

## **Perancangan Mekanisme Interaksi Pemain dan Bot Game *Sumoball* dengan Unity 2020 Menggunakan *Addforce***

Faldi Ramadhan Firmansyah<sup>1</sup>, Alam Rahmatulloh<sup>2</sup>  
<sup>1,2</sup> Program Studi Informatika, Fakultas Teknik, Universitas Siliwangi Jl.  
Mugarsari, Kel. Mugarsari, Kec. Tamansari, Kota Tasikmalaya 46196  
e-mail: <sup>1</sup>207006093@student.unsil.ac.id, <sup>2</sup>alam@unsil.ac.id

### ***Abstrak***

Pada era teknologi ini perkembangan mekanisme *game*, semakin menarik dan unik, sehingga memiliki daya saing dengan perusahaan *game* lainnya. Pada penelitian *game* ini terdapat beberapa mekanisme yang menjadikan hasil *game* tersebut menjadi lebih menarik dan interaktif bagi pemainnya. Salah satunya yang dapat menyumbang hasil itu adalah implementasi mekanisme yang diterapkan dalam interaksi antar pemain dan bot atau AI di dalam sebuah *game*. Sehubungan dengan penelitian ini, penerapan dilakukan dengan mengimplementasikan mekanisme sederhana yang terdiri daripada sifat-sifat dasar untuk setiap objek yang menjadi keterlibatan dalam interaksi tersebut di dalam sebuah *game* berbasis Unity yang dinamakan sebagai "*Sumoball*", yang dimana *game* ini adalah sebuah *game* bergenre *arcade* yang menunjukkan sebuah pertandingan Sumo namun pemainnya dipresentasikan dengan objek bola. Mekanisme ini dibuat dengan menggunakan fungsi yang telah disediakan oleh *game engine* Unity yaitu dikenal dengan *Addforce*. Hasil menunjukkan bahwa untuk menghasilkan mekanisme yang menarik tersebut, bisa diselesaikan dengan beberapa fitur yang disediakan oleh Unity, beserta peran bot sebagai musuh untuk mengalahkan pemain bisa dibuat dengan persamaan yang sederhana.

**Kata kunci:** pengembangan gim, kecerdasan buatan, unity, tingkah laku, bot.

### ***DESIGNING PLAYER AND BOT INTERACTION MECHANISM IN SUMOBALL GAME WITH UNITY 2020 USING ADDFORCE***

### ***Abstract***

*In this technological era, the development of game mechanisms is increasingly interesting and unique, so that it has competitiveness with other game companies. In this game research, there are several mechanisms that make the game results more interesting and interactive for the player. One of them that can contribute to that result is the implementation of mechanisms applied in the interaction between players and bots or AI in a game. In connection with this research, the implementation is done by implementing a simple mechanism consisting of basic properties for each object involved in the interaction in a Unity-based game called "Sumoball", which is an arcade game that shows a Sumo match but the player is represented with a ball object. This mechanism was created using a function provided by the Unity game engine known as Addforce. The results show that to produce an interesting mechanism, it can be solved with several features provided by Unity, along with the role of bots as enemies to defeat players can be made with simple equations..*

**Keywords:** game development, artificial intelligence, unity, behavior, bot.

## **Pendahuluan**

'*Game*' atau gim berasal dari kata bahasa Inggris yang memiliki arti dasar permainan. Menurut Clark C. Abt, Gim adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan "dibatasi oleh konteks tertentu", misalnya peraturan-peraturan yang ada dalam *game* tersebut (Meimaharani & Listyorini, 2015). Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada pihak yang menang dan ada pihak yang kalah, biasanya gim dilakukan dengan tidak serius atau dengan tujuan menghibur.

Perkembangan teknologi meningkat begitu pesat dari berbagai bidang, salah satunya adalah dalam bidang teknologi gim. Teknologi gim yang terlibat tidak hanya mencakup dari meningkatnya estetika grafis yang menjadi daya tarik pemain, namun juga dari kompleksitas mekanisme dan kebutuhan *resource* yang dimiliki untuk mencapai target pengalaman gim yang maksimal dan mengasyikkan (Arfiansyah, 2022). Terdapat banyak tim pengembang yang berusaha mengembangkan produk-produk gim mereka dengan berbagai cara untuk mencapai target tersebut dan berharap mendapatkan persaingan yang baik di dunia pasar.

Namun untuk membuat sebuah mekanisme gim memerlukan daya usaha dan sumber yang cukup besar. Hal ini karena, untuk membuat sebuah gim haruslah menguasai pelbagai logika komputer agar gim tersebut dapat berjalan dengan baik. Penguasaan terhadap ilmu desain dan animasi juga sangat disarankan, sehingga jika pembuatan dilakukan oleh pengembang berskala kecil, akan memakan waktu yang cukup lama.

Salah satu cara untuk mengembangkan gim dengan lebih mudah adalah melalui penggunaan game engine. *Game engine* adalah sistem perangkat lunak yang dirancang untuk bekerja pada konsol gim dan sistem operasi desktop seperti *Microsoft Windows*, *Linux* dan *Mac OS* (Putri, 2016). Salah satu *game engine* yang cukup populer adalah *Unity* dari *Unity Technologies*. Dengan menggunakan *game engine*, maka pengembangan gim akan menjadi lebih cepat, serta mengurangi biaya dan tingkat kerumitan untuk membuat mekanisme gim.

Berdasarkan uraian di atas, peneliti tertarik untuk mempelajari dasar mekanisme gim sederhana menggunakan *Unity*, dengan *framework* tiga dimensi sebagai tajuk yang diteliti. Diharapkan dengan pembelajaran ini peneliti dapat mengetahui ilmu mekanisme gim sederhana yang dapat dijadikan penunjang untuk pengembangan gim seterusnya.

## **Penelitian Terkait**

Terdapat beberapa macam implementasi yang pernah dilakukan terkait dengan pengembangan tingkah laku dalam gim. Salah satunya adalah implementasi pengembangan tingkah

laku kepuasan NPC dalam menerima layanan dari pemain di gim “Mlijo” berbasis seluler menggunakan metode *fuzzy logic* (Prastikasari, Yunus and Aditya Nugraha, 2019). Implementasi lainnya seperti di dalam gim “*Santri on The Road*” yang menggunakan metode *Finite State Machine* pada pemain dan NPC saat menentukan gerakan serta aksi untuk mengambil setiap poin melalui kitab yang muncul (Sifaulloh, Fadila and Nugroho, 2021).

## Metode Penelitian

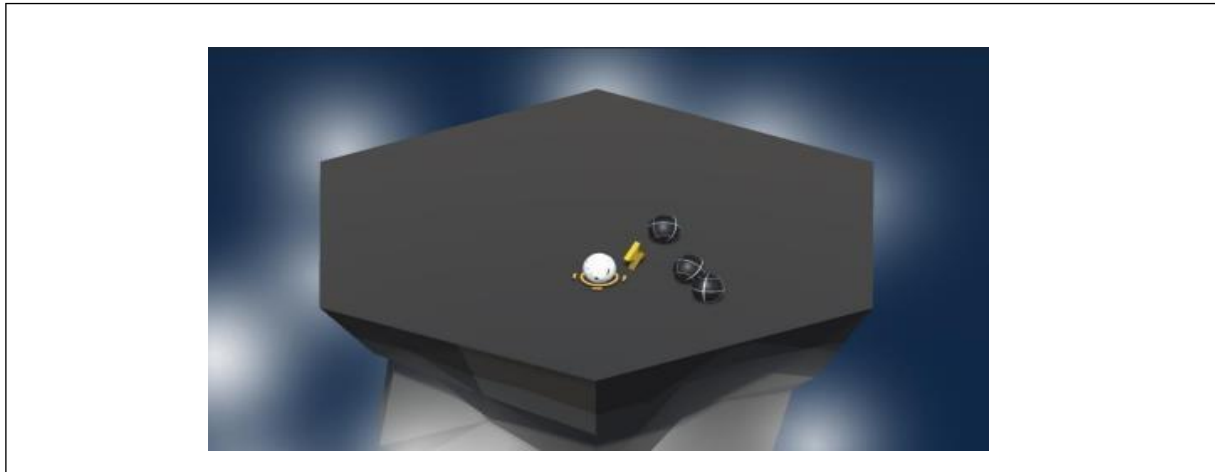
Pengembangan dalam gim ini adalah berupa implementasi sebuah fitur pemrograman bawaan *Unity 3D* yang dikenali sebagai *Addforce*, yang dapat dijadikan sebagai alternatif metode pemberian *Artificial Intelligence* (AI) pada entitas musuh dan sekaligus dapat menjadi fungsi kawalan pergerakan pada entitas pemain dalam gim tersebut tanpa harus membuat algoritma tambahan (Unity Learn, 2018). Metode ini cukup sederhana sehingga disarankan untuk pengembang gim pemula yang ingin mempelajari sifat – sifat NPC (*Non-playable Character*).

*Addforce* adalah salah satu fungsi bawaan dari *Unity* yang bekerja dengan cara menambahkan suatu kekuatan fisika pada sebuah objek. Daya ini biasanya digunakan untuk pemindahan posisi objek tersebut. Contohnya, penembakan *projectile*, peluncuran kendaraan, pendorongan objek dalam sebuah ledakan atau memindahkan karakter. Secara ringkasnya, *Addforce* menambahkan percepatan (*velocity*) pada sebuah objek dalam gim (Matthew, 2021).

Gim ini dibuat dan dikembangkan menggunakan salah satu *game engine* yang populer yaitu *Unity*. Bahasa pemrograman yang digunakan oleh *game engine* tersebut adalah C#. Gim ini terdiri daripada dua entitas utama yang bergerak yaitu pemain dan musuh (Erick, 2023).

## Hasil dan Pembahasan

Gim tersebut diberi judul sebagai “Sumoball” dan tema atau genre dari gim ini adalah Action Arcade 3D yang terinspirasi dari sebuah permainan olahraga tradisional berasal dari Jepang, yaitu Sumo. Namun karakter dalam permainan tersebut diganti dengan bentuk bola yang bisa bergerak.



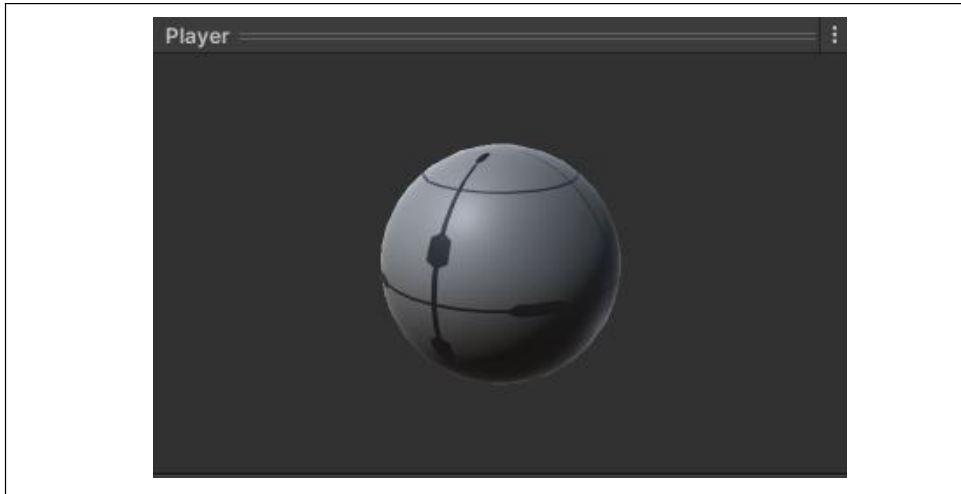
Gambar 1 Tampilan gameplay Sumoball

Pengembangan dalam gim ini adalah berupa implementasi sebuah fitur pemrograman bawaan Unity 3D yang dikenali sebagai *Addforce*, yang dapat dijadikan sebagai alternatif metode pemberian Artificial Intelligence (AI) pada entitas musuh dan sekaligus dapat menjadi fungsi kawalan pergerakan pada entitas pemain dalam gim tersebut tanpa harus membuat algoritma tambahan (Unity Learn, 2018). Metode ini cukup sederhana sehingga disarankan untuk pengembang gim pemula yang ingin mempelajari sifat – sifat NPC (*Non-playable Character*). Gim ini dibuat dan dikembangkan menggunakan salah satu *game engine* yang populer yaitu Unity (Rangan & Hendy, 2023). Bahasa pemrograman yang digunakan oleh game engine tersebut adalah C#. Gim ini terdiri daripada dua entitas utama yang bergerak yaitu pemain dan musuh. Masing– masing model entitas diberi beberapa karakteristik sebagai berikut.

### **Model Permainan (*Player*)**

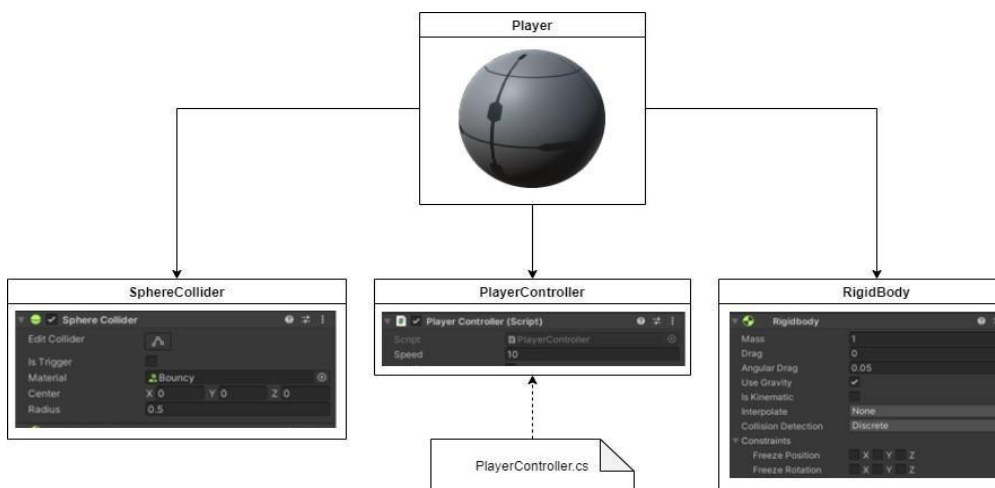
Pemain atau player berperan sebagai entitas utama yang sifatnya dapat dikawal oleh user. Misi *player* adalah untuk melakukan perlawanan terhadap musuh dengan cara menabrak entitas musuh tersebut dan berusaha untuk mengeluarkannya dari area permainan. Selain itu, *player* bisa mendapatkan *power ups* untuk mendapatkan kekuatan impuls yang lebih besar untuk menyingkirkan enemy secara instan.

Seperti yang telah diketahui sebelumnya, *player* adalah salah satu entitas yang mempunyai bentuk bola yang bisa bergerak. Desain yang diberikan kepada *player* cukup simpel, dengan tekstur kulit bola seperti di Gambar 1.

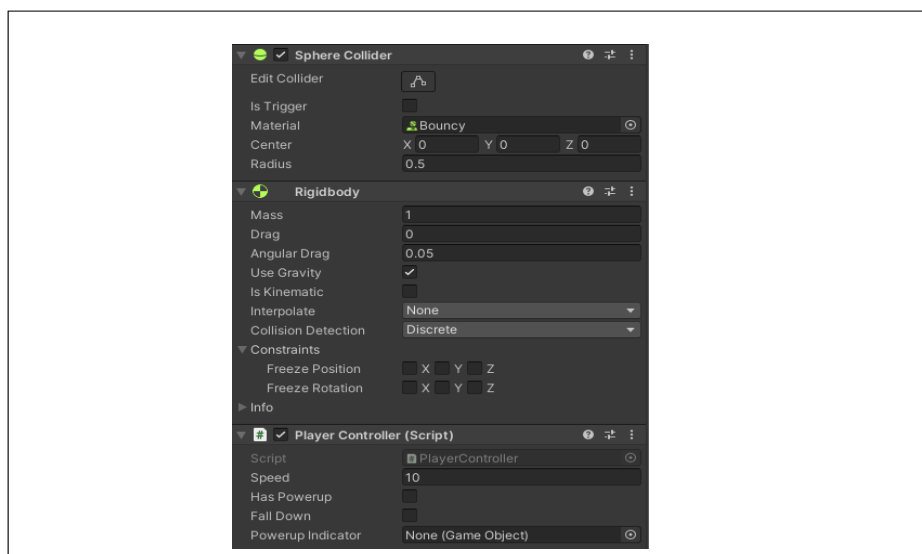


Gambar 2 Desain model player

Seperti komponen fungsi yang diberikan kepada player terdiri daripada komponen seperti yang ditunjukkan dalam Gambar 2.



Gambar 3 Diagram komponen fungsi player



Gambar 4 Komponen fungsi player di navigasi Inspector aplikasi Unity

Setiap komponen fungsi tersebut mempunyai peran yang berbeda-beda terhadap entitas *player*. Ianya sangat penting bagi agar bisa memiliki bentuk berjisim dan bisa dikawal di dalam gim. Adapun perannya adalah sebagai berikut :

a. *Rigidbody*

*Rigidbody* adalah salah satu komponen yang paling umum dan penting digunakan dalam sebuah gim. Komponen ini berperan untuk memberi jisim pada sebuah *player* sehingga terkesan seperti sebuah objek yang padat (keras) dan dapat melambung. Apabila komponen ini tidak digunakan, maka objek tersebut tidak menabrak dan akan tembus dengan objek lainnya (Filip, 2021).

b. *SphereCollider*

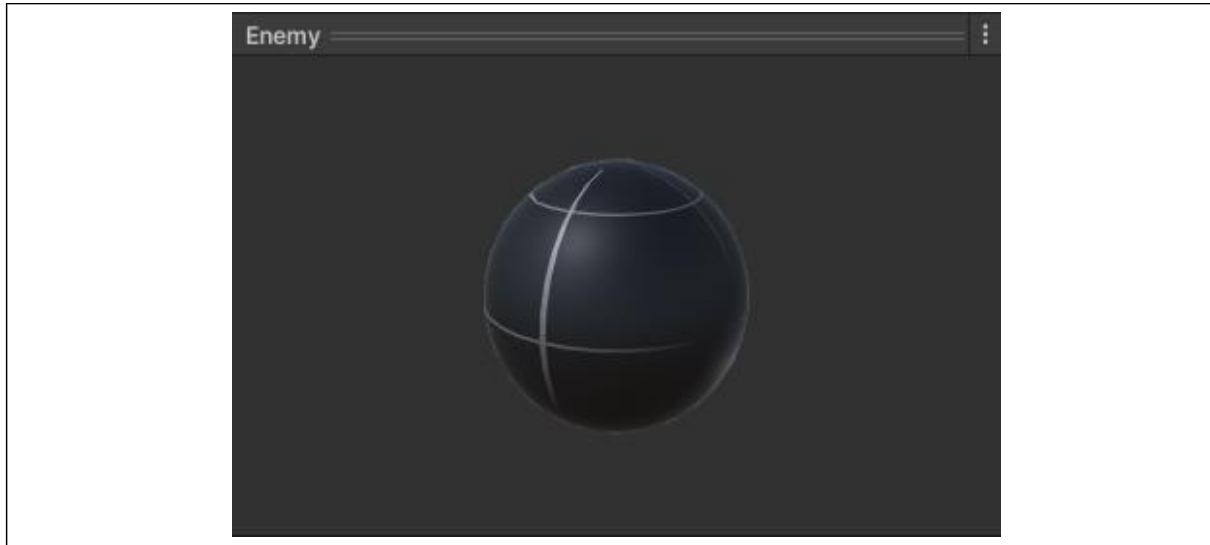
*SphereCollider* adalah salah satu komponen yang berperan untuk menjalankan beberapa perintah yang dibuat oleh pengembang apabila objek menabrak dengan objek lain. Komponen ini khusus diberikan kepada semua objek yang berbentuk bola (İsmail, 2020).

c. *PlayerController*

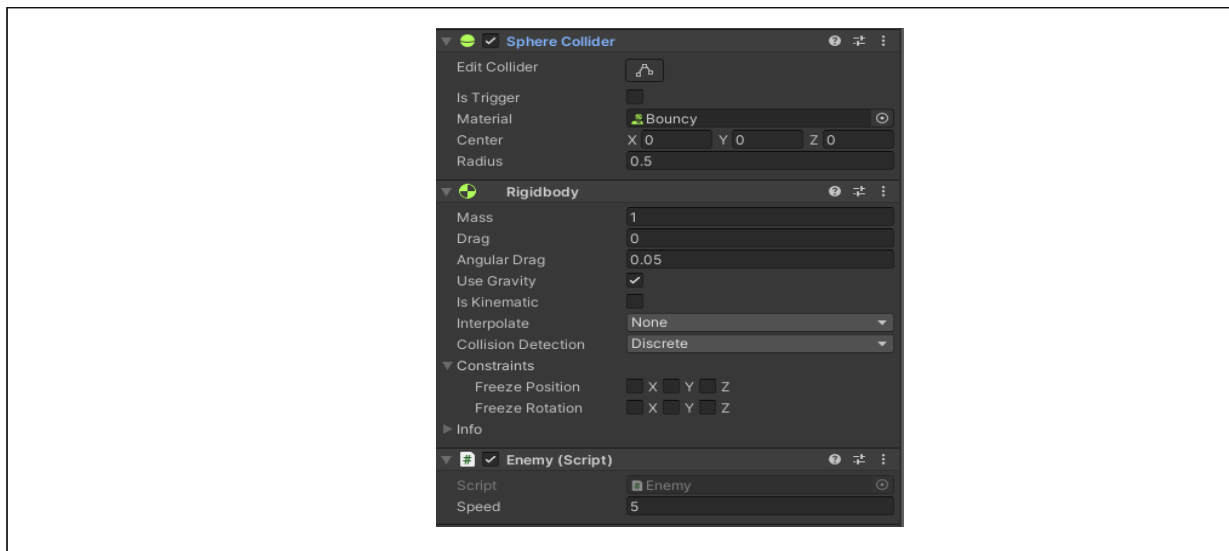
*PlayerController* adalah komponen yang dibuat oleh pengembang itu sendiri yang bersumber dari file kode pemrograman *PlayerController.cs* yang 8 berperan untuk memberikan perintah kawalan kepada *player* agar bisa bergerak (Ramadhani et al., 2023).

## **Model Musuh (*Enemy*)**

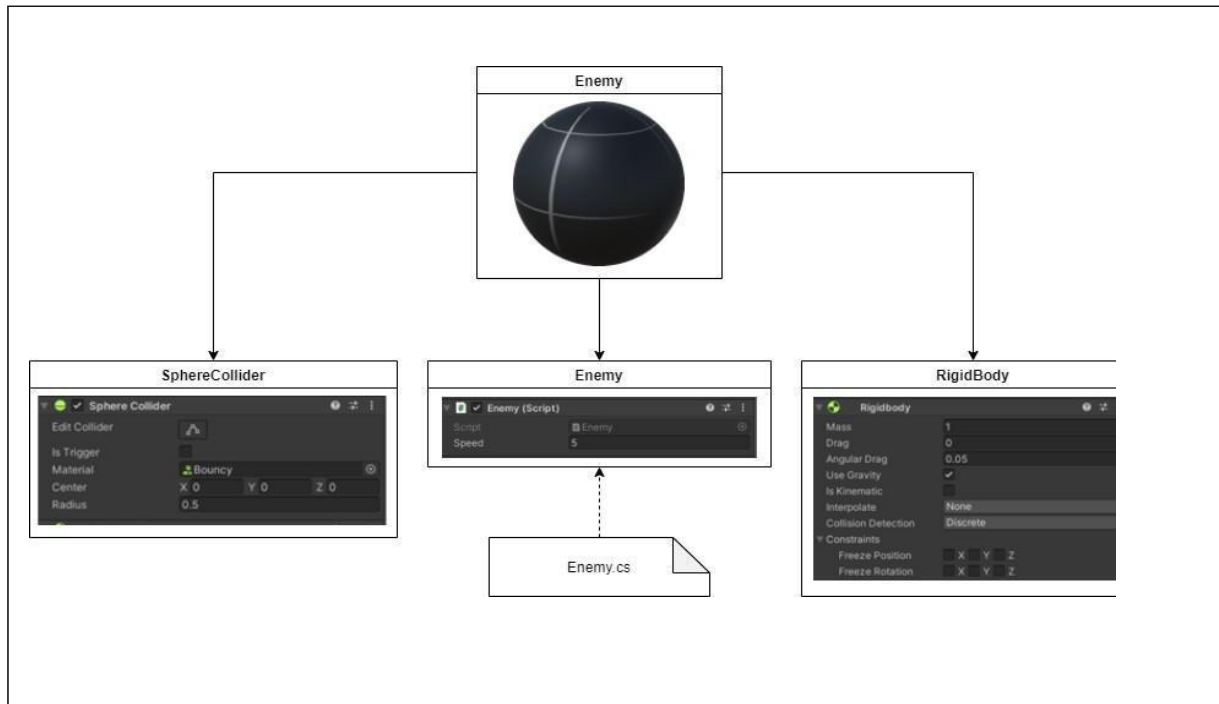
Musuh atau *enemy* berperan sebagai entitas utama kedua yang sifatnya bergerak secara mandiri tanpa kawalan dari user atau merupakan NPC dari gim ini. Misi *enemy* adalah untuk melakukan perlawanan terhadap *player* dengan cara menabrak entitas *player* tersebut dan berusaha untuk mengeluarkannya dari area permainan. Secara umumnya *enemy* memiliki karakteristik yang sama dengan *player*, yaitu salah satu entitas yang mempunyai bentuk bola yang bisa bergerak. Desain yang diberikan kepada *enemy* adalah mirip dengan *player*, namun dengan tekstur kulit bola yang bersifat berlawanan seperti di Gambar 4.



Gambar 5 Desain model enemy



Gambar 6 Diagram komponen fungsi enemy



Gambar 7 Komponen fungsi enemy di navigasi Inspector aplikasi Unity

Komponen fungsi pada *enemy* juga mempunyai kemiripan dengan komponen fungsi yang terdapat pada *player* seperti *Rigidbody* dan *SphereCollider*. Komponen yang berbeda dari *enemy* hanyalah dari komponen kode sumbernya yang berasal dari file kode pemrograman *Enemy.cs*. File kode pemrograman ini 10 terdiri dari perintah *Addforce* yang memungkinkan objek *enemy* untuk memiliki interaksi dengan *player*.

### Implementasi *Addforce*

*Addforce* adalah salah satu fungsi bawaan dari Unity yang bekerja dengan cara menambahkan suatu kekuatan fisika pada sebuah objek. Daya ini biasanya digunakan untuk pemindahan posisi objek tersebut. Contohnya, penembakan *projectile*, peluncuran kendaraan, pendorongan objek dalam sebuah ledakan atau memindahkan karakter. Secara ringkasnya, *Addforce* menambahkan percepatan (*velocity*) pada sebuah objek dalam gim (Matthew, 2021). Fungsi *Addforce* tersebut dimasukkan ke dalam masing-masing file kode pemrograman seperti *PlayerController.cs* dan *Enemy.cs*. Peran *Addforce* dalam menjalankan file kode pemrograman tersebut bertindak secara berbeda untuk setiap file.

#### d. Implementasi pada *PlayerController.cs*

*Addforce* dalam *PlayerController.cs* bertindak sebagai fungsi untuk menambahkan daya



pergerakan pada player dan posisi kamera saat *user* melakukan kawalan melalui *keyboard*. Adapun implementasinya kodenya dapat dilihat di Gambar 6.



```
1 public class PlayerController : MonoBehaviour
2 {
3     public float speed = 15.0f;
4     private Rigidbody playerRb;
5     private GameObject focalPoint;
6
7     // Start is called before the first frame update
8     void Start()
9     {
10        playerRb = GetComponent<Rigidbody>();
11        focalPoint = GameObject.Find("Focal Point");
12    }
13
14    // Update is called once per frame void Update()
15    {
16        float forwardInput = Input.GetAxis("Vertical");
17        playerRb.AddForce(focalPoint.transform.forward * speed * forwardInput);
18    }
19 }
```

Gambar 8 Implementasi Addforce di PlayerController.cs

Kode pemrograman *addforce* tersebut disimpan di dalam kelas fungsi *Update()* sehingga mengharuskan proses implementasi dilakukan di setiap frame saat gim sedang berjalan. Untuk bisa melakukan kawalan *player* berdasarkan posisi kamera, program mengambil komponen *Rigidbody* dari objek *Focal Point*. *Focal Point* berfungsi sebagai penggerak posisi kamera berdasarkan satu titik tengah di area permainan. Hasil pengambilan *Focal Point* tersebut dimasukkan ke dalam fungsi *Addforce* dan diolah melalui persamaan perkalian. Perkalian ini terdiri dari nilai posisi *Focal Point*, percepatan daya dan pembacaan input *keyboard user*.

e. Implementasi pada *Enemy.cs*

*Addforce* dalam *Enemy.cs* bertindak sebagai fungsi untuk menggerakkan objek *enemy* ke objek *player* melalui daya penarikan. Adapun implementasi kodenya dapat dilihat di Gambar 7.

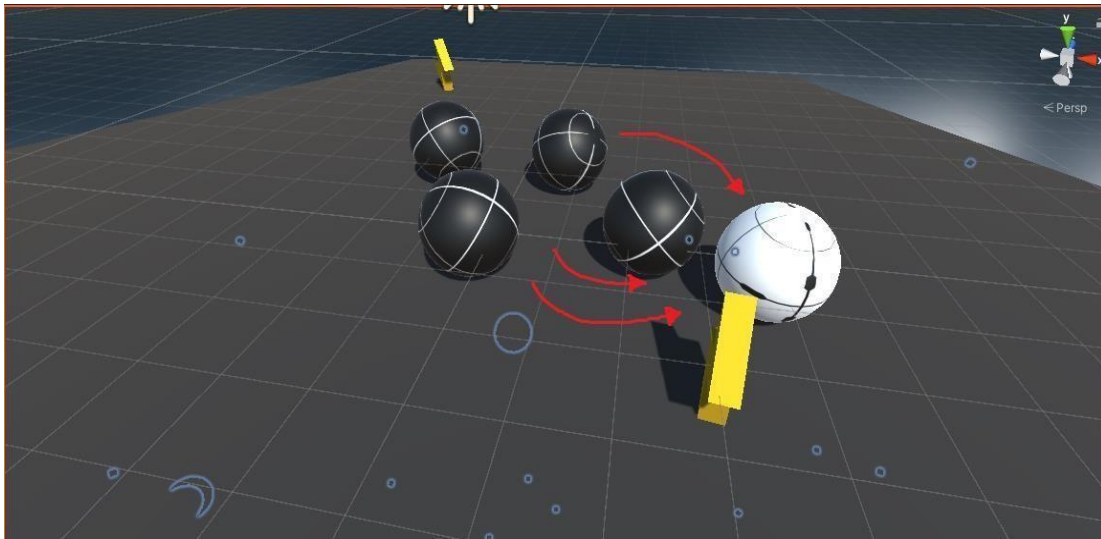
```
1 public class Enemy : MonoBehaviour
2 {
3     public float speed = 0.5f;
4     private Rigidbody enemyRb;
5     private GameObject player;
6     // Start is called before the first frame update
7     void Start()
8     {
9         enemyRb = GetComponent<Rigidbody>();
10        player = GameObject.Find("Player");
11    }
12    // Update is called once per frame
13    void Update()
14    {
15        enemyRb.AddForce((player.transform.position -
16        transform.position).normalized * speed);
17    }
18 }
```

Gambar 9 Implementasi *Addforce* di *Enemy.cs*

Kode pemrograman *Addforce* tersebut disimpan di dalam kelas fungsi *Update()* sehingga mengharuskan proses implementasi dilakukan di setiap *frame* selama gim sedang berjalan. Untuk bisa mendapatkan daya penarikan dari objek *player*, program *Enemy.cs* mengambil komponen *Rigidbody* dari objek tersebut dan diolah melalui fungsi *addforce* melalui persamaan substraksi. Persamaan substraksi ini melakukan proses pengurangan objek *player* dengan objek *enemy* sehingga terjadi daya penarikan fisika. Fungsi normalisasi juga dimasukkan dalam persamaan tersebut untuk menghindari daya tarik eksponensial yang menyebabkan enemy hampir mustahil dikalahkan.

## Hasil Implementasi

Fungsi *Addforce* telah dipastikan memberi pengaruh terhadap tingkah laku *player* dan enemy di dalam gim. Hasil akhir menunjukkan bahwa saat gim sedang berjalan, objek enemy secara langsung mengikuti dan berusaha mengejar *player* 13 untuk dikeluarkan dari arena permainan. Hal ini terjadi akibat wujudnya proses daya tarik yang dihasilkan oleh *Addforce* di antara kedua objek tersebut.



Gambar 10 Hasil Addforce terhadap tingkah laku player dan enemy

Dari hasil tersebut (Gambar 9), dapat dilihat bahwa dengan metode *Addforce* cocok dijadikan sebagai salah satu program NPC sederhana tanpa harus menggunakan algoritma *Artificial Intelligence* lain pada umumnya.

## Kesimpulan

Berdasarkan hasil telah dibahas sebelumnya dapat disimpulkan bahwa pembahasan dalam penelitian ini adalah implementasi *Addforce* dalam pengembangan tingkah laku pemain dan musuh pada gim *action arcade* 3D yang dinamakan sebagai “Sumoball”. Hasil akhir menunjukkan bahwa penggunaan *Addforce* sesuai dijadikan metode untuk pengembangan tingkah laku pemain dan musuh yang sederhana. Penerapan *Addforce* secara konsisten dalam pengembangan gim ini telah menunjukkan dampak positif pada respons dan gerakan pemain dan musuh. Hasil akhir menegaskan bahwa metode ini tidak hanya mempermudah implementasinya, tetapi juga mampu mencapai tujuan pengembangan dengan efisien. Penggunaan *Addforce* sebagai suatu teknik memberikan fleksibilitas yang diperlukan dalam mengontrol tingkah laku entitas di dalam gim, menciptakan pengalaman bermain yang lebih dinamis dan menarik bagi pemain. Selain itu, kesederhanaan metode *Addforce* juga dapat menjadi nilai tambah yang signifikan dalam pengembangan gim, terutama jika dilihat dari sudut pandang efisiensi waktu dan sumber daya. Pengembang gim dapat dengan mudah mengintegrasikan *Addforce* ke dalam proyek mereka tanpa mengorbankan kompleksitas atau kualitas tingkah laku yang dihasilkan. Oleh karena itu, metode ini dapat dianggap sebagai pilihan yang bijak untuk pengembangan gim *action arcade* 3D.

## Referensi

- Filip, J. (2021) *Unity Rigidbody Explained, Medium*. Available at: <https://medium.com/eincode/unity-rigidbody-explained-fb208d0f97f3> (Accessed: October 9, 2021).
- Hendra, S. and Fathiah (2019) "PERANCANGAN GAME ANIMASI ARCADE PERANG JEDI BERBASIS FLASH," *Journal of Informatics and Computer Science*, 5, pp. 84–97.
- İsmail, Ç. (2020) *Colliders and Triggers in Unity 3D, Code In Black*. Available at: <https://www.codinblack.com/colliders-and-triggers-in-unity3d/> (Accessed: October 9, 2021).
- Matthew Robertson (2021) *Unity Rigidbody AddForce, MonkeyKidGC*. Available at: <https://www.monkeykidgc.com/2021/01/unity-rigidbody-addforce.html> (Accessed: September 2, 2021).
- Prastikasari, S., Yunus, A. and Aditya Nugraha, D. (2019) *Implementasi Fuzzy Logic Pada Perilaku NPC (Non-Player Character) dalam Tingkat Kepuasan Pelayanan Pada Game Mljo 2D Berbasis Mobile*. Malang. doi: <https://doi.org/10.21067/jtst.v1i3.3012>
- Sifaulloh, H., Fadila, J.N. and Nugroho, F. (2021) "Penerapan Metode Finite State Machine pada Game Santri on the Road," *Walisono Journal of Information Technology*, 3(1), pp. 11–18. doi:10.21580/wjit.2021.3.1.7135.
- Supriadi, A. and Siregar, I. (2019) "PENGEMBANGAN LOG BOOK PEMBELAJARAN SEPAK BOLA DENGAN PENDEKATAN TAKTIS," *Jurnal Prestasi*, 3(6), pp. 68–72.
- Unity Learn (2018) *Unit 4 - Gameplay Mechanics: Follow the Player, Unity Technologies*. Available at: <https://learn.unity.com/project/unit-4-gameplay-mechanics> (Accessed: September 2, 2021)
- Erick, E. (2023). *Perancangan Game 2D Platformer dengan Gabungan Edukasi Bahasa Pemrograman C Berbasis Desktop Menggunakan Metode Game Development Life Cycle*. Arfiansyah, M. E. (2022). *Sidoarjo Creativepreneur Center Dengan Pendekatan Biophilic Design*.
- Meimaharani, R., & Listyorini, T. (2015). *PURWARUPA GAME EDUKASI PENGENALAN WARNA BERBASIS ANDROID* (Vol. 1, Issue 2).
- Pratama, B., Kriyantono, R., Vol, D. A.-M. (Abdira), & 2022, undefined. (2022). Gim Berbasis Open Source Software sebagai Media Pembelajaran Guru PAUD. *Researchgate.Net*. [https://www.researchgate.net/profile/Prof-Kriyantono/publication/361733348\\_Nomor\\_2\\_Tahun/links/62c25ce9c0556f0d631b782a/Nomor-2-Tahun.pdf](https://www.researchgate.net/profile/Prof-Kriyantono/publication/361733348_Nomor_2_Tahun/links/62c25ce9c0556f0d631b782a/Nomor-2-Tahun.pdf)
- Putri, A. N. (2016). OPTIMASI ALGORITMA BREADTH FIRST SEARCH PADA GAME ENGINE 3D THIRD PERSON SHOOTER MAZE BERBASIS AGEN CERDAS ANDROID. *Jurnal Transformatika*, 14(1), 50–55. <https://doi.org/10.26623/TRANSFORMATIKA.V14I1.349>
- Ramadhani, D., Roedavan, R., & Kurniawan, A. P. (2023). Pengembangan Aplikasi Techpolverse Berbasis Game Engine Unity di Techpolitan Indonesia Persada. *EProceedings of Applied Science*, 10(5). <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/21222/20499>
- Rangan, A. Y., & HENDY. (2023). Penerapan Line of Sight dan Finit State Machine pada Game Platformer "RUN!" *Digital Transformation Technology*, 3(1), 236–247. <https://doi.org/10.47709/DIGITECH.V3I1.2745>