PEMBUATAN *ARTIFICIAL INTELLIGENCE* PADA GAME UCING BELING MENGGUNAKAN ALGORTIMA *RANDOM NUMBER GENERATOR*

ARTIFICIAL INTELLIGENCE MAKING IN UCING BELING GAME USING RANDOM NUMBER GENERATOR ALGORITHM

Mohamad Ari Andrinan[1], Anton Siswo Raharjo Ansori[2], Ashri Dinimaharawati[3]

[1,2,3] Universitas Telkom, Bandung
[1]**ariandrinan@student.telkomuniversity.ac.id**, [2]**raharjo@telkomuniveristy.co.id**,
[3]**ashridini@telkomuniversity.ac.id**

Abstrak

***Artificial intelligence* merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Algoritma *random number generator* adalah suatu algortima yang menghasilkan bilangan-bilangan *random* yang tidak dapat diprediksi. Pada penelitian ini bertujuan untuk membuat kecerdasan buatan pada *game* tradisional ucing beling menggunakan algoritma *random number generator*, dengan membuat nilai *random* yang dihasilkan oleh algoritma *random number generator* lalu diaplikaskan terhadap sumbu (x,y) objek beling yang ada pada *game* ucing beling. Hasil dari penelitian ini adalah algortima *random number generator* sangat cocok diimplementasikan kedalam game ucing beling, karena dalam sepuluh ribu percobaan, tidak ada duplikasi nilai (x.y) yang dihasilkan oleh algoritma *random number generator*.**

Kata kunci : *Random Number Generator*, *Artificial Intelligence*, Ucing Beling, Permainanan Tradisional Sunda.

Abstract

Artificial intelligence is one part of computer science that learns how computers can do work as well as humans do. The random number generator algorithm is an algorithm that generates random numbers that cannot be predicted. In this study aims to create artificial intelligence in traditional games ucing beling using random number generator algorithm, by creating random value generated by random number generator algorithm and then licated against axis (x,y) of beling object in ucing beling game. The result of this study is that the random number generator is very suitable to be implemented into the ucing beling game, because in ten thousand experiments, there is no duplication of values (x.y) generated by the random number generator algorithm.

Keywords: Random Number Generator, Artificial Intelligence, Ucing Beling, Sundanese Traditional Game.

1. Pendahuluan

Permainan Tradisional merupakan budaya negara kita yang diwariskan secara turun menurun, dalam permainan tersebut tidak hanya menguras tenaga fisik tetapi mendidik anak-anak untuk tetap terus bersosialisasi dalam kehidupan sosial [1]. Konsep permainan ucing beling sangat sederhana permainan ini biasanya dimainkan dua sampai lima orang, satu orang akan bertindak menjadi pencari beling yang disembunyikan secara samar oleh pemain lainnya, karena penulis berencana menggunakan *artificial intelligence* maka yang menyembunyikan beling ialah sistem *artificial intelligence* yang ditunjang oleh *algoritma random number generator*. Algoritma *random number generator* adalah program yang berprilaku sebagai variabel acak [2]. Hasil data dari algortima tersebut bertipe angka yang bisa menjadi acuan sumbu (x,y) dari objek beling yang akan disembunyikan.

Dalam penelitian ini penulis menguji apakah algoritma *random number generator* adalah algoritma yang cocok untuk pembuatan *game* ucing beling dan apakah hasil nilai dari algoritma *random number generator* dapat diaplikasikan terhadap sumbu (x,y) objek beling pada *game*.

2. Landasan Teori

2.1 Ucing Beling

Ucing Beling merupakan permainan tradisional dari Jawa Barat, permainan ini biasanya dimainkan oleh dua sampai lima orang anak, dalam permainan ini pemain diharuskan mencari beling yang disamarkan oleh pemain lain. Cara bermain permainan ucing beling sangat mudah pertama para pemain harus membuat batas penyembunyian beling dan mencari alat seperti lidi atau batang pohon yang digunakan untuk mencari beling yang disembunyikan, lalu para pemain diharuskan mencari tempat untuk menyembunyikan beling tersebut, dan mereka akan mencari beling secara bergantian, misal A mencari beling yang disembunyikan B, jika beling sudah ketemu lalu giliran B mencari beling yang disembunyikan C, lalu jika sudah giliran C mencari beling yang disembunyikan A, aturan mainnya seperti itu jika permainan dimainkan oleh tiga orang.

2.2 Algortima *Random Number Generator*

Random Number Generator (RNG) adalah suatu algoritma yang menghasilkan bilangan-bilangan *random* yang tidak dapat diprediksi, yang berarti bahwa angka yang keluar selanjutnya dari pengolahan algoritma tersebut tidak mungkin untuk di prediksi. Dalam kasus ini penulis menggunakan algoritma *random number generator* bawaan dari aplikasi Unity, dengan menggunakan fungsi *RandomRange* terhadap *vector*.

2.3 *Artificial Intelligence*

Artificial Intelligence merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan lebih baik daripada yang dilakukan manusia [3]. Pengaplikasian kecerdasan buatan sangat banyak dan beragam, penggunaan kecerdasan buatan dalam pembuatan *game* adalah salah satu contoh yang paling banyak ditemui dan biasanya di implementasikan kepada *non-playable character* yang membuat berbeda biasanya penggunaan algoritma untuk membangun kecerdasan buatan tersebut [4].

2.4 C#

C# (C-sharp) adalah bahasa pemograman sederhana yang digunakan secara umum, yang berarti C# digunakan untuk berbagai fungsi misalnya pemograman *server-side* pada website, aplikasi desktop, aplikasi *mobile*, pemograman yang beroritentasi objek, pemograman game dan masih banyak lagi. C# sangat bergantung dengan framework yang disebut .NET Framework, framework tersebut yang digunakan untuk meng*compile* atau menjalankan kode bahasa C#.

2.5 Unity *Game Engine*

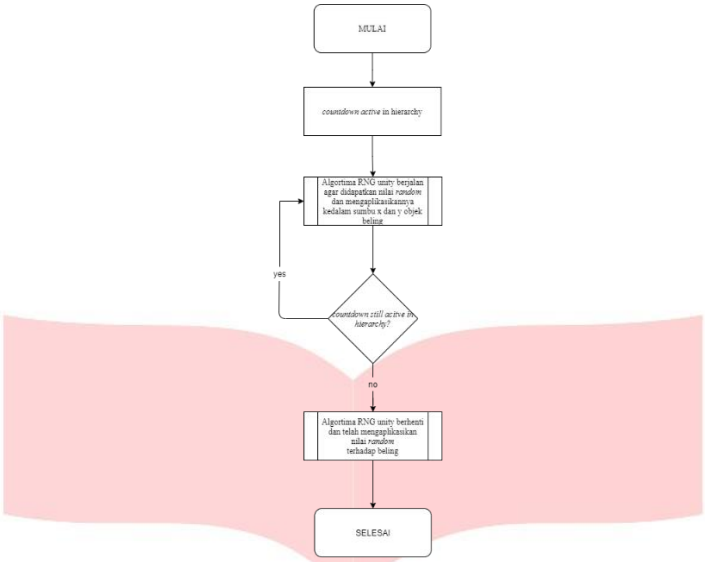
Unity adalah sebuah *tools* yang terintegrasi untuk membuat *game*, arsitektur bangunan dan simulasi, unity tidak dirancang untuk proses desain atau modelling. Unity *game engine* juga dapat membuat suatu *game* berbasis 3D , 2D, dan VR. Unity mampu mengintegrasi C# script, yang akan digunakan dalam pembuatannya. Unity mampu mengaplikasikan UI dan UX kedalam tampilan yang sangat sederhana dan dapat berjalan diberbagai *platform* seperti windows, macOS, linux, tvOS, iOS, Lumin, Android, WebGL, PS4, dan Xbox One.

3. Metodologi Penelitian dan Perancangan

3.1 Desain Sistem

Desain sistem yang dirancang ialah sistem pergerakan atau penempatan *artificial intelligence* pada *game* ucing beling menggunakan algoritma *random number generator* pada perencanaannya penulis akan menggunakan algortima bawaan dari unity, dimana hasil nilai dari pengolahan algoritma *random number generator* tersebut didefiniskan sebagai sumbu x dan y untuk penempatan objek beling pada unity2D [5]. Sehingga tempat munculnya objek beling tiap saat akan berbeda. Cara kerjanya ialah ketika *player* memasuki waktu tunggu untuk bersiap mencari objek beling yang disembunyikan oleh *artificial intelligence* proses dari algoritma *random number generator* akan berkerja untuk mendapatkan nilai acak, sistem unity2D akan mengambil hasil dari angka pengolahan algoritma *random number generator* tesebut sebagai titik sumbu x dan y dari objek beling yang akan disembunyikan, ketika waktu tunggu *player* habis berarti aktifitas dari pemrosesan algoritma random number generator telah selesai, dan *player* pun diharuskan mencari objek beling yang disembunyikan oleh *artificial intelligence*.

3.2 Diagram Blok



Gambar 1. Diagram Blok

Untuk memulai, pertama – tama sistem akan memroses bahwa *countdown active* didalam *hierarchy* setelah itu algoritma *random number generator* akan berjalan sebagai *sub*-proses, algoritma tersebut berfungsi untuk mendapatkan nilai *random* yang akan diaplikasian kedalam sumbu (x,y) objek beling, lalu sistem akan memeriksa kondisi apakah *countdown* masih *active* didalam *hierarchy*, jika masih algoritma *random number generator* akan terus berjalan, setelah itu sistem akan terus memeriksa sampai *countdown* sudah tidak *active* didalam *hierarchy*, setelah *countdown* sudah tidak *active* didalam *hierarchy* maka algoritma *random number generator* akan berhenti dan mengaplikasikan nilai *random* kedalam sumbu (x,y) objek beling.

3.3 Fungsi dan Fitur

1. Membuat sistem *artificial intelligence* yang bisa memindahkan dan menyembunyikan objek beling secara samar menggunakan algortima *random number genarator*, agar permainan ucing beling dapat dimainkan secara digital.

2. Membuat sistem *artificial intelligence* ini dapat diintegrasikan dan aplikasikan kedalam unity2D guna memproses menjadi game uncing beling.

3.4 Implementasi

3.4.1 Implementasi Fungsi Countdown

Pembuatan *countdown* pada *game* ucing beling ini bertujuan untuk membuat waktu tunggu *player* agar bersiap dan juga sebagai kondisi untuk memulai atau berhentinya fungsi algortima *random number generator* yang akan dibuat. Untuk mengimplementasikan fungsi *countdown* penulis membuat satu buah *GameObject*, satu buah *canvas*, dan satu buah *text*. *GameObject* yang diberi nama ”GameController” bertujuan untuk mengontrol *countdown*, *canvas* yang diberi nama ”countdownCanvas” bertujuan untuk menaungi *text* yang diberi nama ”countdownText”.

3.4.2 Implementasi Algortima Random Number Generator

Pada pembuatan algortima *random number generator* yang cocok terhadap vektor, penulis menggunakan algoritma bawaan unity yang memanfaatkan fungsi *RandomRange* yang akan menghasilkan nilai *random* yang akan diimplementasikan terhadap sumbu x dan y, langkah awal untuk pembuatannya ialah membuat satu *GameObject* yang diberi nama “KACA1”, “KACA2”, “KACA3”, “KACA4”, “KACA5”, berikut adalah potongan kode yang disisipkan kedalam *GameObject* yang memuat fungsi *random number generator*.

```
public void SpawnNextCircle()
    {
            Vector3 pos = center + new Vector3(Random.Range(-size.x / 2, size.x / 2),
```

```
                                                Random.Range(-size.y / 2, size.y / 2),
                                                0);
        Instantiate(circlePrefab, pos, Quaternion.identity);
        Debug.Log(pos);
        Destroy(gameObject);
    }
```

Pada pembuatan fungsi dimana bertujuan untuk mulai dan berhenti algoritma *random number generator* yang telah dibuat, penulis membuat kode yang disisipkan juga pada *GameObject* yang diberi nama ”KACA”, berikut potongan kode yang berfungsi untuk memulai atau memberhentikan proses fungsi random number generator yang telah dibuat.

```
    public void selesecountdown()
{
    if (countdownCanvas.activeInHierarchy == true)
    {
        SpawnNextCircle();
    }
    if (countdownCanvas.activeInHierarchy == false)
    {
        Destroy(countdownCanvas.gameObject);
    }
}
```

Pada potongan kode diatas bermaksud untuk mengecek apakah *GameObject* ”Popups” yang akan ditargetkan terhadap *canvas* yang telah dibuat difungsi *countdown* berada pada *hierarchy* atau tidak, jika ada didalam *hierarchy* maka fungsi algoritma random number generator akan terus berjalan sampai *canvas countdown* tidak ada di*hierarchy*.

3.4.3 Implementasi Fungsi Penghitung Beling

Dalam implementasinya fungsi ini mencakup pembuatan fungsi penghitung beling, agar dalam pembuatan game Ucing Beling penulis dapat membuat hal yang harus dilakukan ketika kelima beling berhasil ditemukan. Langkah awalnya membuat suatu fungsi yang bisa mengenal masing masing kaca dengan memberi suatu nama untuk masing-masing *tag* yang telah diberikan penulis, berikut potongan kode yang telah dibuat.

```
public void checknametag()
    {
        e1 = GameObject.FindWithTag("Enemy Circle");
        e2 = GameObject.FindWithTag("Enemy Circle2");
        e3 = GameObject.FindWithTag("Enemy Circle3");
        e4 = GameObject.FindWithTag("Enemy Circle4");
        e5 = GameObject.FindWithTag("Enemy Circle5");
}
```

Setelah itu penulis membuat suatu fungsi untuk memenuhi tujuan dari implementasi fungsi penghitung beling, berikut potongan kode yang telah dibuat.

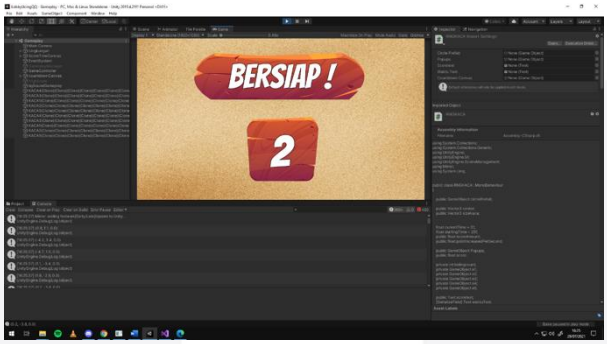
```
public void sentuhenemy()
    {
        Vector3 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
        Vector2 mousePos2D = new Vector2(mousePos.x, mousePos.y);
        RaycastHit2D hit = Physics2D.Raycast(mousePos2D, Vector2.zero);
            //if the hit target the tag of ""
```

```
        if (hit.collider.tag == "Enemy Circle") //repeat for enemy circle2,3,4,5
        {
          if (Input.GetMouseButtonDown(0))
          {
            e1.transform.position = new Vector3(15, 15, 0);
            belingcount++;
            Debug.Log(belingcount);
          }
          //...buat fungsi if seperti diatas untuk fungsi enemey circle2,3,4,5
        }
```

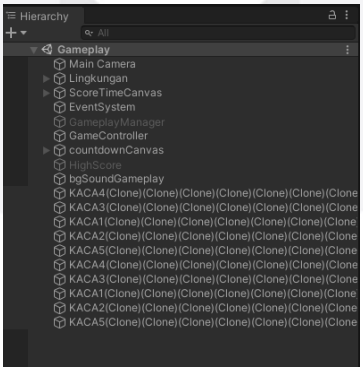
3.5 Pengujian

3.5.1 Pengujian Parameter Alur Algortima

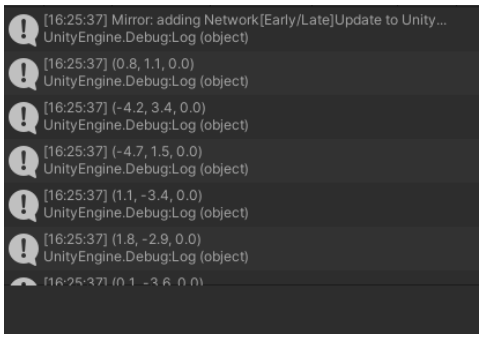
Pengujian parameter alur algortima *random number generator* bertujuan untuk menguji apakah algoritma yang ditulis kedalam *editor* berfungsi sebagaimana yang ditulis dalam desain sistem. Dalam penulisnya alur yang diharapkan yaitu pertama ketika *countdown* canvas *active* di*hierarchy* maka algoritma *random number generator* akan berjalan, ketika *countdown canvas* sudah tidak *active* di*hierarchy* maka algoritma *random number generator* akan berhenti dan mengaplikasikan nilai yang dihasilkan kedalam lima objek beling yang sudah dibuat.



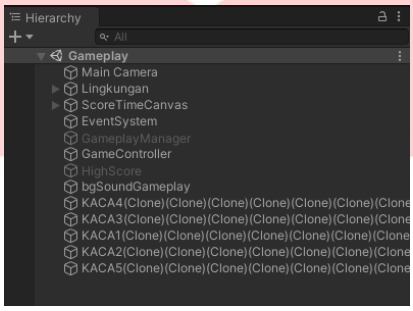
Gambar 2. Game memasuki waktu tunggu.



Gambar 3. *Countdown canvas active in hierarchy*.

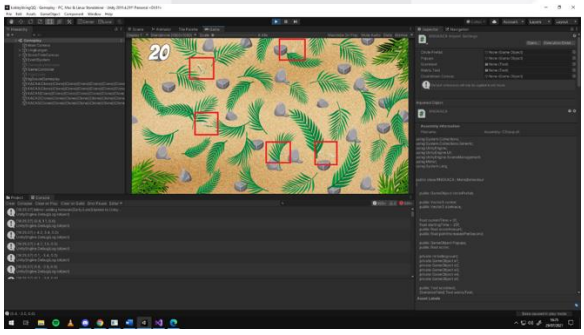


Gambar 4. Algortima pun berjalan terus menerus.



Gambar 5. *Countdown canvas* sudah tidak ada di*hierarchy*.

Gambar 6. Hasil nilai algortima *random number generator*.

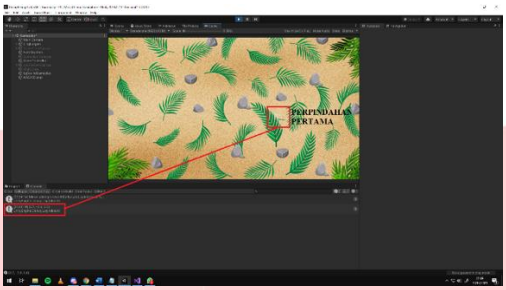


Gambar 7. Hasil dari algortima dapat diaplikaskan kedalam lima objek.

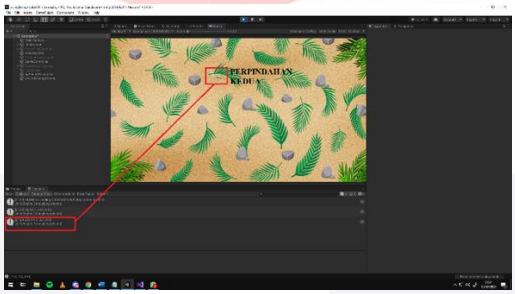
Dapat disimpulkan bahwa alur algortima *random number generator* yang diimplementasikan kedalam *game* Ucing Beling dapat berjalan sesuai yang direncanakan dalam desain sistem.

3.5.2 Pengujian Parameter Perpindahan Objek

Pengujian parameter perpindahan objek bertujuan untuk menguji apakah nilai yang dihasilkan dapat diaplikasikan kedalam sumbu (x,y) objek beling pada *game* dan objek yang berpindah tidak melewati batas yang ditentukan. Untuk menguji apakah nilai yang dihasilkan dapat diaplikasikan kedalam sumbu x dan y objek beling pada *game*, penulis melakukan uji coba dengan cara mencatat output atau nilai keluaran yang dihasilkan oleh fungsi tersebut menggunakan fungsi *DebugLog*, *DebugLog* berfungsi untuk mencatat hasil nilai keluaran kedalam *console* yang tersedia pada Unity.



Gambar 8. Perpindahan objek pertama yang sudah terimplementasikan fungsi *random number generator*.

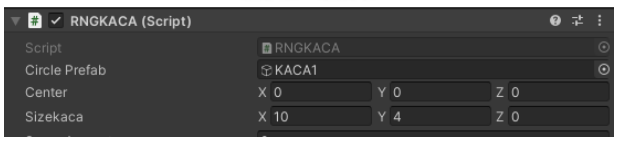


Gambar 9. Perpindahan objek kedua yang sudah terimplementasikan *fungsi random number generator*.

Posisi awal *GameObject* "KACA" adalah (0,0) yaitu berada ditengah tengah-tengah *frame game*, pada perpindahan pertama ketika melakukan algoritma *random number generator* didapat hasil (-1,2.4) dapat dilihat pada Gambar 8, pada perpindahan kedua ketika melakukan algoritma *random number generator* didapat hasil (3.7,-0,6) dapat dilihat pada Gambar 9.Dapat ditarik kesimpulan bahwa nilai random yang dihasilkan algoritma *random number generator* dapat diaplikasikan terhadap objek beling, pada *game* Ucing Beling.

Untuk menguji apakah objek beling pada *game* Ucing Beling berpindah tidak melalui batas yang ditentukan, batas yang penulis maksud disini ialah batas *frame game* Ucing Beling, agar ketika *game* ini dimainkan tidak terjadi *bug* dimana beling diposisikan keluar dari pandangan *player* atau keluar dari *frame game* tersebut.

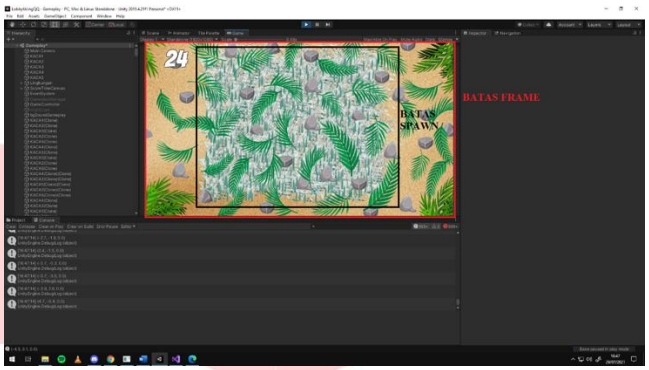
Cara yang akan digunakan penulis ialah dengan cara memanggil fungsi algortima *random number generator* secara terus menerus dan tidak akan berhenti, sebelum memanggil fungsi algoritma *random number generator* secara terus menerus penulis membuat batasan *spawn GameObject* "KACA".



Gambar 10. Batas spawn GameObject "KACA".

Garis x dan garis y yang penulis maksud adalah dari titik tengah frame yaitu (0,0) ditarik garis kearah kanan sebanyak 10 menjadi (10,0), ditarik garis kearah kiri sebanyak 10 menjadi (-10,0), ditarik garis kearah atas sebanyak 5 menjadi (0,4), ditarik garis kearah bawah sebanyak 4 menjadi (0,-4), dan dapat diketahui batas yang penulis atur ialah garis x sepanjang 10 hingga -10 dan garis y sepanjang 4 hingga -4. Ketika batas frame sudah dibuat penulis menulis kembali kedalam kode yang sudah disisipkan digame objek untuk membuat batas spawn, batas spawn dibuat agar spawn kaca tidak mengganggu UI elemen yang lain. Potongan kode berikut bermaksud untuk membuat batas *spawn GameObject* "KACA" tidak melewati dari 5 hingga -5 nilai x dan 4 hinggu -4 nilai y.

```
Vector3 pos = center + new Vector3(Random.Range(-sizekaca.x/2 , sizekaca.x/2 ), Random.Range(-sizekaca.y , sizekaca.y), 0);
```



Gambar 11. Perpindahan objek beling dengan memanggil fungsi algoritma random number generator secara terus menerus.

Dapat ditarik kesimpulan bahwa dengan memanggil fungsi algoritma *random number generator* secara terus menerus sama saja dengan melakukan perpindahan objek secara terus menerus, dan tidak keluar dari batas yang penulis atur sebelumnya.

4. Kesimpulan

Kesimpulan yang didapat dari hasil pengujian dan analisis yang telah dilakukan adalah sebagai berikut:

1. Alur algortima *random number generator* yang diimplementasikan kedalam *game* Ucing Beling dapat berjalan sesuai yang direncanakan dalam desain sistem.
2. Pada posisi awal objek beling yaitu (0,0), dengan proses algoritma *random number generator* yang mengaplikasikan nilai random menjadi (-1,2.4) pada perpindahan pertama dan menjadi (3.7,-0.6) pada perpindahan kedua terbukti bahwa hasil algoritma *random number generator* dapat diaplikasikan kedalam sumbu (x,y).
3. Dengan mengatur batas *frame* sumbu x menjadi 10 hingga -10 dan batas sumbu y menjadi 4 hingga -4, batas *spawn* sumbu x menjadi 5 hingga -5 dan batas sumbu y menjadi 4 hingga -4 lalu memanggil fungsi algoritma *random number generator* secara terus menerus dapat diketahui bahwa perpindahan objek tidak melewati batas yang ditentukan.

Referensi :

[1] William Tedi, “Perubahan Jenis Permainan Tradisional Menjadi Permainan Modern pada anak-anak di Desa Ijuk Kecamatan Belitang Hulu Kabupaten Sekadu,” Jurnal S-1 Sosiologi, vol. 3, no. 4, Desember. 2015.

[2] M. Abutaha, Safwan. El Assad, O. Jallouli, A. Queudet, O. Deforges, “Design of a pseudo-chaotic number generator as random number generator,” in International Conference on Communications, June 2016.

[3] M. Dahria,”Kecerdasan Buatan (Artificial Intelligence)”, Jurnal SAINTIKOM, Agustus 2008.

[4] Abdennour El R, K.W.Wong, M.Price,”ARTIFICIAL INTELLIGENCE FOR COMPUTER GAMES”, International Journal of Computer Games Technology, 2009.

[5] A. S. Milak, E. W. Hidayat, A. P. Aldya,”PENERAPAN ARTIFICIAL INTELLIGENCE PADA NON-PLAYER CHARACTER MENGGUNAKAN ALGORITMA COLLISION AVOIDANCE SYSTEM DAN RANDOM NUMBER GENERATOR PADA GAME 2D BALAP EGRANG”, Jurnal Teknologi Informasi dan Ilmu Komputer, 2020.