

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terdahulu

Berikut adalah penelitian terdahulu dari artikel yang telah di publikasikan tentang aplikasi *game adventure* yang akan saya buat berbasis android, pada kali ini akan membahas tentang aplikasi yang akan di buat.

1. **Fendi (2016).** *“Pembuatan game edukasi Petualangan si gembul sebagai pembelajaran pengenalan daerah solo raya pada anak”.*

Metode pengembangan aplikasi *game* yang digunakan adalah *Water Fall* dimulai dari analisis, perancangan, pembuatan, pengujian dan pemeliharaan. Hasil *game* dari petualangan ini adalah bergenre petualangan (*adventure*).

2. **Inandiastya Herinda, Dyah Ayu Irawati, Ridwan Rismanto (2016).** *“Pengembangan Game Edukasi 2d Platformer Petualangan Rama Sinta Berbasis Android”.*

Memecahkan suatu masalah digunakan cara atau metode tertentu yang sesuai dengan pokok masalah yang akan dibahas metode tersebut dipilih agar penelitian dapat menghasilkan data-data positif. Hasil sebuah aplikasi *Game 2D* dan merupakan *game* edukatif yang bertujuan untuk menyampaikan salah satu kisah pewayangan yang dikemas dengan bentuk cerita bergambar.

3. **Sutariyani, Wisnu Wendanto, Muhammad Nur Ali (2019).** *“Aplikasi game 2D Petualangan Bre Skater Boy Berbasis Android”.*

Merancang dan membangun *game* petualangan berbasis *android* yang dimainkan pada perangkat *mobile phone* berbasis *android*. Aplikasi *Game 2D* Petualangan Bre Skater Boy Berbasis *Android* yang bergenre *adventure* di jalan raya.

4. **Muchammad Rizal Rizqyawan (2017)** *“Pengembangan game adventure satria garuda”.*

*Game Adventure* Satria Garuda adalah *game* yang memadukan *genre* petualangan dan *genre* pertarungan yang menceritakan tentang wilayah indonesia

yang akan dikuasai oleh penjajah yaitu shadow. Tujuan dibuatkan *game* ini yaitu untuk menarik minat masyarakat agar dapat mengetahui bahwa burung garuda merupakan lambang sekaligus Jati kekuatan bangsa Indonesia dan mendorong minat masyarakat untuk memainkan *game* dalam negeri.

## **2.2 Game**

### **2.2.1 Pengertian Game**

*Game* adalah hal aktifitas yang mana banyak sudut pandang dalam menilai *game* tersebut, adakalanya mengatakan *game* berdominan dalam hal atau aktifitas untuk bermain, adapula yang mengatakan *game* menjadi sarana pembelajaran. Seseorang bisa dikatakan menjalankan peran disaat seseorang tersebut bermain *game*. Dalam dunia perindustrian video *game* merupakan media yang efektif yang banyak di kembangkan sehingga menjadi awal mulanya *game*.

*Game* yang memberikan unsur pendidikan atau content pendidikan bisa disebut dengan *game* edukasi. Selain bermain juga memberikan nuansa pendidikan dan memberikan pelajaran yang bermanfaat dalam menunjang ilmu pengetahuan adalah salah satu tujuan dari *game* edukasi. Dengan terpopulernya *game*, maka banyak para pendidik yang mempunyai skil dalam progamer berfikir bahwa mereka mempunyai peluang dan berinisiatif menerapkannya pada kurikulum dengan cara penggunaan perindustrian yang berbasis *game* dengan catatan desain antarmuka yang bersifat interkatif juga mengandung unsur menyenangkan. (Mildayanti : 2016).

### **2.2.2 Jenis Game**

Jenis *game* bias disebut dengan istilah genre *game*. Selain berarti jenis, *genre* juga berarti gaya atau format dari sebuah *game*. Menurut (Samuel, 2014) format sebuah *game* bias murni sebuah *genre* atau bisa merupakan campuran (*hybrid*) dari beberapa *genre* lain.

Jenis jenis *game* yang ada menurut (Samuel, 2014) adalah sebagai berikut :

1. *Maze game*

Jenis *game* ini biasanya menggunakan maze sebagai setting atau latar belakang *game*. Jenis *game* maze ini termasuk jenis *game* yang paling awal muncul, contoh *game* ini adalah *Pacman* dan *Digger*.

## 2. *Board Game*

*Game* jenis ini sama dengan *game* board tradisional seperti monopoli, hanya saja permainan tradisional ini di mainkan di computer.

## 3. *Card Game*

Jenis kartu *game* ini tidak jauh berbeda dari *game* tradisional aslinya. Namun tampilanya lebih bervariasi dari *game* tradisional. *Game* ini juga termasuk *game* yang awal muncul. Contoh *game* ini adalah *Solitaire* dan *Hearts*.

## 4. *Battle Card Game*

Contoh *game* ini yang populer yaitu *Battle Card Pokemon*. *Game* ini jarang ditemukan di Indonesia. Film kartun yang bercerita tentang permainan *battle card* ini pernah ditayangkan di stasiun televisi Indonesia.

## 5. *Quiz game*

Jenis ini merupakan *game* bentuk kuis. Contoh *quiz game* yang pernah beredar yaitu kuis *Who Wants To Be Millionaire*.

## 6. *Puzzle game*

Jenis *game* ini memberi tantangan dengan cara menjatuhkan atau melenyapkan sesuatu dari sisi atas ke bawah atau dari kiri ke kanan. Contoh *game* ini adalah *Tetris*.

## 7. *Shoot Them Up*

*Game* jenis ini biasanya musuh berbentuk pesawat atau bentuk lain yang datang dari arah kanan, kiri atau atas yang harus kita tembak sebanyak dan secepat mungkin. Dulu *game* ini berbentuk dua dimensi (2D), tetapis sekarang sudah berkembang menggunakan efek tiga dimensi (3D).

## 8. *Side Scoller Game*

Saat pertama kali muncul *game* ini berbentuk 2D, sekarang sudah banyak yang di buat menggunakan efek 3D. *Game* jenis ini pemain diharuskan bergerak searah di alur yang sudah disediakan. Diharuskan untuk berjalan, melompat, merunduk

serta menghindar dari rintangan-rintangan. Contoh *game* ini yang populer yaitu *Mario Bros* dan *Prince of Persia*.

#### 9. *Fighting Game*

Jenis *game* ini sesuai dengan namanya berisi tentang pertarungan. Contoh *game* ini yaitu *Street Fighter*, *Samurai Showdown*, *Virtual Figther* dan *Kungfu*.

#### 10. *Racing Game*

Rancing *game* adalah *game* tentang balapan. Contoh *game* ini adalah *Need For Speed Underground* dan *Toca Race Driver*.

#### 11. *Turn-Based Strategi Game*

Pemain dalam *game* melakukan gerakan setelah pemain lain melakukan gerakan jadi saling berganti. Contoh *game* yang terkenal adalah *Empire* dan *Civilization*.

#### 12. *Rael-Time Strategy Game*

*Game* ini seperti *game Turn-Based Strategi* (TBS), namun pada *game* ini pemain tidak perlu menunggu pemain lain. Pemain tercepatlah yang akan menang. Contoh *game* ini yaitu *Warcraft*.

#### 13. *SIM Game*

*Game* merupakan bentuk *game* simulasi, disini pemain membangun sebuah area, kota, negara atau koloni. Contoh *game* ini yaitu *Ship Simulator*, *Train Simulator* dan *Crane Simulator*.

#### 14. *First Person Shooter*

Disebut *First Person Shooter* karena pandangan pemain adalah pandangan orang pertama. Banyak baku tembak dan *game* ini mengutamakan kecepatan gerakan. Contoh *game* ini yaitu *Counterstrike* dan *Doom*.

#### 15. *First Person Shooter 3D Vehicle Based*

*Game* ini sama dengan FPS hanya saja pandangan pemain bukan dari orang pertama, tetapi dari kendaraan atau mesin yang di gunakan. Kendaraan itu bisa berupa tank atau kapal.

#### 16. *Third Person 3D Game*

*Game* ini juga hampir sama dengan FPS hanya sudut pandang pemain merupakan sudut pandang orang ketiga.

### 17. *Role Playing Game*

Jenis *game* ini pemainnya memainkan sebuah tokoh atau karakter. Biasanya ada alur cerita yang harus dijalankan. Contoh *game* ini adalah *Legacy of Kain*, *Blade of Sword* dan *Divinity*.

### 18. *Adventure Game*

*Adventure Game* merupakan *game genre* petualangan. Sepanjang perjalanan pemain akan menemukan peralatan yang akan disimpan dan berguna sebagai petunjuk perjalanan. Contoh *game* ini yaitu *Sam and Max* atau *Beyond and Evil*.

### 19. *Educational and Edutainment*

*Game* ini lebih mengacu pada isi dan tujuan dari *game*. *Game* ini bertujuan memancing minat belajar anak sambil bermain. Contoh *game* ini adalah *Game Bobby Bola*.

### 20. *Sports*

Jenis *game* ini memiliki tema olahraga. *Game* yang menyetengahkan genre olahraga disebut *Sport Game*.

## 2.2.3 Tahapan dalam pembuatan game

Berikut ini tahapan-tahapan pembuatan *game* :

### 1. Menentukan *Genre Game*

Tahap paling awal dalam pembuatan sebuah *game* adalah menentukan *genre* atau tema dari *game* tersebut. ini merupakan salah satu hal yang harus dilakukan agar kedepannya semua fokus dan tidak melenceng dari tema tersebut. Hal ini dilakukan juga untuk menjaga kesolidan antar tiap bagian dari tema tersebut dimana jika semua setuju dan menyukai tema diangkat maka ini juga dapat meningkatkan semangat dalam menyelesaikan *game* tersebut. *Genre game* sendiri tergolong banyak antara lain :

- a) *Action Games*
- b) *Strategi Games*
- c) *Role-Playing Games* (RPG)
- d) *Sport Games*
- e) *Vehicle Simulation*

f) *Construction and Management Simulation*

g) *Adventure Games*

## 2. Menentukan *Tool*

Dalam membuat *game* tentu kita memerlukan suatu *tool* yang digunakan dalam pembuatan *game* diantaranya :

a) *Construct 2*

b) *Game marker studio*

c) *Unity*

d) *Godot gameengine*

e) *Unreal engine 4*

## 3. Menentukan *Game play*

*Gameplay* yang dimaksud disini adalah alur sistem atau sistem dari *game* itu sendiri. Biasanya tampilan-tampilan yang menarik dan dekat dengan kehidupan sehari-hari akan banyak diminati oleh para pencita *gamers*. Terlebih lagi jika *game* tersebut hanya memerlukan spesifikasi yang standar untuk menjalankannya. Disinilah kepandaian *Game Designer* sangat diperlukan dalam menentukan alur dan aturan *game* tersebut.

## 4. Menentukan Grafis

Secara sederhana grafis dapat dibagi menjadi 3 macam, antara lain :

a) Kartun

b) *Semi-Real*

c) *Real*

Dalam memilih grafis kita juga harus menyesuaikan dengan *tool* yang telah ditentukan. Dalam hal ini bagian *Game Designer* dan Artist yang terutama menentukan pengguna grafisnya.

## 5. Menentukan Suara

Penentuan dan penempatan suara dalam sebuah *game* mempunyai pengaruh yang cukup besar dalam meningkatkan ketertarikan para *gamers* terhadap *game* yang dibuat. Dalam hal inilah dibutuhkan keahlian dari bagian Composer untuk menentukan suara/musik untuk *game* tersebut.

## 6. Pembuatan

Sebelum melakukan proses sebuah *game*, adakalanya anda benar-benar sudah mempersiapkan *tool-tool* yang nantinya akan anda gunakan dalam pembuatan sebuah *game*, jika anda tidak mampu untuk menyediakan semuanya anda dapat meminta bantuan dari orang-orang sekitar anda.

## 7. *Publisihing*

Dalam tahapan terakhir ini jika sudah merasa *game* yang dibuat sudah cukup berkompeten, tidak ada salahnya untuk memulai mempublikasikan *game* yang dibuat. Proses publikasian dapat dilakukan dengan cara melalui media-media elektronik, cetak dan sebagainya.

## 2.3 *Android*

*Android* merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis *linux*, android merupakan OS (*Operating System*) mobile yang tumbuh di tengah OS lainnya yang berkembang saat ini. OS lainnya seperti *Windows Mobile*, *i-phone OS*, *Symbian* dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ke tiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka (Nazruddin, 2012).

Berikut adalah *versi Android* diantaranya :

- a. *Android versi 1.1*
- b. *Android versi 1.5 (Cupcake)*
- c. *Android versi 1.6 (Donut)*
- d. *Android versi 2.0/2.1 (Éclair)*
- e. *Android versi 2.2 (Froyo)*
- f. *Android versi 2.3 (Gingerbread)*
- g. *Android versi 3.0 (Honeycomb)*
- h. *Android versi 4.0 (ICS : Ice Cream Sndwich)*
- i. *Android versi 4.1 (Jelly Bean)*

- j. *Android versi 5.0 (Lollipop)*
- k. *Android versi 6.0 (Marshmallow)*
- l. *Android versi 7.0 (Nougat)*
- m. *Android versi 8.0 (Oreo)*
- n. *Android versi 9.0 (pie)*

## **2.4 Unity**

*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 dengan *user interface* yang sederhana. *Tool* yang paling mudah digunakan dalam merancang *game engine* adalah menggunakan *Unity*. Hanya dengan cara memberikan beberapa objek yang dibutuhkan dan juga diberikan beberapa fungsi dari beberapa objek yang diberikan. Dalam *game* yang berkualitas sudah pasti memiliki variabel variabel dalam objeknya. Dalam *Unity* mempunyai beberapa bagian missal, tempat penyimpanan yang meliputi menyimpan gambar, video, suara, tekstur itu berada di *asset*.

Adapun yang berisikan konten konten dalam *game* missal, membuat level, menu, tampilan tunggu, berada di *scenes*. Barang yang berada dalam *assets* yang dipindah ke *scenes* yang dapat bergerak dan di atur adalah *game object*. Reaksi baru yang berada di objek missal *collision*, memunculkan partikel merupakan dari *components*. *Script* yang di gunakan dalam *Unity* adalah *C#*, *javascript*, dan *BOO*. (Tjahyadi, Sinsuw, Tulenan, & Sentinuwo, 2014)

## **2.5 Use Case Diagram**

*Use case* adalah sebuah Teknik yang digunakan dalam pembagunan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem bersangkutan, *use case* menjelaskan interaksi sistem itu sendiri dengan sistem yang ada, sebuah *use case* direpresentasikan dengan urutan langkah yang sederhana.

Perilaku sistem adalah bagaimana sistem beraksi dan bereaksi. Perilaku ini merupakan suatu aktifitas sistem yang bisa dilihat dari luar dan bisa diuji. Perilaku sistem ini dicapture di dalam *use case*. *Use case* sendiri mendeskripsikan sistem, lingkungan sistem, serta hubungan antara sistem dengan lingkungannya.

Deskripsi dari sekumpulan aksi sekuensial yang di tampilkan sistem yang menghasilkan tampak dari nilai ke aktor kusus. *Use case* digunakan untuk menyusun *behavioral things* dalam sebuah model. *Use case* direalisasikan dengan *collaboration*. Secara gambar, sebuah *use case* digambarkan dengan sebuah *ellips* dengan garis penuh, biasanya termasuk hanya namanya.

a. Manfaat *Use Case*

1. Digunakan untuk berkomunikasi dengan end user dan domain expert.
2. Memastikan pemahaman yang tepat tentang requirement/kebutuhan sistem.
3. Digunakan untuk mengidentifikasi siapa yang berinteraksi dengan sistem dan apa yang harus dilakukan sistem.
4. *Interface* yang harus dimiliki sistem.
5. Digunakan untuk verifikasi.

b. Karakteristik

1. *Use case* adalah interaksi atau dialog antara sistem dan actor, termasuk pertukaran pesa dan tindakan yang dilakukan oleh sistem.
2. *Use case* diprakarsai oleh actor dan mungkin melibatkan peran actor lain, *use case* harus menyediakan nilai minimal kepada suatu actor.
3. *Use case* bisa memiliki perluasan yang mendefinisikan tindakan khusus dalam interaksi atau *use case* lain mungkin disisipkan.
4. *Use case class* memiliki objek *use case* yang disebut skenario. Skenario menyatakan urutan pesan dan tindakan tunggal. (Arif, : 2016)

## 2.6 Activity Diagram

*Diagram activity* adalah bentuk visual dari alur kerja yang berisikan aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan dan *concurrency*. Dalam *Unified Modeling Language*, diagram aktifitas di buat untuk menjelaskan aktivitas computer. Diagram aktivitas menggambarkan alur *control*

secara garis besar. Diagram aktivitas memiliki komponen dengan bentuk tertentu, dihubungkan dengan tanda panah. Panah mengarahkan aktivitas terjadi, dari awal sampai akhir. Diagram aktivitas dapat di angap sebagai jenis alir kerja. Umumnya alir kerja tidak memiliki cara untuk menampilkan *concurrency*. Simbol penggabungan dan pemecahan pada diagram aktivitas dapat menjadi solusi untuk pemakaian yang sederhana.

Pada dasarnya *activity diagram* versi UML. Sering digunakan oleh *flowchart*. *Activity diagram* sendiri digunakan untuk menganalisis proses. Diagram ini berhubungan dengan *statechart*. *Diagram statechart* berfokus pada objek yang dalam suatu proses (atau proses menjadi suatu objek), *diagram activity* berfokus pada aktifitas-aktifitas yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. perlu diperhatikan adalah diagram aktivitas menggambarkan aktivitas sistem bukan bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity diagram* bukanlah sebuah *tool* yang sempurna untuk menganalisis masalah dari sistem. (arif, 2016)

## 2.7 Sequence Diagram

*Sequence diagram* (diagram urutan) adalah suatu diagram yang memperlihatkan interaksi-interaksi antara objek didalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek didalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan.

*Sequence diagram* digunaka untuk menggambarkan skenario atau rangkain langkah-langkah yang digunakan sebagai sebuah respon dari suatu kejadian untuk menghasilkan output tertentu. *Sequence diagram* diawali dari apa yang metrigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan *use case diagram*. *Sequence diagram* juga memeperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*. *Sequence diagram* memodelkan

aliran logika dalam sebuah sistem dengan cara yang visual. *Sequence diagram* biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi *method* didalam sebuah sistem. *Sequence diagram* biasanya dipakai untuk memodelkan :

1. Deskripsi tentang sistem yang adapada sebuah *use case* pada *use case diagram*, yang menggambarkan hubungan antar actor dan *use case diagram*.
2. Logika dari *method (operation, function* atau *procedure)*
3. Logika dari *service (high levelmethod)*. (arif, 2016)

## 2.8 Raden Kian Santang

Raden Kian Santang adalah salah satu sosok yang dipercaya oleh masyarakat ditanah sunda yang mengatakan bahwa Raden Kian Santang adalah anak yang sangat terkenal di Kerajaan Sunda yaitu dari Prabu Siliwangi dan Nyi Subanglarang.

Raden Kian Santang sejak kecil dilatih ilmu bela diri, maka dari itu pada saat remaja Raden Kian Santang terkenal sebagai seorang ksatria sakti dipajajaran. Raden Kian Santang memiliki ketertarikan terhadap ilmu bela diri semenjak kecil oleh karena itu Raden Kian Santang mengikuti perguruan ilmu bela diri yang ada. Peran Raden Kian Santang di Pajajaran selain seorang putra raja adalah sebagai senopati/panglima kerajaan. Dari masa kanak-kanak hingga remaja Raden Kian Santang dikenal sebagai sosok yang kuat dan memiliki banyak keingntahuan terhadap hal yang baru. Ia juga sosok yang giat berlatih sehingga tumbuh menjadi seorang remaja yang tangguh, kuat dan pemberani. Bahkan Raden Kian Santang terkenal dan tak terdandingi, ilmu beladiri yang dimilikinya sangatlah kuat dan tak tertandingi oleh siapapun. Ilmu beladiri yang dimilikinya sangatlah kebal dan tak bisa ditangkis senjata atau pukulan apapun sehingga seumur hidupnya ia tak pernah melihat darahnya sendiri. Jangankan darah, segores lukapun tak pernah didapatnya. Hingga akhirnya suatu saat Raden Kian Santang gelisah karena belum menemukan jati dirinya yang sesungguhnya. Raden Kian Santang akhirnya pun mencari tahu siapakah yang harus ia lawan agar mendapat lawan yang dapat menandingi kesaktiannya. Akhirnya Kian Santang Memutuskan untuk

pergi/berkelana untuk mencari lawan yang dapat menandingi kesaktiannya, di dalam perjalanan tersebut Raden Kian Santang bertemu beberapa musuh yang menghadang dan menghalangi perjalanannya. Akhirnya Kian Santang menemukan lawan yang sakti, dan beberapa musuh/pemberontakan berhasil dikalahkan dan akhirnya Raden Kian Santang berhasil mendapatkan pusaka tongkat sakti.(Yundiafi, 1993)

## **2.9 C#**

C# (*csharp*) merupakan bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian inisiatif kerangka *NET Framework*. Bahasa pemrograman ini dibuat berbasiskan Bahasa C++ yang telah dipengaruhi oleh aspek-aspek atau fitur bahasa yang terdapat pada Bahasa-bahasa pemrograman lainnya. Seperti *java*, *delphi*, *visual basic* dan lain-lain dengan beberapa penyederhanaan. (Kadir, : Abdul 2012)

## **2.10 Storyboard**

*Storyboard* adalah suatu sketsa gambar yang disusun denganurut berdasarkan naskah cerita, dengan *storyboard* maka pengarang cerita bisa menyampaikan ide ceritanya secara lebih mudah pada orang lain. karena dengan *storyboard* seorang pembuat cerita dengan mengikuti gambar-gambar yang telah disajikan, sehingga dapat mendapatkan presepsi yang sama dengan ide cerita yang dibuat. *Storyboard* juga didefinisikan sebagai naskah yang penyajiannya berbentuk sketsa gambar secara berurutan, bermanfaat untuk membuat dan memudahkan alur cerita dalam pengambilan gambar. Iscahyadi, (2018).