1. **LANDASAN TEORI**
   1. **Penelitian Terdahulu**

Sebagai bahan pertimbangan dalam penelitian ini akan dicantumkan beberapa hasil penelitian terdahulu oleh beberapa peneliti yang pernah dibaca, diantaranya :

1. **Implementasi Speech Recognition Pada Keamanan Folder Dengan Speech Application Programming Interface (Syarif Husain Hasibuan, 2014, Universitas Sumatera Utara)**

Syarif Husain Hasibuan mengungkapkan bahwa Sebagai pengguna komputer seseorang berhak untuk menjaga kerahasiaan dan kemanan *file* yang dimiliki. Sebab sifat *file* dan dokumen penting sangat sensitif maka dari itu perlu adanya keamanan *file* dan dokumen-dokumen penting dengan menggunakan *password*. Tetapi tidak sedikit *password* menjadi masalah baru bagi pengguna komputer. Seperti mudah ditebak oleh orang lain dan lupa *password*. Pengenalan dan penggunaan suara bisa menjadi aplikasi dalam sistem *security* dalam menjaga *file* atau dokumen penting.

Penelitian ini menerapkan *speech recognition* dengan menggunakan teknik S-API (*Speech Aplication Programing Interface*) didalam keamanan folder guna menjaga keamanan dalam pengaksesan folder.

Hasil dari penelitian ini adalah sebuah *system* keamanan *folder* dengan menggunakan *voice recognition* agar menjaga keamanan dalam pengaksesan *folder*.

Kesimpulan dari penelitian ini adalah :

Penelitian ini menghasilkan sebuah *system* keamanan *folder* dengan menggunakan *voice recognition* agar menjaga keamanan dalam pengaksesan *folder*. dengan menggunakan teknik S-API (*Speech Aplication Programing Interface*) didalam keamanan folder guna menjaga keamanan dalam pengaksesan *folder*.

1. **Penerapan Speech Recognition Pada Permainan Teka-Teki Silang Menggunakan Metode Hidden Markov Model (HMM) Berbasis Desktop (Diyah Puspitaningrum, 2016, Universitas Bengkulu)**

Diyah Puspitaningrum melihat bahwa seiring Perkembangan teknologi saat ini telah memberikan pengaruh yang sangat besar bagi dunia teknologi informasi. Pada perkembangannya sekarang. Aplikasi permainan masih banyak menggunakan *keyboard* atau *mouse* (manual) dan belum otomatis melakukan pengenalan suara dan kebanyakan *game* tersebut hanya memberikan kesan hiburan dan bukan *game* yang bersifat edukasi. Hal inilah yang melatarbelakangi dibuatnya suatu *game* desktop yang bertema teka-teki silang menggunakan suara.

Pada penelitian ini dibuat suatu aplikasi permainan teka-teki silang model baru yang dapat menerima input berupa suara ucapan huruf demi huruf dalam Bahasa Inggris dan outputnya adalah pengucapan kata yang terbentuk juga dalam Bahasa Inggris.

Adapun untuk pengenalan ucapannya digunakan metode *Hidden Markov Model* (HMM) dengan demikian permainan Teka-teki silang TTS ini memiliki nilai hiburan dan nilai edukasi.

Penelitian ini telah menghasilkan aplikasi penerapan *speech recognition* pada permainan Teka-teki silang mengunakan metode *Hidden Markov Model* (HMM) berbasis *desktop*.

Kesimpulan dari penelitian ini adalah :

Pembuatan aplikasi permainan teka-teki silang model baru yang dapat menerima input berupa suara ucapan huruf demi huruf dalam Bahasa Inggris dan outputnya adalah pengucapan kata yang terbentuk juga dalam Bahasa Inggris dengan penerapan *speech recognition* pada permainan Teka-teki silang (TTS) mengunakan metode *Hidden Markov Model* (HMM) berbasis *desktop*.

1. **Perancangan Game Edukasi Untuk Anak Usia Dini (4-6 Tahun) Berbasis Android (Jarot Anton Haryasena, 2013, Universitas Muhammadiyah Surakarta)**

Jarot Anton Haryassena melihat bahwa seiring perkembangan teknologi yang sangat pesat telah mengubah model, pola dan media pembelajaran di dunia pendidikan. Dalam dunia pendidikan sendiri sudah banyak aplikasi multimedia pembelajaran baik untuk anak-anak maupun orang dewasa baik itu dalam bentuk aplikasi ataupun game edukasi. Game edukasi sendiri telah dibuktikan dapat menunjang proses pendidikan, dan *game* edukasi unggul dalam beberapa aspek jika dibandingkan dengan metode pembelajaran konvensional. Salah satu keunggulan yang signifikan adalah adanya animasi yang mampu menarik perhatian anak dalam mengenal suatu hal dan juga dapat meningkatkan daya ingat anak tersebut.

Dalam penelitian ini, peneliti mencoba membuat sebuah game yang mampu digunakan untuk anak usia dini dalam melatih daya ingatnya dan sambil mengenal angka, huruf, hewan, buah dan juga warna.

Pengolahan permainan dalam *game* ini menggunakan Eclipse sebagai media untuk perancangan dan pembuatannya serta menggunakan sistem operasi Android *Jelly Bean* untuk media pengoperasiannya.

Hasil dari penelitian ini adalah sebuah *game* edukasi yang mampu digunakan untuk melatih daya ingat pada anak dan hasil pengujian yang dilakukan bahwa *game* ini bernilai cukup baik untuk melatih daya ingat sambil mengenal angka, huruf, hewan, buah, dan warna dengan menganalisa hasil kuesioner yang diberikan kepada guru di TK Aisyiyah Karangwoni.

Kesimpulan dari penelitian ini adalah :

Game edukasi sendiri telah dibuktikan dapat menunjang proses pendidikan, dan *game* edukasi unggul dalam beberapa aspek jika dibandingkan dengan metode pembelajaran konvensional. Salah satu keunggulan yang signifikan adalah adanya animasi yang mampu menarik perhatian anak dalam mengenal suatu hal dan juga dapat meningkatkan daya ingat anak tersebut. Dalam penelitian ini, peneliti mencoba membuat sebuah *game* yang mampu digunakan untuk anak usia dini dalam melatih daya ingatnya dan sambil mengenal angka, huruf, hewan, buah dan juga warna.

1. **Analisis Dan Perancangan Game Shooting 3D Menggunakan Unity (Suhendrik, 2012, Universitas Bina Nusantara)**

Perkembangan teknologi dalam bidang komputer, banyak digunakan dalam berbagai bidang, termasuk bidang hiburan. Salah satunya penggunaan komputer dalam bidang hiburan ini adalah, dalam bentuk *game*. Game banyak diminati karena membutuhkan interaksi dengan pengguna, oleh sebab itu *game* menjadi lebih menarik.

Pada saat ini banyak sekali genre *game* salah satunya adalah *shooting game.* *Shooting game* adalah sub-genre dari *action game*, dimana pemain memainkan *game* tembak–menembak yang didalamnya biasanya berhubungan dengan senjata untuk menembaki musuh dalam *game* tersebut akan tetapi rata-rata *game* dengan genre yang sejenis mempunyai besar *size* yang relatif besar dan memerlukan spesifikasi komputer yang cukup tinggi.

Maka dari itu penulis tertarik untuk menciptakan *game* 3D dengan genre *shooting game* yang mempunyai ukuran file yang relatif kecil tetapi dengan model yang lebih nyata.

Tujuan dari pembuatan permainan ini adalah untuk membuat *game* 3D dengan ukuran *file* yg *relative* kecil dan dapat dimainkan pada komputer dengan spesifikasi rendah agar dapat dimainkan diberbagai spesifikasi komputer.

Adapun manfaat yang diharapkan dari pembuatan game ini terhadap masyarakat yang memainkan permainan ini adalah :

1. Pemain dapat memainkan *game* 3D dengan spesifikasi komputer yang rendah
2. Menghibur para pemain.
3. Mengurangi stress.

Kesimpulan dari penelitian ini adalah :

Tujuan dari pembuatan permainan ini adalah untuk membuat game 3D dengan ukuran *file* yg *relative* kecil dan dapat dimainkan pada komputer dengan spesifikasi rendah agar dapat dimainkan diberbagai spesifikasi komputer.

1. **Aplikasi Permainan Edukatif Dalam Bahasa Inggris Untuk anak Usia 3-5 Tahun Menggunakan Visual Basic 6.0 (Sepri Minarno, 2012, STMIK Indonesia Banjarmasin)**

Sepri Minarno mengungkapkan bahwa permainan edukasi adalah salah satu cara mudah dan efektif memberikan suatu pelajaran kepada anak – anak pada usia 3-6 tahun kinerja otaknya masih belum berkembang dengan sempurna, tapi jika di usia tersebut kita sudah membiasakan anak untuk belajar maka pelajaran yang diajarkan tersebut akan melekat kuat pada otaknya.

Dengan permainan edukatif anak akan dengan sangat mudah untuk menerima pelajaran karena pelajaran yang diberikan dibarengi dengan bermain yang pastinya untuk anak – anak sangat suka untuk bermain.

Apalagi dalam era modern seperti sekarang ini Bahasa *Internasional* (Bahasa Inggris) sangat penting peranannya dalam kehidupan sehari – hari di masyarakat. Apabila anak sejak dini sudah di kenalkan dengan Bahasa Inggris maka diharapkan anak tersebut untuk kedepannya tidak akan canggung untuk terjun di kehidupan yang sebenarnya.

Oleh karena itu, dengan adanya Aplikasi Permainan Edukatif dalam Bahasa Inggris diharapkan bisa memberikan suatu manfaat bagi anak usia dini untuk mengenal huruf, angka dan bentuk bidang datar dalam Bahasa Inggris sehingga anak dapat bermain sambil belajar dalam Bahasa Inggris.

Kesimpulan dari penelitian ini adalah :

Aplikasi Permainan Edukatif dalam Bahasa Inggris diharapkan bisa memberikan suatu manfaat bagi anak usia dini untuk mengenal huruf, angka dan bentuk bidang datar dalam Bahasa Inggris sehingga anak dapat bermain sambil belajar dalam Bahasa Inggris.

1. **Little Botany: A Mobile Game Utilizing Data Integration to Enhance Plant Science Education ( Suphanut Jamonnak and EnCheng, 2017, Department of Computer Science, College of Arts and Sciences, University of Akron, Akron, OH 44325-4003, USA)**

Perangkat seluler dengan cepat menjadi media baru kehidupan pendidikan dan sosial bagi kaum muda, dan karenanya *game* edukasi seluler telah menjadi mekanisme penting untuk belajar. Untuk membantu anak-anak usia sekolah belajar tentang dunia tanaman yang memukau, kami menyajikan *game* edukasi seluler bernama *Little Botany*, di mana pemain dapat membuat taman virtual sendiri di lokasi mana pun di bumi. Salah satu fitur unik dari *Little Botany* adalah bahwa *game* ini dibangun di atas data dunia nyata dengan memanfaatkan mekanisme integrasi data. Kebun yang dibuat di *Little Botany* ditambahkan dengan data lokasi dunia nyata dan data cuaca waktu nyata. Lebih khusus lagi, *Little Botany* menggunakan data cuaca waktu nyata untuk lokasi taman untuk mensimulasikan bagaimana cuaca mempengaruhi pertumbuhan tanaman. Pemain *game* *Little Botany* dapat belajar memilih tanaman apa yang ditanam, memelihara kebun mereka sendiri, melihat tanaman untuk tumbuh, merawat tanaman setiap hari, dan memanennya.

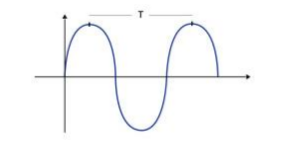
Kesimpulan dari penelitian ini adalah :

Game edukasi seluler bernama *Little Botany,*  di mana pemain dapat membuat taman virtual sendiri di lokasi manapun di bumi. Salah satu fitur unik dari *Little Botany* ini adalah bahwa game ini di bangun di atas dunia nyata dengan memanfaatkan mekanisme integrasi data. Kebun yang di buat di *Little Botany*  ditambahkan dengan data lokasi dunia nyata dan data cuaca waktu nyata.

1. **Pengertian Aplikasi**

Aplikasi berasal dari kata *application* artinya penerapan atau penggunaan. Secara istilah aplikasi adalah *program* siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi lain dan dapat digunakan oleh sasaran yang dituju. Menurut (Jogiyanto, 1999) adalah penggunaan dalam suatu komputer, instruksi (*instuction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

1. **Suara**

** Suara (*Audio*) adalah fenomena fisik yang dihasilkan oleh getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu yang disebut *frekuensi* (Binanto, 2010). Selama bergetar, perbedaan tekanan terjadi di udara sekitarnya. Pola osilasi yang terjadi dinamakan sebagai gelombang. Gelombang mempunyai pola sama yang berulang pada *interval* tertentu, yang disebut sebagai periode. Contoh suara periodik adalah instrumen musik, nyanyian burung sedangkan contoh suara non periodik adalah batuk, percikan ombak dan lain-lain. Gelombang suara terjadi sebagai variasi tekanan dalam sebuah media, seperti udara. Ia tercipta dari bergetarnya sebuah benda, yang menyebabkan udara sekitarnya ikut bergetar. Udara yang bergetar kemudian diterima oleh telinga, menyebabkan gendang telinga manusia bergetar, kemudian otak menafsirkannya sebagai suara. Gelombang suara berjalan melalui udara, sama seperti gelombang yang terjadi di air. Bahkan, gelombang air lebih mudah untuk dilihat dan dimengerti, hal ini sering digunakan sebagai analogi untuk menggambarkan bagaimana gelombang suara berperilaku. Gelombang suara juga dapat ditampilkan dalam grafik XY (Binanto, 2010).

Gambar 2.1 Gelombang Suara (Binanto, 2010)

Pada Gambar 2.1 di atas, suatu grafik gelombang berbentuk dua dimensi, tetapi di dunia nyata gelombang suara berbentuk tiga dimensi. Grafik menunjukkan gelombang bergerak sepanjang jalan dari kiri ke kanan, tapi kenyataannya perjalanan gelombang suara bergerak ke segala arah menjauhi sumber. Kira-kira sama seperti riak air yang terjadi ketika kita menjatuhkan sebuah batu ke dalam kolam. Namun model dua dimensi ini cukup dapat menjelaskan tentang bagaimana suara bergerak dari satu tempat ke tempat lain.

**2.4.1. Bagian-bagian Gelombang Suara :**

Semua gelombang memiliki sifat-sifat tertentu. Ada tiga bagian yang paling penting untuk audio (Binanto, 2010) :

* Panjang gelombang

Jarak antara titik manapun pada gelombang (pada gambar ditunjukkan sebagai titik tertinggi) dan titik setara pada fase berikutnya. Secara harfiah, panjang gelombang adalah jarak yang digambarkan dgn huruf “T”.

* *Amplitudo*

kekuatan sinyal gelombang (*intensity*) Titik tertinggi dari gelombang bila dilihat pada grafik. *Amplitudo* tinggi biasa disebut sebagai *volume* yang lebih tinggi, diukur dalam dB. Nama perangkat untuk meningkatkan *amplitudo* disebut *amplifier*.

* *Frequency*

Frekuensi waktu yang dibutuhkan oleh gelombang bergerak dari satu pase ke pase berikutnya dalam satu detik. Diukur dalam *herz* atau *cycles* per *second*. Semakin cepat sumber suara bergetar, semakin tinggi *frekuensi*.

1. **Representasi Suara**

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur *amplitudo* pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan“*sample*”. *Analog* *to* *Digital* *Conversion* (ADC) adalah proses mengubah amplitudo gelombang bunyi ke dalam waktu *interval* tertentu (sampling), sehingga menghasilkan representasi *digital* dari suara. Dalam teknik sampling dikenal istilah sampling rate yaitu beberapa gelombang yang diambil dalam satu detik. Sebagai contoh jika kualitas CD *Audio* dikatakan memiliki frekuensi sebesar 44.100 Hz, berarti jumlah sampel sebesar 44.100 per detik.

Langkah-langkah dalam proses digitasi adalah:

1. Membuang frekuensi tinggi dari *source signal*.
2. Mengambil *sample* pada *interval* waktu tertentu (*sampling*).
3. Menyimpan *amplitudo* sampel dan mengubahnya ke dalam bentuk diskrit (kuantisasi).
4. Merubah bentuk menjadi nilai biner.

Teknik *sampling* yang umum pada *file* *audio* seperti *Nyquist* *Sampling* *Rate* dimana untuk memperoleh representasi akurat dari suatu sinyal analog secara *lossless*, amplitudonya harus diambil *sample*-nya setidaknya pada kecepatan (*rate*) sama atau lebih besar dari 2 kali lipat komponen *frekuensi* *maksimum* yang akan didengar. Misalnya untuk sinyal analog dengan *bandwith* 15Hz – 10kHz→*sampling rate* = 2 x 10KHz = 20 kHz (Gunawan, 2005).

1. **Speech Recognition**

Speech Recognition adalah sebuah alternatif metode tradisional untuk berinteraksi dengan komputer, seperti input teks melalui *keyboard*. Teknik input data menggunakan *speech recognition* dapat mengganti masukan dari *keyboard* maupun melalui *mouse*. *Speech Recognition* sangat membantu jika seseorang dalam situasi sebagai berikut:

* Orang yang memiliki sedikit keterampilan dalam menggunakan keyboard, dalam arti lambat dalam melakukan pengetikan.
* Orang yang memiliki masalah dengan karakter atau penggunaan kata dan manipulasi dalam bentuk tekstual.
* Orang yang memiliki cacat fisik yang tidak mampu untuk melakukan entri data melalui *keyboard* maupun *mouse*, atau memiliki ketidakmampuan melihat apa yang mereka inputkan. (Kirriemuir, 2003).

1. **Microsoft Speech Recognition**

Microsoft *Speech Recognition* merupakan perangkat yang dikeluarkan oleh Microsoft Coorporation yang memungkinkan kita melakukan perintah pengetikan ke dalam *Personal Computer* (PC) yang beroperasi di bawah sistem operasi Windows XP, Windows 7, Windows 8 maupun Windows 10. Suara yang kita masukkan akan secara otomatis dikonversikan ke dalam bentuk teks.

Dengan menggunakan *speech recognition* memungkinkan kita untuk mengurangi kepenatan pada jari-jari dan pergelangan tangan dan meningkatkan kecepatan dan efisiensi dalam melakukan suatu pekerjaan pengetikan.

1. **Pengertian Game**

Menurut Jense Schell (2008, p37), *game* adalah suatu kegiatan pemecahan

masalah yang dilakukan dengan pendekatan yang bersifat menyenangkan.

Menurut Jesse Schell (2008, p41-43), terdapat empat buah elemen dasar yang membentuk sebuah *game*, yaitu *Mechanics*, *Story*, *Aesthetics*, dan *Technology*. Keempat elemen tersebut disebut *Elemental Tetral*.

1. *Mechanics* merupakan prosedur dan peraturan yang berlaku dalam *game*.

2. *Story* merupakan urutan kejadian yang terjadi dalam *game*.

3. *Aethetics* merupakan bagaimana penampilan sebuah *game* yang dapat

mempengaruhi pengalaman pemain.

4. *Technology* merupakan media yang menghubungkan *mechanics*, *story*, dan

*aethetcics*.

Menurut Lindsay Grace (2005, p1), dalam mendeskripsikan suatu *gameplay*.

Game dibedakan menjadi beberapa jenis, yaitu :

1. *Action*

*Action game* adalah *game* yang menekankan pada intensitas dari aksi sebagai atraksi utamanya. Respon reflek adalah kemampuan utama yang diperlukan dalam memainkan *game* bertipe *action*.

1. *Adventure*

*Adventure game* adalah *game* yang menyediakan eksplorasi dan pemecahan misteri sebagai atraksi utamanya. *Adventure game* umumnya menawarkan cerita yang mengasyikkan. Pemikiran, kreatifitas, dan rasa penasaran adalah kemampuan umum yang diperlukan untuk menjadi pemain *adventure game*.

1. *Puzzle*

*Puzzle game* adalah *game* yang menyediakan *puzzle* sebagai atraksi

utamanya. Game bertipe *puzzle* biasanya diluncurkan dalam anggaran yang rendah melalui *web*.

1. *Role Playing*

*Role Playing game* (RPG) adalah *game* yang menyediakan pemain sebuah peluang untuk menghayati situasi dari karakter *game*. *Role playing game* memiliki karakter yang tergolong banyak, memiliki durasi yang panjang, dan manajemen karakter merupakan hal teknis.

1. *Simulation*

Elemen utama dari *game* bertipe *simulation* adalah kemampuan untuk

menyamai situasi pada dunia nyata. *Simulation game* menyediakan

kesenangan melalui pemeragaan.

1. *Strategy*

*Strategy game* adalah *game* yang menghibur melalui pemikiran dan

pemecahan misteri.

**2.8.1. Game Design**

Menurut Jesse Schell (2008, pXXIV), *game design* adalah suatu aksi untuk

menentukan seperti apakah sebuah *game* seharusnya.

Menurut Bob Bates (2004, p3), pembuatan *game* dimulai dengan membuat sebuah *gameplay treatment* yang dasar, yaitu sebuah dokumen berisi sekitar lima sampai enam halaman. *Treatement* tersebut pada akhirnya akan berkembang menjadi sebuah spesifikasi penuh untuk *game* tersebut, tetapi menulis *treatment* awal akan membantu untuk mematangkan pemikiran akan *game* yang akan dibuat.

Tujuan utama dari membuat *game* desain dokumen adalah untuk menarik

minat pembaca dan untuk menginformasikan kepada pembacanya mengenai gambaran keseluruhan *game* yang akan dibuat sehingga nantinya gambaran-gambaran ini dapat dikumpulkan menjadi suatu *requirement*.

**2.8.2. Game Balancing**

Dalam desain *game*, keseimbangan adalah sebuah konsep hubungan antara

komponen – komponen pada sebuah sisstem dalam *game*. Tujuannya adalah mencegah terjadinya ketidakefektifan dalam *game* tersebut. Sistem yang tidak seimbang hanya akan membuang – buang waktu untuk dikembangkan. Biasanya *game balancing* dilakukan dengan menambahkan sejumlah aturan atau penyesuaian antar komponen. Misalnya, pada *game* *first person shooter* akan ditambahkan lebih banyak senjata beserta amunisinya ketika berada pada *stage* yang lebih sulit dengan jumlah musuh yang lebih banyak**.**

* + 1. **Game Edukasi**

Game berasal dari kata bahasa Inggris yang berarti dasar permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual(*Intellectual Playability Game*) yang juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Dalam *game*, ada target-target yang ingin dicapai pemainnya.

Sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Game bertujuan untuk menghibur, biasanya *game* banyak disukai oleh anak – anak hingga orang dewasa. Game sebenarnya penting untuk perkembangan otak, untuk meningkatkan konsentrasi dan melatih untuk memecahkan masalah dengan tepat dan cepat karena dalam *game* terdapat berbagai konflik atau masalah yang menuntut kita untuk menyelesaikannya dengan cepat dan tepat. Tetapi *game* juga bisa merugikan karena apabila kita sudah kecanduan *game* kita akan lupa waktu dan akan mengganggu kegiatan atau aktifitas yang sedang kita lakukan.

Kata edukasi berasal dari bahasa Inggris yaitu education yang berarti pendidikan. Menurut Marc Prensky (2012, hal. 90), *game* edukasi adalah *game* yang didesain untuk belajar, tapi tetap bisa menawarkan bermain dan bersenangsenang. Game edukasi adalah gabungan dari konten edukasi, prinsip pembelajaran, dan *game* komputer.

Berdasarkan uraian di atas maka dapat disimpulkan *game* edukasi merupakan salah satu jenis *game* yang tidak hanya bersifat menghibur tetapi didalamnya mengandung pengetahuan yang disampaikan kepada penggunanya. Game edukasi dapat digunakan sebagai salah satu media pendidikan yang bisa digunakan sebagai media pembelajaran. Game jenis ini biasa digunakan untuk mengajak penggunanya belajar sambil bermain. Melalui proses belajar ini maka penggunanya dapat memperoleh ilmu pengetahuan, sehingga *game* edukasi merupakan terobosan baru yang digunakan dalam dunia pendidikan. Selain dikarenakan *game* jenis ini memadukan antara sisi belajar dan bermain, *game* jenis ini juga dapat digunakan untuk menarik perhatian anak-anak untuk belajar.

* + - 1. **Prinsip Game Edukasi**

Prinsip *game* edukasi menurut Foremen, beberapa prinsip yang harus di terapkan dalam aplikasi sebuah *game* adalah :

* *Individualization*

Materi pembelajaran (pengetahuan di buat sesuai dengan kebutuhan individual dari pembelajar, sedangkan *game* mengadopsi level individual dari pemain.

* *Feedback Active*

Adanya *feedback* yang sesuai dengan cepat untuk memperbaiki pembelajaran dan mengurangi keditak tahuan pembelajar terhadap materi yang di sampaikan, sedangkan *game* menyediakan *feedback*  dengan cepat dan konstektual.

* *Active Learning*

Adanya kecenderungan untuk menyertakan pelajar secara aktif dalam menciptakan ppenemuan dan pengetahuan baru yang membangun, sedangkan *game* menyediakan suatu lingkungan yang membantu terjadinya penemuan baru tersebut.

* *Motivation*

Pelajar termotivasi dengan *reward* yang diberikan dalam aktivitas permainan, sedangkan *game* melibatkan pengguna berjam-jam untuk mencapai tujuan.

* *Social*

Pengetahuan merupakan suatu proses partisipasi sosial, sedangkan *game* dapat di mainkan dengan orang lain ( seperti *game* *multiplayer*) atau melibatkan komunitas dari pecinta *game* yang sama.

* *Scaffolding*

Pelajar secara berangsur-angsur ditantang dengan tingkat kesulitan yang makin tinggi dan dapat melangkah lebih maju untuk mencapai kemeneangan dari permainan, sedangkan *game* dibangun secara multi level, pemain tidak bisa bergerak ke level yang lebih tinggi sampai dia mampu menyelesaikan permainan di level yang ada

* *Transfer*

Pelajar mengembangkan kemampuan untuk mentransfer pengetahuan dari suatu orang ke orang lain, sedangkan *game* mengijinkan pemain untuk mentransfer informasi dari suatu konteks ke konteks yang lain.

* *Assessment*

Setiap individu mempunyai kesempatan untuk menilai pelajaran mereka sendiri atau membandinkannya dengan orang lain.

* + - 1. **Dampak Positif Game Edukasi**

Ada dampak positif dan negatif sebuah *game*, sekalipun *game* edukasi. Menurut Edward, *game* memiliki banyak dampak positif antara lain :

* Game banyak diguanakn untuk mengajarkan suatu pengetahuan dan membangun keterampilan baik bidang edukasi, bisnis maupun militer
* Game efektif digunakan untuk membangun kemampuan matematika dan membaca pada anak, dibuktikan dengan penelitian yang dilakukan oleh Murphy dan kawan-kawan pada tahun 2002.
* Game terbukti efektif untuk membantu anak-anak penderita ashma dan diabetes mengelola kebiasaan hidup sehat, penelitian dilakukan oleh Lieberman pada tahun 1997 dan McPhershon dan kawan-kawan pada tahun 2006.
* Banyak bisnis menggunakan *game* edukasi untuk membangun keterampilan karyawan mereka seperti Cisco mengajarkan karyawan mereka pengenalan tools dasar dan security network melalui sebuah *game*.

Sedangkan dampak negatif dari *game* edukasi yaitu: dapat menyebabkan ketergantungan atau kecanduan apabila penggunaannya tidak dibatasi, Waktu yang dibutuhkan untuk bermain *game* ini dapat menyita waktu untuk bermain dengan teman sebaya yang dapat melatih jiwa sosial anak.

* 1. **Object Oriented Programming (OOP)**

Menurut Anbay Pillay (2007, p11), *object oriented programming* (OOP) merepresentasikan suatu usaha untuk membuat program lebih dekat dengan model seseorang berpikir bagaimana berurusan dengan dunia. *Object oriented programming* adalah suatu set dari peralatan dan metode yang memperbolehkan *softwareengineers* untuk membuat *software* yang *reliable*, *User friendly*, *maintainable, well documented*, *reasonable*.

Bahasa dari *object oriented programming* menyediakan kebutuhan untuk konsep-konsep *object oriented* yaitu *object* dan *class*, inheritance, dan *polymorphism*.

1. *Object* dan *Class*

Menurut Anban Pillay (2007, p16), *object* adalah suatu *bundle*

perangkat lunak yang terdiri dari *variables* dan *methods*. *Variable* adalah

suatu data yang dinamai olehsebuah *identifier*. *Method* adalah suatu

fungsi yang berkaitan dengan sebuah objek. Sebuah objek juga dikenal

dengan sebutan *instance*. Sebuah *instance* dapat mengacu pada sebuah

objek tertentu.

*Class* adalah sebuah *blueprint* yang mendefinisikan *variables* dan

*methods* yang umum untuk semua objek dari jenis tertentu. *Class*

mendeklarasikan *instance* dari *variables* yang dibutuhkan untuk

menampung *state* dari seluruh objek. *Class* juga akan mendeklarasikan

dan menyediakan implementasi dari *instance* *methods* yang dibutuhkan

untuk beroperasi pada *state* dari objek.

Menghilangkan suatu *variables* objek dengan perlindungan

methodnya disebut dengan *encapsulation*.

*Encapsulation* yang terkait dengan *variables* dan *methods* menjadi sebuah *bundle* perangkat lunak yang rapi adalah hal yang mudah namun merupakan ide yang bagus dan menyediakan dua keuntungan kepada

pengembang *software* :

a. Modularity

*Source* dari *code* untuk sebuah objek dapat ditulis dan dijaga

secara terpisah dari *source code* untuk objek yang lain. Objek

juga dapat dengan mudah beredar melalui sistem.

b. *Information-hiding*

Sebuah objek memiliki sebuah *public interface* yang objek lain

dapat gunakan untuk berkomunikasi dengan objek tersebut.

Objek dapat menjaga informasi yang *private* dan *method* yang

dapat diganti kapan pun tanpa mempengaruhi objek yang

bergantung kepadanya.

2. *Inheritance*

Menurut Anban Pillay (2007, p80), *inheritance* mengacu pada

fakta bahwa sebuah *class* dapat mewarisi sebagian atau seluruh

strukturnya dari *class* lain. *Class* yang diwarisi disebut sebagai *subclass*

dan *class* yang mewarisi disebut *superclass*.Sebuah *subclass* dapat

menambahkan ke dalam struktur dan aksi yang diwariskannya.*Subclass*

juga dapat mengganti atau memodifikasi perilaku yang diwariskan.

3. *Polymorphism*

Menurut Deitel (2002, p472-473), dengan *polymorphism*, memungkinkan untuk merancang dan mengimplementasikan sistem yang mudah ditambahkan (*easily extensible*).*Class* yang tidak terdapat saat pembuatan program dapat ditambahkan dengan sedikit atau tanpa modifikasi pada bagian program yang diproses secara *general*.

Terdapat dua tipe dari *polymorphism*, yaitu :

a. *Overloading*

*Overloading* mengacu pada kemampuan untuk mendefinisikan

beberapa *method* dengan nama yang sama dalam sebuah *class*

selama *method* tersebut memiliki parameter yang berbeda (Deitel,

2002, p288).

b. *Overriding*

*Overriding* terjadi ketika sebuah *subclass* mempunyai *method*

yang memiliki *signature* yang sama dengan *superclassnya*.

*Method* pada *subclass* akan meng-*override* *method* pada

*superclass*. (Deitel, 2002, p455-456).

* 1. **Logika Boolean**

*Logical boolean* Penamaan Aljabar Boolean sendiri berasal dari nama seorang matematikawan asal Inggris, bernama George Boole. Dialah yang pertama kali mendefinisikan istilah itu sebagai bagian dari sistem logika pada pertengahan abad ke-19. Boolean adalah suatu tipe data yang hanya mempunyai dua nilai. Yaitu *true* atau *false* (benar atau salah). Pada beberapa bahasa pemograman nilai *true* bisa digantikan 1 dan nilai false digantikan 0. Dibawah ini teknik yang dipakai dalam logika boolean. Gerbang logika atau sering juga disebut gerbang logika boolean merupakan sebuah sistem pemrosesan dasar yang dapat memproses input-input yang berupa bilangan biner menjadi sebuah output yang berkondisi yang akhirnya digunakan untuk proses selanjutnya. Operator dan logika boolean ada 7, yaitu:

1. Gerbang AND

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Tabel 2.1 Tabel Kebenaran Gerbang AND

Gerbang AND Gerbang AND akan berlogika 1 atau keluarannya akan berlogika 1 apabila semua masukan / inputannya berlogika 1, namun apabila semua atau salah satu masukannya berlogika 0 maka outputnya akan berlogika 0.

1. Gerbang OR

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Tabel 2.2 Tabel Kebenaran Gerbang OR

Gerbang OR akan berlogika 1 apabila salah satu atau semua inputan yang dimasukkan bernilai 1 dan apabila keluaran yang di inginkan berlogika 0 maka inputan yang dimasukkan harus bernilai 0 semua.

3. Gerbang NOT

|  |  |
| --- | --- |
| Input | Output |
| 0 | 1 |
| 1 | 0 |

Tabel 2.3 Tabel Kebenaran Gerbang Not

Gerbang NOT berfungsi sebagai pembalik (Inverter), yang mana outputnya akan bernilai terbalik dengan inputannya.

4. Gerbang NAND

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Tabel 2.4 Tabel Kebenaran NAND

Gerbang NAND akan bernilai / outputnya akan berlogika 0 apabila semua inputannya bernilai 1 dan outpunya akan berlogika 1 apabila semua atau salah satu inputannya bernilai 0.

1. Gerbang NOR

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Tabel 2.5 Tabel Kebenaran NOR

Gerbang NOR merupakan gerbang logika yang outputnya akan berlogika 1 apabila semua inputannya bernilai 0, dan outpunya akan berlogika 0 apabila semua atau salah satu inputannya inputannya berlogika 1.

1. Gerbang XOR

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Tabel 2.6 Tabel Kebenaran XOR

Gerbang XOR merupakan kepanjangan dari Exclusive OR yang mana keluarannya akan berlogika 1 apabila inputannya berbeda, namun apabila semua inputanya sama maka akan memberikan keluarannya 0.

1. Gerbang XNOR

|  |  |  |
| --- | --- | --- |
| Input A | Input B | Output |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Tabel 2.7 Tabel Kebenaran XNOR

Gerbang XOR merupakan kepanjangan dari Exclusive NOR yang mana keluarannya akan berlogika 1 apabila semua inputannya sama, namun apabila inputannya berbeda maka akan memberikan output berlogika 0.

* 1. **Game Engine**

Game engine adalah kerangka dari *game* yang akan kita buat. Game engine yang membuat keputusan dalam menentukan frame sampai menentukan artwork yang ada di dalam scene. Modern 3D *game* engine sekarang dibanjiri dengan script-script yang cermat sehingga memenuhi tujuan yang dimaksudkan (Goldstone, 2009, p1).

* + 1. **Unity Engine**

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan *game* *multi platform* yang didesain untuk mudah digunakan. Unity itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Editor pada Unity dibuat degan *user interface* yang sederhana. Editor ini dibuat setelah ribuan jam yang mana telah dihabiskan untuk membuatnya menjadi nomor satu dalam urutan rankking teratas untuk editor *game*. Grafis pada unity dibuat dengan grafis tingkat tinggi untuk OpenGL dan directX. Unity mendukung semua *format* *file*, terutamanya *format* umum seperti semua *format* dari *art applications*. Unity cocok dengan versi 64-bit dan dapat beroperasi pada Mac OS x dan Windows dan dapat menghasilkan *game* untuk Mac, Windows, Wii, iPhone, iPad dan Android.

Unity secara rinci dapat digunakan untuk membuat *video game 3D, real time* animasi 3D dan visualisasi arsitektur dan isi serupa yang interaktif lainnya. Editor Unity dapat menggunakan *plugin* untuk *web player* dan menghasilkan *game* browser yang didukung oleh Windows dan Mac. *Plugin web player* dapat juga dipakai untuk *widgets* Mac. Unity juga akan mendukung  *console* terbaru seperti PlayStation 3 dan Xbox 360. Pada tahun 2010, telah memperoleh *Technology Innovation Award* yang diberikan oleh *Wall Street Journal* dan tahun 2009, Unity Technology menjadi 5 perusahaan *game* terbesar. Tahun 2006, menjadi juara dua pada *Apple Design Awards*.

Dalam Unity disediakan berbagai pilihan bahasa pemrograman untuk mengembangkan *game*, antara lain *JavaScript, C#,* dan *BooScript*. Namun meskipun disediakan tiga bahasa pemrograman.

*Unity* mensupport pembuatan *game* 2D dan 3D, namun lebih ditekankan pada 3D. Pengembangan *game* lebih ditekankan pada desain dan tampilan visual daripada pemrograman.

Unity membuat produksi *game* menjadi lebih mudah dengan memberikan beberapa logika untuk membangun sekenario *game* yang sudah dibayangkan. Unity salah satu *game engine* yang cukup mudah digunakan, hanya membuat objek dan diberikan fungsi untuk menjalankan objek tersebut. Dalam setiap objek mempunyai variabel, variabel inilah yang harus dimengerti supaya dapat membuat *game* yang berkualitas.

Berikut ini adalah bagian-bagian dalam Unity:

1. *Assets*

*Assets* adalah tempat penyimpanan dalam Unity, suara, gambar, video,

tekstur, semua yang ingin dipakai dalam unity disimpan di dalam *assets*.

2. *Scenes*

*Scenes* adalah sebuah area yang berisikan konten-konten dalam *game*,

seperti membuat sebuah *level*, membuat menu, tampilan tunggu, dsb.

3. *Game Objects*

Ketika sebuah barang di dalam *assets* dipindah ke dalam *scenes*, maka

benda tersebut berubah menjadi *game objects*. Dimana benda tersebut bisa

digerakkan, diatur ukurannya dan diatur rotasinya.

4. *Components*

*Components* dapat dimasukkan ke dalam *game objects* untuk menciptakan

reaksi baru, seperti *collision,* memunculkan partikel, dsb. Intinya

*components* membuat reaksi baru di dalam *game objects*.

5. *Script*

*Script* yang dapat digunakan dalam unity ada tiga, yaitu Javascript, C#

dan BOO. Tetapi Unity tidak menyediakan cara untuk menggunakan script

tersebut. *Script* dalam Unity tidaklah begitu sulit karena *script* tidaklah

bertele-tele, melainkan langsung menunjuk kepada aksi yang ingin

diberikan. Penggunaan *script* dalam Unity harus menggunakan program

lain yang telah disediakan oleh Unity seperti Mono Develop dan Visual Studio.

6. *Prefabs*

*Prefabs* adalah tempat untuk menyimpan satu jenis *game objects*, sehingga

mudah untuk diperbanyak. *Prefabs* juga mempermudah dalam pembuatan

objek-objek yang kompleks, tetapi tujuan utama *prefabs* adalah untuk

mempermudah memunculkan banyak objek dalam seketika.