

## BAB II TINJAUAN PUSTAKA

### A. Pengertian *Game* dan Komponennya

*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, p.2).

*Game* berasal dari kata bahasa Inggris yang berarti dasar permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual yang bisa diartikan sebagai arena keputusan dan aksi pemainnya. (Antonius, 2011, [www.chikhungunya.wordpress.-com](http://www.chikhungunya.wordpress.-com)).

#### 1. Konsep Pengembangan *Game*

Secara garis besar, konsep pengembangan *Game* terbagi menjadi beberapa bagian, yaitu:

##### a. Perancangan Konsep *Game*

Di bagian ini adalah menentukan ide dasar *Game* secara keseluruhan dan jenis dari *Game* yang dikembangkan adalah *Role playing Game* (RPG). Dan pada bagian ini juga ditentukan tujuan dan sasaran *Game* yang dikembangkan misalnya anak-anak, remaja, atau dewasa.

##### b. Implementasi *Engine Game*

*Engine Game* merupakan inti atau mesin utama dari *Game* yang mengendalikan seluruh aktivitas dan interaksi yang terjadi dalam *Game*. Dalam pengembangan *Game* ini, digunakan *RPG Maker VX Ace* sebagai *engine Game*. Biasanya terdapat empat komponen utama yang membentuk *engine Game*. Komponen-komponen tersebut antara lain:

###### 1) *Database* (Basis Data)

Basis data dalam *Game* berguna untuk mendaftarkan dan menyimpan data tentang objek-objek maupun kejadian di dalam *Game*. Misalnya daftar karakter, daftar peta, tingkatan/*level* ataupun data yang disimpan ke dalamnya.

## 2) *Game Objects* (Objek *Game*)

Objek *Game* merupakan objek yang akan pemain kendalikan di dalam *Game* tersebut, bagian ini mengimplementasikan objek *Sprite Game* dan lingkungan pada *Game*. *Sprite* adalah objek yang dapat dikendalikan oleh pemain atau *Artificial Intelligence* (AI). Objek lingkungan pada *Game* adalah latar belakang dan benda. Latar belakang adalah wadah atau tempat di mana objek *Sprite* akan dikendalikan. Benda adalah objek yang dapat berinteraksi dengan *Sprite*.

## 3) *Artificial Intelligence* (Kecerdasan Buatan)

*Artificial intelligence* adalah kecerdasan buatan yang diberikan pada suatu objek agar dapat berlaku secara acak ataupun teratur. Tujuan pemberian kecerdasan buatan adalah agar pemain tidak sembarangan dalam mengambil keputusan dan mengembangkan semangat kompetisi.

## 4) *Game Control* (Kendali *Game*)

Pengendalian *Game* merupakan interaksi antara pemain dengan objek *Game* sebagai hasil keputusan pemain tersebut. Kendali juga membatasi hal apa yang dapat dilakukan oleh objek *Game*.

### c. Penanaman *Visual* dan *Audio* pada *Game*

Aspek *visual* merupakan segala sesuatu yang dapat dilihat oleh pemain. Aspek *visual* dapat berupa model dua dimensi atau tiga dimensi. Model dua dimensi dibuat dengan garis-garis pada bidang, pewarnaan, *masking*, gradasi dan efek-efek percampuran warna. Sedangkan antarmuka biasanya berisi informasi yang diperlukan oleh pemain dalam memainkan *Game* tersebut.

### d. Pengujian

Pengujian dilakukan untuk menjaga mutu *Game* yang dibuat. Pengujian meliputi pengujian *engine Game*, pengujian interaksi *Game* dengan pemain melalui pengendali/*controller*, *debugging*, dan perbaikan.

## 2. Pengertian RPG

RPG adalah salah satu *genre Game*, yang merupakan singkatan dari *Role Playing Game*. Sesuai namanya, dalam *Game* ini *player* berperan sebagai orang lain dan biasanya mengendalikan lebih dari satu tokoh, yang akan dimainkan dalam waktu yang bersamaan. Di dalam RPG, pemain akan mengalami perkembangan dari karakter yang diperankannya baik secara kekuatan, kemampuan bahkan sampai sifat dan emosinya seiring dengan berjalannya plot cerita utama. Di dalam RPG juga banyak percabangan cerita, sehingga sebagian RPG memiliki lebih dari satu *ending*. (Jasson,2009, p.53-54)

### a. Elemen Khas RPG

#### 1). *Storyline* dan *Character Development*

Cerita dalam RPG harus kreatif, inovatif, dan mengalir seiring dengan perkembangan karakter utama dan tokoh pendukungnya. Biasanya cerita dalam RPG mengandung unsur-unsur seperti berikut:

- a) Sang pahlawan
- b) Pasangan sang pahlawan
- c) Kawan-kawan sang pahlawan
- d) Sang pengkhianat
- e) Musuh utama yang tersembunyi

#### 2). *Battle System*

##### a). *Wait-and-see battle*

RPG yang menggunakan sistem ini disebut juga *old-school RPG*, karena memang sistem ini dulu sering digunakan dan pertama kali muncul. Pada sistem, *player* memberi perintah pada setiap *hero* pada posisi masing-masing. Setelah selesai, *player* tinggal *wait and see*. Serangan dilakukan bergiliran antara *hero* dan musuh. Periode ini dinamakan satu ronde.

##### b). *Real-time battle*

Pada RPG ini *player* langsung berhadapan dengan musuh. *Player* tidak memilih berbagai macam perintah pada *heronya*,

namun langsung menjalankan *hero* untuk bertarung. Jadi, tidak ada yang namanya perintah *Attack*, *Defense*, *Magic*, *Item*, dan sebagainya saat bertarung.

3). *Menu system*

*Menu system* berisi petunjuk dan beberapa manual yang ada dalam *RPG* tersebut. Menu ini sangat membantu pemain dalam memahami dan mengkostumisasi karakternya sehingga membuat sang karakter sesuai dengan keinginan sang pemain.

4). *Status Effect*

Berikut status-status yang biasanya muncul pada *Game RPG* yaitu:

a). *Silence*

Status ini menyebabkan karakter pemain yang bersangkutan tidak bisa mengeluarkan kemampuannya sampai efek dari status ini menghilang.

b). *Confuse*

Status ini cukup merugikan karakter pemain. Karakter yang terkena ini tidak bisa membedakan kawan atau lawan, sehingga semua akan dilawan, tidak terkecuali kawan sendiri.

c). *Darkness*

Ini membuat akurasi serangan menjadi berkurang, sehingga serangan biasanya meleset.

d). *Sleep*

Salah satu status yang menjengkelkan, seseorang yang terkena mantra ini akan tertidur sampai beberapa saat.

## **b. Kategori-Kategori *RPG***

1). *Fantasy*

Kategori ini berseting rata-rata di dunia pertengahan Eropa dan menggunakan sihir, serta agama yang berbeda dengan dunia nyata. Contoh: *Final Fantasy*, dan *Lord Of The Ring*.

2). *Science Fiction*

Ini terbalik dari fantasi *RPG*, setingan justru ada di zaman yang jauh lebih maju dari zaman sekarang. Sihir diganti dengan istilah semacam *Telekinesis* atau *Psy*. Contoh: *Star Wars*, dan *Enter the Matrix*.

3). *Historical*

*RPG* ini lebih menekankan aspek sejarah, walaupun sering tertukar dengan *Fantasy RPG*. Yang membedakan, *RPG* ini mengambil seting di dunia nyata, namun di masa lalu. Contoh: *Age of Empire*.

4). *Horror*

Mungkin *RPG* bergenre ini agak jarang, namun yang jelas ada dua versi yaitu manusia biasa melawan makhluk supernatural atau sebaliknya. Contoh: *Resident Evil 4*, *Parasite Eve*, dan *Kult*.

5). *Funny*

*RPG* ini mengandalkan unsur asyiknya bermain tanpa memerlukan cerita yang berat dan kostumisasi karakter yang rumit. Contoh: *Super Mario RPG* dan *Chocobo Dungeon 2*.

6). *Multigenre*

Membedakan *multigenre* susah sekali, karena genre ini tercampur dengan genre lain. Contoh: *Castlevania (horror + fantasy)*, dan *Kingdom Hearts (fantasy + funny)*.

**c. *Rapid Application Development (RAD)***

*Rapid Application Development (RAD)* adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. *RAD* bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem hidup tradisional antara perancangan dan penerapan suatu sistem informasi. (Kendall, 2010, piyaneo.wordpress.com)

Berikut adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi menurut Kendall (2010, piyaneo.wordpress.com ).

1. *Requirement Planning* (Perencanaan Syarat-syarat)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.

2. *RAD Desain Workshop* (*Workshop* Desain RAD)

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambar sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan.

3. *Implementation* (Implementasi)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem di uji coba dan kemudian diperkenalkan kepada organisasi.

## **B. *RPG Maker VX Ace***

Sejarah singkat *RPG maker* (*RPG Tsukuru*) adalah *RPG Authoring Tool* yang dikembangkan oleh sebuah perusahaan yang bernama ASCII (Hingga versi 2000) yang kemudian dibeli lisensinya oleh *Enetbrain*.

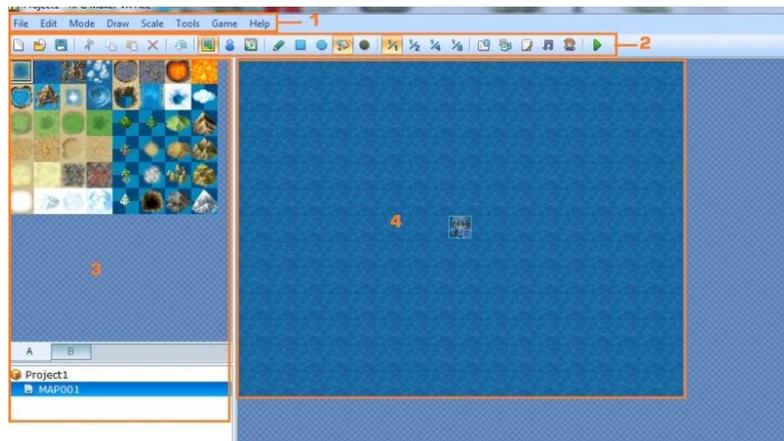
Ada 5 versi *RPG Maker* yang yaitu:

1. *RPG Maker 95-JP*
2. *RPG Maker 2000*
3. *RPG Maker XP*
4. *RPG Maker VX*
5. *RPG Maker VX Ace*

*RPG Maker VX Ace* adalah *Game Engine* buatan *Enterbrain Inc.* *Game Engine* ini adalah salah satu alternatif bagi yang ingin membuat *Game RPG* dengan mudah. *RPG Maker VX Ace* merupakan versi yang paling lengkap dibandingkan *RPG Maker* versi lainnya. Berikut adalah tampilan *RPG Maker VX Ace (RMVXA)* beserta penjelasannya.



Gambar 2.1 Tampilan *Interface RPG Maker VX Ace*



Gambar 2.2 Tampilan *Jendela RPG Maker VX Ace*

Keterangan:

1. *Menu Bar* berfungsi untuk menampilkan menu
2. *Ribbon* merupakan icon tools pada menu. Urutan *ribbon* dari kiri ke kanan:
  - a. *New Project* = Membuat *Project* Baru
  - b. *Open Project* = Membuka *Project* yang telah disimpan
  - c. *Save Project* = Menyimpan *Project* yang dibuat
  - d. *Cut* = Memotong *Event* atau lainnya
  - e. *Copy* = Menyalin *Event* atau lainnya
  - f. *Paste* = Menempelkan Hasil *Cut/Copy*
  - g. *Delete* = Menghapus *Event* atau lainnya
  - h. *Undo* = Membatalkan Perintah
  - i. *Map* = *Tool* yang berguna untuk membuat map dengan cara mengklik *tile set* dan mengkliknya lagi di *Map Sheet*
  - j. *Event* = Membuat *Event*
  - k. *Region* = Menentukan urutan letak *event*
  - l. *Pencil* = Menata *TileSet* satu persatu
  - m. *Rectangle* = Menata *TileSet* dengan bentuk kotak
  - n. *Ellipse* = Menata *TileSet* dengan bentuk Bola
  - o. *Flood Fill* = Mengisi penuh *map* dengan satu *Tile Set*
  - p. *Shadow Pen* = Memberikan bayangan pada dinding atau *tile* lainnya agar terlihat seperti gambar 3D
  - q. Skala *display* 1:1
  - r. Skala *display* 1:2
  - s. Skala *display* 1:3
  - t. Skala *display* 1:4
  - u. *Database* = Mengakses *Database*
  - v. *Resource Manager* = Mengakses *Resource Manager*
  - w. *Script Editor* = Mengakses *Script Editor*
  - x. *Soundtest* = Mengetes OST
  - y. *Character Generator* = Membuat Karakter
  - z. *Play Test* = Mengetes *Game*

### 3. *Tileset* dan *Map Data*

Dibawah *tileset* ada juga deretan Alpabet yaitu dari A-E, fungsi alpabet tersebut adalah untuk mengganti *Tileset*. *Map data* merupakan letak map yang sudah dibuat.

4. *Map Sheet*, disini adalah tempat dimana kalian menaruh *tileset* sehingga terbentuk suatu map dengan cara menggunakan *Tool Map* yang ada di deret *Ribbon*, Caranya mudah tinggal klik *Tileset* yang anda inginkan lalu taruh di *Map* dan kreasikan sesukamu

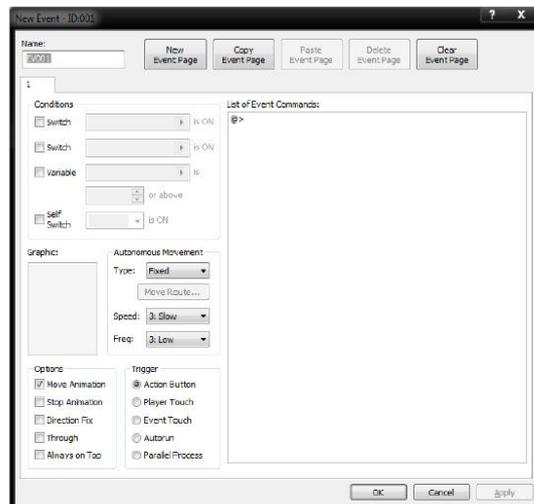
*Event page* adalah jendela pengendalian segala sesuatu yang terjadi dalam *project*. Mulai dari cerita, percakapan, *quest*, bahkan ada beberapa minigame dikendalikan dengan *event*.

#### 1. *Map event*

Adalah *event* yang dipasang di atas *map*. ini adalah tipe *event* yang paling banyak digunakan dan bersentuhan langsung dengan *player*. Biasanya digunakan untuk cerita, percakapan, *quest*, atau *minigame*.

#### 2. *Common event*

Ini adalah *event* yang bisa dipanggil dari *map event* atau *battle event*. bisa juga di set untuk aktif saat penggunaan *item* atau *skill*. bisa diset pad tab '*Common Event*' di database.



Gambar 2.3 Jendela *common event*

Keterangan:

Ketika membuat sebuah *event*, di sudut kiri atas akan terlihat settingan *event page*. berapa *event* yang terpasang dalam *tile* itu dan juga kondisi yang dibutuhkan agar *event* bisa aktif. settingannya antarlain:

- a. *Switch* : ketika *switch* yang didefinisikan disini aktif, maka *event* juga akan aktif
- b. *variable* : data berupa nomor (*integer*). jadi apabila sebuah *variable* mencapai titik tertentu dan *event* diset pada nilai variabel itu, maka *event* akan aktif.
- c. *self switch* : *switch* yang hanya berlaku untuk *event* itu sendiri. jumlah *self switch* tiap *event* ada 5 buah. biasanya digunakan pada pintu, atau chest.
- d. *Items* : *event* akan aktif apabila *party* memiliki *item* yang di set.
- e. *Actor*: *event* akan aktif saat aktor yang telah di set bergabung dalam *party*.
- f. *Action button* : *event* akan aktif apabila *player* menekan *action button* (dalam hal ini tombol 'Z' di *keyboard* atau 'spasi').
- g. *Player touch* : *event* akan aktif saat *player* bersentuhan dengan *event*.
- h. *Event touch* : *event* akan aktif saat *event* menyentuh *player*. (hampir mirip dengan *player touch*)
- i. *Autorun* : begitu *autorun* aktif, *player* akan otomatis kehilangan kendali terhadap aktor sampai *event autorun* dihentikan. dan ingat *event* autorun akan terus melakukan *loop* hinga *event* itu dihentikan baik dengan '*erase event*' atau '*switch*'
- j. *Paralel process* : mirip dengan autorun. Hanya saja, *paralel process* tidak mengakibatkan *player* kehilangan kendali atas aktor.

k. *Event Command* adalah jendela untuk memasukkan perintah.

Total perintah *event* dalam *RMVXA* ada 84. meski yang paling sering digunakan adalah *show text*, *battleprocessing*, *transfer player*, dan *set move route*.

Langkah memasukkan *event command* yaitu:

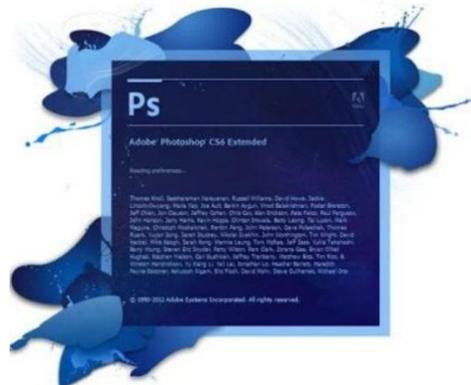
- 1) Buat sebuah *event*
- 2) *Double* klik baris yang bertuliskan "@>"
- 3) Lalu pilih *event command* yang ingin dimasukkan.  
Kalau bingung apa fungsinya, tinggal klik kanan dan klik '*whats this?*'
- 4) Setelah di set tinggal klik ok
- 5) *Event* telah bertambah

### 3. *Battle event*

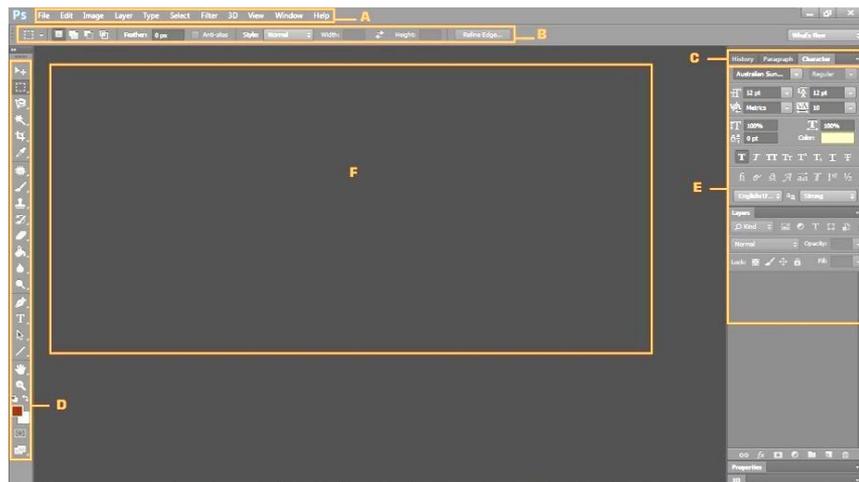
Dari namanya kelihatan jelas. ini adalah *event* yang berlangsung saat *battle* dan bisa diakses lewat *database troops*.

## C. *Adobe Photoshop*

*Adobe Photoshop CS6* merupakan seri terbaru dari program sebelumnya yaitu *Adobe Photoshop CS5*. Dengan perubahan yang menarik dengan fitur terbarunya, menjadikan *Adobe Photoshop CS6* sebagai sebuah program pengolahan gambar dan *photo* yang semakin digemari oleh para *designer* dan *photographer*. (Enterprise, 2012, p.2-5)



Gambar 2.4 *Adobe photoshop CS6*



Gambar 2.5 Tampilan jendela kerja *Photoshop*

Area kerja Adobe Photoshop dapat dilihat pada gambar 2.5, yaitu:

A :*Menu Bar*, berisi perintah utama untuk membuka *file*, *save*, mengubah ukuran gambar, filter dan lain-lain.



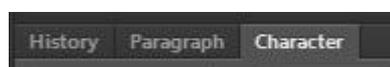
Gambar 2.6 Tampilan *Menu Bar*

B :*Option*, berisi pilihan dari *tool* yang Anda pilih. Misalnya dipilih *move*, maka garis *transform* ada di sini.



Gambar 2.7 Tampilan *Option*

C :*Pallete Well*, cara cepat untuk mengakses *palet brushes*, *tool* resets dan *Layer Comps*. Juga dapat digunakan untuk meletakkan palet yang sering digunakan.



Gambar 2.8 Tampilan *Pallete Well*

D : *Toolbox*, berisi tool untuk menyeleksi dan memodifikasi gambar. *Toolbox* berisi *Rectangular Marquee Tool*, *Elliptical Marquee Tool*, *Lasso Tool*, *Polygonal Lasso Tool*, *Magnetic Lasso Tool*, *Magic Wand Tool*, *Crop Tool*, *Healing Brush Tool*, *Spot Healing Brush Tool*, *Patch Tool*, *Red Eye Tool*, *Clone Stamp Tool*, *Background Eraser Tool*, *Magic Eraser Tool*, *Color Replacement Tool*, *Blur*, *Sharpen*, dan *Smugde Tool*, *Dogde*, *Burn*, dan *Sponge Tool*.



Gambar 2.9 Tampilan *Toolbox*

E : *Palette*, berisi jendela-jendela kecil yang di dalamnya terdapat perintah dan pilihan untuk dokumen/gambar yang sedang dikerjakan.



Gambar 2.10 Tampilan *Pallete*

F : *Gambar*, menampilkan gambar yang sedang dibuat atau diedit.