

BAB 2

LANDASAN TEORI

2.1 Game Structure

Struktur *game* terdiri atas *formal elements* yang berarti elemen-elemen umum yang membentuk sebuah *game* layaknya seperti sebuah sistem, dan *dramatic elements* adalah elemen dari sebuah *game* yang dapat membuat seseorang tertarik untuk memainkannya. (Fullerton, 2014)

Menurut Fullerton (2014), *Formal elements* terdiri dari beberapa elemen.

1. *Players*, menjelaskan tentang siapa saja partisipan dari *game*.
2. *Objectives*, yang mendefinisikan tujuan untuk *game* terbilang selesai.
3. *Procedures*, adalah aksi atau metode bermain yang diperbolehkan oleh *rules* yang dapat dilakukan untuk mencapai *objectives*.
4. *Rules*, mengatur tentang apa yang pemain dapat dan tidak dapat lakukan dalam sebuah permainan dengan mendefinisikan prinsip, objek *game*, dan membatasi permainan.
5. *Resources*, sumberdaya dalam *game* yang tersedia untuk digunakan dalam permainan.
6. *Conflict*, adalah rintangan yang ada dalam *game* guna menghalangi pemain menyelesaikan *objectives* dan mengharuskan pemain untuk beradaptasi.
7. *Boundaries*, mendefinisikan batasan dimana permainan dilakukan.
8. *Outcome*, hasil yang didapatkan dari berhasilnya atau tidak berhasilnya *objective* dicapai pada sebuah *game*.

Sedangkan untuk *dramatic elements* terdiri dari sebagai berikut.

1. *Challenge*, tantangan yang muncul bagi pemain atas *conflict* yang ada dalam *game*.
2. *Play*, mendefinisikan tentang bagaimana *game* dapat dimainkan pada struktur *game* yang kaku. *Play* dapat berdasarkan peraturan, atau berdasarkan kebebasan yang membuat pemain harus serta-merta mengambil keputusan.
3. *Premise*, menjelaskan tentang konsep atau latar mengenai *game* tersebut.
4. *Character*, adalah tokoh-tokoh yang terlibat dalam cerita sebuah *game* dan menjalankan cerita tersebut.
5. *Story*, berisi tentang cerita yang dapat menarik seorang pemain.

2.2 Game Design Document

Game Design Document (GDD) adalah dokumentasi dari sebuah *game design* yang dapat digunakan untuk mengekspresikan visi dari sebuah *game*, mendeskripsikan kontennya, dan mengagag rencana untuk implementasi di masa depan. Tujuan dari dokumentasi *game design* sendiri adalah untuk meyakinkan seluruh bagian tim pengembang berada pada arah dan ide yang sama dalam mengembangkan sebuah *game* (Ryan, 1999). GDD biasanya tidak terlalu formal dan strukturnya dapat bervariasi antar beberapa perusahaan yang bergerak dalam bidang *game* dengan perusahaan lainnya, tetapi terdapat panduan umum yang dapat diikuti, menurut Rollings dan Adam (2013), umumnya GDD terdiri dari beberapa bagian seperti sebagai berikut.

1. Alur cerita.
2. Karakter.

3. Desain *level* atau lingkungan.
4. *Gameplay*.
5. Seni.
6. Suara dan Musik.
7. Antarmuka pengguna dan kontrol game.

2.3 Procedural Content Generation

Procedural Content Generation (PCG) adalah pembuatan konten dalam *game* menggunakan algoritma dengan sedikit atau tidak ada campur tangan dari pengguna. Dalam kata lain, PCG mengacu kepada perangkat lunak yang mampu membuat konten dalam *game* dengan sendirinya, atau bersama dengan satu atau lebih pemain atau desainer. (Shaker dkk., 2016)

Beberapa contoh yang dapat dikatakan sebagai PCG dalam definisi ini menurut Shaker dkk. (2016) adalah sebagai berikut.

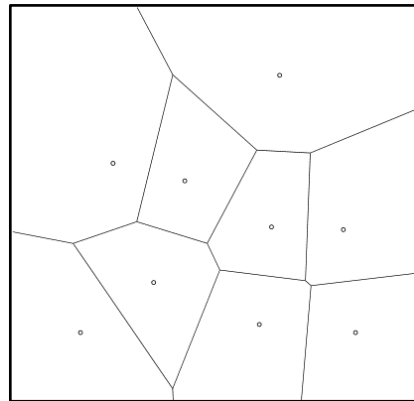
1. Perangkat lunak yang membuat labirin untuk *game* aksi petualangan seperti *The Legend of Zelda* tanpa campur tangan manusia dan dapat membuat level baru setiap kali dijalankan.
2. Sistem yang membuat senjata baru pada *game* pesawat luar angkasa.
3. Program yang membangkitkan permainan papan yang lengkap cukup dengan program itu sendiri.
4. Perantara *game engine* untuk membuat vegetasi dunia *game*.
5. Alat desain grafis yang dapat membantu sebuah pengguna membuat peta *game* strategi, selagi evaluasi terus dilakukan untuk map yang didesain dan memberikan sugesti untuk peta tersebut guna memberikan properti permainan yang sesuai, imbang, dan menarik.

Berikut adalah beberapa contoh yang tidak termasuk PCG pada definisi ini.

1. Pengubah map untuk *game* strategi yang memperbolehkan pengguna menambahkan dan menghilangkan barang, tanpa membangkitkan atau memberikan sugesti lewat algoritma.
2. Kecerdasan buatan untuk permainan papan.
3. *Game engine* yang mempunyai fitur untuk vegetasi secara otomatis.

PCG digunakan untuk beberapa alasan, antara lain adalah mengurangi waktu dan biaya pengembangan, memungkinkan sebuah *game* menjadi adaptif, dan membantu desainer. Beberapa *game* yang PCG adalah Minecraft, Dwarf Fortress, Diablo, Civilization IV, dan Spore.

2.4 Voronoi Diagram



Gambar 2.1 Voronoi diagram dengan 9 titik (Freiknecht dkk., 2017)

Voronoi diagram adalah struktur data yang membuat partisi beberapa titik yang disebut *points* menjadi beberapa daerah yang disebut *sites* pada *Euclidian space*, setiap sel berisi titik yang lebih dekat ke suatu *site* ketimbang *site* yang lainnya. Berikut definisi formal dari *Voronoi diagram*.

$$V(p_i) = \{x \in \mathbb{R}^d : \|x - p_i\| \leq \|x - p_j\|, \forall j \leq n\} \quad (2.1)$$

Dengan $P = \{p_1, \dots, p_n\}$ kumpulan dari titik \mathbb{R}^d . Untuk setiap p_i , asosiasikan daerah *Voronoi* $V(p_i)$. Daerah $V(p_i)$ adalah *intersection* dari $n - 1$ *half-spaces*. Setiap *half-space* berisi p_i dan terikat oleh *bisector* dari p_i dan beberapa titik lain dari P (Wormser, 2008). Lebih detilnya, salah satu langkah untuk membuat *Voronoi diagram* yaitu *naïve approach* dimulai dari membuat *grid*, dan lalu pada beberapa sel di *grid* buatlah untuk menjadi titik yang digunakan sebagai titik utama dari suatu wilayah pada tahap selanjutnya. Kemudian untuk setiap sel yang bukan terpilih menjadi titik, carilah titik yang terdekat dengan jarak ke sel tersebut. Sel lalu menjadi bagian wilayah dari sebuah titik (Mary dkk., 2017). Beberapa cara untuk perhitungan titik terdekat dapat dilakukan, salah satunya adalah *Manhattan distance* dan *Euclidian distance*.

2.5 Leniency

Leniency mendeskripsikan seberapa menyulitkan atau tidak sebuah *level* kepada pemain. Pengukuran dapat dilakukan dengan memberi nilai kepada objek permainan seperti contohnya rintangan, semakin besar nilai yang didapat dapat diartikan *level* tersebut semakin sulit (Smith dan Whitehead, 2010). Untuk *leniency* pada lintasan yang dibuat menggunakan *Voronoi diagram* diukur dengan banyaknya jumlah tikungan yang ada pada sebuah lintasan.

2.6 Linearity

Linearity mengukur profil dari *level* yang dibuat. Profil yang dimaksud adalah seberapa bervariasi *level* yang dibuat dengan perubahan-perubahan yang terjadi pada setiap bagian *level*. Sebuah *level* dapat dikatakan *linear* jika perubahan yang terjadi sedikit atau tidak ada, sebaliknya *level* dikatakan *non-linear* jika perubahan terhadap *level* signifikan (Smith dan Whitehead, 2010). Pengukuran

linearity dari lintasan yang dibuat menggunakan *Voronoi diagram* dilakukan dengan mengukur banyaknya jalan lurus yang dibuat.