Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 3, No. 4, April 2019, hlm. 3252-3257 http://j-ptiik.ub.ac.id

Penerapan *Naïve Bayes* untuk *NPC Braking Decision* pada *Racing Game*

Steven Willy Sanjaya[1], Muhammad Aminul Akbar[2], Tri Afirianto[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]steven.willy24@gmail.com, [2]muhammad.aminul@ub.ac.id, [3]tri.afirianto@ub.ac.id

Abstrak

Racing game merupakan salah satu game yang masih populer hingga saat ini. Pada perkembangannya selalu tidak lepas dari dibutuhkannya NPC (*Non-Player Character*). NPC yang hadir sebagai lawan main dari pemain selalu dihadapkan dengan permasalahan bagaimana membuat NPC yang cerdas. Salah satu permasalahannya adalah *braking decision* yaitu kapan NPC harus mengurangi kecepatannya dengan menggunakan rem. Salah satu metode yang umum digunakan adalah *Brake Zone*, namun metode tersebut kurang efektif karena harus memasang *zone* secara manual di setiap tikungan yang dibutuhkan. Solusi lain seperti *Smart AI System* pada *Racing Game Starter Kit* (RGSK) kurang efektif karena diperlukan konfigurasi yang tepat untuk memperoleh hasil yang optimal. Untuk mengatasi masalah tersebut peneliti menerapkan metode *machine learning* yaitu *Naïve Bayes* dalam *braking decision*. *Naïve Bayes* menggunakan tiga fitur untuk masukan dan dua kelas keluaran yang data latihnya diperoleh dari pemain. Hasil pengujian menunjukkan bahwa hasil *braking decision* dari *Naïve Bayes* mampu membuat kendaraan tidak menabrak pembatas di luar lintasan tanpa menurunkan FPS (Frames per Second) dari *game*. Perolehan waktu setiap lap dari *Naïve Bayes* mampu mengikuti waktu pemain dengan rata-rata 52,5 detik selama 10 lap.

Kata kunci: *racing game, naïve bayes, braking decision, NPC*

Abstract

Racing is a video game genre that is still popular today. Its development processes cannot be separated from the need to have Non-Player Character (NPC) in them. NPCs act as the opponents for the players, and thus the developers are always challenged with the problem of how to make the NPCs smarter than them. One of the problems is related with breaking decision, specifically when the NPCs decided to slow down their speed during races by using brakes. One commonly used method for this type of experiment is the Brake Zone. Although, this method also has its own shortcomings, such as the devs have to manually place the zone themselves in the designated locations for the brake test. Other solution that can be applied is Smart AI System by Racing Game Starter Kit (RGSK), but this also has its problem in which to get the best result, a proper configuration is needed. To resolve the problem, researcher proposes the method of machine learning, Naïve Bayes for the braking decision. Naïve Bayes use three features for the data input, and two output class in which the data will be obtained from the player. The test result showed that the braking decision from Naïve Bayes was able to prevent the vehicle from crashing with the outer wall without dropping the game's FPS (Frames per Second). Time acquisition each lap from Naïve Bayes was able to keep up with the player's time at an average of 52,5 seconds during 10 laps.

Keywords: *racing game, naïve bayes, braking decision, NPC*

1. PENDAHULUAN

Racing game atau *racing video game* merupakan salah satu jenis *game* yang sudah ada sejak tahun 1969. *Racing game* merupakan salah satu bagian dari genre *Vehicle Simulations*, yang memiliki tujuan untuk memberikan sebuah pengalaman bagaimana mengemudikan berbagai kendaraan baik nyata maupun imaginer (Rollings & Adams, 2003). *Racing Game* sesuai namanya berfokus pada siapa yang paling cepat sampai ke garis *finish*. Hingga saat ini, *racing game* memiliki peminat yang cukup besar, terbukti dengan suksesnya berbagai game seperti seri Need For Speed

yang keseluruhan seri pada tahun 2009 sudah terjual 100 juta kopi (Totu, 2009) dan hingga saat ini *sequel* masih terus dibuat yang *game* terkininya adalah Need For Speed Payback yang rilis 6 November 2017 (Electronic Arts, 2017). Seri lain yang juga sukses hingga saat ini adalah Mario Kart dari Nintendo yang sudah terjual lebih dari 126 juta kopi dari 9 *game* yang sudah dirilis (VGsales, 2018) dan game terkininya yaitu Mario Kart 8 Deluxe yang rilis pada 28 April 2017 sudah terjual 10,35 juta kopi (Nintendo, 2018) dan menjadi juara dalam Weekly Software Charts untuk game dengan penjualan sebesar 86.595 pada 28 Juli 2018 (VGChartz, 2018).

Pengembangan *Racing Game* tidak lepas dari dibutuhkannya NPC (*Non-Player Character*) yang hadir dalam bentuk *auto-vehicle* sebagai lawan main dari pemain. Pada NPC tersebut pasti ada sebuah kecerdasan buatan yang diberikan dalam bentuk *Pathfinding* agar NPC dapat mengemudi di jalur yang sesuai. Metode yang sering digunakan adalah melakukan konfigurasi *Waypoints* dan pembagian *grid* dengan A* (Bennett & Sagmiller, 2014). Perkembangan fisika *game* yang semakin maju, menyebabkan kendaraan dapat bergerak meyerupai kendaraan asli pada dunia nyata, sehingga ketika melewati tikungan harus mengurangi kecepatan. Pada kasus ini hadir komponen lain pada NPC yaitu *Braking Decision* tentang kapan NPC harus menggunakan rem. Salah satu metode yang sering digunakan adalah *brake zone* (Bennett & Sagmiller, 2014), namun metode tersebut kurang efektif karena setiap lintasan harus dilakukan konfigurasi secara manual dengan memasang *brake zone* untuk setiap tikungan pada semua lintasan yang berbeda. Penggunaan *brake zone* salah satu contohnya ada di Racing Game Starter Kit (RGSK) v1.0.1 yang saat ini sudah terdepresiasi. Pengembangan lanjutan pada RGSK menggunakan *Smart AI System* yang dapat dikustomisasi (Intense Game, 2015), namun memiliki performa waktu yang kurang optimal akibat sulitnya memperoleh konfigurasi yang tepat. Permasalahan yang terdapat pada *brake zone* juga dapat diselesaikan menggunakan *machine learning*, salah satunya adalah *Naïve Bayes*.

Naïve Bayes merupakan klasifikasi probabilitas yang menggunakan Teorema Bayes dengan sebuah asumsi bahwa fitur-fitur yang mendeskripsikan objek secara statistik tidak terikat satu sama lain atau independen (Wang & Shang, 2017). Asumsi yang dimiliki oleh *Naïve Bayes* jarang sekali terpenuhi dalam kondisi-kondisi di dunia nyata, meski begitu pendekatan *Naïve Bayes* masih dapat berfungsi dengan bagus meskipun asumsinya tidak terpenuhi (Brownlee, 2016). Proses latih dan uji pada *Naïve Bayes* juga dapat dilakukan dengan cepat dan lebih toleran terhadap *missing data* dibanding dengan klasifikasi *Bayes Network*.

Metode *Naïve Bayes* sudah banyak diterapkan dalam berbagai bidang. Pada bidang kesehatan *Naïve Bayes* juga sering digunakan untuk membantu diagnosis penyakit, contohnya pada penyakit jantung (Jabbar & Samreen, 2016). Pada klasifikasi teks juga sudah banyak digunakan terutama dalam *spam filtering* dan terus dikembangkan seperti dipakai pada lirik lagu (An, Sun, & Wang, 2017). Saat ini selain bidang-bidang tersebut, *Naïve Bayes* juga sudah mulai diteliti pada bidang *game* contohnya pada prediksi strategi RTS *Game* (Frandsen dkk, 2010) dengan akurasi 80% dan prediksi hasil pada DOTA 2 (Wang & Shang, 2017) dengan akurasi 85,33%.

Berdasarkan dari tinjauan di atas, penulis tertarik untuk melakukan penelitian mengenai penerapan *Naïve Bayes* untuk NPC *Braking Decision* dalam *Racing Game*, perbedaan dengan penelitian sebelumnya ada pada perubahan genre dari strategi menjadi *racing* dan pada fitur rem yang mengganti *brake zone* serta *smart AI system* dengan *Naïve Bayes*. Penerapan *Naïve Bayes* terbukti efektif dalam berbagai bidang meskipun fitur yang dipakai tidak sepenuhnya independen. *Naïve Bayes* memiliki kecepatan perhitungan yang cepat dan tidak memakan *resource* komputasi yang besar (Paryudi, Ashari, & Tjoa, 2013). Sehingga diharapkan *Naïve Bayes* efektif untuk menjadi NPC *Braking Decision*, yang dapat diterapkan di berbagai *Racing Game* dengan performa akurat dan efisien.

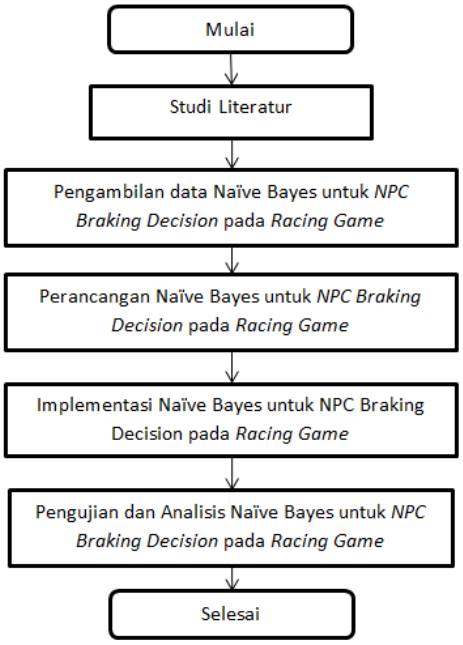
2. METODOLOGI

Diagram alir penelitian penerapan *Naïve Bayes* untuk NPC *braking decision* pada *racing game* ditunjukkan pada Gambar 1.

Pada tahap studi literatur dilakukan dengan mempelajari buku-buku literatur dan *paper* yang berkaitan dengan NPC, *racing game,* dan *Naïve Bayes*. Studi literatur dapat berupa media cetak maupun media elektronik.

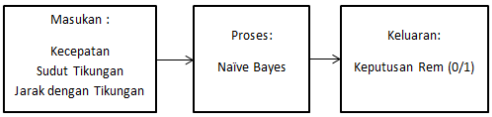
Tahap kedua dilakukan pengambilan data latih yang dibutuhkan untuk proses belajar

metode *Naïve Bayes*. Proses pengambilan data dilakukan pada pemain manusia dengan melakukan *capture* setiap 0,02 detik dan pada saat rem digunakan. Data yang disimpan adalah atribut yang dipakai dan juga keputusan yang diambil oleh pemain apakah menekan rem atau tidak. Syarat dari pemain adalah tidak pernah keluar jalur saat bermain.



Gambar 1. Diagram Alir Penelitian

Tahap ketiga dilakukan perancangan dengan menjelaskan secara lebih detail mengenai fitur sebagai masukan dan kelas yang menjadi keluaran *Naïve Bayes*. Setelah itu dilakukan perancangan untuk NPC *Braking Decision* dengan menggunakan metode *Naïve Bayes*. Rancangan proses *Naïve Bayes* yang dilakukan untuk memperoleh keluaran kelas rem dan tidak rem ada pada Gambar 2.



Gambar 2. Diagram Perancangan

Tahap keempat dilakukan implementasi sesuai dengan yang telah dirancang sebelumnya pada proses perancangan dengan menggunakan bahasa *C#* pada *game engine* Unity 3D. Implementasi dilakukan dengan membuat kelas untuk *Naïve Bayes* (Persamaan 1) dan diaplikasikan langsung ke dalam RGSK yang metode *braking decision* dilakukan setiap *fixedUpdate* (setiap *framerate frame* yang tetap, digunakan untuk interaksi-interaksi *physics*). Bagian *Brake Zone* dinonaktfikan dan diganti dengan *Naïve Bayes*.

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \quad (1)$$

Persamaan 1 merupakan persamaan dari *Naïve Bayes* yang diimplementasikan. $p(C_k \mid x)$ merupakan probabilitas *posterior*, $p(C_k)$ merupakan probabilitas *prior*, $p(x| C_k)$ merupakan *likelihood* yaitu peluang kemunculan nilai nilai fitur (x) pada kelas tertentu (C_k) dan terakhir p(x) merupakan *evidence* yaitu peluang munculnya fitur (x) dari seluruh pengamatan.

Tahap kelima dilakukan pengujian *game* yang telah diimplementasikan. Pengujian dilakukan dengan mencoba pada setiap tikungan pada *circuit*. Proses pengujian untuk menentukan apakah NPC *Braking Decision* tersebut bekerja, dilakukan dengan menghitung waktu tempuh NPC, menyimpan jalur yang diambil untuk melihat apakah keluar jalur dan menabrak dinding.

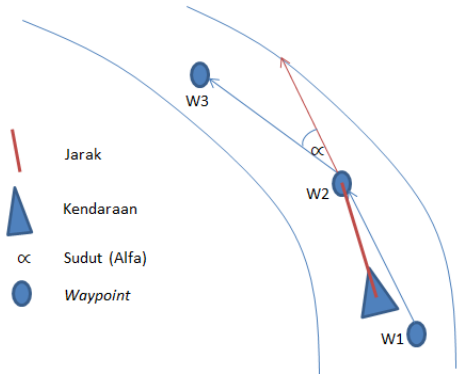
3. PERANCANGAN

3.1 Penentuan Fitur Naïve Bayes

Pada penelitian ini digunakan tiga fitur yang berpengaruh terhadap *braking* decision. Fitur pertama adalah kecepatan kendaraan. Nilai kecepatan kendaraan diperoleh dari variabel currentSpeed yang menyimpan data kecepatan mobil saat ini. Fitur ini sangat berpengaruh terhadap keputusan pengambilan rem, karena dalam kecepatan rendah penggunaan rem lebih jarang meskipun akan melewati tikungan yang tajam. Pada kecepatan tinggi rem menjadi lebih dibutuhkan terutama ketika akan melewati tikungan.

Fitur kedua adalah jarak kendaraan dengan tikungan. Data tersebut diperoleh dengan cara menghitung jarak antara posisi kendaraan saat ini dengan tikungan yang akan dilewatinya. Pada sepanjang lintasan terdapat *waypoints* yang membantu dalam proses navigasi. Setiap *waypoint* memiliki informasi posisi dan pada tiap tikungan pasti terdapat *waypoint*. Sehingga informasi jarak antara kendaraan dengan tikungan dapat diperoleh dari perhitungan antara posisi *waypoint* selanjutnya dengan posisi kendaraan.

Fitur ketiga adalah sudut pada tikungan. Nilai fitur sudut pada tikungan merupakan seberapa besar sudut tikungan yang akan segera dilewati oleh kendaraan. Nilai tersebut diperoleh dengan cara menghitung sudut yang terbentuk dari dua vektor. Kedua vektor tersebut diperoleh dari tiga *waypoint.* Untuk memperjelas peroleh sudut dan jarak pada tikungan dapat dilihat pada ilustrasi yang terdapat pada Gambar 3.



Gambar 3. Ilustrasi Sudut dan Jarak Tikungan

3.2. Penentuan Kelas Naïve Bayes

Penentuan kelas yang terbentuk sebagai keluaran dari *Naïve Bayes* pada NPC *Braking Decision* terdiri dari dua buah kelas, yaitu kelas rem dilakukan dan kelas tidak melakukan rem yang ditandai dengan angka 0 dan 1. Pada angka 1 dilakukan pengereman kendaraan sementara pada angka 0 tidak dilakukan pengereman kendaraan. Pada data latih, ketika gas tidak ditekan juga dianggap melakukan rem dan masuk ke kelas 0.

4. IMPLEMENTASI

Pada tahap implementasi, perancangan yang sudah dilakukan diterapkan ke dalam *method NavigateAI* yang berada dalam *fixedUpdate*. *Naïve Bayes* yang telah melalui tahap latih melakukan tahap uji setelah menerima ketiga fitur yang telah ditentukan dan menghasilkan kelas hasil klasifikasi. Hasil tersebut selanjutnya diteruskan ke dalam *FeedInput* yang menangani *throttle*, *brake* dan *steering*.

Tabel 1. *Pseudocode Braking Decision*

No	Pseudocode
1	float mybrake
2	float distance = jarak kendaraan dengan waypoint selanjutnya (w2)
3	var w3w2 = vektor antara w3 dan w2
4	var w2w1 = vektor antara w2 dan w1
5	float angle = sudut antara w3w2 dan w2w1
6	mybrake = Tahap Uji Bayes 3 fitur (currentSpeed, angle, distance)
7	Kirim mybrake ke penanganan keputusan rem

Tabel 1 merupakan *pseudocode* dari *braking decision* yang telah dirancang. Pada baris 1 merupakan deklarasi variabel *mybrake* yang digunakan untuk menyimpan hasil uji. Baris 2 menghitung jarak antara kendaraan dengan tikungan. Baris 3-5 menghitung sudut pada tikungan. Baris 6 mengirim tiga fitur yang telah diperoleh untuk masuk dalam tahap uji *Naïve Bayes*, kelas hasil uji disimpan dalam *mybrake* yang selanjutnya dikirim ke proses selanjutnya untuk menangani keputusan rem pada baris 7. Variabel *w1 w2 w3* mengacu pada Gambar 3.

5. PENGUJIAN DAN ANALISIS

Pada tahap pengujian dan analisis akan dipaparkan hasil pengujian berupa waktu tempuh dan jalur yang diambil dari NPC *Naïve Bayes* dan dibandingkan dengan data awal pemain yang digunakan dalam proses latih.

Pada bagian ini dibahas mengenai waktu tempuh dari NPC dan pemain yang dipakai dalam tahap latih. Tabel 2 menunjukkan waktu tempuh pemain selama 2 lap. Tabel 3 menunjukkan waktu tempuh NPC yang dijalankan selama 10 lap. Pengambilan pada pemain hanya dilakukan pada 2 lap karena kondisi dari lap ke-1 ketika melewati garis *start* dimulai dari berhenti, sementara lap ke-2 ketika kembali ke garis awal tidak dimulai dari berhenti. Sehingga lap ke-2 dan seterusnya diasumsikan memiliki waktu yang serupa.

Tabel 2. Waktu Tempuh Pemain

Lap	Waktu (mm:ss)
1	00:59,86
2	00:51,30

Tabel 3. Waktu Tempuh NPC

Lap	Waktu (mm:ss)
1	01:00,98
2	00:51,47
3	00:51,82
4	00:51,54
5	00:51,48
6	00:51,52
7	00:51,67

8	00:51,35
9	00:51,82
10	00:51,35
Total	08:45,00
Rata-rata	00:52,50

Dari segi perolehan waktu, *Naïve Bayes* dapat mengimbangi waktu dari pemain pada lap ke-2 hingga lap ke-10. Pemain memiliki waktu 51,30 detik, sementara pada lap ke-2 hingga lap ke-10 NPC memiliki waktu yang berkisar pada 51 detik. Data tersebut menunjukkan bahwa NPC *Naïve Bayes*, berhasil belajar dan dapat memperoleh waktu yang mirip dengan pemain meskipun pada lap ke-1 NPC tertinggal sekitar 1 detik dibanding waktu yang diperoleh oleh pemain. Perbedaan yang ada disebabkan karena NPC hanya belajar mengenai keputusan rem dari pemain saja, sehingga dapat terjadi perbedaan dalam proses *steering*. Selama 10 lap, NPC juga tidak pernah menabrak dinding yang ada diluar lintasan.

6. KESIMPULAN

Dalam Penelitian ini dapat diambil kesimpulan bahwa:

1. Perancangan *Naïve Bayes* untuk NPC *braking decision* pada *racing game* dilakukan dengan menggunakan tiga fitur yaitu kecepatan kendaraan, jarak dengan tikungan, dan sudut pada tikungan. Sementara kelas keluaran terdiri dari dua kelas yaitu kelas 0 (tidak dilakukan pengereman) dan kelas 1 (dilakukan pengereman).
2. Penerapan Naïve Bayes untuk NPC *braking decision* pada *racing game* dilakukan pada *FixedUpdate* Unity, fitur-fitur masukan ditangkap di setiap *frame FixedUpdate* dan tahap uji *Naïve Bayes* dilakukan langsung setelah memperoleh ketiga fitur sebagai masukan dan menghasilkan kelas hasil prediksi yang menentukan apakah rem digunakan atau tidak.
3. Performa dari NPC *braking decision* menggunakan *Naïve Bayes* memiliki hasil yang positif, dari segi kecepatan NPC *Naïve Bayes* mampu mengikuti pemain dengan hanya selisih 1 detik pada lap pertama dan seimbang untuk lap selanjutnya serta tidak pernah menabrak dinding.

NPC *braking decision* dapat dikembangkan lebih lanjut lagi dengan mengubah kelas keluaran menjadi kontinyu dari 0 hingga 1 dengan menggabungkan dengan metode lain atau menggunakan metode lain, serta menambah fitur yang digunakan sebagai masukan dari metode yang dipilih.

7. DAFTAR PUSTAKA

An, Y., Sun, S., & Wang, S., 2017. Naive Bayes Classifiers for Music Emotion Classification Based on Lyrics. *International Conference on Computer and Information Science*, Volume 16, pp. 635-638.

Bennett, C., & Sagmiller,D.V., 2014. *Unity AI Programming Essentials*. Birmingham: Packt Publishing.

Brownlee, J., 2016. *Naive Bayes for Machine Learning*. [online] Tersedia di: <https://machinelearningmastery.com/naive-bayes-for-machine-learning/> [Diakses 28 Agustus 2018].

Electronics Arts, 2017. *Need For Speed Payback*. [online] Tersedia di: <https://www.origin.com/sgp/en-us/store/need-for-speed/need-for-speed-payback> [Diakses 7 September 2018].

Frandsen, F., Hansen, M., Sorensen, H., Sorensen P., Nielsen, J. & Knudsen, J., 2010. *Predicting Player Strategies in Real Time Strategy Games*. S2. Aalborg University.

Intense Games, 2015. *Racing Game Starter Kit*. [online] Tersedia di: <https://forum.unity.com/threads/v2-0-beta-racing-game-starter-kit-easily-create-racing-games.337366/> [Diakses 22 November 2018].

Jabbar, M.A., & Samreen, S., 2016. Heart disease prediction system based on hidden naïve bayes classifier. *International Conference on Circuits, Controls, Communications and Computing*, p.49.

Nintendo, 2018. *Top Selling Title Sales Units*. Tersedia di: <https://www.nintendo.co.jp/ir/en/finance/software/index.html> [Diakses 7 September 2018].

Paryudi, I., Ashari, A., & Tjoa, A.M., 2013. Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool. *International Journal of Advanced Computer Science and Applications*, Volume 4, pp. 33-39.

Rollings, A. & Adams, E., 2003. *Andrew Rollings and Ernest Adams on Game Design*. San Fransisco: New Riders.

Totu, F., 2009. *100 million Need for Speed Games Have Been Sold to This Day*. [online] Tersedia di: <https://news.softpedia.com/news/100-million-Need-for-Speed-Games-Have-Been-Sold-to-This-Day-125015.shtml> [Diakses 7 September 2018].

VGChartz, 2018. *Weekly Sofware Charts*. [online] Tersedia di: <http://www.vgchartz.com/> [Diakses 7 September 2018].

VGsales, 2018. *Mario Kart*. [online] Tersedia di: <http://vgsales.wikia.com/wiki/Mario_Kart> [Diakses 7 September 2018].

Wang, K., & Shang, W., 2017. Outcome Prediction of DOTA2 Based on Naive Bayes Classifier. *International Conference on Computer and Information Science*, Volume 16, pp. 591-593.