



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Video Game Design**

*Video Game* dirancang berdasarkan ide dan peraturan untuk menciptakan mekanisme permainan yang ada. Semua pemain menginginkan permainan yang dimainkan bersifat baik. Permainan yang baik diciptakan karena bersifat menyenangkan (Rogers, 2010). Ketika merancang sebuah permainan diperlukan beberapa hal yang dapat dipertimbangkan sebagai berikut (Rogers, 2010).

1. *Ideas*, bagaimana cara mendapatkannya dan di mana untuk dapat diingat.
2. *Target Audience*, di mana permainan ini cocok di target usia.
3. *Brainstorming*, bagaimana cara merancang sebuah permainan.
4. *Fun*, memulai permainan yang bersifat menyenangkan dan menghindari sifat *Un-Fun*.
5. *Stories*, alur cerita yang bersifat naratif.

*Top-down Shooting* atau disebut juga sebagai *Shoot'em up* merupakan subgenre dari *video games* yang mengkombinasikan *shooter* dengan aksi. Permainan ini diciptakan untuk pemain yang berhadapan dengan musuh yang berjumlah banyak dan menghindari tembakan atau serangan dari musuh. Pemain harus mengandalkan ketepatan waktu ketika bereaksi terhadap musuh untuk dapat menyelesaikan permainan (Bielby, 1990). Dengan menggunakan *Top-down Shooting*, seorang desainer dapat mendesain konsep permainan yang bersahabat dan tidak bersifat *graphics*.

Dalam perancangan permainan, meskipun terdapat berbagai tipe dan *genre* yang berbeda, pada dasarnya terdapat persamaan elemen dari setiap permainan yang dibuat. Elemen tersebut dibagi menjadi dua yaitu *Formal Elements* dan *Dramatic Elements* (Fullerton, 2008).

*Formal Elements* adalah elemen yang merancang struktur permainan secara keseluruhan. Tanpa adanya elemen ini, maka sebuah permainan tidak dapat dikatakan sebagai permainan. Berikut adalah *Formal Elements* menurut Fullerton (2008).

1. *Players*

*Players* merupakan peserta yang berinteraksi dengan permainan. Interaksi ini yang menentukan tipe pemain, jumlah pemain dan peran pemain di permainan.

2. *Objectives*

*Objectives* merupakan tujuan dari permainan untuk menyelesaikan masalah pada permainan sehingga mendorong motivasi pemain untuk mencapai sesuatu yang dapat diselesaikan oleh pemain.

3. *Procedures*

*Procedures* mendefinisikan tahapan atau instruksi kepada pemain untuk dapat mencapai *Objectives* pada permainan.

4. *Rules*

Merupakan aturan yang diperbolehkan atau dilarang dalam permainan yang mengatur tindakan dari pemain.

5. *Resources*

*Resources* Merupakan elemen yang ada pada sebuah permainan yang berhubungan dengan pemain untuk membantu pemain dalam permainan.

6. *Conflict*

*Conflict* merupakan permasalahan yang terjadi selama permainan berlangsung baik dari awal hingga akhir dari permainan berdasarkan *Rules* permainan.

7. *Boundaries*

*Boundaries* merupakan batasan dari permainan yang terkait dengan ruang permainan pada permainan.

8. *Outcome*

*Outcome* merupakan hasil dari permainan. Hasil tersebut dapat menghasilkan kondisi akhir dari permainan, atau sebuah hasil yang dapat dihitung selama permainan berlangsung.

*Dramatic Elements* adalah elemen yang bertujuan untuk menarik perhatian pemain untuk menciptakan sebuah pengalaman permainan yang bersifat dramatis.

Berikut adalah *Dramatic Elements* menurut Fullerton (2008).

1. *Challenge*

*Challenge* merupakan tantangan yang ada pada sebuah permainan. Tantangan tersebut harus membangun tensi permainan selama pemain menyelesaikan permainan yang dapat memberikan kepuasan terhadap pemain ketika menyelesaikan tantangan.

2. *Play*

*Play* mendefinisikan jenis dari permainan. Terdapat dua kategori jenis permainan berdasarkan kebebasan dalam aktivitas permainan yaitu *free-*

*form play* dan *rule-based play*. *Free-form play* merupakan kategori berdasarkan improvisasi dari pemain. *Rule-based play* lebih mengutamakan aturan dari permainan.

### 3. *Premise*

*Premise* merupakan latar belakang cerita dari permainan yang membangun sebuah kondisi dari permainan.

### 4. *Character*

*Character* merupakan jenis karakter yang berpengaruh pada alur cerita permainan untuk berpartisipasi dalam permainan.

### 5. *Story*

*Story* merupakan cerita pada permainan yang terjadi dari awal permainan hingga akhir dari permainan dimana pemain berperan pada cerita tersebut dan bertanggung jawab untuk menyelesaikan alur cerita.

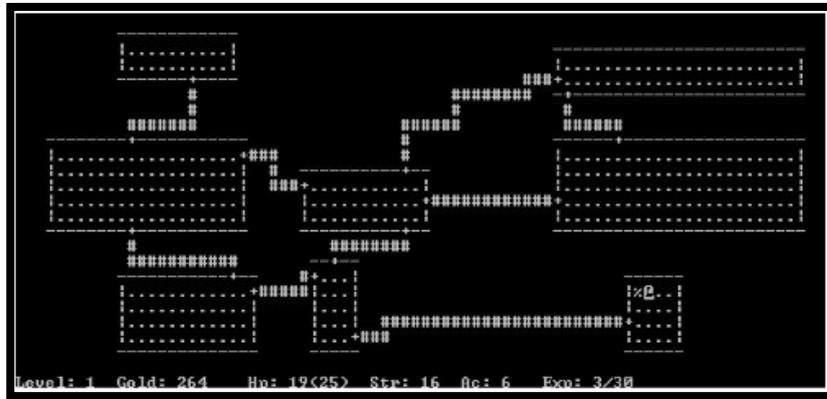
### 6. *World Building*

*World Building* merupakan deskripsi dari *game space* dimana pemain berada pada permainan yang menggambarkan karakteristik dari dunia yang berada pada permainan.

## **2.2 Roguelike**

*Roguelike* merupakan subgenre dari *video game* yang mengandalkan *Procedural Content Generation* pada *Dungeon Crawling Generation* dan sistem *permadeath* di mana mekanisme permainan yang berakhir karena kematian permanen pada pemain dan memulai permainan baru tanpa menyimpan kondisi terakhir pemain yang mati sebelumnya. Genre ini bermula dari permainan *Rogue*

pada tahun 1980 yang didesain menggunakan *simple text (ASCII Graphics)* dan menghasilkan sebuah *tiles* yang digambarkan sebagai konten labirin.



Gambar 2.1 *Rogue* Merupakan permainan yang mengandalkan *PCG* dalam menciptakan ruangan (Craddock, 2015)

### 2.3 Drunkard Walk Algorithm

*Drunkard Walk* atau disebut juga sebagai *Random Walk* adalah salah satu teknik pembuatan konten dari *Procedural Dungeon Generation* yang menghasilkan konten berdasarkan proses yang dilakukan secara *random*. *Random Walk* dimulai dari mengambil titik awal dari sebuah *grid* dan bergerak terus-menerus secara acak dan mengisi *empty grid* pada *grid* yang tersedia sehingga menghasilkan sebuah ruang labirin yang menjadi *level* dari suatu permainan tanpa ada sisa *grid* yang kosong (Yang, 2010).

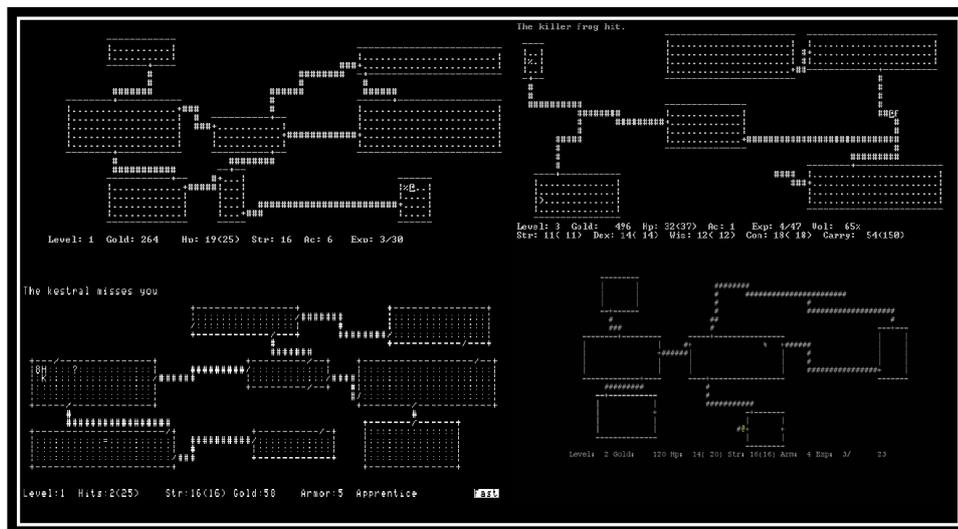
$$S_N = \sum_{i=1}^N X_i = X_1 + \dots + X_N \quad \dots(2.1)$$

Berdasarkan rumus *Random Walk* diatas, nilai  $S_N$  menunjukkan jumlah setiap langkah yang berjalan secara acak berdasarkan langkah  $X_i$ , kemudian  $S_N$  membentuk sebuah jalan berdasarkan seluruh penjumlahan  $X_i$  yang merupakan gambaran dari *Random Walk* (Yang, 2010).

*Random Walk* juga dapat disebut sebagai *Drunkard Walk* berdasarkan skenario pemabuk yang berjalan di jalan. Setiap langkah pemabuk yang berjalan secara acak baik ke depan, ke belakang, maupun ke samping. Jika pemabuk berjalan pada sebuah lapangan bola, dia akan berjalan kemana saja secara acak dan meninggalkan bekas jejak kaki yang membentuk sebuah motif *2D Random Walk* (Yang, 2010). Di mana dijelaskan pada rumusan berikut.

$$S_{t+1} = S_t + \omega_t \quad \dots(2.2)$$

Pada rumusan diatas,  $S_t$  merupakan lokasi sekarang berdasarkan nilai  $t$ , dan  $\omega_t$  merupakan sebuah langkah atau *random variable* yang memiliki nilai distribusi kemungkinan  $S_t$  akan berjalan selanjutnya (Yang, 2010). Kemungkinan yang dihasilkan akan membentuk sebuah labirin yang digambarkan pada gambar berikut.



Gambar 2.2 Beberapa hasil dari *Drunkard Walk* yang membentuk sebuah labirin pada permainan *Roguelike* (Craddock, 2015).

Pada Gambar diatas, merupakan beberapa hasil dari *Drunkard Walk* yang berjalan secara acak sehingga terbentuk sebuah tingkat permainan yang berbeda dan tidak dapat diprediksi pola permainan yang ada. Bermula pada sebuah *walker*

yang berjalan sendirian dan kemudian bercabang secara terus menerus hingga mencapai sebuah titik di mana *walker* sudah tidak dapat berjalan lagi. Pola permainan seperti ini mampu memberikan kejutan juga terhadap pemain sehingga pemain tidak dapat berekspektasi kemana arah musuh berjalan. Selain pemain yang dimulai secara acak, terdapat juga kemungkinan jumlah dan letak musuh yang ada, *item drops* yang diberikan atau bahkan tidak ada sama sekali. Hal ini sudah diterapkan pada beberapa permainan berbasis *roguelike* seperti *Nuclear Throne*, *Rogue*, *Minecraft*, dan *enter the gungeon*.



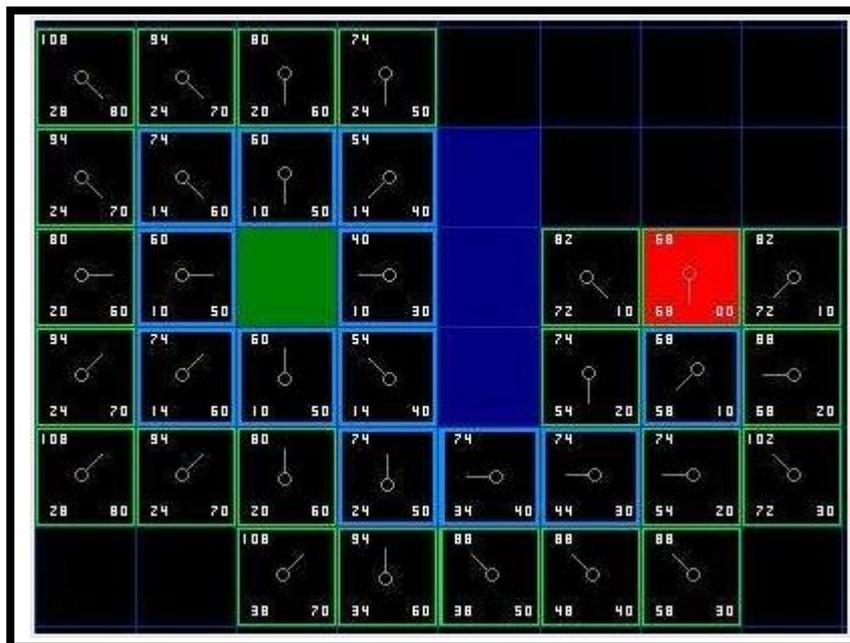
Gambar 2.3 *Nuclear Throne* merupakan salah satu contoh permainan yang menggunakan *Drunken Walk Algorithm* (Vlambeer,2016).

Salah satu contoh permainan yang memanfaatkan *Drunkard Walk* adalah permainan *Nuclear Throne*, di mana *Walker* yang digunakan lebih dari satu, sehingga dapat menciptakan sebuah *level* permainan yang besar. Serta menciptakan beberapa konten kejutan seperti variasi musuh yang berbeda, *spawn boss* yang bisa hadir kapan saja, *loot chest* yang berisi amunisi atau obat, serta beberapa *obstacle* yang berinteraksi dengan pemain dan juga musuh.

## 2.4 Pathfinding A\*

Sebuah robot dapat digunakan untuk memenuhi segala kegiatan yang berkaitan dengan pencarian dan penyelamatan orang-orang yang terjebak dalam situasi yang berbahaya. Hal ini diabstraksikan pada robot yang menerapkan teknik pencarian pada situasi sebagai labirin yang harus diselesaikan masalahnya untuk menemukan target yang ingin dicari (Liu & Gong, 2011). Selain berfungsi untuk menyelamatkan target, robot juga dapat digunakan sebagai musuh atau pencari target di sebuah permainan apabila pemain sebagai target mencapai suatu batas pendeteksi dari sebuah musuh.

Algoritma A\* merupakan salah satu bagian dari *Pathfinding* yang sering digunakan sebagai *Artificial Intelligence* untuk *game programming* (Fridenfalk, 2014). Pada algoritma ini, musuh berperan sebagai *pathfinder* untuk mencari pemain yang berperan sebagai target. Untuk mencapai target, algoritma ini mencari potensi *path* yang memiliki ukuran terkecil sehingga menciptakan nilai efisiensi ketika mencari target.



Gambar 2.4 Hasil dari pergerakan algoritma *A\* Pathfinding* dari *Patrick Lester* (Lester, 2013).

Untuk mencari nilai terkecil diperlukan nilai *F-cost* terkecil, di mana *F-cost* merupakan hasil nilai dari penjumlahan nilai *movement cost* (G) dan *estimated movement cost* (H). *movement cost* (G) merupakan pergerakan yang ditempuh dari titik awal pergerakan menuju sebuah objek yang ada di *grid*. *estimated movement cost* (H) merupakan estimasi pergerakan objek menuju titik akhir suatu tujuan. setelah mencari nilai *F-cost* terkecil, objek akan melakukan pencarian berdasarkan *F-cost* terkecil hingga menemukan target sesuai dengan total jarak *F-cost* terkecil (Lester, 2013).

## 2.5 Game User Experience Satisfaction Scales

*GUESS* adalah sebuah metode untuk mengukur validasi dari suatu permainan berdasarkan sifat dan karakteristik dari suatu permainan. Metode ini memiliki 9 *subscales* yang terdiri atas 55 jenis pertanyaan berdasarkan *subscales* yang ada yaitu *Usability/playability*, *Narratives*, *Play engrossment*, *Enjoyment*, *Creative Freedom*, *Audio aesthetics*, *Personal gratification*, *Social connectivity*, dan *Visual aesthetics* (Phan, dkk., 2016). Berikut adalah jenis *subscales* menurut Phan (2016).

### 1. *Usability/Playability*

Kemudahan pada permainan yang dapat dimainkan dengan menyelesaikan *objectives* pada permainan tersebut dan mengurangi kesulitan pada pemain untuk mengakses kontrol dan antarmuka pada permainan.

### 2. *Narratives*

Penjelasan sebuah cerita pada permainan dan kemampuan permainan dalam mengembangkan minat dan perasaan pada pemain.

3. *Play Engrossment*

Kemampuan permainan untuk mencapai seberapa jauh permainan dapat menarik perhatian dan minat pemain.

4. *Enjoyment*

Kemampuan permainan yang mampu memberikan kesenangan dan kegembiraan yang dirasakan oleh pemain selama bermain.

5. *Creative Freedom*

Kemampuan permainan yang menunjukkan sejauh mana permainan dapat meningkatkan kreativitas dan rasa ingin tahu pada pemain untuk mengekspresikan individualitasnya ketika bermain.

6. *Audio Aesthetics*

Keunikan dari berbagai suara yang dimasukkan kedalam permainan yang memberikan keunikan suasana dalam bermain.

7. *Personal Gratification*

Kemampuan permainan yang dapat memotivasi pemain dalam meningkatkan keinginan untuk dapat menyelesaikan permainan.

8. *Social Connectivity*

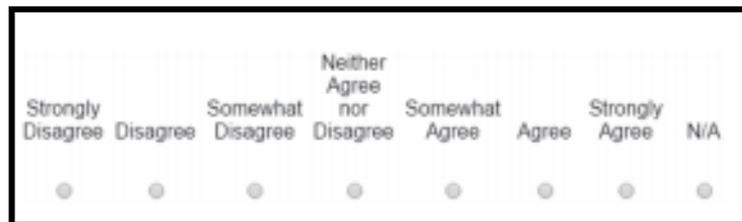
Kemampuan permainan yang memfasilitasi hubungan social antara pemain melalui alat dan fitur pada permainan.

9. *Visual Aesthetics*

Kemampuan permainan yang menunjukkan daya tarik pemain melalui penampilan dan grafik dari permainan.

*GUESS* diciptakan sebagai bentuk evaluasi dari suatu permainan berdasarkan *user experience* yang didapat oleh pemain. Dari segala aspek yang ditanyakan, pemain memberikan sebuah hasil pengukuran berdasarkan 9 aspek dengan opsi penghilangan aspek apabila permainan tidak memiliki salah satu dari semua aspek yang ada (Phan, dkk., 2016).

Untuk menguji permainan berdasarkan skala tersebut diperlukan suatu skala *likert* untuk mengukur kepuasan dalam permainan. Berdasarkan skala *likert*, pemain diberi beberapa pertanyaan yang dapat memilih satu opsi saja di setiap pertanyaan yang ada, dimulai dari jawaban yang tidak setuju hingga sangat setuju (Phan, dkk., 2016).



Gambar 2.5 Skala *likert* yang dijadikan pengukuran pada kuesioner *GUESS* (Phan, dkk., 2016).

Kriteria Intepretasi Skor yang digunakan untuk mengolah jawaban skala *likert* adalah sebagai berikut.

9. Sangat Tidak Setuju (Kategori 1) dengan syarat  $\geq 0\%$  dan  $< 14\%$
10. Tidak Setuju (Kategori 2) dengan syarat  $\geq 14\%$  dan  $< 28\%$
11. Kurang Setuju (Kategori 3) dengan syarat  $\geq 28\%$  dan  $< 42\%$
12. Netral (Kategori 4) dengan syarat  $\geq 42\%$  dan  $< 56\%$
13. Cukup Setuju (Kategori 5) dengan syarat  $\geq 56\%$  dan  $< 70\%$
14. Setuju (Kategori 6) dengan syarat  $\geq 70\%$  dan  $< 84\%$

15. Sangat Setuju (Kategori 7) dengan syarat  $\geq 84\%$

Untuk menghitung rata-rata nilai dari pertanyaan yang diolah digunakan rumus sebagai berikut.

$$\begin{aligned} \text{Hasil} = & (\text{Jumlah sangat tidak setuju} \times 1 + \text{Jumlah tidak setuju} \times 2 \\ & + \text{Jumlah kurang setuju} \times 3 + \text{Jumlah netral} \times 4 \\ & + \text{Jumlah cukup setuju} \times 5 + \text{Jumlah setuju} \times 6 \quad \dots(2.3) \\ & + \text{Jumlah sangat setuju} \times 7) \div (\text{Jumlah sampel} \\ & \times \text{Jumlah skala}) \times 100 \end{aligned}$$