

MERANCANG GAME OTO LAJU UNTUK MOBILE DAN PC

Razky Lawani^{1)*} Mugairun I. Bano^{2)*}

^{1)*} Program Studi Teknologi Rekayasa Perangkat Lunak, Program Vokasi Universitas Negeri Gorontalo
Email: razkylawani517@gmail.com mugairunbano004@gmail.com

Asal Negara: Indonesia

ABSTRAK

Game bertema Endless run sangatlah populer pada masanya, contohnya: (Subway Surfer, Temple Run Dan sebagainya). Game bertema Endless Run singkatnya merupakan game side scrolling dimana game akan berjalan dengan lintasan yang tak terhingga dan Player harus menghindari segala rintangan yang ada. Begitu pula dengan game yang akan dibahas pada jurnal kali ini. Dibuatnya game ini bertujuan untuk meningkatkan fokus dan juga melatih kesabaran dari para player yang memainkan game ini. Dengan mengumpulkan data-data dari internet dan juga media sosial yang bernama Youtube, maka terbentuklah game ini. Inilah Game yang bernama Oto Laju, dimana di game ini kalian akan berperan sebagai pengendara mobil F1 dan menghindari rintangan berupa mobil-mobil lain. Seperti yang kalian ketahui bahwa game bertema endless run tidak memiliki alur cerita ataupun gameplay yang susah seperti kebanyakan game soulslike. Digame ini kalian hanya akan mengumpulkan skor sebanyak-banyaknya, dan bertanding dengan teman kalian beradu skor siapa yang paling banyak. Game ini sangat layak dicoba untuk mewarnai hari-hari kalian, Bukan karena grafiknya yang bagus atau sistem kompetitif yang sangat seru. Tapi dengan gameplay yang simple dan mudah dimengerti, tidak seperti kebanyakan game di era sekarang yang harus belajar berbagai macam hal yang ada didalam suatu game.

Kata kunci: Endless Run, Oto Laju, F1 Racing, Skor Tertinggi.

ABSTRACT

Endless run themed games were very popular in their time, for example: (Subway Surfer, Temple Run and so on). Endless Run themed games in short are side scrolling games where the game will run with an infinite track and the Player must avoid all the obstacles that exist. Likewise with the game that will be discussed in this journal. The creation of this game aims to increase focus and also train the patience of the players who play this game. By collecting data from the internet and also social media called Youtube, this game was formed. This is a game called Oto Laju, where in this game you will act as an F1 car driver and avoid obstacles in the form of other cars. As you know that endless run themed games don't have a storyline or difficult gameplay like most soulslike games. In this game you will only collect as many scores as possible, and compete with your friends to see who scores the most. This game is really worth trying to color your days, not because of its good graphics or a very exciting competitive system. But with gameplay that is simple and easy to understand, unlike most games in this era that have to learn various kinds of things in a game.

Keywords: Endless Run, Oto Laju, F1 Racing, Highest Score

1. PENDAHULUAN

Endless Run atau Berlari Tanpa akhir merupakan tema dari sebuah game, dimana pemain akan menggunakan karakter tentu dan menghindari berbagai macam rintangan yang ada. Tujuan dari tema game ini sangatlah simpel dan tidak ribet, yaitu mengumpulkan skor tertinggi.

Genre ini berawal dari video game yang bergulir secara vertikal pada tahun 1970-an, terutama game balap. Pemain selalu bergerak maju, menghindari rintangan dan kendaraan lain. Taito's Speed Race, yang dirilis pada tahun 1974, adalah yang pertama. Pada akhir 1970-an dan awal 1980-an, konsep yang sama digunakan dalam permainan ski.

Game platform yang berorientasi pada kendaraan, seperti Jump Bug (1981) dan Moon Patrol (1982), menambahkan lompatan dan tembakan sebagai cara untuk menghadapi rintangan dalam level yang terus bergulir.

Game rumahan B.C.'s Quest for Tires (1983) menggunakan gameplay gulir paksa dan lompat dari Moon Patrol. Ide dikejar tanpa henti oleh rintangan, monster, atau bos yang tidak bisa dihancurkan untuk mendorong kemajuan sangat dipengaruhi oleh adegan batu besar dari film Raiders of the Lost Ark pada tahun 1981. Tema ini muncul dalam game seperti Draconian (1984) di mana pemain harus menghindari rintangan sambil dikejar oleh naga raksasa yang tak terkalahkan. Ini akan menjadi tema yang berulang dalam gameplay endless running.

2. METODOLOGI PENELITIAN

Pada penelitian kali ini menggunakan metode penelitian Literatur Digital, yaitu dengan mengumpulkan, menganalisis, mencari informasi berbasis digital. Seperti website-website informasi ataupun jurnal-jurnal yang lain.

3. HASIL DAN PEMBAHASAN

3.1 Penginstalan Unity 2018

Pada project kali ini Engine yang digunakan adalah Unity versi 2018.4.36f1. Kalian bisa mendownloadnya di website resmi unity yaitu : <https://unity.com/download>



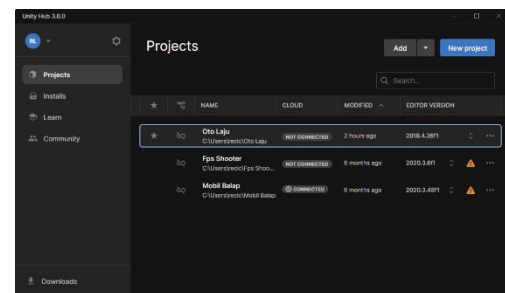
1. Download the Unity Hub

Follow the instructions onscreen for guidance through the installation process and setup.

[Download for Windows](#)
[Download for Mac](#)
[Instructions for Linux](#)

Gambar 1. Download Unity for Windows

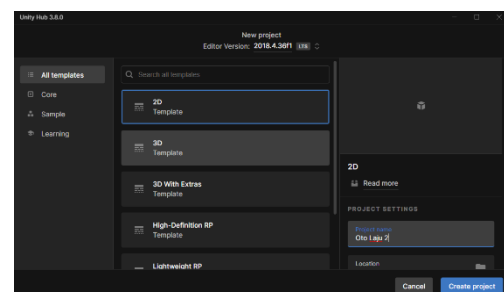
Pastikan untuk mendownload Unity untuk Windows. Kemudian install dan akan muncul tampilan seperti dibawah ini:



Gambar 2. Tampilan Unity hub

3.2 Pembuatan Project Oto Laju

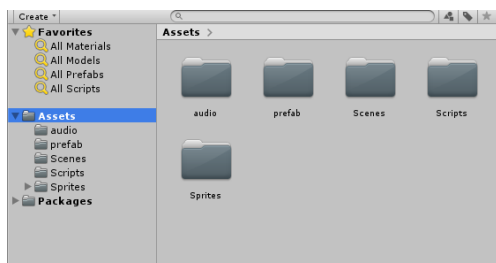
Setelah melakukan penginstalan, dilanjutkan dengan membuat project yang bernama Oto Laju.



Gambar 3. Pembuatan project Oto Laju

3.3 Folder-Folder yang penting

Jika sudah masuk ke projectnya, Tentunya harus membuat folder-folder penting terlebih dahulu. Untuk kelancaran pembuatan Game ini.



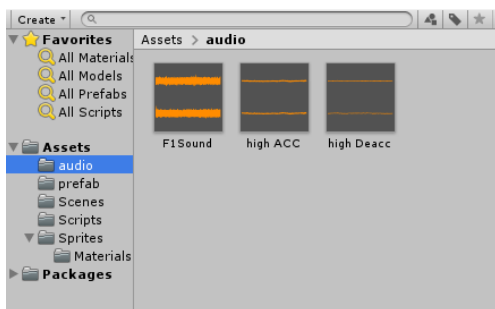
Gambar 4. Folder-Folder Penting

Seperti yang tertera pada Gambar 4., didalam folder Assets dibuatlah Folder-Folder lain seperti Audio, Prefab, Scenes, Scripts, Sprites.

- Audio – Menyimpan audio mobil yang akan dikendalikan oleh Player.
- Prefab – Berisi texture dari setiap musuh yang akan menghalangi Player.
- Scenes – Menyimpan beberapa halaman game yang akan digunakan untuk keperluan Project ini.
- Scripts – Mempunyai file Script atau file perintah untuk keperluan game.
- Sprites – Berisi semua texture yang ada didalam game ini.
- Flowchart – Berisi tentang penjelasan secara lengkap dari game yang dibuat.

Berikut penjelasan lebih lengkapnya, File apa saja yang ada didalam folder yang sudah dibuat :

A. Folder Audio



Gambar 5. Isi Folder Audio

Seperti yang kalian lihat pada Gambar 5., dimasukkan 3 file audio. Namun yang digunakan hanya 1 dari ketiga file yaitu F1 Sound.

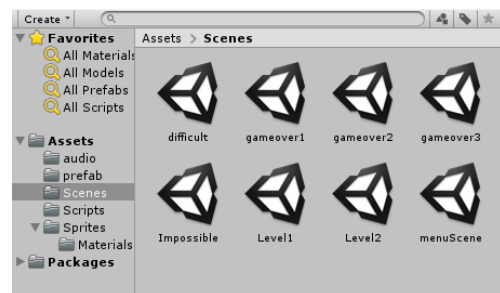
B. Folder Prefab



Gambar 6. Isi Folder Prefab

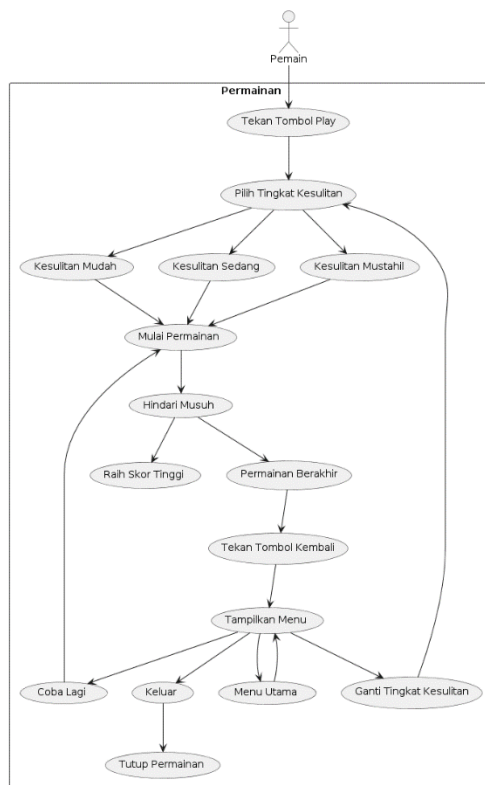
Di folder Prefab hanya digunakan 5 tekstur untuk musuhnya. Karena takutnya tekstur dari musuh yang keenam tidak akan muncul.

C. Folder Scenes



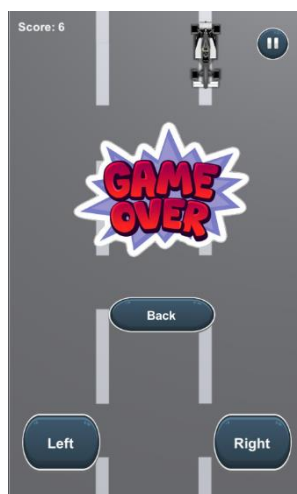
Gambar 7. Isi Folder Scene

Folder kali ini berbeda, karena isi dari folder atau file Scenes ini sangat berperan penting dalam penyempurnaan game ini. Di sini, terdapat tiga tingkat kesulitan yang berbeda yaitu: Easy, Medium, dan Impossible.



Gambar 8. UML Game OtoLaju

Scene Level1 sebagai Easy, Level2 sebagai Medium, dan Level3 sebagai Impossible. Setiap Scene tersebut memiliki tampilan game over yang rumit namun simpel: gambar game over sebenarnya adalah tombol yang diubah sedemikian rupa agar terlihat seperti gambar. Fungsi tombol ini, yang merupakan gambar game over, dihilangkan untuk penyempurnaan.



Gambar 9. Tampilan Game Over

Begitu juga dengan Scene gameover yang ada. Berikut perbedaan tingkat kesulitannya

Easy Medium Impossible



Gambar 10, 11 dan 12. Perbedaan setiap tingkat kesulitan

Bisa dilihat perbedaan setiap difficult yang ada.

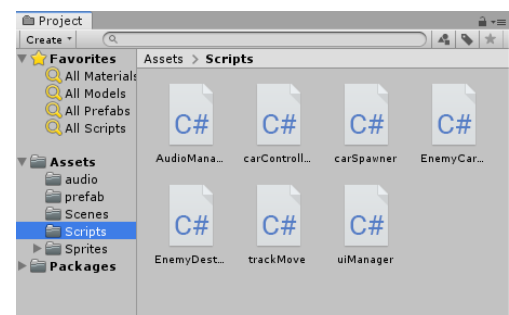
Untuk file Scenes menuScene dan difficult akan memperlihatkan tampilan seperti ini :



Gambar 12 dan 13. Tampilan Difficult dan MenuScene

D. Folder Scripts

Folder ini berisi file script/perintah agar gamenya bisa berjalan sedemikian rupa dan sangat rawan terjadi error discript ini.



Gambar 13. File yang ada didalam folder Scripts

Ada 7 File scripts disini yang berbeda-beda fungsinya, Lebih lengkapnya sebagai berikut

- AudioManager

```

using UnityEngine;
using System.Collections;

public class AudioManager : MonoBehaviour {

    public AudioSource carSound;

    // Use this for initialization
    void Start () {

    }

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

    }

}

```

Gambar 14. Scripts AudioManager

Simpelnya Codengan diatas bisa mengatur kapan audio ini akan berfungsi. Seperti saat player mulai bergerak audionya akan hidup atau saat player sudah kalah maka audionya akan mati.

- CarController

```

if UNITY_ANDROID
    currentPlatformAndroid = true;
#else
    currentPlatformAndroid = false;
#endif

am.carSound.Play ();

// Use this for initialization
void Start () {
    position = transform.position;

    if(currentPlatformAndroid == true){
        Debug.Log ("Android");
    }
    else {
        Debug.Log ("Windows");
    }
}

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

    if(currentPlatformAndroid == true){
        //Android spesifik code
    }
    else{
        position.x += Input.GetAxis ("Horizontal") * carSpeed * Time.deltaTime;
        position.x = Mathf.Clamp (position.x,-2.5f,2.5f);
        transform.position = position;

        position = transform.position;
        position.x = Mathf.Clamp (position.x,-2.5f,2.5f);
        transform.position = position;
    }

    void OnCollisionEnter2D(Collision2D col){
        if(col.gameObject.tag == "Enemy Car"){
            //Destroy (gameObject);
            gameObject.SetActive(false);
            ui.gameOverActivated();
            am.carSound.Stop();
        }
    }

    public void MoveLeft(){
        rb.velocity = new Vector2(-carSpeed, 0);
    }

    public void MoveRight(){
        rb.velocity = new Vector2(carSpeed, 0);
    }

    public void SetVelocityZero(){
        rb.velocity = Vector2.zero;
    }
}

```

Gambar 15. Scripts CarController

Intinya Semua script diatas berperan untuk membuat player dapat bergerak kekiri ataupun kekanan menggunakan arrow yang ada dikeyboard atau menggunakan tombol yang sudah tersedia bagi mobile/android.

- CarSpawner

```

using UnityEngine;
using System.Collections;

public class CarSpawner : MonoBehaviour {

    public GameObject[] cars;
    int carNo;
    public float spawnRate = 1.0f;
    public float delayTimer = 0f;
    float timer;

    // Use this for initialization
    void Start () {
        timer = delayTimer;
    }

    // Update is called once per frame
    void Update () {
        timer -= Time.deltaTime;
        if(timer <= 0f){
            Vector3 carPos = new Vector3(Random.Range (-2.2f,2.8f), transform.position.y,transform.position.z);
            carNo = Random.Range (0,cars.Length);
            Instantiate (cars[carNo],carPos,transform.rotation);
            timer = delayTimer;
        }
    }
}

```

Gambar 16. Scripts CarSpawner

Kode-kode diatas berfungsi untuk membuat mobil musuh player bisa Muncul dengan otomatis setiap detik, dan waktu kemunculannya juga bisa disesuaikan.

- EnemyCarMove

```

using UnityEngine;
using System.Collections;

public class EnemyCarMove : MonoBehaviour {

    public float speed = 5f;

    // Use this for initialization
    void Start () {

    }

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

        transform.Translate (new Vector3(0,1,0) * speed * Time.deltaTime);
    }
}

```

Gambar 17. Scripts EnemyCarMove

Kode yang rumit diatas berfungsi agar mobil musuh dapat bergerak secara vertikal yaitu dari atas kebawah untuk menghalangi Player.

- EnemyDestroyer

```

using UnityEngine;
using System.Collections;

public class EnemyDestroyer : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

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

    }

    void OnCollisionEnter2D(Collision2D col){
        if(col.gameObject.tag == "Enemy Car"){
            Destroy (col.gameObject);
        }
    }
}

```

Gambar 18. Scripts EnemyDestroyer

Nah untuk kode yang satu ini dia berfungsi agar game tetap bisa berjalan lancar dimana, musuh yang melewati player akan terhapus oleh kode EnemyDestroyer ini yang berbentuk area dan bisa diatur untuk kelebarannya.

- TrackMove

```

using UnityEngine;
using System.Collections;

public class trackMove : MonoBehaviour {

    public float speed;
    Vector2 offset;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        offset = new Vector2(0,Time.time * speed);
        GetComponent<Renderer> ().material.mainTextureOffset = offset;
    }

}

```

Gambar 19. Scripts TrackMove

Script TrackMove ini berfungsi untuk menggerakkan track atau jalannya dengan konsep looping sehingga menciptakan ilusi seolah-olah jalannya bergerak.

- UiManager

```

using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class UiManager : MonoBehaviour {

    public Button[] buttons;
    public Text scoreText;
    bool gameOver;
    int score;

    // Use this for initialization
    void Start () {
        gameOver = false;
        score = 0;
        InvokeRepeating("scoreUpdate",1.0f,0.5f);
    }

    // Update is called once per frame
    void Update () {
        scoreText.text = "Score: " + score;
    }

    void scoreUpdate(){
        if(gameOver == false){
            score += 1;
        }
    }

    public void gameOverActivated(){
        gameOver = true;
        foreach(Button button in buttons){
            button.gameObject.SetActive(true);
        }
    }

    public void Back1()
    {
        Application.LoadLevel("gameover1");
    }

    public void Back2()
    {
        Application.LoadLevel("gameover2");
    }

    public void Back3()
    {
        Application.LoadLevel("gameover3");
    }

    public void Retry1()
    {
        Application.LoadLevel("Level1");
    }
}

```

```

public void Retry2()
{
    Application.LoadLevel("Level2");
}

public void Retry3()
{
    Application.LoadLevel("Impossible");
}

public void Play()
{
    Application.LoadLevel("difficult");
}

public void Easy(){
    Application.LoadLevel("Level1");
}

public void Medium()
{
    Application.LoadLevel("Level2");
}

public void Impossible()
{
    Application.LoadLevel("Impossible");
}

public void Pause(){
    if(Time.timeScale == 1){
        Time.timeScale = 0;
    }
    else if(Time.timeScale == 0){
        Time.timeScale = 1;
    }
}

public void MainMenu(){
    Application.LoadLevel("menuScene");
}

public void Exit(){
    Application.Quit();
}

```

Gambar 20. Scripts UiManager

Di antara semua script, UiManager menjadi pusat yang menghubungkan semua script tersebut. Di UiManager, konsep game over dan tingkat kesulitan dibuat, serta semua scene dihubungkan menjadi satu kesatuan. Tanpa UiManager, tidak mungkin beralih ke scene selanjutnya atau lainnya. Itulah kenapa UiManager sangat penting dan rawan sekali terjadi error.

E. Folder Sprites

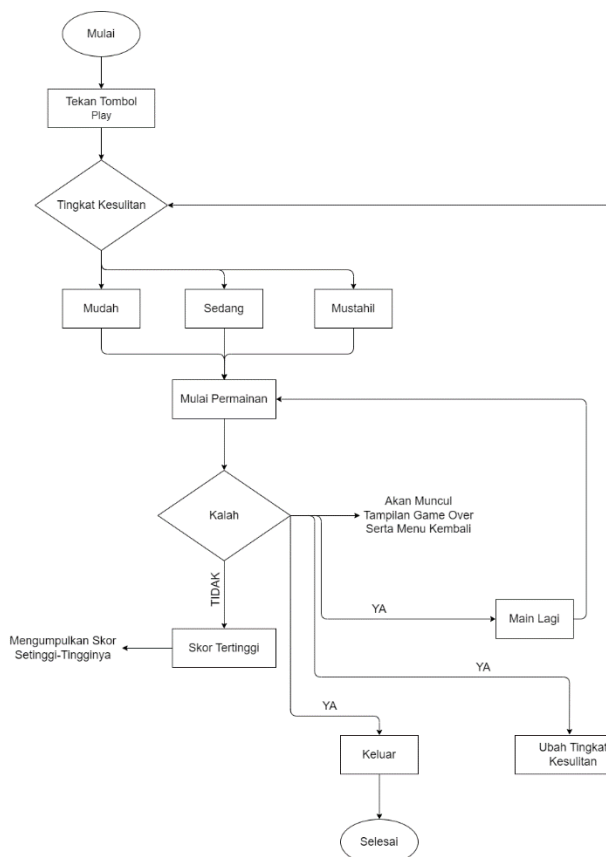


Gambar 21. Folder Sprites

Disinilah semua gambar-gambar yang kalian lihat didalam game. Folder Sprites menjadi Folder yang tak kalah penting. Karena tanpa adanya Folder Sprites kalian tidak akan pernah melihat Gambar ataupun Tampilan

menu pada saat kalian memainkan Game OtoLaju.

F. Flowchart Game OtoLaju



Gambar 22. Flowchart Game OtoLaju

4. KESIMPULAN

Berdasarkan hasil pembahasan dalam merancang game Oto Laju, digunakan software engine Unity 2018 dan bahasa pemrograman C#. Beberapa kali uji coba telah dilakukan untuk memastikan game ini berjalan dengan baik di Android maupun di PC.

DAFTAR PUSTAKA

P. M. A. WP, F. S. Wahyuni, and A. F. Setiawan, "GAME EDUKASI 'PENGENALAN TRANSPORTASI' 2D BERBASIS ANDROID," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, pp. 2136-2143, 2023. doi: <https://doi.org/10.33365/jti.v18i1.3407>.

[1] K. F. Hakim, D. Pasha, dan Q. J. Adrian, "Desain 2D Game Platform Petualangan Gajah Berbasis Android," *JURNAL INFORMATIKA DAN REKAYASA PERANGKAT LUNAK (JATIKA)*, vol. 4, pp. 41-46, 2023. <https://doi.org/10.33365/jatika.v4i1.2462>.

[2] M. G. Wellson and W. T. Atmojo, "IMPLEMENTASI METODE GDLC PADA

GAME TAXI RUSH MENGGUNAKAN UNITY ENGINE," *JURNAL TEKNOINFO*, vol. 18, pp. 201-214, 2024. DOI: <https://doi.org/10.33365/jti.v18i1.3407>.

[3] M. K. Nisfi, I. F. Astuti, and D. M. Khairina, "Membangun game adventure quiz 2D khas daerah Kalimantan Timur: 'Bejalanan'," in **Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi**, vol. 2, pp. 91-95, 2017, doi: <https://core.ac.uk/download/pdf/268075052.pdf>.

[4] M. Budiwansyah and Malabay, "Pembuatan Game Zombie Smasher dengan Unity berbasis Android," *Jurnal IKRAITH-INFORMATIKA*, vol. 7, pp. 116-125, 2023. [Online]. Available: <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive>.

[5] B. A. Wicaksono, A. Mahmudi, dan K. Auliasari, "Perancangan game Alien Warfare 2D menggunakan metode Finite State Machine (FSM) berbasis Android," *Jurnal Mahasiswa Teknik Informatika*, vol. 7, pp. 2945-2951, 2023. doi: <https://doi.org/10.36040/jati.v7i5.7555>

[6] R. Sussi, R. M. Prasojoe, N. Fitriyanti, and K. M. Shihab, "Pembuatan Game Online BoMClean sebagai Media Pembelajaran Kebersihan Lingkungan," *Jurnal Edukasi dan Penelitian Informatika*, vol. 5, pp. 113-118, 2019, doi: <http://dx.doi.org/10.26418/jp.v5i1.29874>.

[7] A. R. Wijaya, I. Kanedi, and R. Zulfiandry, "PEMBUATAN GAME THE LEGEND OF BENGKULU MENGGUNAKAN RPG MAKER VX ACE," *Jurnal Media Infotama*, vol. 19, pp. 69-78, 2023, doi: <https://doi.org/10.37676/jmi.v19i1.3807>.

[8] N. Nasution, F. B. Nasution, and M. A. Hasan, "PKM Pelatihan Pembuatan Game Menggunakan Unity untuk Siswa SMK di Kota Pekanbaru," *Journal of Computer Science Community Service*, vol. 2, pp. 117-127, 2022. DOI: <https://doi.org/10.31849/jcoscis.v2i2.9151>.

[9] K. Rahmadi, M. Fitriyasyah, and M. Fitriyasyah, "PENERAPAN METODE ON TRIGGER PADA APLIKASI GAME 2D ANDROID," *Informatics, Science and Technologies Journal*, vol. 10, pp. 1-5, 2022.

