

BAB 2

LANDASAN TEORI

2.1 Platformer

Platformer adalah subgenre permainan di mana pemain mengontrol sebuah karakter untuk melewati berbagai rintangan berupa *platform* yang dapat lebih tinggi ataupun lebih rendah tergantung dengan desain level [12]. *Platformer* juga sering disebut *Jump and Run Games* dikarenakan sifat alami dari game *Platformer* yang membutuhkan pemain untuk menggunakan kemampuan melompat karakter untuk melewati rintangannya. Game seperti *The Legend of Zelda* dimana mekanik loncat karakter dilakukan secara otomatis tidak termasuk sebagai *Platformer*.

2.2 2D Endless Runner Game

Endless Runner Game adalah genre permainan di mana pemain mengontrol sebuah karakter yang terus berjalan tanpa berhenti dalam dunia procedural yang tidak terbatas[3]. Tujuan dari game ini biasanya adalah *Highscore* yang menuntut pemain untuk mendapatkan skor setinggi-tingginya. Tingkat kesulitan level juga semakin sulit seiring tingginya skor pemain. Level dalam 2D endless runner adalah dunianya, di mana pada level terdapat blok ataupun rintangan. Rintangan yang akan terus dihadapi oleh pemain sehingga player harus melewati semua rintangan tersebut untuk mendapat skor yang setinggi mungkin. Pada game 2D endless runner, kamera umumnya bersifat 2D side-scoller seperti pada game *Flappy Bird*.

2.3 Procedural Content Generation

Procedural Content Generation adalah cara untuk membuat atau menciptakan konten yang bervariasi menggunakan sebuah algoritma [14]. Konten dapat berupa music, peraturan game, level, gambar, video dan lainnya. Salah satu contoh game yang menggunakan PCG adalah Minecraft, Game yang dibuat oleh Markus Persson. Ketika pemain memulai gamenya, Sebagian dari map dibangun secara procedural dan lanjut membangun ketika player berjalan mendekati batas map yang telah dibangun.

2.4 Genetic Algorithm

Genetic Algorithm (GA) adalah algoritma pencarian yang menggunakan pendekatan dengan teori evolusi Darwin, di mana setiap kandidat solusi dianggap sebagai individu dengan kode genetik yang merepresentasikan karakteristiknya. Individu dengan nilai tertinggi mempunyai kemungkinan lebih tinggi untuk dilanjutkan ke generasi selanjutnya atau yang biasa disebut dengan elitism dan mempunyai kemungkinan tinggi untuk menurunkan sebagian dari kode genetiknya dengan melakukan crossover dengan individu lainnya. Pada algoritma GA mempunyai tahap-tahap berupa menginisialisasi populasi individu dengan ukuran N dengan kode genetik yang acak dan selanjutnya untuk setiap generasi atau iterasi, sebagian dari individu yang terbaik dilanjutkan ke generasi selanjutnya (elitism). Dan kemudian sisa populasi yang baru adalah hasil dari crossover dengan pasangan yang dipilih secara acak dari generasi sebelumnya. Pemilihan pasangan ini didasarkan nilai fitness individu dimana semakin tinggi fitness individu, semakin besar kemungkinan dipilih sebagai pasangan. Populasi baru tersebut akan dilakukan mutasi berdasarkan parameter persenan mutasi yang telah ditentukan, begitu juga dengan jumlah iterasi.

2.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) adalah algoritma yang berfungsi untuk pencarian. Dimana algoritma tersebut menggunakan banyak partikel atau individu, dikelompokkan dalam segerombolan [11]. Kandidat solusi untuk optimasi masalah terdapat pada masing-masing partikel ini. Secara konsep, penggunaan pbest&gbest pada Particle Swarm Optimization (PSO) mirip dengan operasi crossover pada Genetic Algorithm (GA). Selain itu, PSO juga menggunakan konsep nilai fitness.

Terdapat Langkah-langkah dalam algoritma PSO sebagai berikut:

1. Tentukan terlebih dahulu jumlah partikel yang akan digunakan
2. Tentukan kecepatan partikel dan posisi secara acak.

3. Mengevaluasi nilai fitness menggunakan rumus yang telah ditentukan dari masing-masing partikel berdasarkan posisinya
4. Menentukan partikel dengan fitness terbaik untuk dijadikan sebagai Gbest
5. Pbest awal sama dengan posisi awal
6. Menggunakan Pbest dan Gbest yang sudah ada, perbarui kecepatan partikel dengan rumus berikut :

$$V_i(t) = V_i(t-1)C_1R_1(X_i^L - X_i(t-1)) + C_2R_2(X^G - X_i(t-1)) \quad (2-1)$$

Keterangan :

t = iterasi ke-t

i = indeks partikel

V = kecepatan partikel

V_i = kecepatan partikel pada suatu indeks

X = posisi partikel

R_1, R_2 = nilai acak, range [0-1]

C_1, C_2 = konstanta positif yang sering disebut learning factor

X^L = local best dari suatu partikel

X^G = global best dari seluruh kawanan

7. Perbarui posisi masing-masing partikel dengan rumus berikut:

$$X_i(t) = V_i(t) + X_i(t-1) \quad (2-2)$$

8. Evaluasi kembali nilai fitness dari tiap partikel
9. Tentukan partikel dengan fitness terbaik sebagai GBest.
10. Untuk tiap partikel, tentukan PBest dengan membandingkan posisi sekarang dengan PBest pada iterasi sebelumnya
11. Mengecek apakah semua posisi partikel konvergen. Jika ya, Maka berhenti. Partikel disebut konvergen apabila semua posisi partikel berada pada posisi yang sama atau sangat berdekatan dengan partikel gBest serta tidak memiliki perkembangan posisi pada iterasi berikutnya. Posisi partikel yang konvergen menandakan bahwa solusi telah ditemukan.