KURAWAL Jurnal Teknologi, Informasi dan Industri *Volume 4 Nomor 2 - Oktober 2021*
***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

PENERAPAN METODE ALGORITMA *SHUFFLE RANDOM* PADA *GAME 2D* PERTUALANGAN PEMUDA DESA

Fransiskus Heri [1)]
Amak Yunus [2)] **, Alexius Endy Budianto** [3)]

Teknik Informatika Universitas PGRI Kanjuruhan Malang, Jl. S. Supriadi No 48Bandungrejosari Sukun Malang Jawa Timur
email : fransiskusheri1997@gmail.com[1)], amakyunus@unikama.ac.id[2)], endybudianto@yahoo.com[3)]

Abstrak

Game 2D Pertualangan Pemuda Desa adalah karakter seorang pemuda desa yang sangat prosesif untuk menjaga kawasannya dari para pendatang yang akan merusak atau mengganggu kawasan tempat tinggalnya. Game ini memiliki genre action dimana game ini dapat berpindah ke kiri, kanan, menyerang dan melompat. Untuk dapat menjadikan sebuah game yang menarik maka sangat diperlukan musuh yang mempunyai sebuah kemampuan kecerdasan buatan untuk menampilkan game agar lebih terlihat alami dan efektif. Pada game ini, musuh memiliki kemampuan masing-masing yaitu terdiri dari menyerang, mengejar, serta memiliki kemampuan untuk muncul dengan posisi yang berubah atau teracak. Untuk dapat mengatur serta mengontrol pengacakan, penulis menggunakan algoritma shuffle random yang berfungsi untuk mengatur pengacakan urutan posisi kemunculan pada musuh dalam game yang akan dibuat oleh peneliti.

Kata Kunci:

Shuffle Random, Game 2D, Pertualangan, Pemuda Desa.

Abstract

Village Youth Adventure 2D Game is the character of a village youth who is very processive to protect his area from immigrants who will damage or disturb the area where he lives. This game has an action genre where this game can move left, right, attack and jump. To be able to make an interesting game, it is very necessary for an enemy who has an artificial intelligence ability to display the game to make it look more natural and effective. Inthis game, enemies have their own abilities, which consist of attacking, chasing, and having the ability to appear in a changed or random position. To be able to regulate and control this behavior, the author uses a random shuffle algorithm which functions to regulate the randomization of the order of appearance of the enemy in the game that will be made by the researcher.

***Keywords*:**

Shuffle Random, Game 2D, Adventure, Village Youth.

1. PENDAHULUAN

Game merupakan salah satu media hiburan yang dapat digunakan oleh setiap kalanganuntuk menghilangkan rasa jenuh. *Game* juga memiliki manfaat lain seperti membantu pengembangan otak, melatih memecahkan masalah, meningkatkan konsentrasi serta jugamelatih kecepatan. Ada beberapa genre yang terdapat dalam sebuah game seperti *action,strategy, role playing, sport, vehicle simulations, construction and management simulations, adventure, artifical life and puzzle games* (Adam, 2010:390-590).

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

Game 2D Pertualangan Pemuda Desa merupakan karakter seorang pemuda desa yang sangat prosesif untuk menjaga kawasannya dari para pendatang yang akan merusak atau mengganggu kawasan tempat tinggalnya. *Game* ini memiliki genre *action* dimana *game* ini dapat berpindah ke kiri, kanan, menyerang dan melompat.

Selain itu di dalam sebuah *game* juga terdapat sebuah *obstacle* atau bisa diartikan secara harfiah adalah rintangan yang harus dilewati oleh *player* untuk dapat menyelesaikan *level* pada sebuah *game*. *Obstacle* merupakan unsur penting yang harusada pada sebuah *game* karena selain memberikan manfaat, obstacle juga dapatmempengaruhi minat pemain saat bermain *game*. Bentuk *obstacle* pada sebuah *game* juga bermacam-macam, seperti jebakan yang terdapat dalam *game*, waktu yang ditentukan untuk menyelesaikan *game*, teka-teki yang ada di dalam *game*, musuh yang kuat, dan lain sebagainya.

Salah satu cara untuk membuat sebuah *obstacle* pada *game* adalah dengan menerapkan algoritma tertentu. Algoritma *shuffle random* merupakan salah satu algoritma yang dapat diterapkan pada *game* yang akan dibuat. Menurut Andrea dalam jurnal Senaik (2015), *shuffle random* merupakan metode pengacakan urutan *indeks* dari sebuah *array*. Pengacakan ini di analogikan seperti pengocokan terhadap *dek kartu*, dimana semua kartu dikocok sehingga susunannya teracak. Menurut Trabani (2019)menyatakan bahwa algoritma *shuffle random* berfungsi dengan baik untuk digunakan sebagai pengacakan posisi dari *indeks array*. Menurut *Jurnal Implementasikan AlgoritmaShuffle Random pada Pembelajaran Panca Indra Berbasis Android* (2019) menyatakan Metode *shuffle random* lebih efektif karena dapat berdiri sendiri tanpa harus menambahkan metode lain sperti metode *fisher-yates random* karena dalam *bahasa pemrograman* metode *shuffle random* berfungsi untuk mengacak angka dan juga dapat mengacak array string, Sedangkan Menurut *Jurnal Penerapan Algorima Fisher-Yates Random untuk Mengacak Soal Penerimaan Forum Studi Mahasiswa Informatika Universal Nasional* (2020) menyatakan Algoritma ini mungkin memiliki banyak keuntungan tetapi masih memiliki kekurangan yang ditunjukkan untuk algoritma iniadalah algoritma ini tidak dapat berdiri sendiri dan harus membutuhkan algoritma lain guna untuk menyempurnakan metodenya. Dalam *bahasa pemrograman* metode *fisher- yates random* hanya berfungsi untuk mengacak angka.

Program yang bisa mengendalikan serta jadi kecerdasan buatan musuh disebut dengan *Artificial intelligence*. *Artificial Intelligence* merupakan suatu faktor yang dibutuhkan dalam pengembangan permainan ialah membuat *game* lebih dinamis serta alami Seri Suriani (2015). Pada mayoritas permainan *Artificial Intelligent* kerap digunakan serta mempunyai ketergantungan interaksi terhadap pemain, sehingga *Artificial Intelligent* mempunyai kedudukan yang berarti untuk menambah ketertarikan pengguna dalam bermain Mcgee Kevin & Abraham (2010). Dengan menggunakanmusuh yang mengimplementasikan kecerdasan buatan maka sebuah *game* terlihat lebih seru karena permainan ini tidak lagi monoton melainkan terlihat lebih menantang saat dimainkan.

Berdasarkan paparan diatas, dapat disimpulkan bahwa penggunaan AI di dalam *Game* dibutuhkan untuk meningkatkan tantangan dalam *Game* dan juga bahwa obstacle merupakan *elemen* penting yang dibutuhkan dalam sebuah *game*, begitu juga pada algoritma *shuffle random* banyak diterapkan pada *game* yang berjenis monoton sehingga pemain merasa bosan dan tidak puas. Oleh karena itu, penulis mendapatkan sebuah ide untuk menerapkan algoritma *Shuffle Random* pada musuh sebagai *obstacle* dan *AI*, yakniagar musuh yang ada dalam *game* muncul dalam posisi teracak seperti yang disusun dalam skripsi berjudul "**Penerapan metode algoritma *Shuffle Random* untukmenentukan posisi kemunculan musuh secara acak pada *game 2d* pertualangan pemuda desa**"**.**

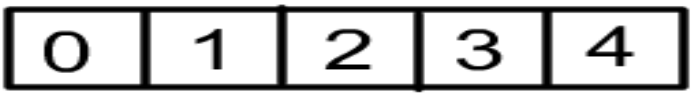
***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

Dengan ini nantinya *game* akan terasa unik dan menantang saat dimainkan, terutama saat pemain akan berhadapan dengan musuh, karena tingkat kesulitan utama terletak pada pengacakan posisi kemunculan yang akan digunakan musuh.

2. METODE / ALGORITMA

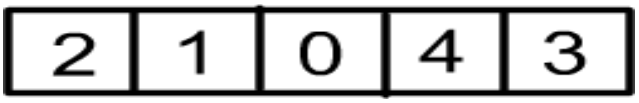
Shuffle random adalah pengacakan urutan *indeks* dari sebuah *record* atau *array*. Pengacakan ini diibaratkan pengocokan pada dek kartu, semua kartu dikocok sehingga susunannya teracak. Contoh lain misalkan A adalah *array* 5 x 1, A = [1 2 3 4 5] maka proses*shuffle random* akan mengacak susunan indek dari *array* A menjadi A1 = [5 1 3 2 4] ataupunmenjadi susunan *array* yang lain. Dalam bahasa pemrograman fungsi *shuffle random* tidak hanya dapat mengacak angka, tetapi juga dapat mengacak *array string* ataupun campuran *string* dan angka.

1) Deklarasi indeks pada kode program
   Pada mendeklarasikan sebuah array menggunakan angka 0 sampai 4. Contoh tahap deklarasi jika dituliskan kedalam bentuk kode program dapat dilihat sebagai berikut:
   String[] Arr =(0,1,2,3,4);
   Di dalam urutan array dari variabel diatas, urutan pertama atau urutan ke 0 jikamenurut perhitungan urutan array adalah 0, sedangkan indeks terakhir atau urutan ke 4. Contoh dari indeks array yang sudah dibuat dapat dilihat pada gambar 1.



Gambar 2.1 Indeks array yang belim diacak

2) Pengacakan pada *indeks array* yang sudah di deklarasikan
   Penerapan dari shuffle random dilakukan dengan cara memanggil fungsi dari shuffle random yang sudah dibuat.
   Arr ← *ShuffleRandom*(Arr)
   Fungsi diatas adalah pengacakan yang dipanggil untuk mengacak isi atau indeks dari Arr yang sebelumnya adalah Arr = (0,1,2,3,4) berubah menjadi Arr = (2,1,0,4,3) atau bentuk susunan array yang lain. Gambar 2 merupakan hasil dari *shufflerandom* yang dilakukan pada *indeks array* Arr yang telah di deklarasikan sebelumnya.



Gambar 2.2 Indeks Array yang sudah di acak

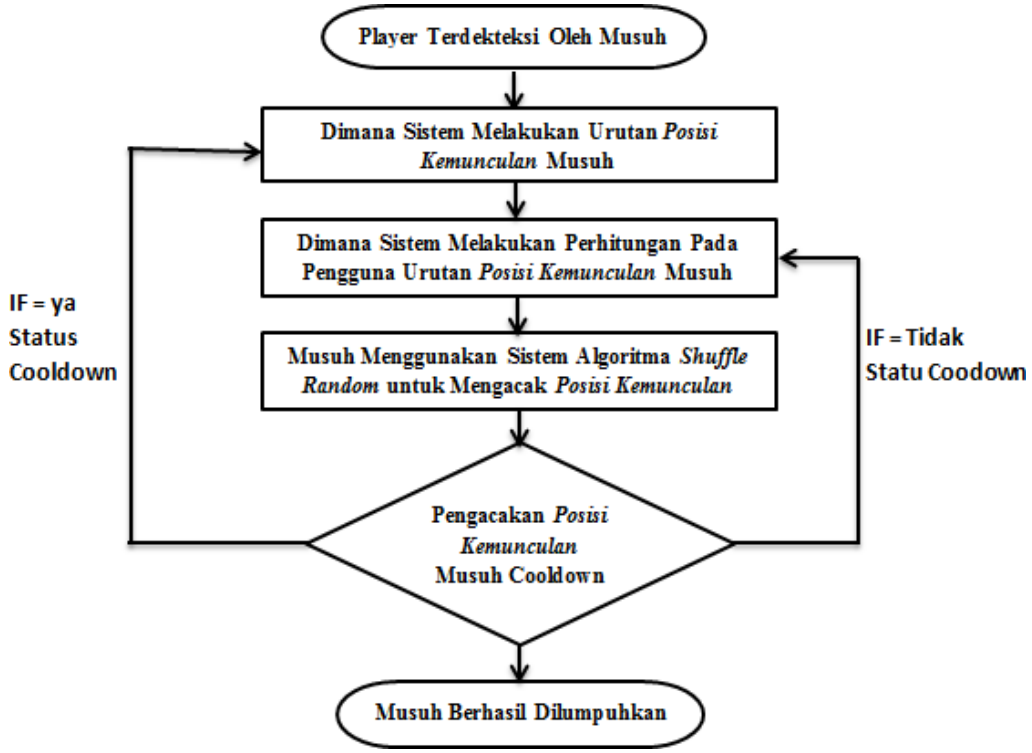
Berikut ini akan dijelaskan langkah-langkah dari penerapan *shuffle random* terhadap kemunculan yang akan digunakan musuh:

1. Ketikan *player* terdeteksi oleh musuh, maka musuh akan muncul untuk memasukkan area bertarung.
2. Sistem melakukan urutan pengacakan kemunculan yang akan digunakan musuh.
3. Sistem akan melakukan perhitungan mundur jeda kemunculan pada musuh.
4. Musuh dalam *game* ini menggunakan sistem pengacakan kemunculan.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

5. Ketika musuh baru saja melakukan proses pengacakan kemunculan akan mendapatkan status *cooldown*.

Cara kerja *shuffle random* terhadap proses pengacakan kemunculan yang akan digunakan oleh musuh dapat dilihat gamber 2.3



Gambar 2.3 cara kerja Algoritma Shuffle Random

Selanjutnya adalah membuat dan menerapkan metotode *Shuffle Random* sebagai metode pengacakan terhadap urutan posisi kemunculan pada musuh ke dalam *script* kode program "Sistem Pengacakan" pada game yang telah dibuat. Struktur dari fungsi metode *Shuffle Random* yang telah diterapkan pada program pengacakan posisi kemunculan terdapat pada "Sistem Pengacakan" sehingga dapat di lihat pada gambar 2.4

```
public void AcakUrutanSpawn()
{
    for (int PosisiArray = 0; PosisiArray < UrutanSpawnMusuh.Length; PosisiArray++)
    {
        int PosisiAcakan = UrutanSpawnMusuh[PosisiArray];
        int PengacakanArray = Random.Range(PosisiArray, UrutanSpawnMusuh.Length);
        UrutanSpawnMusuh[PosisiArray] = UrutanSpawnMusuh[PengacakanArray];
        UrutanSpawnMusuh[PengacakanArray] = PosisiAcakan;
    }
}
}
```

Gambar 2.4 Program Sistem Pengacakan

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

3. HASIL DAN PEMBAHASAN

Pada bab ini hasil dan pembahasan penelitian serta gambaran Metode Algoritma *Shuffle Random* yang sudah dibuat. Berikut pembahasan nya:

3.1. Spesifikasi produk

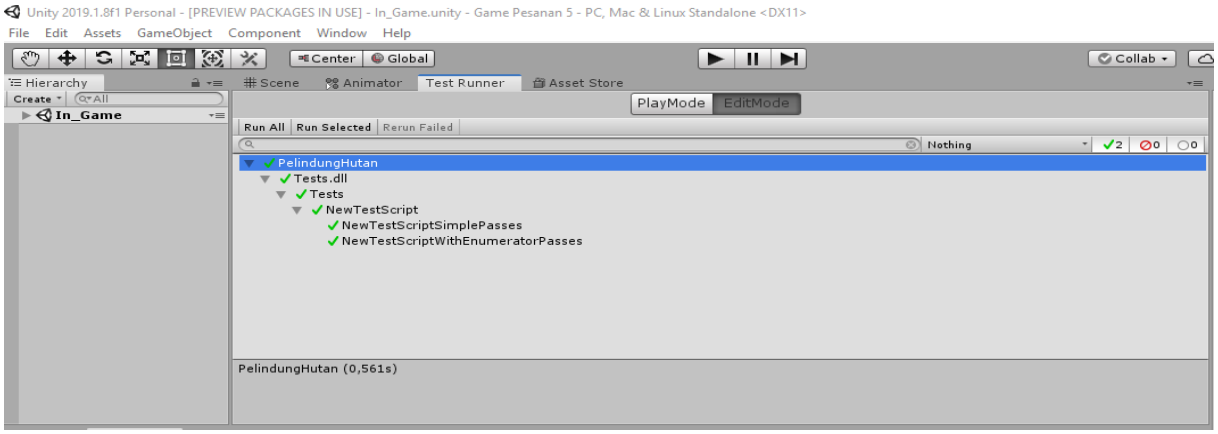
Berikut ini merupakan *spesifikasi* dari produk yang telah dibuat dilakukan uji cobaterhadap produk:

1. Produk yang dibuat adalah sebuah game action 2D yang sudah dikemas dalam betukAPK.
2. Produk dapat digunakan pada device Android versi 5 diatasnya.
3. Produk ini dapat dimainkan tanpa menggunakan koneksi internet.
4. Produk dibuat dengan mengambil salah satu pemuda desa di kalimantan barat sebagai protagonis dari jalannya cerita game.
5. Produk memiliki size data 70MB saat masih dalam kemasan APK, dan memiliki size data 95MB setelah diinstal.

3.2. Unit Testing

Pengujian ini digunakan untuk menguji kesesuaian dari komponen yang ada pada setiap *unit software*, pada penelitian ini, penulis akan mengecek setiap komponen yang ada pada *game* untuk kemudian dapat diambil sebagai sebuah kesimpulan apakah komponen tersebut berfungsi sebagai mestinya.

Pengujian *Unit Testing* ini langsung menggunakan *Test Runner* yang ada pada *software Unity*.



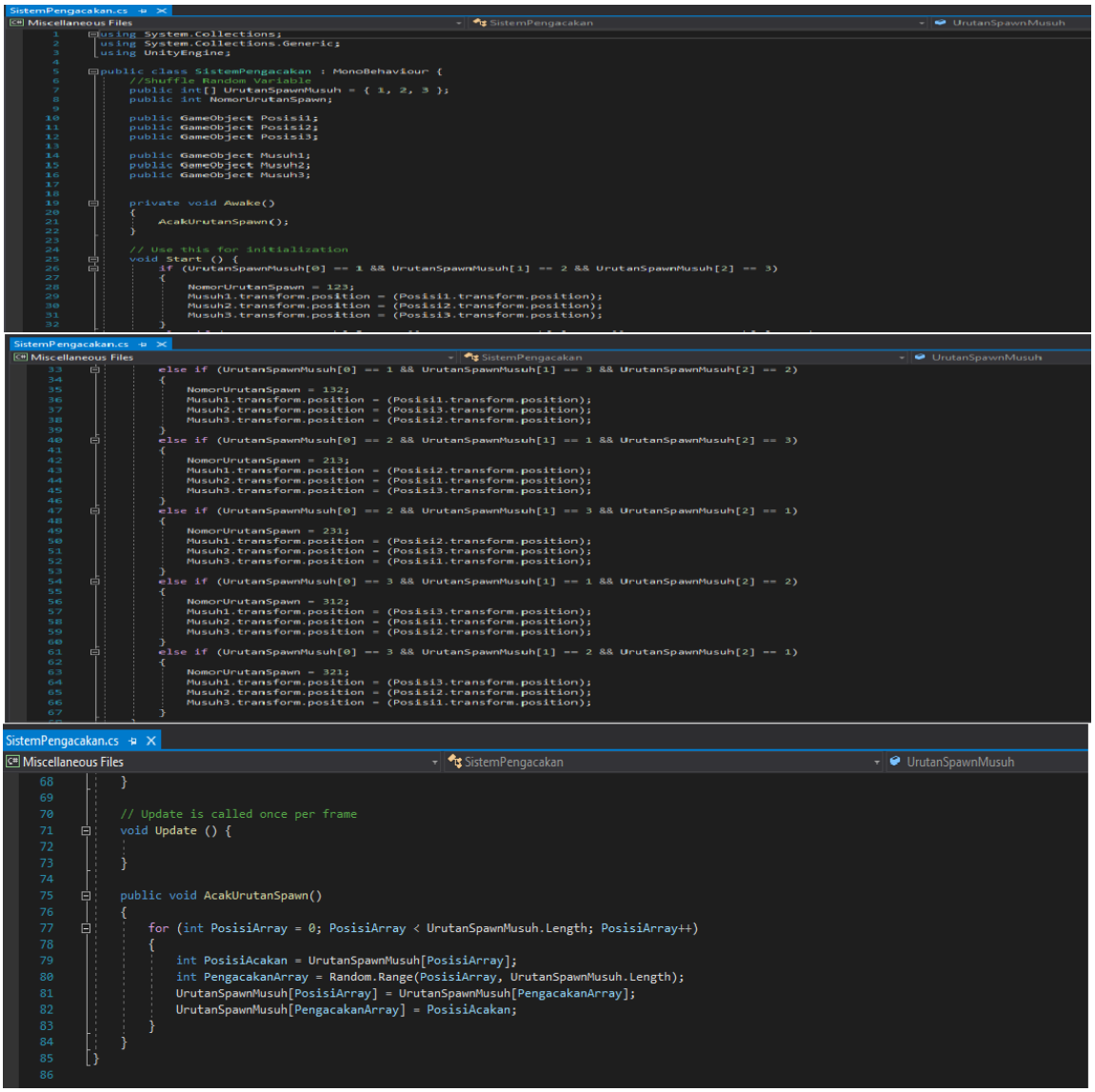
Gambar 3.1 Hasil Pengujian *Unit Testing*

Unity Test Runner adalah alat yang menguji kode dari target, pada pengujian kali ini target pengujian adalah target program yang terdapat dalam *game*.

1. Uji Shuffle Random

Pengujian ini digunakan untuk menguji kesesuaian tingkat kesulitan pada fungsi algoritma *shuffle random* yang diterapkan pada musuh untuk mengetahuai apakah metode ini sudah berfungsi sebagai mestinya. Pengujian ini langsung dilakukan pada unity untuk memastikan apakah algoritma *shuffle random* berjalan seperti mestinya.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SistemPengacakan : MonoBehaviour {
    //Shuffle Random Variable
    public int[] UrutanSpawnMusuh = { 1, 2, 3 };
    public int NomorUrutanSpawn;

    public GameObject Posisi1;
    public GameObject Posisi2;
    public GameObject Posisi3;

    public GameObject Musuh1;
    public GameObject Musuh2;
    public GameObject Musuh3;


    private void Awake()
    {
        AcakUrutanSpawn();
    }

    // Use this for initialization
    void Start () {
        if (UrutanSpawnMusuh[0] == 1 && UrutanSpawnMusuh[1] == 2 && UrutanSpawnMusuh[2] == 3)
        {
            NomorUrutanSpawn = 123;
            Musuh1.transform.position = (Posisi1.transform.position);
            Musuh2.transform.position = (Posisi2.transform.position);
            Musuh3.transform.position = (Posisi3.transform.position);
        }
        else if (UrutanSpawnMusuh[0] == 1 && UrutanSpawnMusuh[1] == 3 && UrutanSpawnMusuh[2] == 2)
        {
            NomorUrutanSpawn = 132;
            Musuh1.transform.position = (Posisi1.transform.position);
            Musuh2.transform.position = (Posisi3.transform.position);
            Musuh3.transform.position = (Posisi2.transform.position);
        }
        else if (UrutanSpawnMusuh[0] == 2 && UrutanSpawnMusuh[1] == 1 && UrutanSpawnMusuh[2] == 3)
        {
            NomorUrutanSpawn = 213;
            Musuh1.transform.position = (Posisi2.transform.position);
            Musuh2.transform.position = (Posisi1.transform.position);
            Musuh3.transform.position = (Posisi3.transform.position);
        }
        else if (UrutanSpawnMusuh[0] == 2 && UrutanSpawnMusuh[1] == 3 && UrutanSpawnMusuh[2] == 1)
        {
            NomorUrutanSpawn = 231;
            Musuh1.transform.position = (Posisi2.transform.position);
            Musuh2.transform.position = (Posisi3.transform.position);
            Musuh3.transform.position = (Posisi1.transform.position);
        }
        else if (UrutanSpawnMusuh[0] == 3 && UrutanSpawnMusuh[1] == 1 && UrutanSpawnMusuh[2] == 2)
        {
            NomorUrutanSpawn = 312;
            Musuh1.transform.position = (Posisi3.transform.position);
            Musuh2.transform.position = (Posisi1.transform.position);
            Musuh3.transform.position = (Posisi2.transform.position);
        }
        else if (UrutanSpawnMusuh[0] == 3 && UrutanSpawnMusuh[1] == 2 && UrutanSpawnMusuh[2] == 1)
        {
            NomorUrutanSpawn = 321;
            Musuh1.transform.position = (Posisi3.transform.position);
            Musuh2.transform.position = (Posisi2.transform.position);
            Musuh3.transform.position = (Posisi1.transform.position);
        }
    }

    // Update is called once per frame
    void Update () {

    }

    public void AcakUrutanSpawn()
    {
        for (int PosisiArray = 0; PosisiArray < UrutanSpawnMusuh.Length; PosisiArray++)
        {
            int PosisiAcakan = UrutanSpawnMusuh[PosisiArray];
            int PengacakanArray = Random.Range(PosisiArray, UrutanSpawnMusuh.Length);
            UrutanSpawnMusuh[PosisiArray] = UrutanSpawnMusuh[PengacakanArray];
            UrutanSpawnMusuh[PengacakanArray] = PosisiAcakan;
        }
    }
}
```

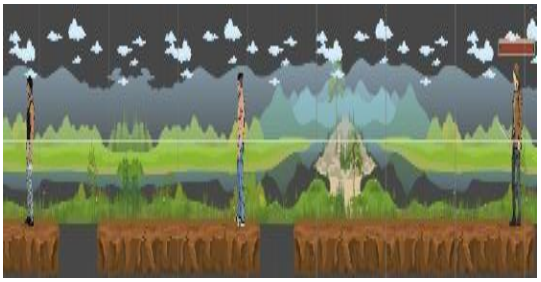
Gambar 3.2 Script Metode Shuffle Random Pada Musuh

2. Uji Skenario

Uji skenario merupakan pengujian perbandingan antara *software* yang sudah diterapkan Algoritma *Shuffle Random* dan software sebelum diterapkan Algoritma *Shuffle Random.* Dari hasil pengujian ini peneliti dapat menyimpulkan perbedaan kedua *software* yang telah diterapkan algoritma tersebut.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

Tabel 1. Sebelum diterapkan Metode *Shuffle Random*

Skenario Pengujian	Hasil yang diharapakan	Hasil Pengujian	Kesimpulan
Saat Permainan akan pertama kali untuk dijalankan	Musuh akan melakukan proses pengacakan *posisi kemunculan*		Tidak Valid
Player berada pada jarak <= 10 dariMusuh	Musuh akan melakukan aksiuntuk menyerang.		Valid
Game terdapat Bossmusuh dan memiliki anak Buah 2	Yang diacak *posisi kemunculan* untuk meningkatkan tingkat kesulitan yangdilewatkan player dengan memiliki kekuatan skill masing-masing.		Tidak Valid
Bos musuh memiliki senjata berupa pedang untukmelawan player	Agar bos musuh memiliki kekuatan yang sulit untuk dikalahkan olehplayer.		Tidak Valid
Player beradapada jarak >3dari musuh	Musuh akan mengeluarkan skill untuk menyerang player		Tidak Valid

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

Tabel 2. Sudah diterapkan Metode *Shuffle Random*

Skenario Pengujian	Hasil yang diharapakan	Hasil Pengujian	Kesimpulan
Saat Permainan akan pertama kali untuk dijalankan	Musuh akan melakukan proses pengacakan *posisi kemunculan*		Valid
Player berada pada jarak <= 10 dariMusuh	Musuh akan melakukan aksi untuk menyerang.		Valid
Game terdapat Boss musuh dan memiliki anak Buah2	Yang diacak *posisi kemunculan* untuk meningkatkantingkat kesulitan yang dilewatkan player denganmemiliki kekuatan skill masing- masing.		Valid
Bos musuh memilikisenjata berupa pedang untuk melawan player	Agar bos musuh memiliki kekuatan yang sulit untuk dikalahkan oleh player.		Valid
Player berada pada jarak >3 darimusuh	Musuh akan mengeluarkanskill untuk menyerang player		Valid

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

3. Pengujian Pertama User Acceptance Test (UAT)

Berikut hasil dari data yang telah diperoleh dari Uji Pertama ini:

Tabel 3. Hasil Pengujian Lapangan Tebatas

No	Pernyataan	Jawaban					Pecahan	Persentase	Total Persentase
		STS X1	TS X2	CS X3	S X4	SS X5			
1	Tampilan interfaces yang terdapat pada game ini lebih menarik untuk dilihat?		4	3	20	10	3,5	70%	73,3%
2	Menu yang terdapat pada game ini lengkap dan sesuai?	1		6	12	20	3,9	78%	
3	Tombol kontrol untuk bergerak dalam melakukan aksi terhadap karakter terletak dengan rapi dan mudah untuk di tekan?	1			20	15	3,6	72%	
4	Efek suara yang terdapat di menu dalam game terasa seusai dan nyaman untuk didengar?	1			28	10	3,9	78%	76,7%
5	Obstacle atau rintangan yang terdapat di dalam permainan game ini terasa menantang dan seru?	1		3	20	15	3,8	76%	
6	Efek suara ketika karakter saat sedang melakukan aksi terdengar lebih bagus?	1			32	5	3,8	76%	
7	Proses pertarungan pada game ini ketika player bertarung melawan musuh terasa seru?	1		3	16	20	4	80%	68,5%
8	Perilaku pengacakan yang digunakan oleh musuh susah untuk di tebak?	1	4	6	16	10	3,5	70%	
9	Penggunaan Shuffle Random pada musuh dapat menambah tantangan bermain pada game ini?	1			24	15	4	80%	
10	Damage atau kerusakan yang dihasilkan antara karakter dan musuh yang ada di level adalah balance atau sama rata?	5		9	8		2,2	44%	

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

Nilai yang didapat dari masing-masing aspek adalah UI & UX 73,3%, SFX & BGM 78,7% dan Tingkat Kesulitan 68,5%.

Dari paparan diatas maka dapat disimpulkan bahwa nilai persentase dari aspek UI & UX terbilang baik karena menyentuh angka 73,3% dan SFX & BGM terbilang baik karena menyentuh angka 78,7%. Sedangkan dari aspek Tingkat kesulitan mendapat angka persentase yang cukup baik, karena mendapatkan angka 68,5%.

Karena masing-masing aspek nilai diatas rata-rata 56%, maka dapat disimpulkan tidakterdapat revisian pada masing-masing aspek. Namun karena data yang baru saja didapat hanyasebagai sampel yakni hanya diambil 10 orang orang saja maka masih belum bisa dijadikan sebagai kesimpulan akhir, sehingga pengambilan data akan dilakukan kembali dengan subjek lebih banyak. Hasil dari pengujian lapangan terbatas dalam bentuk tabel dapat dilihat diatas.

4. Pengujian Kedua User Acceptance Test (UAT)

Berikut hasil dari data yang telah diperoleh dari Uji kedua ini:

Tabel 4. Hasil Pengujian Lapangan Lebih Luas

No	Pernyataan	Jawaban					Pecahan	Persentase	Total Persentase
		STS X1	TS X2	CS X3	S X4	SS X5			
1	Tampilan interfaces yang terdapat pada game ini lebih menarik untuk dilihat?	6	2	3	56	40	3,6	72%	72%
2	Menu yang terdapat pada game ini lengkap dan sesuai?	6		9	52	40	3,6	72%	
3	Tombol kontrol untuk bergerak dalam melakukan aksi terhadap karakter terletak dengan rapi dan mudah untuk di tekan?	6		6	52	45	3,63	73%	
4	Efek suara yang terdapat di menu dalam game terasa seusai dan nyaman untuk didengar?	6		3	52	50	3,7	74%	73%
5	Obstacle atau rintangan yang terdapat di dalam permainan game ini terasa menantang dan seru?	6		6	60	35	3,6	72%	
6	Efek suara ketika karakter saat sedang melakukan aksi terdengar lebih bagus?	6		3	60	40	3,63	73%	
	Proses pertarungan pada game ini								

7	ketika player bertarung melawan musuh terasa seru?	6		3	56	45	3,66	73%	70%
8	Perilaku pengacakan yang digunakan oleh musuh susah untuk di tebak?	6	2	9	48	40	3,5	70%	
9	Penggunaan Shuffle Random pada musuh dapat menambah tantangan bermain pada game ini?	6			60	45	3,7	74%	
10	Damage atau kerusakan yang dihasilkan antara karakter dan musuh yang ada di level adalah balance atau sama rata?	8	6	6	44	30	3,13	63%	

Berdasarkan data hasil uji lapangan lebih luas nilai yang didapatkan masing-masing aspek ialah UI & UX 72%, SFX & BGM 73% dan Tingkat Kesulitan 70%.

Jika dibandingkan dengan hasil pengujian lapangan terbatas, maka pada aspek UI & UX mengalami penurunan nilai yang cukup signifikan, dimana pada pengujian lapangan terbatas aspek UI & UX mendapatkan nilai persentase 73,3%. Selanjutnya dari aspek SFX & BGM juga mengalami penurunan nilai, dimana pada pengujian lapangan terbatas aspek SFX & BGM mendapatkan nilai persentase 76,7%. Sedangkan dari aspek Tingkat Kesulitan mengalami peningkatan nilai yang signifikan, dimana pada pengujian lapangan terbatas aspek Tingkat Kesulitan mendapatkan nilai persentase 68,5%.

Pada pengujian lapangan lebih luas ini, peneliti telah menyematkan algoritma shuffle random di dalam game, sehingga hasil yang didapatkan adalah persentase dari tingkat kesulitan memperoleh peningkatan yang cukup signifikan.

Dalam pengujian lapangan lebih luas ini hanya diambil dari 30 orang saja, maka masih belum bisa dijadikan sebagai kesimpulan akhir, sehingga pengambilan data akan dilakukan kembali dengan subjek yang lebih banyak.

5. Pengujian Ketiga User Acceptance Test (UAT)

Berikut hasil dari data yang telah diperoleh dari Uji ketiga ini:

Tabel 5. Hasil Pengujian Operasional

No	Pernyataan	Jawaban					Pecahan	Persentase	Total Persentase
		STS X1	TS X2	CS X3	S X4	SS X5			
1	Tampilan interfaces yang terdapat pada game ini lebih menarik untukdilihat?	7	2	21	220	100	3,9	78%	77%
2	Menu yang terdapat pada game ini lengkap dan sesuai?	7		30	224	85	3,8	76%	

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

3	Tombol kontrol untuk bergerakdalam melakukan aksi terhadap karakter terletak dengan rapi dan mudah untuk di tekan?	7		30	192	125	3,9	78%	
4	Efek suara yang terdapat di menu dalam game terasa seusai dannyaman untuk didengar?	7	2	21	220	100	3,9	78%	77%
5	Obstacle atau rintangan yang terdapat di dalam permainan game ini terasamenantang dan seru?	7	2	30	212	95	3,8	76%	
6	Efek suara ketika karakter saatsedang melakukan aksi terdengar lebih bagus?	7		21	240	80	3,9	78%	
7	Proses pertarungan pada game iniketika player bertarung melawan musuh terasa seru?	7	2	27	220	90	3,8	76%	74%
8	Perilaku pengacakan yang digunakan oleh musuh susah untuk di tebak?	7	6	39	200	85	3,7	74%	
9	Penggunaan Shuffle Random padamusuh dapat menambah tantangan bermain pada game ini?	7	2	21	232	85	3,8	76%	
10	Damage atau kerusakan yang dihasilkan antara karakter dan musuh yang ada di level adalah balance atau sama rata?	11	18	30	184	70	3,5	70%	

Berdasarkan data hasil uji lapangan lebih luas nilai yang didapatkan masing-masing aspek ialah UI & UX 77%, SFX & BGM 77% dan Tingkat Kesulitan 74%.

Jika dibandingkan dengan hasil pengujian lapangan terbatas, maka pada aspek UI & UX mengalami peningkatan nilai persentase, dimana pada pengujian lapangan terbatas aspek UI & UX mendapatkan nilai persentase 73,3%. Selanjutnya dari aspek SFX & BGM juga mengalami peningkatan nilai persentase, dimana pada pengujian lapangan terbatas aspek SFX & BGM mendapatkan nilai persentase 76,7%. Sedangkan dari aspek Tingkat Kesulitan mengalami peningkatan niliai persentase, dimana pada pengujian lapangan tebatas aspek Tingkat Kesulitan mendapatkan nilai persentase 68,5%.

Jika dibandingkan dengan hasil pengujian lapangan lebih luas, maka pada aspek UI & UX mengalami peningkatan nilai persentase, dimana pada pengujian lapangan lebih luas aspek

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

UI & UX mendapatkan nilai persentase 72%. Selanjutnya dari aspek SFX & BGM juga mengalami peningkatan nilai persentase, dimana pada pengujian lapangan lebih luas aspek SFX & BGM mendapatkan nilai persentase 73%. Sedangkan dari aspek Tingkat Kesulitan mengalami peningkatan niliai persentase, dimana pada pengujian lapangan lebih luas aspek Tingkat Kesulitan mendapatkan nilai persentase 70%.

4. KESIMPULAN

Berdasarkan hasil dari penelitian dan pengembangan yang telah diperoleh, maka dapat diambil kesimpulan sebagai berikut:

- Penelitian dan pengembangan ini menghasilkan *game 2D action* yang memiliki *obstacle* atau rintangan yang unik yakni pengacakkan terhadap kemunculan posisi yang digunakan oleh musuh.
- Penerapan metode algoritma *shuffle random* berpengaruh pada peningkatan jumlah nilai persentase pada aspek Tingkat Kesulitan. Peningkatan nilai tersebut dapat dilihat dari hasil masing-masing aspek penelitian, dimana pada penelitian I metode algoritma *shuffle random* belum diterapkan di dalam game, persentase yang diperoleh dari aspek tingkat kesulitan 68,5%. Selanjutnya pada penelitian II setelah metode algoritma *shuffle random* sudah diterapkan di dalam game, persentase yang diperoleh tingkat kesulitan meningkat menjadi 70%. Juga di penelitian III persentase yang didapatkan dari aspek tingkat kesulitan kembali mengalami peningkatan nilai persentase menjadi 74%.
- Perbaikan bug yang terdapat dalam game berpengaruh terhadap aspek UI & UX dan aspek SFX & BGM. Hal ini dapat dilihat dari masing-masing aspek penelitian, dimana pada pengujian I aspek UI & UX mendapatkan nilai persentase 73,3% dan aspek SFX & BGM mendapatkan nilai persentase 76,7%. Selanjutnya pada penelitian II aspek UI & UX dan SFX & BGM mengalami penurunan nilai persentase, dimana aspek UI & UX mendapatkan nilai persentase 72% dan aspek SFX & BGM mendapatkan nilai persentase 73%. Tetapi pada penelitian III aspek UI & UX dan SFX & BGM memperoleh peningkatan nilai persentase, dimana aspek UI & UX mendapatkan nilai persentase 77% dan aspek SFX & BGM mendapatkan nilai persentase 77%.

Berdasarkan kesimpulan diatas, maka disarankan beberapa hal tersebut:

- Diharapkan pada penelitian atau pengembangan selanjutnya untuk menambahkan fitur-fitur pada *game* ini seperti fitur pencapai seperti koleksi item dan lain sebagainya agar parapemain tidak mudah untuk merasakan bosan dalam bermain.
- Diharapkan kepada peneliti atau pengembangan selanjutnya untuk menambahkan opsi resolusi layar yang lebih banyak agar terlihat begitu menarik.
- Diharapkan pada penelitian atau pengembangan selanjutnya untuk menambahkan level pada *game* ini agar alur cerita yang ada pada *game* ini dapat terus berlanjut.

***e-issn :* 2615-6474**
p-issn : 2620-3804
URL : https://jurnal.machung.ac.id/index.php/kurawal

5. REFERENSI

[1] Amelia Yusnita, Sefty Wijayanti, Putri Alysia Felita. (2017). IMPLEMENTASI ALGORITMA *SHUFFLE RANDOM* PADA *EDUGAME MAGIC TIME* BERBASIS *UNIVERSAL WINDOWS PLATFORM* (UWP)

[2] Alvian, P.M., dan Dwi, K., (2015). PEMBUATAN *GAME ANDROID 2D* PERTUALANGAN MR.KENTANG MENGGUNAKAN *UNITY*. Vol. 21 No 2. Desember 2015

[3] Bartolomius Harpad, Salmon, Yohanes Rombe Paran. (2019). PENERAPAN ALGORITMA *SHUFFLE RANDOM PADA GAME* EDUKASI TEBAK LAGU DAERAH KALIMANTAN TIMUR

[4] Fendy, A.P., (2016) PEMBUATAN *GAME* EDUKA "PERTUALANGAN SI GEMBUL" SEBAGAI PEMBELAJARAN PENGENALAN DAERAH SOLO RAYA PADA ANAK. Vol. 7 No 2 November 2016