ataupun keterampilan, melainkan juga harus ditanamkan pada anak didik nilai – nilai dan norma – norma susila yang tinggi dan luhur. Dari pengertian di atas diketahui bahwa mendidik lebih luas dari pada mengajar. Mengajar hanyalah alat atau sarana dalam mendidik .dan mendidik harus mempunyai tujuan dan nilai – nilai yang tinggi [4].

II.3.1 Unsur – unsur Edukasi

Unsur – unsur edukasi adalah [5]:

1. Anak didik, yaitu pihak yang menjadi objek utama edukasi.
2. Pendidik, yaitu pihak yang menjadi subjek dari pelaksanaan edukasi.
3. Materi, yaitu bahan atau pengalaman belajar yang disusun menjadi kurikulum.
4. Alat edukasi, yaitu tindakan yang menjadi kelangsungan mendidik.
5. Lingkungan, yaitu keadaan yang berpengaruh terhadap hasil edukasi.
6. Dasar dan landasan edukasi, yaitu landasan yang menjadi fundamental dari segala kegiatan edukasi.

II.3.2 Faktor – faktor yang membatasi kemampuan edukasi

Adapun faktor-faktor yang membatasi kemampuan edukasi diantaranya [5]:

1. Faktor anak

   Di dalam anak didik terdapat potensi-potensi yang butuh edukasi dari luar
2. Faktor pendidik

   Guru mempunyai metode penyampaian yang berbeda dan beragam
3. Faktor lingkungan

   Lingkungan sangat berpengaruh baik positif maupun negatif

II.4 Game Education

Game edukasi adalah game digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interaktif [6].

Kriteria perancangan game edukasi yang ideal dibagi menjadi enam *point*, yaitu :

1. Rasa ingin tahu, fantasi dan kontrol
   a. Motivasi intrinsik, menyenangkan untuk digunakan. Pengguna memiliki kontrol terhadap permainan.

b. Integrasi antara materi edukasi dan aspek fantasi serta permainan

c. Pengguna memegang kendali permainan dan dapat mengatur kecepatan bermain.

d. Mendorong keingintahuan, pengguna dapat melakukan eksplorasi bebas. Permainan mengandung rahasia tersembunyi.

e. Pengguna dapat menyimpan kemajuan permainan.

f. Menyediakan lebih dari jalur. Ada beberapa cara untuk memenangkan permainan.

g. Pengguna dapat melakukan *trial-error* untuk menyelesaikan permainan.

h. Keberhasilan ditentukan oleh pengetahuan, bukan kebetulan.

i. Task disajikan secara *incremental* dan bertahap.

j. Simulasi realistik dunia.

k. Materi edukasi disesuaikan dengan materi dunia nyata.

l. Personalisasi, yaitu pengguna dapat mengubah karakter dan objek-objek.

m. Menyediakan banyak pilihan dan tema.

n. Hindari pengulangan, hindari *drill-practice*.

o. Sediakan elemen tidak terduga dan kejutan.

p. Poin tidak hilang saat menjawab pertanyaan.

2. Tantangan

a. Kinerja pemain terukur dan didefinisikan dengan baik.

b. Tantangan diberikan secara terus menerus
tapi disesuaikan dengan tingkat pemahaman pemain.

c. Kompleksitas semakin berkembang sejalan dengan kemampuan pengguna. Tersedia banyak level.

d. Memonitor kinerja pengguna, berdasarkan hasil *monitoring* ini kesulitan kemudian disesuaikan.

e. Pengguna dapat melihat *progress*nya setiap saat.

f. Menyediakan *hint* dan instruksi untuk membantu pengguna.

g. Bantuk pengguna menemukan *'flow state'* yaitu keseimbangan antara tantangan dengan kemampuan untuk menghadapi tantangan.

3. Sosialisasi
   a. Pemain dapat berkolaborasi bersama.

   b. Pemain dapat saling berkompetisi.

   c. Pemenang bisa lebih dari satu (*multiple winner*)

4. Pedegogi
   a. Menyebutkan secara eksplisit target umur dan menyesuaikan rancangan permainan sesuai umur tersebut.

   b. Aktivitas pembelajaran dilaksanaan bersamaan dengan permainan.

   c. Objektif pembelajaran jelas.

   d. Memberi kesempatan kepada guru dan orangtua untuk ikut berperan.

   e. Memberikan petunjuk yang jelas sehingga pemain dapat berkonsentrasi kepada isi permainan, bukan bagaimana cara menggunakan permainan.

f. Menyediakan tutorial sehingga pemain tidak perlu membaca manual.

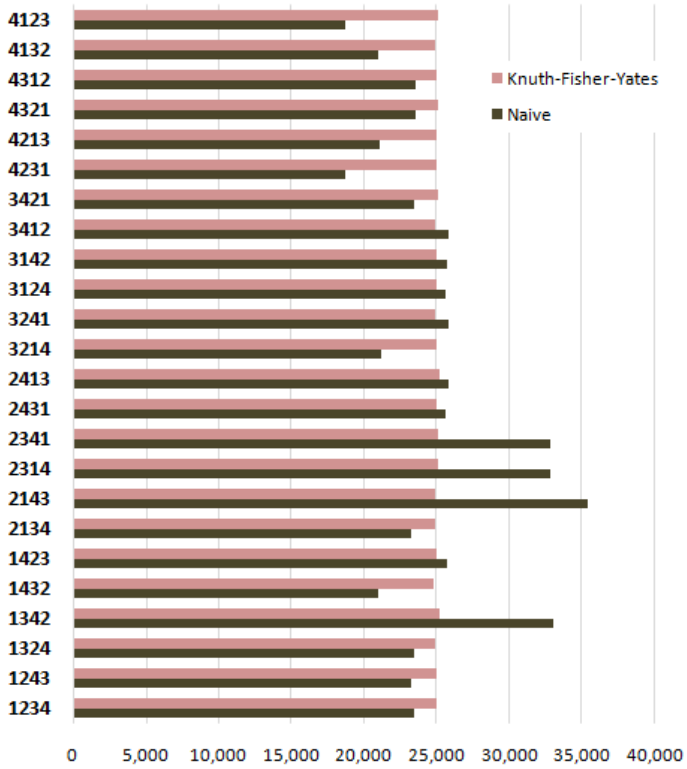
5. Teknologi
    a. Menggunakan teknologi yang tersedia di sekolah dan masyarakat umum (tidak membutuhkan *hardware* yang terlalu tinggi)
    b. Memberikan lisensi yang terjangkau sekolah.
    c. Menggunakan antarmuka yang intuitif.
6. Pengguna anak-anak dan yang berkebutuhan khusus
    a. Memberikan petunjuk yang dibacakan selain petunjuk tertulis.
    b. Gambar, objek, layar tertata rapi.
    c. Permainannya sendiri merupakan aktivitas yang penting (*play for the sake of play*)
    d. Transformasi terlihat. Saat anak melakukan interaksi, ada *instant feedback*.
    e. *Input* dan *output* yang mudah terlihat.
    f. Tantangan diberikan secara *gradual*.
    g. Permainan menyenangkan untuk diulang, beberapa karakter, musik mudah teringat oleh anak.
    h. Objek dan suara diambil dari kehidupan sehari-hari yang mudah dikenali.
    i. Menginspirasi anak, bahkan setelah komputer dimatikan.

II.5 Shuffle Fisher Yates

Fisher-Yates Shuffle (dinamai berdasarkan penemunya, Ronald Fisher dan Frank Yates) digunakan untuk mengubah urutan masukan yang diberikan secara

acak. Permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama [7].

Algoritma ini dinyatakan bias karena permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama, hal ini dibuktikan dengan percobaan mengacak suatu set kartu yang dilakukan berulang – ulang seperti yang bisa dilihat pada gambar II.1.



Gambar II. 1 Hasil perbandingan algoritma shuffle [8]

Pada gambar II.1 dilakukan tes terhadap 4 kartu, yang kemudian diacak sebanyak 600.000 kali. Keterangan pada sumbu Y menunjukkan permutasi / kemungkinan kombinasi kartu yang muncul, sedangkan keterangan pada sumbu X menunjukkan jumlah kemunculan kombinasi itu muncul. Warna pink menunjukkan hasil dari algoritma Fisher-Yates shuffle sedangkan warna abu-abu kehijauan menunjukkan algoritma Naïve shuffle. Hasilnya algoritma Fisher-Yates menghasilkan nilai yang hampir sama untuk setiap kemungkinan kombinasi kartu, sedangkan pada algoritma Naïve kombinasi kartu tertentu muncul jauh lebih sering dan sebagian lagi muncul lebih sedikit dibanding kombinasi kartu lain.

Metode Fisher-Yates secara umum adalah :

1. Ketika masih ada elemen tersisa untuk diacak
2. Ambil elemen secara acak dari elemen yang tersisa
3. Kemudian tukar dengan elemen saat ini.

Gambar II.2 menunjukkan algoritma fisher – yates shuffle yang ditulis dalam bahasa javascript.

```
function shuffle(array) {
  var m = array.length, t, i;

  // While there remain elements to shuffle…
  while (m) {

    // Pick a remaining element…
    i = Math.floor(Math.random() * m--);

    // And swap it with the current element.
    t = array[m];
    array[m] = array[i];
    array[i] = t;
  }

  return array;
}
```

Gambar II. 2 Algoritma Fisher - Yates Shuffle [9]

II.6 Tools

Pada sub bab ini akan dijelaskan *tools* yang digunakan dalam pembangunan game edukasi *adventure* teori masuik ini.

II.6.1 Object Oriented Programing

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. Tidak mengherankan bahwa konsep dasarnya sederhana [10].

Pendekatan berorientasi objek diciptakan (atau lebih tepatnya, berkembang) 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 :

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.

II.7 *Unifed Modeling Language* (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.[11]

UML menyediakan jenis-jenis diagram untuk memodelkan pemrograman berorientasi objek atau *Object Oriented Programing (OOP)*. Dimana terdapat beberapa diagram untuk menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sebuah sistem. Berikut ini adalah jenis-jenis diagram yang digunakan [11]:

1. *Use case diagram*

   Use case diagram adalah sebuah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk *use case* dan *actor*. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sebuah sistem aplikasi.
2. *Sequence diagram*

Sequence diagram merupakan sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah sistem. Interaksi tersebut berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri dari dimensi horizontal (objek-objek) dan dimensi vertical (waktu). Diagram ini juga menggambarkan urutan *event* yang terjadi, dan lebih detail dalam menggambarkan aliran data yang dikirimkan atau diterima. Namun, diagram ini kurang mampu menjelaskan detail dari sebuah algoritma, seperti *loop* dan *branching*.

3. *Class diagram*

   Class diagram adalah sebuah diagram yang menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Class itu sendiri yaitu dekripsi dari kelompok *object- object* dengan *property*, perilaku (operasi), dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah sistem.

4. *Activity* Diagram

   Activity Diagram adalah sebuah diagram yang menggambarkan rangkaian aliran dari aktifitas yang digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

II.8 Flash

Adobe Flash dahulu bernama Macromedia Flash adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe Systems. Adobe Flash digunakan untuk membuat gambar *vektor* maupun animasi gambar tersebut. Berkas yang dihasilkan dari perangkat lunak ini mempunyai *file extension* .swf dan dapat diputar di penjelajah web yang telah dipasangi Adobe Flash Player. Flash menggunakan bahasa pemrograman bernama *Action Script* yang muncul pertama kalinya pada Flash 5.

Sebelum tahun 2005, Flash dirilis oleh Macromedia. Flash 1.0 diluncurkan pada tahun 1996 setelah Macromedia membeli program animasi *vektor* bernama

FutureSplash. Versi terakhir yang diluncurkan di pasaran dengan menggunakan nama 'Macromedia' adalah Macromedia Flash 8. Pada tanggal 3 Desember 2005 Adobe Systems mengakuisisi Macromedia dan seluruh produknya, sehingga nama Macromedia Flash berubah menjadi Adobe Flash.

Adobe Flash merupakan sebuah program yang didesain khusus oleh Adobe dan program aplikasi standar *authoring tool professional* yang digunakan untuk membuat animasi dan bitmap yang sangat menarik untuk keperluan pembangunan situs web yang interaktif dan dinamis. Flash didesain dengan kemampuan untuk membuat animasi 2 dimensi yang handal dan ringan sehingga flash banyak digunakan untuk membangun dan memberikan efek animasi pada website, CD Interaktif dan yang lainnya. Selain itu aplikasi ini juga dapat digunakan untuk membuat animasi logo, *movie*, *game*, pembuatan navigasi pada situs web, tombol animasi, banner, menu interaktif, interaktif form isian, *e-card*, *screen saver* dan pembuatan aplikasi-aplikasi web lainnya.

Dalam Flash, terdapat teknik-teknik membuat animasi, fasilitas *action script*, *filter*, *custom easing* dan dapat memasukkan *video* lengkap dengan fasilitas *playback FLV*. Keunggulan yang dimiliki oleh Flash ini adalah ia mampu diberikan sedikit code pemograman baik yang berjalan sendiri untuk mengatur animasi yang ada didalamnya atau digunakan untuk berkomunikasi dengan program lain seperti HTML, PHP, dan Database dengan pendekatan XML, dapat dikolaborasikan dengan web, karena mempunyai keunggulan antara lain kecil dalam ukuran file outputnya.

Movie-movie Flash memiliki ukuran *file* yang kecil dan dapat ditampilkan dengan ukuran layar yang dapat disesuaikan dengan keingginan. Aplikasi Flash merupakan sebuah standar aplikasi industri perancangan animasi web dengan peningkatan pengaturan dan perluasan kemampuan integrasi yang lebih baik. Banyak fitur-fitur baru dalam Flash yang dapat meningkatkan kreativitas dalam pembuatan isi media yang kaya dengan memanfaatkan kemampuan aplikasi tersebut secara maksimal. Fitur-fitur baru ini membantu kita lebih memusatkan perhatian pada desain yang dibuat secara cepat, bukannya memusatkan pada cara kerja dan penggunaan aplikasi tersebut.

Flash juga dapat digunakan untuk mengembangkan secara cepat aplikasi-aplikasi web yang kaya dengan pembuatan *script* tingkat lanjut. Di dalam aplikasinya juga tersedia sebuah alat untuk men-*debug script*, dengan menggunakan *code hint* untuk mempermudah dan mempercepat pembuatan dan pengembangan isi *Action Script* secara otomatis [12].