

2. LANDASAN TEORI

2.1. Tinjauan Studi

Penelitian sebelumnya berguna bagi penulis untuk dijadikan pedoman serta menjadi pegangan penelitian yang akan penulis lakukan sehingga nantinya dengan adanya penelitian yang sebelumnya dapat membantu dan mempermudah penulis dalam menyelesaikan penelitiannya sesuai dengan tema dan membuat system yang lebih kompleks dan menarik pengguna. Penelitian yang terkait dan yang digunakan sebagai referensi *game* Battleship adalah Rancangan bangun aplikasi *Game* Battleship *Multiplayer* pada jaringan LAN (M.Fikri 2013).

Pada Jurnal yang berjudul Rancangan bangun aplikasi *Game* Battleship *Multiplayer* pada jaringan LAN, Tujuan penelitian adalah untuk membuat sebuah permainan battleship yang dapat dimainkan menggunakan jaringan LAN. Metode yang di gunakan dalam penelitian tersebut adalah *Object Oriented Analysis*. Cara kerja Metode Object Oriented Analysis adalah mempelajari permasalahan dengan menspesifikasikannya atau menobeservasi permasalahan tersebut. Bahasa pemrograman yang digunakan adalah *Visual Basic* 6.0. Pembuat jurnal juga melakukan beberapa wawancara yang dilakukan dipusat permainan *game* dalam seminggu mengenai *game* apa yang sering dimainkan dan *game* apa yang perlu dikembangkan.

2.2. Definisi *Game*

Game sudah banyak dikenal oleh manusia sejak dahulu, hampir semua orang pernah memainkan *game* dikarenakan saat ini *game* merupakan kebutuhan bagi manusia, berikut adalah beberapa definisi daripada *game* sebagai berikut:

- A. *Game* merupakan suatu sistem atau program dimana satu atau lebih pemain mengambil keputusan melalui kendali pada objek didalam *game* untuk suatu tujuan tertentu (Jasson,2009).

- B. *Game* adalah suatu perbuatan yang mengandung keasikan dan dilakukan atas kehendak diri sendiri, bebas tanpa paksaan dengan bertujuan untuk memperoleh kesenangan pada waktu melakukan kegiatan tersebut (Ahmadi dan Sholeh 2005).
- C. *Game* adalah suatu kekuatan yang sangat menyenangkan dan dapat merupakan cara atau alat pendidikan yang bersifat mendidik (Andang Ismail, 2006).
- D. *Game* adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, ditentukan oleh aturan, yang menghasilkan hasil yang terukur (Salen dan Zimmerman, 2003).
- E. *Game* adalah salah satu kebutuhan sekaligus menjadi masalah besar bagi para pengguna komputer, hal tersebut dikarenakan untuk dapat menjalankan suatu *game* dengan nyaman, maka dibutuhkan spesifikasi komputer yang relatif tinggi dibanding komputer untuk penggunaan biasa (Wijaya Ariyana & Deni Arifianto, 2009).

2.3. *Genre Game*

Andrew Rollings dan Dave Morris (2004), *game* memiliki *genre* yang luas, antara lain :

- A. *Sports – Type game* ini terdiri dari *game* yang menciptakan tiruan dari *sports* asli jadi pemain bisa merasakan seperti bermain *sports* asli hanya dengan memainkan *game*-nya.
- B. *Action – Type game* ini menyediakan pemain dengan beberapa rintangan dan musuh yang harus dikalahkan dengan menggunakan reaksi cepat sang pemain dan biasanya tidak memberikan banyak teka-teki pada pemain.
- C. *Adventure – Type game* ini terdiri dari *game* yang ada beberapa tantangan untuk pemain selama pemain mengendalikan 1 atau lebih karakter yang berkelana tempat yang disediakan.
- D. *Strategy* – Pemain diberi kesempatan untuk memikirkan strategi untuk menyelesaikan masalah atau mendapatkan gol, biasanya *setting*-nya tentang peperangan.

- E. *Simulation – Type* ini menyediakan simulasi dari aktifitas asli, jadi pemain diharapkan untuk mempunyai pengalaman yang sama tanpa melibatkan aktifitas yang asli.
- F. *Puzzle – Game* ini tentang pemain mencoba untuk menyelesaikan teka-teki yang diberikan oleh *game*.
- G. *Role Playing Game (RPG)* – Dalam jenis ini pemain diharapkan untuk mengikuti cerita dan diberikan 1 karakter atau lebih untuk dikendalikan, karakter akan diberikan perjalanan dan akan mendapatkan pengalaman dengan mengalahkan musuh.
- H. *Educational* – Belajar sambil melaksanakan permainan.
- I. *Role Playing Game (RPG), video game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh atau peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain dalam berbagai parameter yang biasanya ditentukan dengan naiknya *level*. Pengembang *game* Jepang biasanya membuat RPG dengan arah cerita linear seolah karakter kita adalah tokoh dalam cerita itu, seperti *game* Final Fantasy, Dragon Quest dan Xenogears. Sedangkan pengembang *game* RPG Eropa, cenderung membuat karakter bebas memilih jalan cerita sendiri secara non-linear, seperti *game* Farcry, Elder Scroll , dan Fallout.

Beberapa *genre game* juga dapat dikombinasikan dalam tiap-tiap *game* karena *game* sangat kaya atas daya tampung dan tidak bisa dimasukan 1 *genre*.

2.4. Mode Permainan

Berdasarkan *mode, game* dibagi menjadi beberapa jenis, yaitu:

- A. *Multiplayer Online. Game* yang dapat dimainkan oleh banyak orang secara bersamaan, dari sekedar chatting ataupun melakukan misi bersama-sama. Contoh: Closers, VR chat, World of Warcraft, Ayo Dance, Lineage.

- B. *Casual Games*, *game casual* tidak kompleks, dimainkan santai dan umumnya sangat mudah untuk dimainkan. Contoh: Plant Vs Zombies, Zuma, Feeding Frenzy, Jewel Quest.
- C. *Edugames*, *video game* jenis ini dibuat dengan tujuan sebagai alat pembelajaran, umumnya *game* ini di tujukan untuk anak anak agar dapat membantu proses pembelajaran agar mereka dapat dengan mudah memahami apa yang di ajarkan. *Developer* harus memperhitungkan segala aspek agar *game* ini dapat mendidik, menambah pengetahuan dan meningkatkan keterampilan bagi yang memainkannya. Contoh: Animal Quiz, The Shape Train.
- D. *Advergemes*. Jenis *game* yang biasanya mudah dimainkan ini mengusung dan menampilkan produk atau *brand* mereka baik secara terang-terangan maupun tersembunyi. Di era sekarang ini, dunia periklanan memang sudah tidak lagi terbatas pada TV, koran, majalah, *billboard* dan radio, *video game* sekarang telah menjadi sarana iklan atau membangun *brand-awareness* yang efektif. Contoh *advergemes*: A-Mild, Rexona teens. (M Corry Fice & Very Zian, 2015)

2.5. Action Games

Action Games adalah Jenis *game* yang pertama yaitu permainan aksi (*Action games*), jenis *game* yang satu ini adalah *game* yang menggunakan refleks, akurasi, dan waktu yang tepat untuk menyelesaikan sebuah tantangan. Ini mungkin merupakan *genre* dasar dari sebuah permainan, dan yang memiliki permainan terbanyak. Dalam permainan aksi, biasanya terdapat pertempuran. (Expro ,2010).

Setiap *player* berhasil menyelesaikan tantangan dengan tepat, maka *player* akan dapat membuka stage selanjutnya. Dimana stage selanjutnya akan semakin susah dan memiliki kesulitan yang lebih tinggi daripada sebelumnya.

2.6. Firebase & Firebase API (Application Programming Interface)

API adalah fungsi fungsi untuk menggantikan bahasa yang digunakan dalam system calls dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh programmer.

Firebase adalah penyedia layanan *cloud* dengan *back-end* sebagai servis yang berbasis di San Fransisco, California. Firebase membuat sejumlah produk untuk pengembangan aplikasi *mobile* ataupun *web*. Firebase di dirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara realtime di tahun 2012 (Evangelist, 2015).

Firebase memiliki banyak library yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, iOS, *Javascript*, *Java*, Objective-C dan Node.JS. *Database* Firebase juga bersifat bisa diakses lewat REST API dan data binding untuk beberapa *framework Javascript* seperti halnya AngularJS, ReactJS, Ember.JS, dan Backbone.JS. REST API tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima push notification dari server. Pengembang juga bisa menggunakan *database* ini untuk mengamankan data mereka menggunakan server Firebase dengan rules yang ada.

2.7. Unity Engine

Unity *game engine* merupakan sebuah *game engine* dan *editor*, yang memungkinkan pengembang membuat *object*, meng-import *external asset* dan menggabungkan semuanya dengan menggunakan kode secara efisien dan cepat.

Editor pada Unity dibuat dengan *user interface* yang sederhana. Unity dapat beroperasi pada Mac OS dan Windows dan dapat menghasilkan *game* untuk Mac, Windows, Wii, iPhone, iPad dan Android. Umumnya Unity digunakan untuk membuat video *game* 3D, tetapi tidak sedikit yang menggunakan Unity untuk membuat animasi 3D dan visualisasi arsitektur. (Fajar Andriansyah, 2013).

Berikut ini adalah bagian – bagian dalam Unity:

1. *Assets*

Assets adalah tempat penyimpanan dalam Unity seperti suara, gambar, video, tekstur, semua yang dapat 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, dan sebagainya.

3. *Game Objects*

Ketika sebuah objek di dalam *assets* dimasukkan ke dalam *scenes*, maka objek tersebut berubah menjadi *game objects*. Dimana objek 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, dan sebagainya. Intinya *components* membuat reaksi baru di dalam *game objects*.

5. *Script*

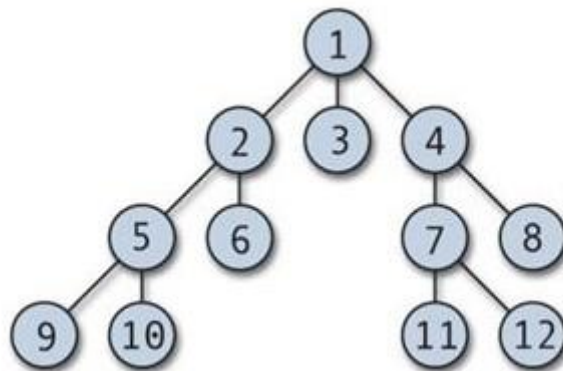
Script yang dapat digunakan dalam unity ada tiga, yaitu *javascript*, *C#* dan *BOO*. Pengolahan *script* dalam Unity secara *default* menggunakan *text-editor* Monodevelop.

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. (Rizky, 2013).

2.8. Metode / Algoritma BFS(Breadth-First Search)

Breadth-first search adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara preorder yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya. Metode penelusuran ini memeriksa semua node (simpul) pohon pencarian, dimulai dari simpul akar. Simpul-simpul dalam tingkat diperiksa seluruhnya sebelum pindah ke simpul di tingkat selanjutnya. Proses ini bekerja dari kiri ke kanan, baru bergerak ke bawah. Ini berlanjut sampai ke titik tujuan (Nurdiansah Wijaya, 2016) seperti pada gambar 2.1. Langkah metode BFS dimulai dari angka 1 setelah itu melanjutkan pencarian ke no 2, setelah itu ke no 3 dan seterusnya hingga menemukan titik tujuan.



Gambar 2.1 Metode pencarian BFS dari titik Ke 1 hingga ke 12