

Implementasi *Dynamic Difficulty Adjustment* pada *Racing Game* Menggunakan Metode *Behaviour Tree*

Isthofo Aslim Sofyan¹, Muhammad Aminul Akbar², Tri Afirianto³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹isthofiaslim@gmail.com, ²muhammad.aminul@ub.ac.id, ³tri.afirianto@gmail.com

Abstrak

Video games merupakan hiburan dan tantangan. Tanpa tantangan video games akan mudah diselesaikan dan membosankan. Namun apabila tantangan terlalu sulit dapat membuat pemain frustrasi dan menyerah. Hal ini berhubungan dengan flow-state yaitu ketika kemampuan pemain dan tantangan dari game setara. Pada umumnya setiap *game* menyediakan pengaturan tingkat kesulitan. Tingkat kesulitan yang disediakan biasanya disediakan dalam bentuk pilihan dari tingkat kesulitan mudah (*Easy*), sedang (*Medium*), dan sulit (*Hard*). Sayangnya model pengaturan seperti ini bersifat statis sehingga menimbulkan ketidaksetaraan antara pemain dan tantangan dari game. Untuk Menyelesaikan masalah tersebut *Dynamic Difficulty Adjustment (DDA)* diterapkan dalam penelitian ini. Dengan menerapkan *DDA* tingkat kesulitan permainan secara otomatis menyesuaikan kemampuan pemain seiring dengan berjalannya permainan. Untuk mendukung penerapan *DDA*, *Behaviour Tree* digunakan untuk membantu proses adaptasi AI terhadap kemampuan pemain. Hasil penerapan *Behaviour Tree* dan *DDA* menghasilkan permainan yang tidak membosankan dan tidak terlalu sulit untuk pemain.

Kata kunci: *Dynamic Difficulty Adjustment, Behaviour Tree, Racing Game, Difficulty*

Abstract

Video games is all about entertainment and challenge. Without challenge video games will be too easy and boring. But if the challenge is too hard it can make the player frustrated and give up. This related to flow-state where the goal is to find balance between player skills and game challenge. In general, almost every game has difficulty level settings. The difficulty usually ranged from easy, medium, to hard. Unfortunately, this type of difficulty is static, giving inequalities between players and AI. To solve this problem Dynamic Difficulty Adjustment is applied in this research. By applying DDA the difficulty of the game will automatically adjusts to player's ability as the game progress. To support DDA implementation, Behavior Tree is used to help AI to adapt to player's ability. By implementing both method, the game used in this research become less boring and challenging for player.

Keywords: *Dynamic Difficulty Adjustment, Behaviour Tree, Racing Game, Difficulty*

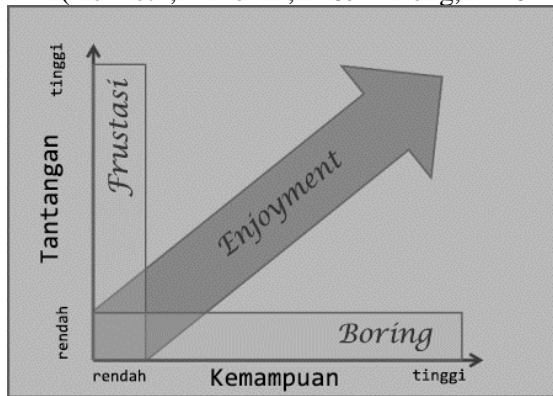
1. PENDAHULUAN

Video games merupakan hiburan dan tantangan. Video games tidak akan menyenangkan jika tidak ada tantangannya. Apabila tantangannya terlalu mudah maka akan jadi membosankan, Namun jika tantangannya terlalu sulit maka akan membuat kebanyakan orang frustrasi. Hal ini berhubungan dengan flow-state yaitu ketika kemampuan pemain dan tantangan dari game setara. Game pada umumnya menyediakan Pengaturan tingkat kesulitan (*difficulty setting*) untuk mengatur seberapa menantang sebuah permainan. Tingkat

kesulitan yang disediakan biasanya berkisar dari mudah (*Easy*), sedang (*Medium*), sampai yang sulit (*Hard*). Adanya pengaturan tingkat kesulitan ini ditujukan untuk memilihkan tantangan yang cocok bagi pemula maupun yang sudah mahir (Kenneth, Torkil, & Long, 2011).

Tingkat kesulitan game yang statis bisa menyebabkan ketidaksetaraan antara pemain dengan tantangan dari game. Untuk menyelesaikan masalah ini dibuatlah sistem *Dynamic Difficulty Adjustment (DDA)* (Kenneth, Torkil, & Long, 2011). *DDA* adalah alternatif dari *Difficulty Settings* statis yang harus ditentukan oleh pemain sebelum memulai

permainan, dengan adanya DDA pemain tidak perlu repot mengatur difficulty sebelum bermain. DDA berfungsi sebagai Difficulty Settings yang bekerja secara otomatis berdasarkan kemampuan pemain. Dengan adanya DDA pemain bisa merasakan permainan yang lebih menyenangkan dan menghibur. Hal ini berhubungan dengan Game Flow / Balance Match yaitu keadaan ketika kemampuan pemain dan tantangan game setara seperti pada Gambar 1 (Kenneth, Torkil, & Long, 2011).



Gambar 1 Enjoyment Flow

2. Dynamic Difficulty Adjustment (DDA)

Dynamic Difficulty Adjustment (DDA) adalah sistem yang mengatur difficulty secara dinamis, hal ini berhubungan dengan *Enjoyment Flow*, yang bertujuan untuk mendapatkan keseimbangan antara kemampuan pemain dengan tantangan game. Jika tingkat kemampuan lebih tinggi dari tingkat tantangan maka game akan menjadi membosankan, begitu juga sebaliknya jika tingkat kemampuan lebih rendah dari tingkat tantangan maka game akan membuat kebanyakan orang frustrasi. Dengan adanya DDA tingkat tantangan akan selalu berganti, beradaptasi dengan kemampuan pemain untuk mencapai enjoyment dalam permainan (Robin & Vernell, 2004)

3. Behaviour Tree (BT)

Behaviour Tree adalah sebuah cara agent berpikir dan berperilaku menggunakan struktur hirarki yang menyimpan perilaku dan tingkah agen. BT dapat digambarkan sebagai pohon dari node hirarki yang mengatur alur perilaku yang dilakukan AI, Behaviour Tree yang menjalankan tugas secara berurutan dari Find Ball, Pick Ball, kemudian Place Ball. Pada pohon terdapat daun, daun disini dianalogikan sebagai perintah untuk AI, sedangkan rantingnya adalah berbagai macam node utilitas yang mengatur AI yang

dipilih berdasarkan perintah dari daun, hingga turun ke batang pohon untuk memilih urutan perintah yang cocok pada situasi yang diberikan oleh daun.

4. Node pada Behavior Tree

Behaviour Tree memiliki beberapa tipe node. Penjelasan tipe node dapat dilihat di Tabel 1.

Tabel 1 Rancangan Analisa Komputasi

Type Node	Penjelasan
Composite	Adalah node yang mewakili satu cabang dari Behaviour Tree dan menjadi acuan untuk bagaimana cabang tersebut berjalan
Decorator	Juga dikenal sebagai Conditional. Biasanya ini terhubung dengan node lain dan menentukan apakah node itu bisa dijalankan atau tidak
Tasks	Adalah daun dari Behaviour Tree. Node yang menjalankan tugas dan tidak punya koneksi lanjutan
Service	Biasanya terhubung dengan node Composite dan akan menjalankan tugas pada preferensi yang ditentukan selama cabang tree yang terhubung berhasil berjalan
Root	Adalah Starting point dari Behaviour Tree, node ini tidak bisa dihubungkan dengan node Decorator atau node Service.

5. PERANCANGAN

Genre permainan yang digunakan dalam penelitian ini adalah Racing Game. Agen AI yang dimaksud disini adalah karakter yang akan menjadi lawan pemain dalam balapan. AI disini memiliki 5 tingkat kesulitan, dimana setiap tingkat kesulitan akan memiliki attribute value yang berbeda – beda. Tingkat kesulitan AI akan ditentukan dari kemampuan pemain dan akan berganti – ganti sesuai tingkat kemampuan pemain.

4.1 Perancangan Agen AI

Agen AI dirancang dengan membagi tingkat kesulitan menjadi lima jenis kesulitan AI. Yaitu *dif1*, *dif2*, *dif3*, *dif4*, *dif5*. Tingkat kesulitan tersebut berbeda-beda berdasarkan *Acceleration Sensivity*, *Acceleration Wander*, *Brake Sensitivity*, dan *Nitro Probability*. Untuk menentukan *difficulty AI*, dibutuhkan input dari

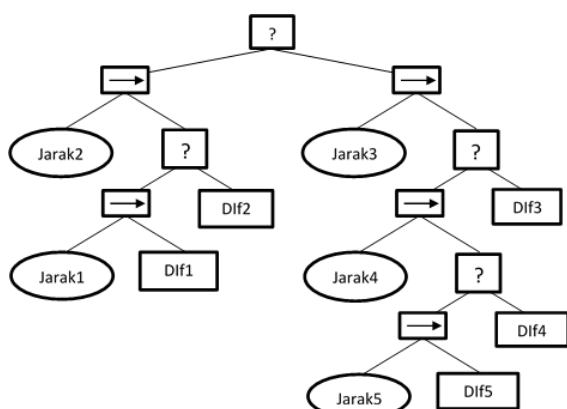
pemain berupa jarak antara pemain dengan AI. Perubahan *difficulty* AI akan diproses ulang pada setiap *checkpoint* yang dilewati oleh pemain.

Tabel 2 Nilai Jarak

Jarak	Tingkat Kesulitan	Penjelasan
Jarak1 < (-50)	Dif1	Jarak antara kendaraan Pemain dan AI lebih kecil dari (-50) unit jarak pada unity
Jarak2 < (0)	Dif2	Jarak antara kendaraan Pemain dan AI lebih kecil dari (0) unit jarak pada unity
Jarak3 > 50	Dif3	Jarak antara kendaraan Pemain dan AI lebih besar dari 50 unit jarak pada unity
Jarak4 > 100	Dif4	Jarak antara kendaraan Pemain dan AI lebih besar dari 100 unit jarak pada unity
Jarak5 > 150	Dif5	Jarak antara kendaraan Pemain dan AI lebih besar dari 150 unit jarak pada unity

Dapat dilihat pada Tabel 2, jika jarak antara pemain dan AI DDA lebih kecil dari (-50), maka tingkat kesulitan AI DDA akan diatur menjadi Dif1, jika jarak antara pemain dan AI DDA lebih kecil dari (0) dan tidak lebih kecil dari (-50) maka tingkat kesulitan akan diatur menjadi Dif2, dan begitu seterusnya.

4.2 Perancangan Behavior Tree



Gambar 2 Perancangan Behaviour Tree

Gambar 2 merupakan representasi kondisi perubahan tingkat kesulitan AI dalam bentuk Behavior Tree berdasarkan selisih jarak antara AI dan pemain.

6. IMPLEMENTASI

Agan AI disini menggunakan Game Object pada game engine Unity. Terdapat dua Agen AI yaitu AI Statis yang menggunakan *difficulty* statis, dan AI DDA yaitu AI yang sudah diterapkan sistem *Dynamic Difficulty Adjustment*. Dapat dilihat pada Gambar 3, AI statis menggunakan mobil berwarna biru dan AI DDA menggunakan mobil berwarna putih.



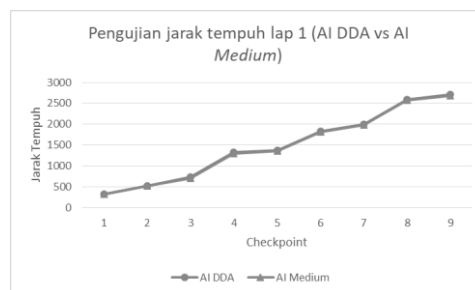
Gambar 3 AI DDA vs AI Statis

7. PENGUJIAN

Pada penelitian ini akan dilakukan pengujian efektifitas antara AI DDA terhadap AI statis dan pengujian antara AI DDA dan AI statis melawan pemain. Dari setiap skenario akan dilakukan uji coba balap sebanyak 3 lap dengan 9 checkpoint dari setiap lap dan dicatat nilai jarak tempuh dari setiap checkpoint yang dilewati. Nilai jarak dari masing-masing *difficulty* akan dijumlahkan dan dihitung rata-rata dari setiap skenario, kemudian dibuat grafik dari setiap skenario.

7.1 Pengujian AI DDA vs AI Statis

Hasil pengujian AI DDA melawan AI statis dengan tingkat kesulitan *medium* dapat dilihat pada gambar 4,5, dan 6.



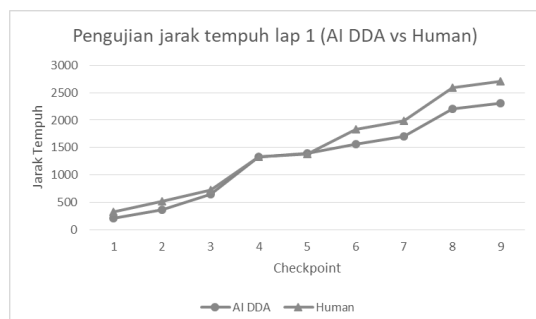
Gambar 4 Hasil Pengujian AI DDA vs AI Medium

Dapat dilihat pada Gambar 4 AI *medium* berada pada posisi depan, kemudian setelah melewati checkpoint 2, AI DDA membalap AI *medium* dan merebut posisi depan karena menyesuaikan dengan kemampuan AI *medium*. Adanya saling balap antara AI DDA dan AI

medium menunjukkan bahwa AI DDA efektif terhadap lawan dengan kemampuan sedang.

7.2 Pengujian Pemain vs AI DDA

Hasil pengujian pemain melawan AI DDA dapat dilihat pada Gambar 7,8, dan 9.



Gambar 5 Hasil Pengujian 1 AI DDA vs Pemain

Dapat dilihat pada Gambar 7 pemain berada pada posisi depan, kemudian setelah melewati *checkpoint* 3, AI DDA membalap pemain dan merebut posisi depan dengan menaikkan kemampuan AI DDA, namun pada *checkpoint* 5 pemain kembali merebut posisi depan dan memperbesar selisih jarak antara pemain dan AI DDA. Adanya saling balap antara pemain dan AI DDA menunjukkan bahwa AI DDA efektif terhadap lawan pemain berkemampuan handal.

8. KESIMPULAN DAN SARAN

Berdasarkan pengujian yang dilakukan terhadap implementasi DDA pada AI *racing game* dengan metode *Behaviour Tree*, maka dapat diambil kesimpulan sebagai berikut:

1. Penerapan *Dynamic Difficulty Adjustment* pada AI *racing game* menghasilkan AI dinamis yang bisa beradaptasi sesuai kemampuan lawan menggunakan masukan selisih jarak pada *Behaviour Tree* dan menghasilkan keluaran untuk mengubah parameter atribut AI.
2. Dari hasil pengujian AI dinamis dan AI statis melawan pemain yang dilakukan, didapat bahwa AI statis bertingkah laku *monotone* dan tidak bisa menyesuaikan dengan kemampuan pemain, sedangkan AI dinamis bertingkah laku secara dinamis dan dapat mengimbangi kemampuan pemain dengan baik sehingga membuat permainan jadi menyenangkan.

Saran untuk mendapatkan hasil yang lebih baik dari penelitian ini, dilakukan beberapa perbaikan sebagai berikut:

1. Diharapkan pada penelitian selanjutnya, dapat menambah atribut pada AI DDA supaya dapat berjalan lebih baik.
2. Penambahan parameter untuk perbandingan AI DDA dengan AI statis pada pengujian, sehingga rasio perbandingan data uji yang didapatkan akan meningkat.

9. DAFTAR PUSTAKA

- Alexander, S., Francisco, M. G., Matthew, J., Robert, M., & Norman, I. B. (2011). Parameterizing Behavior Trees.
- Andrade, G., Santana, H., Ramalho, G., & Corruble, V. (2005). Extending Reinforcement Learning to Provide Dynamic Game Balancing.
- Csikszentmihalyi, M. (1991). Flow: The Psychology of Optimal Experience.
- Intense Games. (2015, Juni 30). Racing Game Starter Kit. Retrieved from [assetstore.unity.com: https://assetstore.unity.com/packages/templates/racing-game-starter-kit-22615](https://assetstore.unity.com/packages/templates/racing-game-starter-kit-22615) [Diakses 17 Juni 2018]
- Kenneth, S., Torkil, O., & Long, H. (2011). Dynamic Difficulty Adjustment Using Behaviour Trees.
- Meniku. (2016, Juni 02). NPBehave - An event driven Behavior Tree Library for code based AIs in Unity. Retrieved from [unitylist.com: https://unitylist.com/p/3ob/NP-Behave](https://unitylist.com/p/3ob/NP-Behave) [Diakses 21 Juli 2018]
- Michele, C., & Petter, O. (2017). Behavior Trees in Robotics and AI.
- Mirna, P., Victor, d., & Luiz, C. (2016). Dynamic Difficulty Adjustment on MOBA Games.
- Rabin, S. (2008). AI Game Programming Wisdom 4. First Edition.
- Robin, H., & Vernell, C. (2004). AI for Dynamic Difficulty Adjustment in Games.
- Simon, T., & Nic, M. (2013). An Architecture Overview for AI in Racing Games.