

BAB 2 TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Herlambang et al., (2019) pernah melakukan penelitian tentang penerapan *Neural Network* pada NPC *Braking Decision* dalam *Racing Game*. Penelitian ini dilakukan untuk mengetahui bagaimana *neural network* dapat dimanfaatkan sebagai algoritma pembelajaran untuk mesin pada gim mobil balap. Dimana perilaku yang dibentuk setelah pembelajaran tersebut akan membuat mobil mampu mendapatkan waktu optimal dalam lintasan yang akan diujikan. Dari penelitian tersebut hasil waktu tempuh yang didapat AI *neural network* memiliki selisih sebanyak 72 detik jika dibandingkan dengan AI milik RGSK, berarti 12 persen lebih cepat dari keseluruhan waktu tempuh masing-masing. Hal tersebut dapat terjadi hanya karena perbedaan jumlah dan waktu pengereman yang dilakukan kedua AI tersebut. AI *neural network* cenderung lebih sedikit melakukan pengereman terutama dibagian tikungan pada lintasan.

Penelitian mengenai Artificial Intelligence juga pernah dilakukan oleh Michael pada 2020. Penelitian ini merupakan penerapan dua jenis *Artificial Intelligence*, yaitu *Behavior Tree* dan *Fuzzy Waypoint Tactics*, pada karakter musuh dan karakter pembantu milik musuh dalam sebuah gim bernama *War of Zombies*. Pada penelitian ini musuh dapat memilih dan menyesuaikan tindakan-tindakan yang diambil berdasarkan kondisi yang ada. Mulai dari menentukan jenis serangan musuh terhadap karakter *player* berdasarkan

lingkup jarak karakter, dimana posisi yang baik untuk bersembunyi, sampai pada menentukan kapan dan bagaimana karakter pembantu melakukan tugasnya atau kabur. Pada gim bertema strategi seperti *war of zombies* ini, *fuzzy logic* sangat efektif dalam membantu AI dalam mengambil keputusan bagi setiap karakter

Pada 2015 Marvin pernah melakukan penelitian mengenai Pengembangan *Game Simulator* Balap Mobil Menggunakan Teknik *Artificial Intelligence*. Pada penelitian ini gim diuji oleh lebih dari dua puluh pengguna, yang berdasarkan ulasan pengguna mereka menyatakan bahwa benar-benar menikmati permainan. Diamati bahwa terdapat jumlah pemain yang sama yang mampu memenangkan perlombaan melawan mobil yang dikendalikan oleh gim. Ini menunjukkan bahwa permainan tidak hanya memberikan hiburan dalam bermain gim, hal itu juga memberikan pengalaman yang cukup menarik dan *gameplay* yang menantang. Secara umum, dapat disimpulkan bahwa *platform* Unity mendukung pengembangan efisien dari gim *racing*. Unity mendukung penerapan pencarian rute untuk mobil ke dalam jalur dengan sistem *waypoint*, *physics engine*, dan perhitungan *vektor* fungsi, yang semuanya tidak tersedia jika implementasi dilakukan dengan menggunakan teknik pencarian AI tradisional

Penelitian tentang *racing game* pernah dilakukan oleh Yoppy et al., (2018). Penelitian ini membandingkan Algoritma a^* dan *dynamic pathfinding* dengan *dynamic pathfinding algorithm* untuk NPC pada *car racing game*. Penelitian ini menggunakan Algoritma A^* yang melakukan pencarian rute

terpendek dengan titik-titik yang ada di setiap lintasan secara keseluruhan. Hasil dari penelitian ini adalah gabungan *dynamic pathfinding algorithm* dan Algoritma A* yang dapat diimplementasikan pada *Car Racing Game*. Pada penelitian ini NPC berhasil menggunakan gabungan Algoritma A* dan DPA melewati *area* yang sama. Posisi NPC berhasil melewati tikungan. NPC menabrak *obstacle* yang sudah berpindah dari posisi awal. NPC tidak berhasil mengikuti *Node A** yang melewati ruang antara *obstacle* dan dinding pembatas lintasan analisa perbandingan Algoritma A* dan DPA dengan DPA mendapatkan hasil yang lebih baik dari NPC yang hanya menggunakan Algoritma DPA pada lintasan perlombaan kosong dan lintasan perlombaan yang memiliki *obstacle*.

Hamid dan Cüneyt (2011) juga pernah melakukan penelitian tentang Kecerdasan Buatan menggunakan Jaringan Saraf Buatan Evolusioner. Penelitian ini memungkinkan pengembang gim untuk membuat pilihan berdasarkan informasi tentang apakah akan mencoba dan menerapkan EANN dalam AI game balap atau tidak, untuk manfaat yang diinginkan EANN dapat memberikan hasil yang lebih dari gim *racing* dengan konvensional AI. Hasil penelitian ini menunjukkan dua hal utama, pertama adalah bahwa algoritma NEAT dapat belajar menavigasi sirkuit dengan cukup baik dan bahkan balapan di sekitar sirkuit sederhana seperti *A-Speedway* lebih baik daripada AI standar. Hal kedua yang ditunjukkannya adalah bahwa pada sirkuit kompleks dibutuhkan waktu lebih lama untuk mengembangkan perilaku umum, dan ini tidak sebagus teknik AI standar. Dari analisis hasil cukup jelas

bahwa EANN tidak lebih efektif dalam balapan mobil yang dikendalikan komputer di sekitar sirkuit.

2.2 Dasar Teori

2.2.1 *Game*

Game berasal dari kata bahasa Inggris yang memiliki arti dasar permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada pihak yang menang dan ada pihak yang kalah, biasanya game dilakukan dengan tidak serius atau dengan tujuan menghibur (Riyadi, 2013).

Game merupakan suatu metode bermain yang digunakan untuk kesenangan ataupun sebagai alat pembelajaran (pendidikan) (Horton, 2015). Seiring perkembangan teknologi, tantangan dari gim terus ditingkatkan dengan memberikan *gameplay* baru ataupun fitur baru, pada saat yang sama, gim tersebut harus tetap menarik untuk dimainkan dan tidak memberi tantangan yang mustahil untuk dimainkan. Karena itu seiring dengan perkembangan teknologi, variasi gim menjadi luas dan dikategorikan. Salah satu perkembangan gim yang hingga saat ini terus berkembang adalah *video games* (Lumaris et al., 2019).

2.2.2 *Game Genre*

Genre pada suatu *game* memperlihatkan pola umum tantangan dari *game* tersebut. Dengan perkembangan informasi seperti sekarang, *genre* dari gim masih terus berkembang, sebagai contoh simulasi menari (*dance simulation*)

yang diperkenalkan oleh *desainer* gim dari Jepang (Tanjung, 2013). Namun secara umum gim dapat dibagi menjadi beberapa jenis , yaitu:

1. *Action games*, Sebuah gim action membutuhkan pemain dengan kecepatan *reflex*, akurasi, dan ketepatan waktu untuk menghadapi sebuah rintangan. Gim jenis ini biasanya meliputi tantangan fisik, teka-teki (*puzzle*), dan beberapa konflik lainnya.
2. *Action-Adventure*, Gim *action-adventure* menggabungkan unsur-unsur dari gim *action* dan gim *adventure*, biasanya menampilkan rintangan yang berjangka, dimana user dapat mengeksplorasi satu tempat atau lebih, melawan musuh yang ditemui, berinteraksi dengan karakter yang ditemui, dan terkadang memecahkan teka-teki.
3. *Real Time Strategy (RTS)*, merupakan gim dimana user harus melibatkan masalah strategi, taktik, dan logika. Contoh gim jenis ini adalah *Age of Empire*, *War Craft*, dan sebagainya.
4. *Fighting game*, merupakan gim yang terdapat dua karakter yang bertarung untuk memperoleh kemenangan. Contoh gim jenis adalah *Mortal Kombat*, *Weird Fighters*, *Tekken*, dan sebagainya
5. *Racing game*, merupakan gim yang berjenis balapan untuk bermain gim ini di butuhkan teknik agar dapat berhasil untuk menyelesaikan sebuah misi dengan waktu tercepat. Contoh gim jenis ini adalah *Need For Speed Shift*, *Puppy Racing*, *Speed Warrior*, dan sebagainya
6. *Role Playing Games (RPG)*, kebanyakan gim jenis ini melibatkan masalah taktik, logika, dan eksplorasi penjelajahan. Dan juga kadang

meliputi teka-teki. Contoh dari gim ini adalah *Final Fantasy*, *Ragnarok*, *Lord of The Rings*, dan sebagainya.

7. *Slide scrolling games*, pada jenis gim ini karakter dapat bergerak ke samping diikuti dengan gerakan background. Contoh gim tipe seperti ini adalah *Super Mario*, *Metal Slug*, dan sebagainya.
8. *Puzzle games*, ditujukan untuk memecahkan suatu masalah tertentu. Jenis gim ini mengharuskan *user* untuk berfikir menggunakan logika yang biasanya dibatasi oleh waktu.
9. *Shooting game*, merupakan gim yang bertipe menembak musuh atau sasaran tertentu. Contoh gim jenis ini adalah *Time crisis*, *House of The Dead*, dan sebagainya.
10. *Casual game*, merupakan sebuah gim yang dibuat untuk semua kalangan, dan dapat dimainkan setiap orang tanpa harus memiliki kemampuan khusus. Contoh gim ini adalah *Abduction! World* (Riyadi, 2013).

2.2.3 Artificial Intelligence

Artificial Intelligence atau Kecerdasan Buatan adalah kecerdasan yang dimiliki oleh mesin yang dapat bertindak seperti layaknya manusia. Pada gim, AI bertindak dalam mengendalikan karakter musuh dan karakter pembantu dari musuh agar dapat bermain melawan *player* (Christie et al., 2020). “kecerdasan buatan” atau dikenal dengan istilah *artificial intelligence* (AI) diciptakan pada tahun 1950-an namun pengembangannya relatif lambat. Penerapan kecerdasan buatan dalam gim berkembang setelah 20 tahun

kecerdasan buatan ini diciptakan. Hal ini dikarenakan keterbatasan mesin komputasi pada masa itu dalam melakukan perhitungan dan serta ruang penyimpanan. Di sisi lain, proses penciptaan gim masih sangat sederhana (Troy, 2015).

Kata *intelligence* atau “kecerdasan” sendiri berdasar KBBI memiliki arti: 1 perihal cerdas; 2 perbuatan mencerdaskan; kesempurnaan perkembangan akal budi (seperti kepandaian, ketajaman pikiran). Dalam kondisi ini, agen cerdas dalam suatu gim ditanamkan suatu pengetahuan (*knowledge base*) tentang lingkungan, aturan permainan, musuh, dan mungkin juga kemampuan untuk menambahkan pengetahuan seiring dengan perjalanan gim itu sendiri. Namun bukan berarti bahwa agen cerdas ini menjadi sangat reaktif, akan tetapi agen cerdas ini justru diharapkan mampu melakukan aksi untuk meningkatkan kualitas gim dalam memecahkan suatu skenario

2.2.4 Sistem Waypoint dengan Perhitungan Vektor

Pada *pathfinding*, *waypoint* (Liden, 2002) biasanya lebih dikenal dengan sebutan representative point atau node. Developer akan menaruh waypoint sebagai cara untuk memberitahu karakter area yang bisa dilewati. Lalu untuk mengetahui rute terdekat maka tiap koneksi antar waypoint akan diberi bobot seperti jarak atau waktu, dan developer akan menggunakan algoritma seperti djikstra atau A* untuk mencari rute terdekat (Prasetyo et al., 2019).

Sistem *waypoint* sederhana digunakan untuk mengendalikan mobil. Sistem *waypoint* mencakup sekumpulan titik jalan dan setiap titik jalan adalah koordinat dalam ruang 3D yang mewakili posisi kunci di arena dalam gim. Sistem ini digunakan oleh banyak gim balap mobil awal karena efektivitas dan kesederhanaan. Dalam *genre racing*, *waypoint* disimpan sebagai daftar posisi yang diurutkan dalam ruang 3D menggunakan daftar generik dari *namespace System.Collections.Generic* (Microsoft, 2021). Sistem gim membaca daftar *waypoint* ini sebagai masukan dan kemudian pindah melalui daftar sampai semua *waypoint* yang dimiliki telah dilewati oleh mobil yang dikendalikan gim. Sistem hanya beralih ke titik jalan berikutnya dalam daftar ketika mendeteksi bahwa mobil yang dikendalikan permainan berada dalam jarak tertentu dari *waypoint* saat ini. Bola merah (Gambar 2.1) di sepanjang *arena* menunjukkan titik jalan yang menjadi posisi kunci di *track*. Untuk membelokkan mobil ke *waypoint* saat ini, sistem melakukan serangkaian perhitungan vektor, menentukan jumlah kemudi sistem yang diperlukan untuk mobil. Perhitungan ini didasarkan pada vektor awal yang dibuat oleh kedua posisi titik arah saat ini dan posisi mobil itu sendiri saat ini. Rangkaian perhitungan ini menghasilkan vektor keluaran, yang menyediakan dasar untuk tingkat keluaran kemudi dan pengereman dari mobil, namun selama pengembangan gim, menggunakan metode *waypoint* dengan vektor perhitungan tidak efektif dalam mengendalikan mobil. Karena hubungan *nonlinier* antara vektor *input* dan *output*, mobil tidak dapat dikendalikan menggunakan metode ini saja, oleh

karena itu sistem *conditional monitoring* diadopsi untuk menambah metode *waypoint* dengan perhitungan vector (Chan et al., 2015) ..



Gambar 2.1 Waypoint Dalam Racing Gim

2.2.5 *Conditional Monitoring System*

Karena hubungan antara vektor *input* dan *output nonlinier*, tingkat *output* kemudi dan pengereman dari mobil yang dikendalikan gim tidak dapat ditentukan hanya dari *output* yang dihitung vektor dalam sistem *waypoint*, perlu untuk sistem gim menggunakan sistem *conditional monitoring* yang selanjutnya dapat menyempurnakan tingkat keluaran kemudi dan pengereman diterapkan pada mobil. Dengan kata lain, tergantung pada nilai vektor keluaran pada awal yang dihasilkan dari perhitungan vektor sistem *waypoint*, kemudi dan pengereman *output* yang diterapkan pada mobil disesuaikan dengan kondisi sistem *monitoring*. Ketika sistem *waypoint* dasar ditingkatkan dengan sistem *conditional monitoring*, mobil mampu melintasi lintasan dengan hasil yang memuaskan. Pada penambahkan sistem *conditional monitoring*, desain alternatif untuk menurunkan dan menggabungkan fungsi matematika ke dalam sistem

permainan juga dipertimbangkan. Karena ini mungkin memerlukan manipulasi matematika yang kompleks atau pemodelan fungsi matematika menggunakan teknik seperti jaringan saraf tiruan, alternatif ini akhirnya ditinggalkan (Chan et al., 2015) ..

2.2.6 Artificial Intelligence Perception

Metode ini digunakan sistem gim dalam mengendalikan mobil adalah mendeteksi pemicu dan *artificial intelligence perception*. *Trigger detection* adalah mekanisme populer yang digunakan dalam *video game* kontemporer, dan mendukung pendeteksian *object* gim tertentu ketika berada di sekitar *object* gim lain. Metode ini dapat menentukan apakah mobil yang dikendalikan permainan dapat berlari ke beberapa objek seperti dinding. Untuk mencegah tabrakan agar tidak terjadi, sistem permainan dapat menyesuaikan kemudi dan tingkat keluaran pengereman yang diterapkan pada mobil. Mekanisme yang mendukung fitur ini adalah *trigger area* yang ditambahkan ke mobil. Jika sebuah dinding masuk ke *trigger area* ini, sistem permainan akan menyadari di mana dinding itu sehubungan dengan mobil dan itu akan sesuaikan tingkat *output* yang sesuai untuk mengarahkan mobil menjauh dari dinding. Dengan penambahan *trigger area* ini, gim mampu mengendalikan mobil sehingga dapat menavigasi dengan baik. Mobil di tengah Gambar 2.2 yang disorot di wireframe mesh (hijau) memiliki deteksi *trigger area* yang didefinisikan sebagai persegi panjang (hijau) di sekitarnya. Ketika sebuah rintangan atau mobil lain masuk ke area deteksi pemicu, tingkat *output* mobil yang disorot akan disesuaikan

sehingga untuk mengarahkan mobil yang disorot menjauh dari rintangan. Teknik alternatif yang dipertimbangkan untuk diterapkan pada game ini adalah teknik *ray casting*. *Ray-casting* adalah mekanisme yang umum digunakan dalam *video game* kontemporer, yang melibatkan: menggambar sinar atau garis dari suatu objek ke arah tertentu, yang dalam gim *racing* akan menjadi arah dari tujuan mobil. Garis ini akan digunakan untuk menentukan apakah mobil akan bertabrakan dengan benda lain atau tidak, namun mekanisme *ray-casting* tidak memadai karena hanya dapat mendeteksi tabrakan dalam satu arah saat pemicu mekanisme deteksi dapat mendeteksi tabrakan ke segala arah (Chan et al., 2015) ..



Gambar 2.2 Trigger Area

2.2.7 Unity

Unity adalah *software* pengembangan gim yang awalnya dirancang untuk pembuatan game 3D, namun juga sangat mungkin dan mudah untuk

membuat gim 2D dan aplikasi menggunakan *software* ini (Glover et al., 2021). Karena lisensi gratisnya, sejumlah besar *asset*, berbagai macam dokumen pendukung, dan *tutorial* mengadopsi penggunaan Unity, hal ini menjadikan Unity menjadi salah satu *software* pembuat gim paling populer. Meskipun *software* ini dirancang untuk pengembangan game, namun Unity Lab juga mendukung penelitian di berbagai bidang yang dapat diterapkan pada game (mis. *Virtual Reality* (VR), AI). Pada tahun 2017 Unity merilis versi pertama dari ML-Agents mereka sebagai *add-on* untuk Unity yang memungkinkan pengembang dan peneliti gim bereksperimen dengan berbagai algoritma ML. *Toolkit* ini masih dalam pengembangan aktif, dan diharapkan menjadi bagian dari mesin utama dalam waktu dekat (Metz, 2020).