

Penerapan *Finite State Machine* pada Squid Game Berbasis Dekstop Menggunakan Unity 3D

Muhamad B. Kresna Bayu¹, Uus Muhamad Husni Tamyiz², Lise Sri Andar³

Program Studi Teknik Informatika STT Wastukencana Purwakarta
Jl Cikopak No. 53 Kec. Babakancikao, Purwakarta 41151

¹bayukreshna00@gmail.com

²mrtamyiz@gmail.com

³lise@wastukencana.ac.id

Abstract— *The emergence of modern games has resulted in children tending to play gadgets and leave traditional games. This study aims to develop a traditional 3D themed game that adapts the desktop-based Squid Game series. Squid Game 3D will be developed by applying the Finite State Machine (FSM) control system design method. In designing the game, the researchers used the Multimedia Document Life Cycle (MDLC) development method which consisted of the concept, design, material collection, assembly, and testing. The concept adapts the Red Light Green Light game found in the Squid Game film. Game designed using UML, Storyboard, and FSM. Materials needed in include writing titles, pictures of characters and images as game backgrounds. Furthermore, the game is made by coding and creating an interface using Unity 3D, Visual Studio Code, and 3D Viewer. As a result, a desktop-based 3D Squid Game was successfully developed next tested using the black box testing. and the result was that all system functions in Squid Game 3D were successfully run.*

Keywords— *FSM, Squid Game, Unity 3D, MDLC, Black Box Testing.*

Abstrak— *Munculnya game modern mengakibatkan anak-anak cenderung bermain gadget dan meninggalkan permainan tradisional. Penelitian ini bertujuan untuk mengembangkan game 3D bertema permainan tradisional yang mengadaptasi serial Squid Game berbasis dekstop. Squid Game 3D akan dikembangkan dengan menerapkan metode perancangan sistem kontrol Finite State Machine (FSM). Dalam merancang game tersebut peneliti menggunakan metode pengembangan Multimedia Document Life Cycle (MDLC) yang terdiri dari tahap konsep, desain, pengumpulan bahan, pembuatan, dan pengujian. Konsep game mengadaptasi permainan Red Light Green Light pada film Squid Game. Game didesain menggunakan UML, Storyboard, dan FSM. Bahan yang diperlukan diantaranya tulisan judul, gambar karakter, dan gambar lain sebagai latar belakang game. Selanjutnya game dibuat dengan melakukan pengkodean dan pembuatan antarmuka menggunakan aplikasi Unity 3D, Visual Studio Code, dan 3D Viewer. Alhasil, Squid Game 3D berbasis dekstop berhasil dikembangkan lalu diuji dengan black box testing dimana semua fungsi sistem pada Squid Game 3D sukses dijalankan.*

Kata kunci— *FSM, Squid Game, Unity 3D, MDLC, Black Box Testing.*

I. PENDAHULUAN

Perkembangan teknologi tentunya memberikan banyak dampak bagi kehidupan manusia misalnya dalam hal hiburan [1]. Seperti permainan, yang awalnya para anak-anak berkumpul di luar rumah untuk bermain permainan tradisional seperti kelereng, lompat tali, petak umpet dan sebagainya. Namun kini hal tersebut sudah sangat jarang ditemukan. Sebab, munculnya permainan modern yakni sebuah game yang dimainkan secara *offline* maupun *online*. Akibatnya, anak-anak cenderung meninggalkan permainan tradisional.

Game dimainkan dengan aturan tertentu sehingga ada hasil akhir menang dan kalah, biasanya tujuannya adalah untuk hiburan. Misalnya permainan pada tantangan pertama di serial Squid Game yang bernama Red Light Green Light atau Lampu Merah Lampu Hijau. Dalam bahasa Korea, permainan ini disebut dengan “Mugunghwa kkoci pieot seumnida” yang berarti bunga Mugung telah mekar. Permainan tersebut dimainkan oleh sekelompok orang yang salah satunya adalah penjaga. Apabila penjaga mengucapkan “Mugunghwa kkoci pieot seumnida” tersebut, para pemain harus segera diam membeku. Siapa yang bergerak akan langsung kalah dan jika di Squid Game akan mendapat tembakan, sedangkan yang berhasil sampai ke garis finish terlebih dahulu akan menjadi pemenangnya [2].

Game yang baik diciptakan menggunakan metode pengembangan yang tepat, salah satunya ialah metode MDLC (*Multimedia Development Life Cycle*). Metode MDLC merupakan metode yang sesuai dalam merancang dan mengembangkan suatu aplikasi media yang merupakan gabungan dari media gambar, suara, video, dan animasi. Maka, metode ini cocok untuk diterapkan dalam pengembangan sistem pada aplikasi multimedia misalnya game. Sebab tiap-tiap tahapannya dapat saling bertukar posisi sesuai dengan kebutuhan penelitian [3].

Pembuatan game umumnya juga menggunakan *artificial intelligecne* salah satunya yakni Finite State Machine (FSM) [4]. *Finite State Machine* ialah metode perancangan sistem kontrol yang menggambarkan prinsip kerja sistem dengan menerapkan tiga hal, yaitu: *state*, *event*, dan *action* [5]. Penerapan algoritma *Finite State Machine* ini juga berguna untuk menentukan dan mendesain berbagai macam respon pada model game [6]. Kelebihan lainnya dari *Finite State Machine* ini adalah sederhana, efisien, dan mudah dikembangkan [7].

Penelitian ini akan mengimplementasikan metode FSM ke dalam pembuatan game Red Light Green Light dengan judul “Squid Game 3D” yang merupakan game platform yang mengadaptasi permainan di serial Squid Game. Game tersebut akan dikembangkan berbasis desktop menggunakan Unity 3D dan Visual Studio Code. Permainan Red Light Green Light secara garis besar bercerita dimana pemain sebagai karakter utama harus menuju garis finish dengan cara berlari dan menghindari tembakan dari boneka penjaga sebagai (*Non Player Character*) (NPC). Game tersebut akan dikembangkan menggunakan metode MDLC (*Multimedia Development Life Cycle*). Diharapkan hasil pembuatan Squid Game 3D mampu memberikan hiburan yang meyenangkan.

II. PENELITIAN TERKAIT

Penerapan metode FSM ke dalam game Pengenalan Unsur Kimia berpengaruh pada perilaku musuh digunakan untuk penghitungan pemodelan selanjutnya dan merubah performa objek-objek pada game [8]. Metode FSM yang diimplementasikan pada game “The Mahasiswa” terdapat pada percakapan karakter tokoh NPC untuk menentukan tingkat kesulitan pertanyaan yang diberikan sesuai jumlah jawaban yang berhasil dijawab benar oleh pemain [9]. Serial Squid Game tentang permainan tradisional asal Korea Selatan menunjukkan hegemoni akan paham kapitalisme menyebabkan masyarakat percaya bahwa kekayaan dan status sosial dapat memberikan kekuasaan [10]. Aplikasi Unity 3D berhasil diterapkan pada pembuatan game Last Night bergenre Survival Arcade dengan menggunakan Bahasa Pemrograman C# [11]. Unity 3D juga berhasil diterapkan pada game Covid-19 sebagai media edukasi Covid-19. [12].

III. ANALISIS DAN PERANCANGAN GAME

A. Analisis Kebutuhan

Kebutuhan sistem adalah penetapan fungsi yang dapat dilakukan oleh sistem, sehingga dapat memilih perangkat yang akan digunakan pada pembuatan game Squid Game 3D. Perangkat lunak yang dibutuhkan antara lain yakni :

- 1) Unity 3D
- 2) Visual Studio Code
- 3) 3D Viewer

Sedangkan perangkat keras yang dibutuhkan yakni :

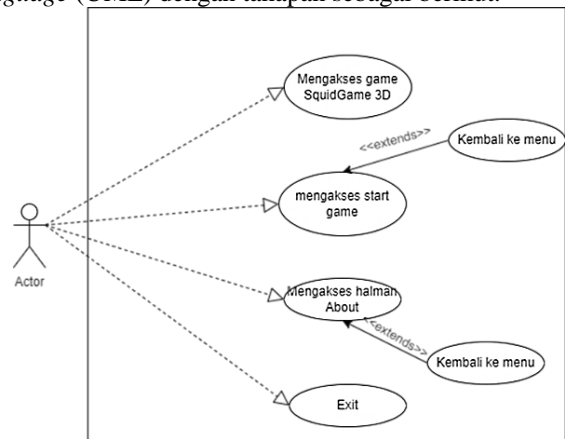
- 1) PC desktop
- 2) Prosesor i5 3470
- 3) VGA GTX 750
- 4) RAM 8gb
- 5) PSU aerocool

B. Konsep Game

Concept (konsep) merupakan tahap dalam menentukan tujuan, genre, dan pola permainan dalam pembuatan game. Game berjudul “Squid Game 3D” ini bertujuan untuk mengenalkan salah satu tantangan yang ada di serial Squid Game yakni Red Light, Green Light ke dalam bentuk game berbasis Windows yang dapat dimainkan oleh masyarakat umum. Game ini bergenre platformer dengan pola permainan Red Light Green Light.

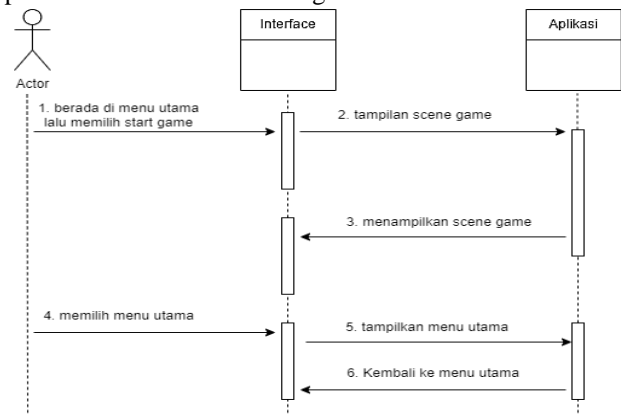
C. Desain Game

Perancangan sistem dibangun dengan *Unified Modeling Language* (UML) dengan tahapan sebagai berikut.



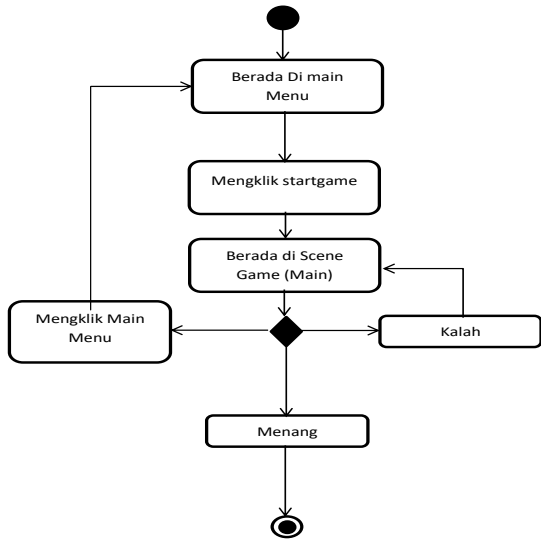
Gambar. 1 Use Case Diagram Squid Game 3D

Gambar 1 merupakan *use case diagram* pada game “Squid Game 3D” dengan aktornya sebagai pemain / pengguna. *Use case* utama pada game tersebut adalah menu utama yang terdiri dari 3 tiga bagian yakni *Start Game*, *About*, dan *Exit*. Selanjutnya *sequence diagram* pada Start Game adalah sebagai berikut.



Gambar. 2 Squence Diagram Squid Game 3D

Kemudian, activity diagram mengakses start game ditunjukkan pada gambar 3 berikut.



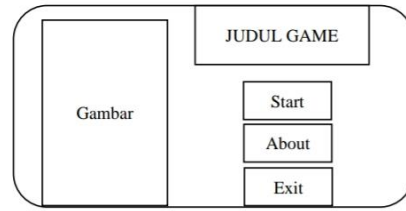
Gambar. 3 Activity Diagram Squid Game 3D

Selain UML adapula desain storyboard Squid Game 3D sebagai berikut.

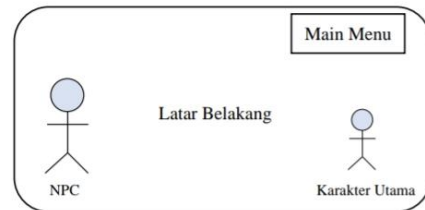
TABEL I
STORYBOARD SQUID GAME

Object		Keterangan
Text	<ul style="list-style-type: none"> Main menu You Win 	<ul style="list-style-type: none"> You Win akan muncul ketika player sampai garis finish
Suara	<ul style="list-style-type: none"> Suara Green Light Suara Red Light Dark Thing The Island Dr Sinister 	<ul style="list-style-type: none"> Suara greenlight akan muncul saat greenlight Suara redlight akan muncul saat redlight Backsound Backsound
Animasi	<ul style="list-style-type: none"> Animasi Awan Animasi boneka Animasi player lari Animasi player iddle Animasi bullet Animasi mati 	

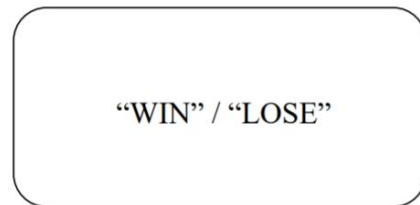
Interface atau antarmuka merupakan salah satu bagian yang penting dalam suatu aplikasi. Berikut merupakan desain tampilan antarmuka pada halaman awal game, credit scene, dan akhir game.



Gambar. 4 Rancangan Antarmuka Awal Squid Game 3D



Gambar. 5 Rancangan Antarmuka Gameplay Squid Game 3D



Gambar. 6 Rancangan Antarmuka Akhir Squid Game 3D

D. Pengumpulan Bahan

Bahan – bahan yang dibutuhkan dalam pembuatan Squid Game 3D, disajikan pada tabel berikut:

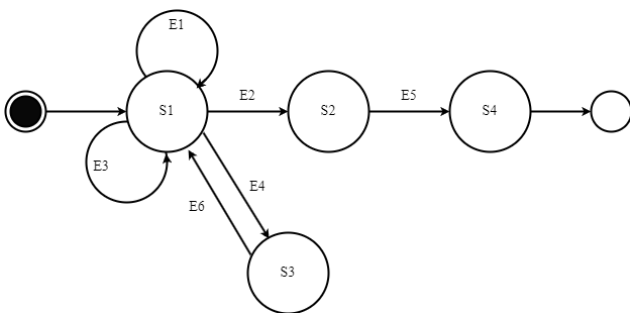
TABEL II
ASET SQUID GAME

No.	Jenis Data	Nama File	Aset
1.	Gambar	Tulisan Squid Game	
2.	Gambar	Kertas	
3.	Gambar	Karakter Player	
4.	Gambar	Karakter NPC	
5.	Gambar	Pohon	
6.	Audio	Backsound	Dark Thing
7.	Audio	Backsound	The Island Dr Sinister

IV. HASIL DAN PEMBUATAN GAME

A. Pembuatan Game

Pada tahap pembuatan (*assembly*) ini dilakukan pembuatan semua objek atau bahan multimedia dengan kata lain yaitu pengkodean. Pengkodean dalam pembuatan game “Squid Game 3D” menggunakan aplikasi visual studio code yang didasarkan pada tahap *design*, seperti *storyboard*, bagan alir, dan/atau struktur navigasi. Sedangkan perancangan sistem kontrol untuk karakter utama maupun NPC yang ada di game ini menerapkan metode *Finite State Machine* (FSM). Pada pembuatan game ini lebih berfokus pada pergerakan karakter dengan metode *Finite State Machine*. Berikut algoritma FSM untuk karakter utama.



Gambar. 7 Algoritma FSM untuk Pemain

Keterangan Gambar 12 :

- S1 = Pemain diam
- S2 = Pemain berlari
- S3 = Pemain mati
- S4 = Pemain finish
- E1 = “Greenlight” pemain diam
- E2 = “Greenlight” pemain jalan
- E3 = “Redlight” pemain diam
- E4 = “Redlight” pemain jalan
- E5 = Pemain sampai finish
- E6 = Pemain mati

Transisi tiap *state* pada gambar 12 akan disajikan pada tabel III *State* awal berada pada kolom sebelah kiri dimulai dari S1 dan berurutan ke bawah sampai S4. *Event* berada pada kolom bagian atas dimulai dari E1 dan berakhir di E6. *State* akhir berada pada kolom tengah dan tidak berurutan karena *state* akhir ditentukan oleh *state* awal dan *event*.

TABEL III
TRANSISI PLAYER

S	INPUT					
	E1	E2	E3	E4	E5	E6
S1	S1	S2	S1	S3		
S2					S4	
S3						S1
S4						

Berikut adalah hasil pengkodean menggunakan metode *Finite State Machine* bagi perilaku Player.

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

public class PlayerController : MonoBehaviour
{
    private CharacterController characterController;
    public float movementSpeed;

    public Animator animator;
    public readonly string moveAnimParameter = "Move";

    public DollController dollController;

    public bool isDead;

    private void Awake()
    {
        characterController = GetComponent<CharacterController>();
    }

    void Update()
    {
        Movement();
    }

    void Movement()
    {
        if (isDead) return;
        float moveX = Input.GetAxis("Horizontal");
        float moveZ = Input.GetAxis("Vertical");
        Vector3 move = new Vector3(moveX, 0, moveZ);

        if (moveX != 0 || moveZ != 0)
        {
            if (!dollController.isGreenLight)
            {
                dollController.ShootPlayer(transform);
                print("Kamu Ditembak mati");
            }
        }

        characterController.Move(move * movementSpeed * Time.deltaTime);

        float moveAnim = new Vector2(moveX, moveZ).magnitude;
        animator.SetFloat(moveAnimParameter, moveAnim);

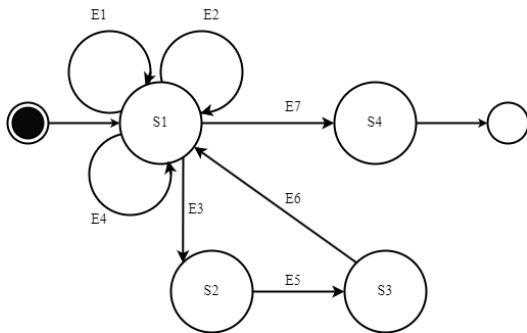
        if (moveX == 0 || moveZ == 0) return;
        float heading = Mathf.Atan2(moveX, moveZ);
        transform.rotation = Quaternion.Euler(0, heading * Mathf.Rad2Deg, 0);
    }
}
```

```

}

public void Dead()
{
    isDead = true;
    print("Player mati");
}
}
    
```

Pergerakan NPC (*Non Player Character*) yakni boneka (*doll*) ketika terdengar kata “*Green Light*” maka NPC membiarkan pemain berlari. Namun, jika yang muncul kata “*Red Light*” dan pemain bergerak, maka NPC akan menembak pemain tersebut. Berikut ini adalah algoritma FSM perilaku NPC yakni boneka (*doll*).



Gambar. 8 Algoritma FSM untuk NPC

Keterangan Gambar 13 :

- S1 = NPC diam
- S2 = NPC menembak
- S3 = NPC berhasil menembak
- S4 = NPC tidak menembak
- E1 = “Greenlight” pemain diam
- E2 = “Greenlight” pemain jalan
- E3 = “Redlight” pemain jalan
- E4 = “Redlight” pemain diam
- E5 = NPC menembak pemain
- E6 = Pemain mati
- E7 = Pemain finish

Transisi tiap *state* pada gambar 13 akan disajikan pada tabel IV *State* awal berada pada kolom sebelah kiri dimulai dari S1 dan berurutan ke bawah sampai S4. *Event* berada pada kolom bagian atas dimulai dari E1 dan berakhir di E7. *State* akhir berada pada kolom tengah dan tidak berurutan karena *state* akhir ditentukan oleh *state* awal dan *event*.

TABEL IV
TRANSISI NPC

S	INPUT						
	E1	E2	E3	E4	E5	E6	E7
S1	S1	S1	S2	S1			S4
S2					S3		
S3						S1	
S4							

Berikut adalah hasil pengkodean menggunakan metode *Finite State Machine* bagi perilaku NPC.

```

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

public class DollController : MonoBehaviour
{
    public float minTimer, maxTimer;
    public bool isGreenLight = true;

    public Animator animator;

    public readonly string greenLightAnim = "GreenLight";

    public Transform shotPoint;
    public GameObject bulletPrefab;
    public bool hasShot;

    // Start is called before the first frame update
    void Start()
    {
        StartCoroutine(ChangeLightCoroutine());
    }

    IEnumerator ChangeLightCoroutine()
    {
        yield return new WaitForSeconds(Random.Range(minTimer, maxTimer));

        if (isGreenLight)
        {
            isGreenLight = false;
            animator.SetBool(greenLightAnim, false);
            print("Lampu Merah, Player gableh jalan!!");
        }
        else
        {
            isGreenLight = true;
            animator.SetBool(greenLightAnim, true);
            print("Lampu Hijau, Player boleh jalan!!");
        }
        StartCoroutine(ChangeLightCoroutine());
    }

    public void ShootPlayer(Transform playerTarget)
    {
        if (hasShot) return;
        GameObject bulletGO = Instantiate(bulletPrefab, shotPoint.position, Quaternion.identity);
        bulletGO.GetComponent<BulletMovement>().playerTarget = playerTarget;
        hasShot = true;
    }
}
    
```

Dalam membuat game ini penulis juga menggunakan 3D versi 2021.3.19f1 dan Visual Studio Code. Game yang dibuat berdasarkan tahap perancangan desain, berikut hasil tampilan game yang telah dibuat.



Gambar. 9 Tampilan Awal Squid Game



Gambar. 10 Tampilan Scene Gameplay Squid Game



Gambar. 11 Tampilan Akhir Squid Game

B. Pengujian Game

Tahap pengujian (*testing*) adalah tahap untuk menguji beberapa fungsi utama yang ada pada game, apakah fungsi tersebut berfungsi sesuai yang diharapkan atau tidak. Pengujian yang dilakukan pada game Squid Game 3D adalah dengan pengujian *alpha* menggunakan metode *black box testing* hasilnya disajikan pada tabel berikut:

TABEL V
HASIL PENGUJIAN PADA SCENE MENU UTAMA

No	Ket	Action	Fungsi	Hasil
1.	Tombol	Tombol Start Game	Memulai permainan memindahkan ke scene game play	Sukses
2.	Tombol	Tombol about	Memindahkan scene ke scene about	Sukses
3.	Tombol	Tombol exit	Keluar dari permainan	Sukses

Pada tabel V yang di uji yaitu tombol play yang berfungsi untuk menampilkan scene game play, tombol about yang berfungsi menampilkan scene about dan tombol exit yang berfungsi untuk mengakhiri permainan dan keluar ke windows.

TABEL VI
HASIL PENGUJIAN PADA SCENE GAME

No	Ket	Action	Fungsi	Hasil
1.	Key	Tombol A	Jalan ke kiri	Sukses
2.	Key	Tombol D	Jalan ke kanan	Sukses
3.	Key	Tombol S	Jalan ke belakang	Sukses
4.	Key	Tombol W	Jalan ke depan	Sukses
5.	Menu	Tombol Menu Utama	Kembali ke scene menu utama	Sukses
6.	Mati	Reset scene game	Reset scene game play	Sukses
7.	Menang	Muncul text "you Win"	Game berakhir	Sukses

Pada tabel VI yang di uji berupa tombol pergerakan karakter, tombol menu, state ketika player mati, state ketika player menang. Tombol tersebut telah di uji dan berhasil semua.

TABEL VII
HASIL PENGUJIAN PADA SCENE ABOUT

No	Ket	Action	Fungsi	Hasil
1.	Tombol	Tombol main menu	Mengembalikan scene ke main menu	Sukses

Pada tabel VII dilakukan pengujian alpha pada menu about yang berisikan tombol main menu. Pengujian pada tombol main menu berhasil.

V. KESIMPULAN

Aplikasi "Squid Game 3D" berbasis dekstop berhasil dibuat berdasarkan rancangan serta implementasi dan pengujian yang telah dilakukan. Metode *Finite State Machine* yang diimplementasikan dalam game "Squid Game 3D" terdapat pada pergerakan yang ada pada karakter tokoh NPC (*Non Player Character*) dan tokoh utama. FSM berguna dalam menentukan tindakan dan gerakan tokoh utama dalam mengambil keputusan untuk berlari atau diam agar dapat menghindari serangan NPC yakni boneka penjaga yang sulit ditebak.

REFERENSI

[1] Armanda, A. F., & Rizqi, M. (2020). Game 3D Perjuangan Rakyat Surabaya Dalam Pertempuran 10 November 1945 Dengan Belanda Menggunakan Unreal Engine. *Journal of Animation and Games Studies*, 6(2), 87–96.

- [2] Oey, S. (2021). 7 Permainan Anak di Squid Game, Berani Coba? Retrieved from Ruparupa: <https://www.ruparupa.com/blog/permainan-squid-game/>
- [3] Hormansyah, D. S., Astiningrum, M., & Asyraq, F. A. (2020). Implementasi FSM (Finite State Machine) Pada Game Surabaya Membara. *JIP (Jurnal Informatika Polinema)*, 6, 11–17.
- [4] Jones, M. T. 2015. *Artificial Intelligence: A Systems Approach: A Systems Approach*. Jones & Bartlett Learning.
- [5] Rafi, A., & Tahtawi, A. (2016). Implementasi Metode Finite State Machine (FSM) sebagai Dasar Algoritma Robot Line Follower. *Jurnal Enjinerig Dan Ekonomi (JEE)*, 1, 6–11.
- [6] Rahadian, M. F., Suyatno, A., & Maharani, S. (2016). Penerapan Metode Finite State Machine Pada Game “THE RELATIONSHIP.” *Jurnal Informatika Mulawarman*, 11(1), 14–22.
- [7] Ramadijanti, N., Setiowati, Y., & Fathoni, K. (2015). Penerapan Finite State Machine Pada Perilaku Karakter. *FT UGM*, 30-36.
- [8] Bimantoro, T., & Hanny, H. (2016). Pemodelan Perilaku Musuh Menggunakan Finite State Machine (FSM) Pada Game Pengenalan Unsur Kimia. *Journal of Applied Intelligent System*, Vol.1, No. 3.
- [9] Hernawan, S. R. (2018). Penerapan Metode Finite State Machine Pada Game “The Mahasiswa” Guna Membangun Perilaku Non Playable Character. Skripsi Universitas Islam Indonesia.
- [10] Kusumayanti, D. D., Mukharomah, K. N., Wijayanti, E. C., Andriaka, R., & Sasli, S. A. (2022). Analisis Aspek-Aspek Hegemoni Dan Dominasi Sosial Ekonomi Dalam Serial ‘Squid Game.’ *Jurnal Entitas Sosiologi*, 11(2).
- [11] Anshori, I. F. (2022). Perancangan Game “Last Night” Menggunakan Unity 3D. *Jurnal Responsif*, 4(2), 168–172.
- [12] Haromunthe, J. P., & Riyanto, J. (2022). Pembuatan Game 3D Corona Survival Menggunakan Unity Berbasis Android. *OKTAL : Jurnal Ilmu Komputer Dan Science*, 1(09), 1355–1359.