

BAB 2

LANDASAN TEORI

2.1 Survival Horror

Survival horror merupakan genre *game* yang menggabungkan dua aspek utama, yaitu *survival* (bertahan hidup) dan *horror* (ketakutan). Aspek *survival* mengacu pada tantangan yang dihadapi pemain untuk bertahan dalam kondisi berbahaya dengan sumber daya yang terbatas. Pemain harus mengelola kebutuhan dasar, mengeksplorasi lingkungan, serta memanfaatkan alat, senjata, dan tempat berlindung untuk menghadapi berbagai ancaman [18]. Sementara itu, aspek *horror* berkaitan dengan upaya menciptakan rasa takut melalui elemen-elemen yang bersifat material, nyata, atau memiliki potensi untuk terjadi dalam kehidupan nyata, seperti serangan zombie, monster, wabah virus, atau pandemi [19]. Desain *game survival horror* yang efektif mampu menciptakan pengalaman tak terlupakan dan berdampak kuat kepada pemain, dengan narasi yang dibangun melalui interaksi antara protagonis, antagonis, dan latar tempat. Hal tersebut dapat dicapai dengan membangun atmosfer yang menegangkan melalui penggunaan audio dan visual yang efektif, serta menyajikan kejadian-kejadian mengerikan yang memicu emosi pemain. Salah satu ciri khas *game survival horror* adalah keberadaan musuh dalam bentuk monster. Desain monster yang semakin jauh dari wujud, rupa, atau suara manusia dapat meningkatkan rasa takut pemain dan menimbulkan kesan tidak nyaman. Saat menghadapi musuh, pemain akan terus menyesuaikan strategi, baik untuk meningkatkan kemampuan menyerang maupun untuk menghindari dari ancaman yang ada [17].



Gambar 2.1. *Gameplay Survival Horror*

Sumber: [8]

2.2 Wave Function Collapse

Wave Function Collapse (WFC) merupakan algoritma *Procedural Content Generation* yang diusulkan oleh Maxim Gumin. Terinspirasi dari mekanika kuantum, algoritma WFC mengadopsi prinsip superposisi (*superposition*), di mana semua objek awalnya memiliki seluruh keadaan yang mungkin terjadi, direpresentasikan sebagai fungsi gelombang. Proses selanjutnya melibatkan tahap pengamatan (*observation*), yaitu pemilihan objek dengan jumlah kemungkinan keadaan yang paling sedikit untuk ditetapkan (*collapse*) menjadi satu keadaan yang pasti. Objek yang telah ditetapkan akan menjalani tahap pembaruan (*propagation*) untuk memperbarui kemungkinan keadaan objek-objek di sekitarnya. Seluruh tahap akan diulang secara iteratif hingga semua objek teramati dan mencapai keadaan yang pasti, menandakan bahwa fungsi gelombang telah sepenuhnya runtuh [20].

Algoritma WFC dapat diimplementasikan melalui dua metode, yaitu *Simple Tiled Model* dan *Overlapping Model*. Penelitian ini menggunakan *Simple Tiled Model* dikarenakan metode tersebut lebih sesuai untuk menghasilkan *map* menggunakan kumpulan *tile* yang telah diatur sebelumnya, sementara metode *Overlapping Model* cenderung lebih sesuai untuk digunakan pada proses pembuatan pola tekstur gambar. Metode *Simple Tiled Model* bekerja dengan menggunakan batasan keterhubungan (*adjacency constraint*) untuk menggabungkan *tile* berdasarkan kesamaan sisi. Batasan ini dapat ditentukan secara manual atau dihasilkan secara otomatis oleh program dengan mempertimbangkan rotasi atau pencerminan *tile* agar mencakup seluruh kemungkinan arah. Hasil dari metode *Simple Tiled Model* dapat dilihat pada Gambar 2.2 [20, 21].



Gambar 2.2. Metode *Simple Tiled Model*

Sumber: [20]

Cara kerja algoritma WFC menggunakan metode *Simple Tiled Model* dapat dijabarkan sebagai berikut [21]:

1. Siapkan sebuah *tileset* yang berisi semua *tile* yang akan digunakan, dengan masing-masing *tile* memiliki sebuah kumpulan informasi tepi yang digunakan untuk menyesuaikan hubungan antar *tile*. Tahap ini dapat didefinisikan melalui Rumus 2.1 dan 2.2.

$$t \in T, \quad t = (e_{east}, e_{west}, e_{north}, e_{south}) \quad (2.1)$$

$$e_{dir}(t), \quad dir \in \{east, west, north, south\} \quad (2.2)$$

T merupakan *tileset* yang memiliki kumpulan *tile* t . $e_{dir}(t)$ merupakan informasi tepi *tile* (*edge*) yang mencakup sisi pada arah timur (*east*), barat (*west*), utara (*north*), dan selatan (*south*).

2. Inisialisasi sebuah *grid* dengan ukuran $M \times N$ yang berisi kumpulan sel yang belum teramati. Sel yang belum teramati memiliki semua pilihan *tile* dari *tileset*. Tahap ini dapat didefinisikan melalui Rumus 2.3.

$$g_{m,n} = T, \quad \forall m \in [1, M], \forall n \in [1, N] \quad (2.3)$$

$g_{m,n}$ merupakan sel pada *grid* dengan panjang M dan lebar N yang memiliki nilai awal *tileset* T .

3. Pilih sel dengan nilai entropi terendah menggunakan formula *Shannon entropy* [22] yang dapat didefinisikan melalui Rumus 2.4.

$$E = - \sum_{i=1}^N P_i \log_2 P_i \quad (2.4)$$

Nilai entropi (E) diperoleh berdasarkan besarnya probabilitas pemilihan salah satu *tile* di dalam sel (P_i). Sel dengan pilihan *tile* paling sedikit memiliki probabilitas pemilihan yang paling besar dan akan menghasilkan nilai entropi yang rendah.

4. Tetapkan salah satu kemungkinan *tile* yang tersedia di dalam sel yang telah dipilih. Tahap ini dapat didefinisikan melalui Rumus 2.5.

$$g_{m,n} \leftarrow \{t\}, \quad t \in g_{m,n} \quad (2.5)$$

5. Perbarui keadaan sel-sel tetangga (*neighbors*) berdasarkan hasil sel yang telah ditetapkan dengan menghilangkan pilihan *tile* yang tidak memenuhi *adjacency constraint* untuk *edge* pada masing-masing sisi. *Adjacency constraint* tersebut dapat didefinisikan melalui Rumus 2.6–2.9.

$$e_{east}(g_{m,n}) = e_{west}(g_{m+1,n}) \quad (2.6)$$

$$e_{west}(g_{m,n}) = e_{east}(g_{m-1,n}) \quad (2.7)$$

$$e_{north}(g_{m,n}) = e_{south}(g_{m,n+1}) \quad (2.8)$$

$$e_{south}(g_{m,n}) = e_{north}(g_{m,n-1}) \quad (2.9)$$

Untuk *neighbors* pada setiap sisi sel ($g_{m+i,n+j}$), *edge* pada setiap sisi *neighbors* ($e_{dir}(g_{m+i,n+j})$) harus sesuai dengan *edge* pada sisi yang berlawanan arah dari sel yang telah ditetapkan ($e_{dir}(g_{m,n})$).

6. Ulangi langkah 3-5 hingga semua sel teramati dan memenuhi *adjacency constraint* untuk setiap *edge* di dalamnya.

Cara kerja algoritma yang telah dijelaskan menerapkan *adjacency constraint* yang bersifat lokal. Namun, pendekatan tersebut memiliki sebuah limitasi dikarenakan hasil yang didapatkan sangat bergantung pada proses penetapan *tile* secara acak tanpa mengetahui keadaan *tile* lain secara global. Untuk mengatasi hal tersebut, digunakanlah batasan tambahan dalam bentuk *non-local constraints* untuk mendapatkan hasil yang lebih terkendali dan lebih mendekati desain yang dibuat secara manual. Batasan-batasan tersebut dapat dijabarkan sebagai berikut [23]:

1. Global Constraints

Batasan ini terdiri dari dua jenis yang dapat dijabarkan sebagai berikut:

- **Global Maximum Constraint:**

Jumlah maksimum *tile* dan elemen *game* yang dapat digunakan di dalam *grid*.

- **Global Minimum Constraint:**

Jumlah minimum *tile* yang harus digunakan di dalam *grid* sebelum proses penetapan dijalankan, baik dengan posisi acak maupun ditentukan secara manual.

2. **Inter-layer Constraints**

Batasan ini mengatur penempatan elemen *game* (seperti struktur bangunan atau musuh) agar sesuai dengan lapisan *tile* di bawahnya. Elemen *game* tidak dapat ditempatkan pada *tile* tertentu jika tidak memenuhi kriteria yang ditetapkan oleh elemen tersebut. Misalnya, sebuah objek pohon harus ditempatkan pada *tile* rumput dan tidak dapat ditempatkan pada *tile* air.

3. **Distance Constraints**

Batasan ini mengatur jarak antara elemen-elemen *game*. Elemen *game* yang ditempatkan pada sebuah *tile* tidak boleh berada di dalam rentang jarak elemen yang telah ditempatkan sebelumnya.



2.3 Game User Experience Satisfaction Scale

Game User Experience Satisfaction Scale (GUESS) merupakan metode pengukuran tingkat kepuasan pemain terhadap *game* yang dikembangkan oleh Mikki H Phan et al. Metode tersebut mengevaluasi 55 pernyataan yang mencakup sembilan aspek di dalam *game*, yaitu *Usability/Playability*, *Narratives*, *Play Engrossment*, *Enjoyment*, *Creative Freedom*, *Audio Aesthetics*, *Personal Gratification*, *Social Connectivity*, dan *Visual Aesthetics*. GUESS telah digunakan secara luas untuk mengevaluasi tingkat kepuasan pemain dalam berbagai macam domain, seperti *healthcare simulation*, *mixed reality*, *social interaction*, dan *virtual reality gaming*. Namun, metode tersebut memiliki kelemahan dalam hal waktu penyelesaian yang lama dikarenakan jumlah pernyataan yang banyak, sehingga kurang sesuai bila digunakan untuk melakukan tahap penilaian atau iterasi desain yang cepat [24].

Untuk mengatasi hal tersebut, dikembangkan versi singkat dari GUESS, yaitu GUESS-18, yang meringkas 55 pernyataan menjadi 18 pernyataan dengan mempertahankan kesembilan aspek *game* yang dievaluasi. GUESS-18 memberikan hasil penilaian yang sama validnya dengan versi lengkap GUESS, tetapi dengan waktu penyelesaian yang lebih singkat dan penggunaan yang lebih sesuai untuk sebuah penelitian. Penjelasan lebih lanjut mengenai kesembilan aspek *game* yang dievaluasi dapat dilihat pada Tabel 2.1 [25].

Tabel 2.1. Aspek dan Pernyataan dari GUESS-18

Aspek	Deskripsi	Pernyataan
<i>Usability/Playability</i>	Kemampuan <i>game</i> dalam memberikan kemudahan terhadap aspek antarmuka dan kontrol kepada pemain dengan gangguan atau hambatan yang minimal.	<ol style="list-style-type: none"> 1. <i>I find the controls of the game to be straightforward.</i> 2. <i>I find the game's interface to be easy to navigate.</i>
Lanjut pada halaman berikutnya		

Tabel 2.1 Aspek dan Pernyataan dari GUESS-18

Aspek	Deskripsi	Pernyataan
<i>Narratives</i>	Kemampuan <i>game</i> dalam menarik minat dan membentuk emosi pemain melalui cerita atau narasi yang disampaikan.	<p>3. <i>I am captivated by the game's story from the beginning.</i></p> <p>4. <i>I enjoy the fantasy or story provided by the game.</i></p>
<i>Play Engrossment</i>	Kemampuan <i>game</i> dalam mempertahankan perhatian dan ketertarikan pemain.	<p>5. <i>I feel detached from the outside world while playing the game.</i></p> <p>6. <i>I do not care to check events that are happening in the real world during the game.</i></p>
<i>Enjoyment</i>	Kemampuan <i>game</i> dalam memberikan kesenangan dan kepuasan kepada pemain.	<p>7. <i>I think the game is fun.</i></p> <p>8. <i>I feel bored while playing the game.</i></p>
<i>Creative Freedom</i>	Kemampuan <i>game</i> dalam mendorong kreativitas, rasa ingin tahu, dan kebebasan berekspresi pemain.	<p>9. <i>I feel the game allows me to be imaginative.</i></p> <p>10. <i>I feel creative while playing the game.</i></p>
Lanjut pada halaman berikutnya		

Tabel 2.1 Aspek dan Pernyataan dari GUESS-18

Aspek	Deskripsi	Pernyataan
Audio Aesthetics	Kemampuan <i>game</i> dalam memperkaya pengalaman bermain pemain melalui aspek suara.	<p>11. <i>I enjoy the sound effects in the game.</i></p> <p>12. <i>I feel the game's audio (e.g., sound effects, music) enhances my gaming experience.</i></p>
Personal Gratification	Kemampuan <i>game</i> dalam mendorong motivasi pemain dan keinginan untuk terus bermain hingga mencapai suatu tujuan.	<p>13. <i>I am very focused on my own performance while playing the game.</i></p> <p>14. <i>I want to do as well as possible during the game.</i></p>
Social Connectivity	Kemampuan <i>game</i> dalam memfasilitasi hubungan sosial antar pemain melalui fitur dan alat yang disediakan.	<p>15. <i>I find the game supports social interaction (e.g., chat) between players.</i></p> <p>16. <i>I like to play this game with other players.</i></p>
Visual Aesthetics	Kemampuan <i>game</i> dalam memberikan grafis dan tampilan yang menarik kepada pemain.	<p>17. <i>I enjoy the game's graphics.</i></p> <p>18. <i>I think the game is visually appealing.</i></p>

Setiap pernyataan dari GUESS-18 dapat dijadikan sebagai subskala yang diukur menggunakan skala Likert 7 poin (1 = Sangat Tidak Setuju sampai 7 = Sangat Setuju) dengan urutan poin dan nilai yang dapat dilihat pada Tabel 2.2.

Tabel 2.2. Skala Likert 7 Poin

Poin	Nilai
Sangat Tidak Setuju	1
Tidak Setuju	2
Cenderung Tidak Setuju	3
Netral	4
Cenderung Setuju	5
Setuju	6
Sangat Setuju	7

Nilai untuk setiap subskala ($\bar{x}(\%)$) dapat dihitung dengan merata-ratakan jumlah nilai masing-masing poin (n_i) berdasarkan jumlah responden pemain (N) yang dapat didefinisikan melalui Rumus 2.10.

$$\bar{x}(\%) = \left(\frac{\sum_{i=1}^7 (n_i \times i)}{N \times 7} \right) \times 100\% \quad (2.10)$$

Hasil rata-rata total seluruh nilai subskala dapat digunakan untuk menentukan nilai GUESS-18 secara keseluruhan, dengan interpretasi predikat nilai yang dapat dilihat pada Tabel 2.3 [26].

Tabel 2.3. Predikat Nilai GUESS-18

Rentang Nilai	Predikat
< 15%	Sangat Buruk
15%-28%	Buruk
29%-42%	Cukup Buruk
43%-56%	Cukup
57%-70%	Cukup Baik
71%-84%	Baik
> 84%	Sangat Baik