

BAB IV

HASIL DAN PEMBAHASAN

4.1 Penerapan Interaksi Sepeda Dengan *Virtual Reality*

Game “Simulator Sepeda VR” merupakan game 3D yang menggabungkan antara aktivitas manusia di kehidupan nyata dan interaksinya di dalam game. Dengan dipakainya VR sebagai salah satu fitur dalam game ini memberikan pengalaman bermain seperti layaknya di dunia virtual. Dan dengan menerapkan VR untuk game simulasi bersepeda ini akan menambah tingkat imersif (*immersive*) game tersebut.

Untuk menerapkan interaksi antara sepeda dengan VR adalah dengan menggunakan perangkat berbasis Arduino. Perangkat berbasis Arduino dihubungkan dengan sensor magnetik yang diletakkan pada roda belakang sepeda, dan menggunakan modul joystick analog yang diletakkan di belakang setang sepeda. Hal ini memungkinkan Arduino untuk membaca setiap perputaran roda dan sudut perputaran setang, yang nantinya akan dikonversi menjadi input dan dikirim ke dalam game VR melalui transmisi Bluetooth.

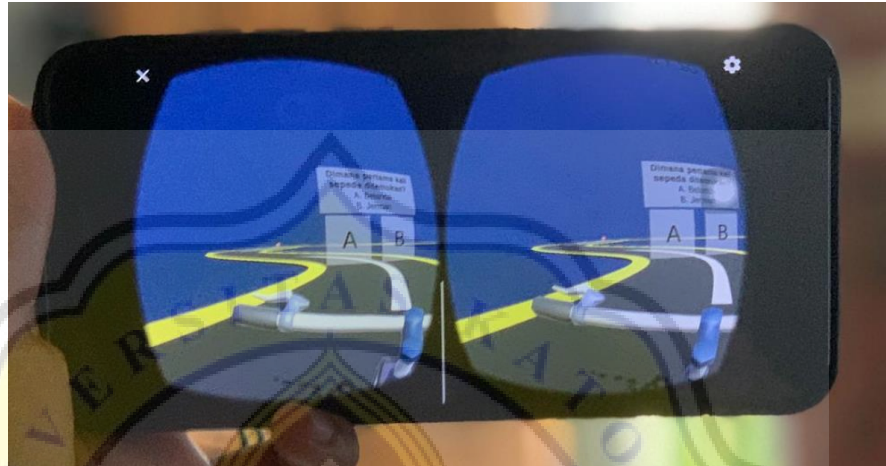
a. Perangkat VR



Gambar 4.1 Perangkat VR untuk smartphone

Gambar 4.1 merupakan gambar perangkat VR yang digunakan untuk mendukung penggunaan konsep *virtual reality* pada game ini.

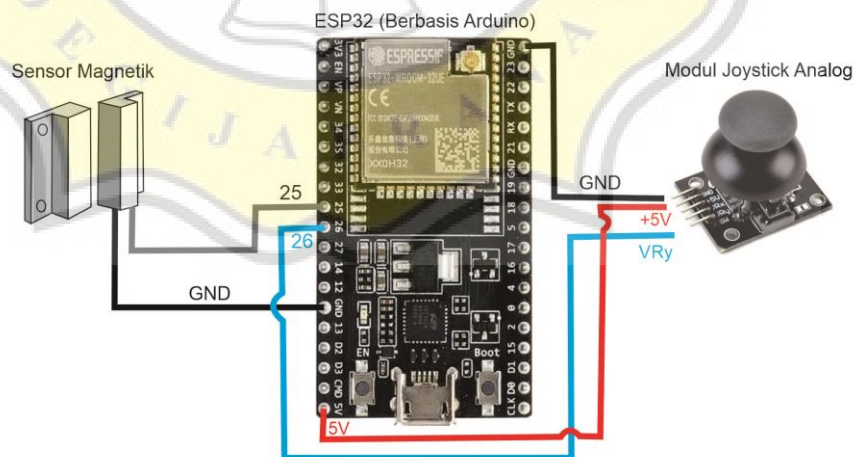
b. Tampilan VR



Gambar 4.2 Tampilan game dalam format VR

Gambar 4.2 merupakan *screenshot* dari tampilan game, tampilan dengan format seperti pada gambar dibutuhkan untuk nantinya dapat dilihat menggunakan perangkat VR.

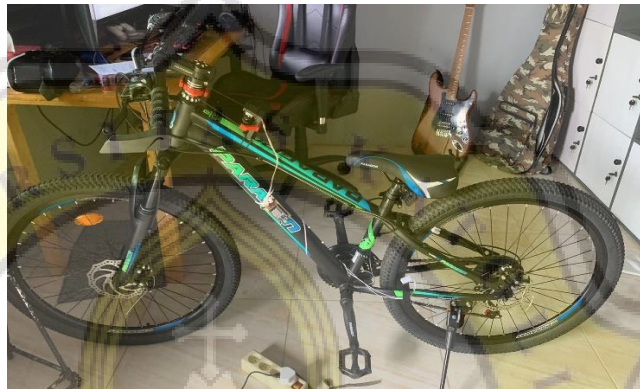
c. Rangkaian Sensor dan Perangkat ESP32



Gambar 4.3 Rangkaian Sensor dan Perangkat ESP32

Gambar 4.3 adalah susunan rangkaian untuk menghubungkan sensor-sensor yang dibutuhkan dan disambungkan ke perangkat ESP32. Penggunaan pin akan mempengaruhi variabel program yang akan ditanam di perangkat ESP32. Rangkaian ini nantinya akan terhubung menggunakan bluetooth dengan ponsel android yang berisi game “Simulator Sepeda VR”.

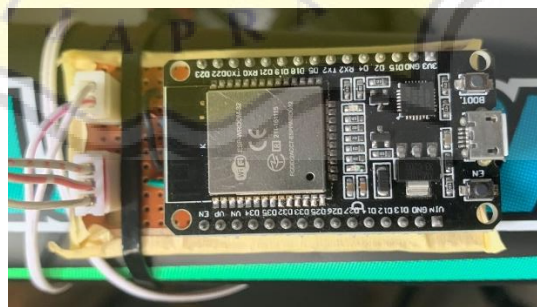
d. Sepeda



Gambar 4.4 Perangkat sepeda

Gambar 4.4 merupakan perangkat sepeda yang digunakan manusia (user) untuk memainkan game ini. Sepeda dipasang perangkat berbasis arduino untuk mengubah putaran roda dari sepeda menjadi input di dalam game.

e. Penerapan Pada Sepeda



Gambar 4.5 Perangkat berbasis Arduino

Gambar 4.5 merupakan perangkat berbasis arduino yang dipasang di perangkat sepeda untuk membaca putaran roda dan

dikonversi menjadi input. Hal ini dapat terwujud dengan adanya sensor magnet untuk membaca setiap putaran roda.



Gambar 4.6 Sensor magnetik yang sudah terpasang

Gambar 4.6 merupakan sensor magnet yang berfungsi untuk membaca setiap putaran roda. Dengan adanya sensor ini, maka dalam setiap putaran roda akan dibaca selayaknya tombol yang ditekan. Sehingga dapat dijadikan sebagai input dalam game.



Gambar 4.7 Modul analog joystick yang sudah terpasang

Gambar 4.7 menunjukkan modul analog joystick yang berfungsi untuk membaca putaran setang sepeda sehingga dapat dikonversi

menjadi angka yang nantinya bisa diproses untuk menjadi input di dalam game.

```
#include <Arduino.h>
#include <BleGamepad.h>

#define BUTTONPIN 25

BleGamepad bleGamepad("BicycleController");

int previousButton1State = HIGH;

const int potPin = 26;
const int numberOfPotSamples = 5;
const int delayBetweenSamples = 4;
const int delayBetweenHIDReports = 5;

void setup()
{
  pinMode(BUTTONPIN, INPUT_PULLUP);
  bleGamepad.begin();
}

void loop()
{
  if (bleGamepad.isConnected())
  {
    // Button
    int currentButton1State = digitalRead(BUTTONPIN);

    if (currentButton1State != previousButton1State)
    {
      if (currentButton1State == LOW)
      {
        bleGamepad.press(BUTTON_1);
      }
      else
      {
        bleGamepad.release(BUTTON_1);
      }
    }
    previousButton1State = currentButton1State;

    // Axis
    int potValues[numberOfPotSamples];
    int potValue = 0;

    // Populate readings
    for (int i = 0; i < numberOfPotSamples; i++)
    {
      potValues[i] = analogRead(potPin);
      potValue += potValues[i];
      delay(delayBetweenSamples);
    }

    potValue = potValue / numberOfPotSamples;

    int adjustedValue = map(potValue, 0, 4095, 32737, -32737);

    bleGamepad.setX(adjustedValue);
    delay(delayBetweenHIDReports);
  }
}
```

Gambar 4.8 Script untuk Arduino

Gambar 4.9 Lanjutan script untuk Arduino

Gambar 4.8 dan 4.9 adalah script yang digunakan oleh perangkat arduino untuk membaca *value* dari sensor magnet dan modul analog joystick dan mengubahnya menjadi input yang dapat dibaca oleh Unity.

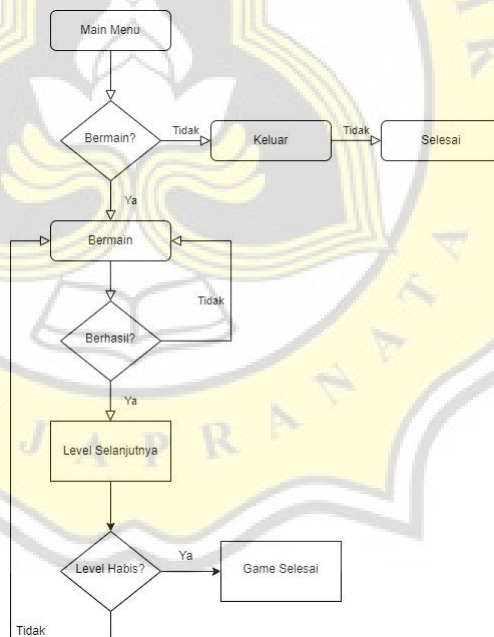
4.2 Mewujudkan Rancang Bangun “Game Simulator Bersepeda Berbasis VR” dengan Unity Engine

Untuk mewujudkan rancang bangun “Game Simulator Bersepeda Berbasis VR”, penggunaan *Unity Engine* sangatlah tepat. Karena *Unity Engine* merupakan *game engine* yang mendukung tampilan 2D maupun 3D. Sebelum nantinya dikembangkan di dalam *Unity Engine* ada beberapa rancangan yang harus dipersiapkan, diantaranya:

a. Game Design

“Game Simulator Bersepeda Berbasis VR” adalah game yang memiliki 4 level di dalamnya, yaitu “Trivia”, “Collect Trophy”, “Time Rush”, dan “Time Trial”. Setiap level akan menyajikan cara bermain yang berbeda, dan pengalaman yang berbeda.

Game flow utama adalah pemain menggerakkan sepeda dan melewati tantangan di setiap level dan berlanjut ke level yang lain.



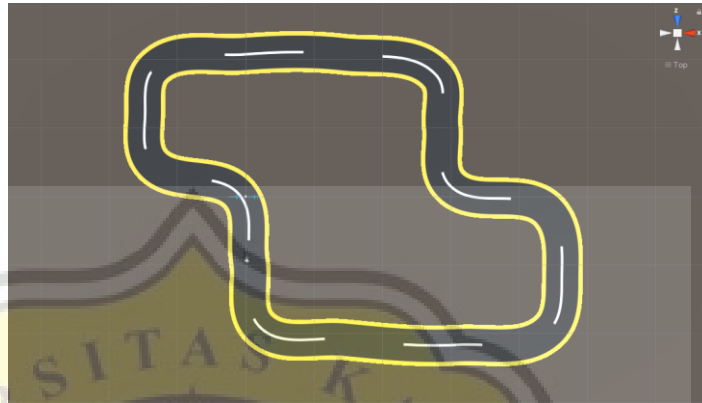
Gambar 4.10 Game Flow

b. World Design

Karena “Game Simulator Bersepeda Berbasis VR” menyajikan 4 level yang berbeda, maka rute atau jalan yang ditampilkan juga berbeda. Berikut

adalah bentuk rute atau jalan dari masing-masing level yang dibuat menggunakan “Path Tool” dari Unity.

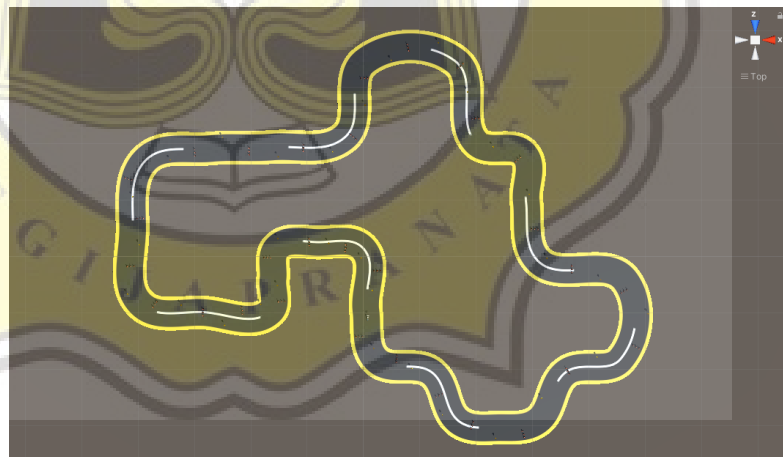
i. Rute Level 1 “Trivia”



Gambar 4.11 Rute untuk level 1 "Trivia"

Gambar 4.11 merupakan rute untuk level 1 “Trivia”, bentuk rute sederhana sehingga pemain dapat berfokus pada pertanyaan yang disajikan di level 1.

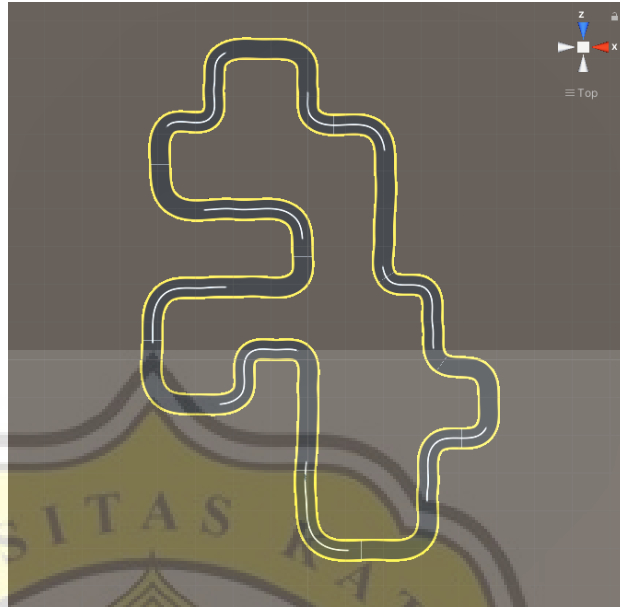
ii. Rute Level 2 “Collect Trophy”



Gambar 4.12 Rute untuk level 2 "Collect Trophy"

Gambar 4.12 merupakan rute untuk level 2 “Collect Trophy”, bentuk rute sedikit lebih menantang sehingga pemain dapat mencoba manuver yang bisa dilakukan dan juga untuk mengumpulkan “Trophy”

iii. Rute Level 3 “Time Rush”



Gambar 4.13 Rute untuk level 3 "Time Rush"

Gambar 4.13 merupakan rute untuk level 3 “Time Rush”, bentuk rute banyak belokan sehingga memberikan tantangan untuk pemain dalam melewati rute dengan batas waktu yang ditetapkan.

iv. Rute Level 4 “Time Trial”



Gambar 4.14 Rute untuk level 4 "Time Trial"

Gambar 4.14 merupakan rute untuk level 4 “Time Trial”, bentuk rute mirip dengan rute di level 3, namun dengan cara bermain yang berbeda, disini akan dicatat waktu tercepat pemain dalam menyelesaikan rute dari titik start hingga titik finish.

c. Implementasi Mekanik Game

Ketika game design dan world design sudah ada, saatnya untuk implementasi mekanik game. Ada beberapa mekanik dan fitur yang perlu dibuat diantaranya adalah *Player Controller*, *Bike Controller*, *Level Manager*, dan implementasi Google Cardboard SDK.

i. *Player Controller*

Player controller disini bertujuan untuk menerima input dari pemain dan menyampaikan datanya dalam bentuk *boolean* dan *Vector2* menuju ke *Bike Controller*.

```
[RequireComponent(typeof(BikeController))]
@ Unity Script | 0 references
public class PlayerController : MonoBehaviour
{
    BikeController bikeController;
    public float wheelRadius = 0.34f;
    private bool hasPassed;
    private float timeBtwPassings;
    private float inputValues = 0;

    @ Unity Message | 0 references
    private void Start()
    {
        bikeController = GetComponent<BikeController>();
    }

    @ Unity Message | 0 references
    private void Update()
    {
        bikeController.setRotationAndSpeed(new Vector2(Input.GetAxis("Horizontal")/4, GetThrottle()));
    }

    1 reference
    private float GetThrottle() {
        if (Input.GetButton("Throttle") && hasPassed == false) {
            hasPassed = true;
            inputValues = wheelRadius * 2 * Mathf.PI / timeBtwPassings;
            timeBtwPassings = 0f;
        }
        else if (!Input.GetButton("Throttle")) {
            hasPassed = false;
        }

        if (timeBtwPassings > 5) {
            inputValues = 0f;
        }
        timeBtwPassings += Time.deltaTime;

        return inputValues;
    }
}
```

Gambar 4.15 Script *PlayerController*

Gambar 4.15 adalah script untuk membaca input dari pemain melalui fungsi *GetThrottle()* dan mengirimnya ke

Bike Controller menggunakan `setRotationAndSpeed` yang terdapat pada fungsi `Update()`.

ii. *Bike Controller*

Bike controller disini adalah mekanik yang berguna untuk menggerakkan sepeda di dalam game. Dalam game ini, karena karakter menggunakan sepeda, maka yang dibutuhkan adalah sepeda dapat bergerak maju dan berbelok ke kanan atau ke kiri. Oleh karena itu, bike controller yang dibuat akan berbasis komponen *Wheel Collider*.

```
public class BikeController : MonoBehaviour
{
    [Header ("GameObjects")]
    public GameObject rearWheel;
    public GameObject frontWheel;
    public GameObject crank;
    public GameObject pedall;
    public GameObject pedalR;
    public GameObject fork;
    public Transform centerOfMass;

    [Header("Values")]
    public float oneRotationSpeed = 2.7f;
    public float crankMultiplier = 2f;
    public List<AxleInfo> axleInfos;
    public float maxMotorTorque;
    public float maxSteeringAngle;
    [Range(0,1)]
    public float relativeLeanAmount = 0f;
    public Transform leftWheels;
    public Transform rightWheels;

    private float rotationValue = 0f;
    private float maxSpeed = 0f;
    public float rotSpeed = 10;

    private Vector3[] wheelPositions;
    private Rigidbody rb;
    private Quaternion startForkRot;
    private Vector3 upDirection = Vector3.up;

    @ Unity Message | 0 references
    void Start()
    {
        rb = GetComponent<Rigidbody>();
        rb.centerOfMass = centerOfMass.localPosition;
        startForkRot = fork.transform.localRotation;
        wheelPositions = new Vector3[axleInfos.Count];
        for (int i = 0; i < axleInfos.Count; i++)
        {
            wheelPositions[i] = axleInfos[i].wheel.center;
        }
    }

    @ Unity Message | 0 references
    void Update()
    {
        RotateMeshes();
        RotateFork();
    }
}
```

Gambar 4.16 Script *BikeController*

```

@ Unity Message | 0 references
public void FixedUpdate()
{
    ApplyWheelForce();
    RotateStraight();
}

1 reference
void RotateMeshes()
{
    RotateObject(crank, 1);
    RotateObject(pedall, -1);
    RotateObject(pedalR, -1);
    RotateObject(rearWheel, crankMultiplier);
    RotateObject(frontWheel, crankMultiplier);
}

1 reference
void RotateFork()
{
    fork.transform.localRotation = startForkRot;
    fork.transform.RotateAround(fork.transform.position, fork.transform.up, maxSteeringAngle * rotationValue);
}

0 references
void Lean()
{
    upDirection = Vector3.Normalize(Vector3.up + transform.right * maxSteeringAngle * relativeLeanAmount * rotationValue * rb.velocity.magnitude / 100);
}

1 reference
void ApplyWheelForce()
{
    float motor = maxMotorTorque;
    float steering = maxSteeringAngle * rotationValue;

    leftWheels.localPosition = - Vector3.up * relativeLeanAmount * rotationValue * rb.velocity.magnitude;
    rightWheels.localPosition = Vector3.up * relativeLeanAmount * rotationValue * rb.velocity.magnitude;

    foreach (AxleInfo axleInfo in axleInfos)
    {
        if (axleInfo.steering)
        {
            axleInfo.wheel.steerAngle = steering;
        }
        if (axleInfo.motor && rb.velocity.magnitude < maxSpeed)
        {
            axleInfo.wheel.motorTorque = motor;
        }
        else if (axleInfo.motor)
        {
            axleInfo.wheel.motorTorque = 0;
        }
    }
}

```

Gambar 4.17 Lanjutan script BikeController

```

1 reference
public void setRotationAndSpeed(Vector2 value)
{
    rotationValue = value.x;
    maxSpeed = value.y;
}

1 reference
private void RotateStraight()
{
    Vector3 axis = Vector3.Cross(transform.forward, Vector3.Cross(upDirection, transform.forward));
    float angle = Vector3.Angle(transform.up, axis);
    Vector3 direction = Vector3.Cross(transform.up, axis);
    if (angle > 0.001)
    {
        rb.AddTorque(direction * angle * 100);
    }
}

5 references
void RotateObject(GameObject obj, float multiplier)
{
    obj.transform.Rotate(Time.deltaTime * rb.velocity.magnitude * (360f / oneRotationSpeed) * multiplier, 0, 0);
}

[System.Serializable]
2 references
public class AxleInfo
{
    public WheelCollider wheel;
    public bool motor;
    public bool steering;
}

```

Gambar 4.18 Lanjutan script Bike Controller

Gambar 4.16 – 4.18 adalah script yang berfungsi untuk membaca data input yang telah dikirim oleh PlayerController dan menjadikan data tersebut sebagai acuan untuk menggerakkan objek sepeda dalam game.

iii. *Level Manager*

Level manager adalah kumpulan dari beberapa script untuk mengatur bagaimana setiap level disajikan, karena ada 4 (empat) level yang berbeda, maka akan ada 4 script yang harus dibuat, namun karena adanya kemiripan antara level 1 dan level 2, maka script bisa disederhanakan menjadi 1 (satu) script saja. Akhirnya hanya ada 3 script yang harus dibuat untuk mengatur level, diantaranya adalah sebagai berikut.

i. *PointManager*

Dikarenakan kemiripan dari level 1 dan level 2 yang sama-sama mengumpulkan poin, maka hanya akan ada 1 (satu) script.

```
Unity Script | 6 references
public class PointManager : MonoBehaviour
{
    public int score;
    public int maxScore;
    public Text scoreText;

    public bool useHealth;
    public int health = 3;
    public Text healthText;

    public UnityEvent onWin;
    public UnityEvent onLose;

    Unity Message | 0 references
    private void Update()
    {
        scoreText.text = $"{score} / {maxScore}";

        if (score >= maxScore)
        {
            onWin.Invoke();
        }

        if (score < 0)
        {
            onLose.Invoke();
        }

        if (useHealth)
        {
            if (health <= 0)
            {
                onLose.Invoke();
            }

            healthText.text = $"Life: {health}";
        }
    }
}
```

Gambar 4.19 Script *PointManager*

```

2 references
public void IncreasePoint()
{
    score++;
}

1 reference
public void DecreasePoint()
{
    score--;
}

1 reference
public void DecreaseHealth()
{
    health--;
}

0 references
public void LoadSceneByIndex(int index)
{
    SceneManager.LoadScene(index);
}

```

Gambar 4.20 Lanjutan script PointManager

Gambar 4.19 – 4.20 merupakan manager untuk mengatur level 1 dan level 2. Dalam script ini terdapat beberapa fungsi untuk menambah dan mengurangi point, serta mengurangi health untuk digunakan di level 2.

ii. TimeRushManager

```

public class TimeRushManager : MonoBehaviour
{
    public float timeAmount = 5f;
    public Text timerText;

    public UnityEvent onWin;
    public UnityEvent onLose;

    private float currentTime;
    private bool isStarted;

    [Unity Message | 0 references]
    private void Start()
    {
        isStarted = false;
        currentTime = timeAmount;
        timerText.text = currentTime.ToString();
    }

    [Unity Message | 0 references]
    void Update()
    {
        if (isStarted)
        {
            if (currentTime > 0)
            {
                currentTime -= Time.deltaTime;
                timerText.text = currentTime.ToString("0.0");
            }
            else
            {
                currentTime = 0f;
                Debug.LogError("Time Out!");
                isStarted = false;
                onLose.Invoke();
            }
        }
    }
}

```

Gambar 4.21 Script TimeRushManager

```

1 reference
public void CrossStart()
{
    isStarted = true;
}

1 reference
public void CrossCheckpoint()
{
    currentTime += timeAmount;
}

1 reference
public void Finish()
{
    onWin.Invoke();
}

0 references
public void LoadSceneByIndex(int index)
{
    SceneManager.LoadScene(index);
}
}

```

Gambar 4.22 Lanjutan script TimeRushManager

Gambar 4.21 dan 4.22 adalah script yang mengatur level 3. Pada script ini, terdapat fungsi yang akan menambah waktu ketika player melewati *Checkpoint*. Serta fungsi yang mendeteksi waktu habis sebelum *Checkpoint*.

iii. TimeTrialManager

```

public class TimeTrialManager : MonoBehaviour {
    public Text timerText;

    public UnityEvent onWin;
    public UnityEvent onLose;

    private float currentTime;
    private bool isStarted;

    @Unity Message | 0 references
    private void Start() {
        isStarted = false;
        currentTime = 0;
    }

    @Unity Message | 0 references
    private void Update() {
        if (isStarted) {
            currentTime += Time.deltaTime;
            timerText.text = TimeSpan.FromSeconds(currentTime).ToString(@"mm\:ss\.:ff");
        }
    }

    1 reference
    public void OnCrossStart() {
        isStarted = true;
    }

    1 reference
    public void OnCrossFinish() {
        isStarted = false;
        onWin.Invoke();
    }

    0 references
    public void SaveBestTime() {
        Debug.Log(currentTime);
        float currentBestTime = PlayerPrefs.GetFloat("BestTime", 180f);

        if (currentTime < currentBestTime) {
            PlayerPrefs.SetFloat("BestTime", currentTime);
        }
    }

    0 references
    public void LoadSceneByIndex(int index) {
        SceneManager.LoadScene(index);
    }
}

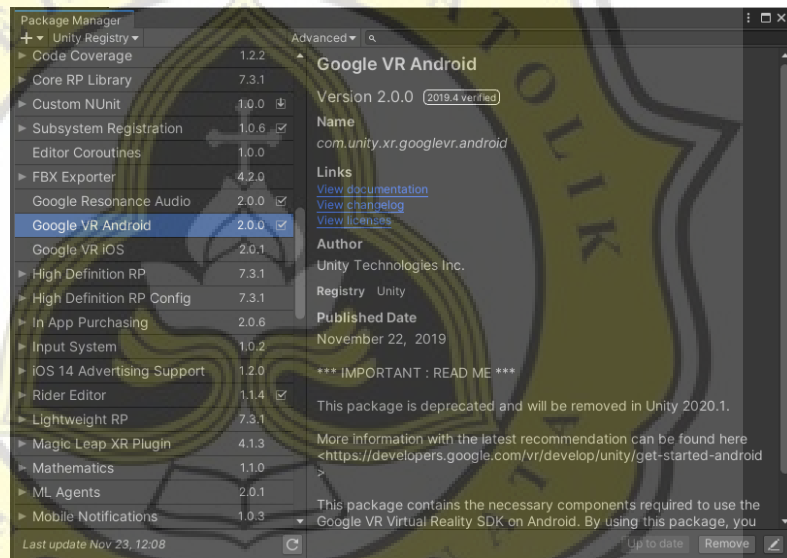
```

Gambar 4.23 Script TimeTrialManager

Gambar 4.23 adalah script yang mengatur level 4. Pada script ini, terdapat fungsi yang akan memulai waktu ketika player melewati *Start*. Serta fungsi yang mencatat waktu saat mencapai *Finish*.

iv. Implementasi *Google Cardboard SDK*

Dalam mewujudkan konsep VR dalam game ini, maka harus ada implementasi *Google Cardboard SDK*. Hal terpenting dalam implementasi *Google Cardboard SDK* di dalam Unity adalah dengan menambahkan *package* *Google VR Android* dari *package manager*.

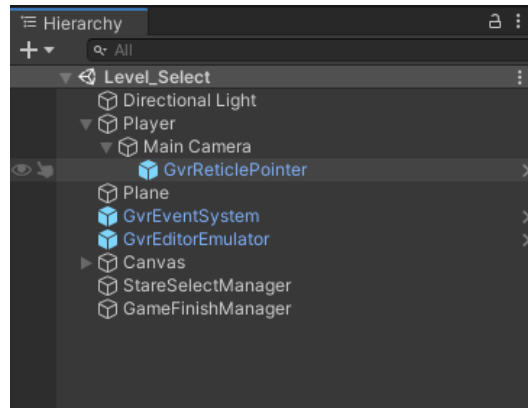


Gambar 4.24 Penambahan package *Google VR Android*

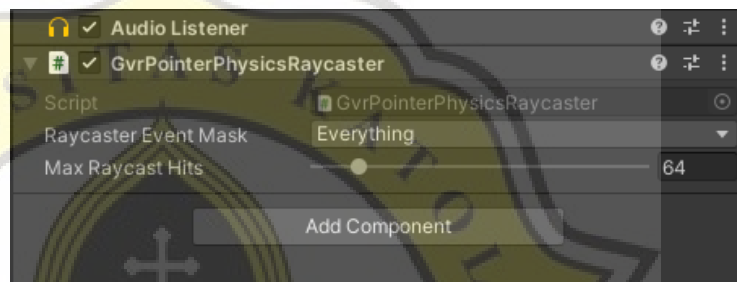
Gambar 4.24 menunjukkan langkah penambahan *Google VR Android* dari *package manager* Unity.

Ketika *Google Cardboard SDK* atau *Google VR Android* sudah ditambahkan, maka akan ada folder *Google VR* beserta beberapa *prefabs*.

Untuk menerapkan *Google Cardboard VR* atau *Google VR Android* ke dalam game yang sudah dibuat, adalah dengan memasukkan beberapa *prefabs* dan *component* ke dalam *scene* game.



Gambar 4.25 Prefabs di dalam Scene untuk implementasi VR



Gambar 4.26 Component di dalam GameObject MainCamera

Gambar 4.25 dan 4.26 menunjukkan beberapa prefabs dan component yang sudah dimasukkan ke dalam scene game dan juga ke dalam *GameObject*.

Ketika semua prefabs dan komponen telah dimasukkan ke dalam scene, maka scene tersebut sudah terimplementasi VR. Sehingga fitur Google VR sudah diterapkan ke dalam game.

4.3 Mengumpulkan respon pengguna ketika menggunakan “Game Simulator Bersepeda Berbasis *Virtual Reality*”

Untuk membuktikan bahwa Rancang Bangun “Game Simulator Bersepeda Berbasis VR” dapat memberikan pengalaman baru dan menarik, maka dilakukan pengujian dan pengumpulan data kepada pemain. Beberapa pertanyaan diberikan kepada pemain dan harus dijawab oleh pemain. Berikut pertanyaan dan jawaban dari pengujian dan pengumpulan data.

1. Apakah memainkan game "**Simulator Sepeda VR**" memberikan **pengalaman baru** dalam menggunakan sepeda? Berikan alasan!
(*Performance Expectancy*)

Kesimpulan untuk pertanyaan 1 adalah pengguna merasakan pengalaman baru dalam bersepeda setelah mencoba game "Simulator Sepeda VR".

Tabel 4.1 Jawaban pemain untuk pertanyaan 1

Nama	Pendapat
M Syahnas Lutfi	Ya ini merupakan pengalaman baru bermain game sepeda dengan menggunakan VR
Daniel Santosa	Ya, game ini memberikan pengalaman bersepeda di dalam ruangan menggunakan VR dan kedepannya saya berharap game ini akan terus dikembangkan sehingga pemain dapat bersepeda di spot spot bersepeda yang bagus di dunia nyata.
Millen	Ya, Dengan menggunakan VR saya merasa lingkungan disekitar saya berubah dan pengalaman bersepeda menjadi lebih immersive ditambah lagi saya tidak perlu keluar rumah untuk berolahraga
Edbert Liniarto	Ya, karena selain pengalaman yang baru bisa juga untuk berolahraga sepeda walaupun tidak seperti kalau bersepeda di jalanan
Glenn Hariyanto	Ya karena simulator sepeda VR merupakan kegiatan yang langka di temukan
Miftahuzzaka	Pasti, karena teknologi simulator sepeda cenderung lebih sedikit

(lanjutan tabel)

Nama	Pendapat
Syahnas Anjas	Ya ini memberikan saya pengalaman baru dalam bermain game VR
Jeffri Harjayanto	Ya, dengan memainkan game ini saya bisa merasakan bersepeda di dunia maya
Sigit	Iya, karena seperti mengendarai sepeda "in real life" namun tetap di dalam rumah
Hugo Apy	Iya, karena belum pernah memakai VR dalam bersepeda

2. Apakah memainkan game "**Simulator Sepeda VR**" memberikan **variasi tantangan lebih dari bersepeda biasa**? Berikan alasan!
(*Performance Expectancy*)

Kesimpulan untuk pertanyaan 2 adalah pengguna setuju dengan pernyataan bahwa game "Simulator Sepeda VR" memberikan variasi tantangan yang lebih dibandingkan dengan bersepeda secara konvensional.

Tabel 4.2 Jawaban pemain untuk pertanyaan 2

Nama	Pendapat
M Syahnas Lutfi	Dengan beberapa tantangan dan pertanyaan di dalam game Simulator Sepeda VR ini membuat game menjadi menarik untuk dimainkan
Daniel Santosa	Ya, game ini membuat pemain berolahraga dan belajar di waktu yang sama
Millen	Ya, pada game Simulator sepeda VR ini terdapat beberapa stage yang saya coba dari situ saya sudah mendapatkan pengalaman tambahan selain berolahraga tapi juga mendapatkan trivia hal hal mengenai sepeda

(lanjutan tabel)

Nama	Pendapat
Edbert Liniarto	Iya, karena unik dan tantangan di setiap level berbeda-beda
Glenn Hariyanto	ya karena saya dapat merasakan hal biasa tetapi digital
Miftahuzzaka	Pastinya, bisa merasakan adventure tanpa harus memikirkan resiko terhadap diri dan pasti lebih safety tapi tidak lupa dengan keseruan adventure
Syahnas Anjas	Ini sangat memberikan tantangan baru dari bermain sepeda biasa
Jeffri Harjayanto	Ya, karena di simulator vr ini saya bisa mengasah kemampuan bersepeda saya di dunia maya tanpa perlu tempat yang besar
Sigit	Iya, lebih menantang. Karena ada beberapa tantangan yang tidak memungkinkan di dunia nyata.
Hugo Apriy	Iya, tantangan dalam game cukup bervariasi dan cukup unik

3. Apakah game "**Simulator Sepeda VR**" mudah untuk **dimainkan**? Berikan alasan! (*Effort Expectancy*)

Kesimpulan untuk pertanyaan 3 adalah pengguna merasa mudah dalam memainkan game "Simulator Sepeda VR".

Tabel 4.3 Jawaban pemain untuk pertanyaan 3

Nama	Pendapat
M Syahnas Lutfi	Sangat mudah dimainkan, seperti menggunakan sepeda biasanya
Daniel Santosa	ya sangat mudah untuk dimainkan dan pertanyaan di game sendiri cukup menantang

(lanjutan tabel)

Nama	Pendapat
Millen	Dapat terbilang cukup mudah karena memang seperti naik sepeda seperti biasa, akan tetapi untuk awalnya memang harus membiasakan terlebih dahulu, perasaan takut kehilangan keseimbangan ketika ingin belok
Edbert Liniarto	Mudah, karena hanya perlu duduk mengayuh sepedanya saja, dan mengarahkan playernya untuk menjawab pertanyaan dan menghindari rintangan
Glenn Hariyanto	iya karena tinggal mengenakan vr lalu melakukan aktivitas tersebut
Miftahuzzaka	Bisa dibilang mudah, tergantung rute atau adventure yang di pilih
Syahnas Anjas	Sangat mudah untuk dimainkan
Jeffri Harjayanto	Iya, rig sepeda benar-benar kuat dan tidak mudah jatuh, kontrol juga seperti mengendarai sepeda biasa
Sigit	Iya, karena hanya tinggal menggunakan sepeda seperti biasanya
Hugo Apriy	Iya, pengoperasian sangat mudah, hanya tinggal gowes dan belok

4. Apakah anda **mudah menyelesaikan** game "**Simulator Sepeda VR**"? Berikan alasan! (*Effort Expectancy*)

Kesimpulan untuk pertanyaan 4 adalah pengguna dapat melewati level yang ada di dalam game "Simulator Sepeda VR".

Tabel 4.4 Jawaban pemain untuk pertanyaan 4

Nama	Pendapat
M Syahnas Lutfi	Mudah untuk menyelesaikan game ini

(lanjutan tabel)

Nama	Pendapat
Daniel Santosa	mudah untuk dimainkan dan menyenangkan
Millen	cukup mudah, diluar dari kurang terbiasa dengan setup permainan menyelesaikannya dapat dibilang cukup mudah, mungkin kalo sudah terbiasa menyelesaikannya bisa dalam hitungan dibawah 1 menit.
Edbert Liniarto	Mudah, karena player bisa mengontrol sepedanya sesuai selera sendiri
Glenn Hariyanto	iya karena mudah untuk memahami sistem dari game tersebut
Miftahuzzaka	Awalnya saya pikir akan susah, setelah dicoba ternyata mudah
Syahnas Anjas	Mudah untuk menyelesaikan misi yang terdapat di dalam game ini
Jeffri Harjayanto	sedikit susah, karena memang saya belum terbiasa menggunakan VR
Sigit	Iya, semua tantangannya tidak terlalu sulit
Hugo Apy	Iya, karena pengoperasian mudah, menyelesaikan game juga tidak terlalu susah

5. Apakah **cara memainkan** game "**Simulator Sepeda VR**" menarik? Berikan alasan! (*Hedonic Motivation*)

Kesimpulan untuk pertanyaan 5 adalah pengguna merasa cara memainkan game "Simulator Sepeda VR" menarik, karena penggunaan VR dalam bersepeda bisa dikatakan baru.

Tabel 4.5 Jawaban pemain untuk pertanyaan 5

Nama	Pendapat
M Syahnas Lutfi	Menarik sekali, dengan beberapa fitur yang ada di dalam game ini sangat menarik

(lanjutan tabel)

Nama	Pendapat
Daniel Santosa	sangat menarik, bisa bersepeda tanpa harus peduli cuaca di luar ruangan
Millen	sangat menarik, karena dengan mengkombinasikan Virtual Reality dan static bicycle terasa kalau kita seperti memiliki kontroler berbentuk sepeda, tentu saja ini juga berpengaruh pada gameplay yang dimana saya merasa kalau saya benar benar bersepeda
Edbert Liniarto	Menarik, selain player bisa sedikit berolahraga tetapi juga bisa melatih kemampuan berpikir disaat menghindari rintangan, mengambil piala untuk scorenya , dan menjawab pertanyaan dekat dengan obstaclenya
Glenn Hariyanto	iya karena jarang terjadi di "daily basis"
Miftahuzzaka	Menarik, karena merasakan sensasi berkendara dan adventure
Syahnas Anjas	Sangat menarik untuk memainkan game ini karena sebuah pengalaman baru bermain game
Jeffri Harjayanto	Sangat menarik, menggabungkan benda fisik dan VR membuat hal menjadi lebih menarik karena biasanya VR dimainkan menggunakan controller dan kondisi fisik/badan static cuma pandangan saja yang berubah tp dengan Simulator Sepeda VR saya merasa benar-benar seperti menaiki sepeda
Sigit	Iya, karena ada hal2 baru untuk dimainkan
Hugo Apry	Menarik, karena sebelumnya tidak terpikirkan bahwa bermain sepeda bisa diberi inovasi seperti ini

6. Apakah **level** yang terdapat pada game "**Simulator Sepeda VR**" menarik? Berikan alasan! (*Hedonic Motivation*)

Kesimpulan untuk pertanyaan 6 adalah pengguna tertarik dengan level yang ada dalam game "Simulator Sepeda VR" karena tidak terlalu sulit dan cukup bervariasi. Beberapa pengguna memberi saran untuk menambah level.

Tabel 4.6 Jawaban pemain untuk pertanyaan 6

Nama	Pendapat
M Syahnas Lutfi	Menarik di setiap level mempunyai tantangan yang berbeda beda
Daniel Santosa	cukup menarik, dari mulai tingkat kesulitan hingga design game cukup bagus
Millen	untuk model levelnya cukup menarik akan tetapi versi sekarang masih sangat kosong.
Edbert Liniarto	Menarik, karena setiap level mempunyai cara memainkan yang berbeda
Glenn Hariyanto	iya karena terdapat banyak perbedaan di setiap levelnya
Miftahuzzaka	Menarik, karena setiap level memberikan petualangan masing masing
Syahnas Anjas	Level yang berbeda setiap naik membuat game ini menjadi sangat menarik
Jeffri Harjayanto	menarik, untuk versi sekarang akan tetapi sangat diharapkan agar kedepannya ditambah dengan environment yang mendukung misal level dibuat seperti tanjak gunung atau jalan di taman agar tidak bosan
Sigit	Iya, karena tidak terlalu sulit dan lumayan bervariasi
Hugo Apry	Cukup menarik dan kalau bisa lebih banyak lagi levelnya

7. Jika memiliki akses untuk memainkan game "**Simulator Sepeda VR**", apakah kedepannya akan **terus mencoba memainkan?** Berikan alasan! (*Behavioral Intention*)

Kesimpulan untuk pertanyaan 7 adalah pengguna akan terus memainkan game "Simulator Sepeda VR" untuk waktu yang akan datang jika memiliki fasilitas yang diperlukan.

Tabel 4.7 Jawaban pemain untuk pertanyaan 7

Nama	Pendapat
M Syahnas Lutfi	Saya akan terus mencoba game ini
Daniel Santosa	ya,saya tidak terlalu suka bersepeda karena bersepeda di luar ruangan tergantung oleh cuaca dan kondisi jalur, dengan game ini saya bisa bersepeda kapan saja
Millen	Tentu saja, game-game seperti ini sangat cocok untuk saya yang jarang keluar rumah tapi ingin merasakan pengalaman bersepeda
Edbert Liniarto	Iya, karena sangat menarik dimainkan sambil berolahraga
Glenn Hariyanto Putra	iya karena saya ingin mencoba menaiki sepeda sedangkan tidak punya sepeda
Miftahuzzaka	Pasti, tidak perlu ke luar rumah soalnya
Syahnas Anjas	Ya saya akan terus memainkan game ini
Jeffri Harjayanto	iya, dapat melakukan bersepeda di dalam rumah merupakan hal yang menarik dan terlebih lagi tidak akan terkena panas karena di rumah
Sigit	Iya, karena menarik dan bisa sekalian berolahraga
Hugo Apy	Iya, apalagi jika tantangan bisa semakin ditambah

8. Jika memiliki akses untuk memainkan game "**Simulator Sepeda VR**", apakah kedepannya akan **terus menggunakannya sebagai sarana olahraga**? Berikan alasan! (*Behavioral Intention*)

Kesimpulan untuk pertanyaan 8 adalah pengguna setuju bahwa game "Simulator Sepeda VR" dapat digunakan sebagai sarana berolahraga alternatif.

Tabel 4.8 Jawaban pemain untuk pertanyaan 8

Nama	Pendapat
M Syahnas Lutfi	Game ini selain membuat tantangan di dalam game, juga perlu diketahui gamen ini secara tidak langsung sangat membantu kita untuk berolahraga
Daniel Santosa	ya, karena dapat bersepeda dalam ruangan sangat membantu proses pembakaran kalori
Millen	Tentu, untuk saya yang selalu berada dirumah dan jarang keluar pastinya saya kurang berolahraga dengan adanya simulasi seperti ini saya punya alternatif untuk berolahraga
Edbert Liniarto	Ya, supaya saat berolahraga di rumah tidak perlu bosan hanya mendengarkan lagu saja
Glenn Hariyanto Putra	iya karena bisa membuat saya sehat
Miftahuzzaka	Iyaa tentu saja, adventure and tetap bisa dengan keluarga
Syahnas Anjas	Bisa buat sarana olahraga karena sangat simpel dan menyenangkan

(lanjutan tabel)

Nama	Pendapat
Jeffri Harjayanto	Mungkin, karena sedikitnya waktu kosong membuat sulit untuk berolahraga, saya tetap lebih memilih aplikasi lebih simple untuk berolahraga, dibandingkan harus melakukan setup untuk memainkan game simulator sepeda VR, meski begitu jika saya tidak perlu melakukan setup maka game ini akan menjadi sarana olahraga yang baik
Sigit	Iya, karena seperti menggunakan sepeda biasa dan juga sebuah terobosan baru dalam bersepeda
Hugo Apriy	Iya, bisa menggantikan bersepeda konvensional yang harus keluar rumah dan kontak dengan orang lain

i. Kesimpulan dari respon pengguna

Dari keseluruhan respon pengguna yang sudah mencoba memainkan “Simulator Sepeda VR”, dapat disimpulkan bahwa “Simulator Sepeda VR” memberikan pengalaman baru dan merasa lebih tertantang (PE). Pengguna juga merasa mudah dalam memainkan dan menyelesaikan game “Simulator Sepeda VR” (EE). Pengguna merasa bahwa cara memainkan dan level di dalam game “Simulator Sepeda VR” bisa dibilang menarik (HM). Dikarenakan oleh ketiga hal tersebut, pengguna berencana untuk terus memainkan game “Simulator Sepeda VR” (BI).

ii. **Data waktu yang dikumpulkan dari pengujian pengguna game**

Tabel 4.9 Data waktu dari pengguna

Nama	Waktu
Millen	2:56:50
Max	2:58:37
Antonius	3:01:12
Chacha	3:03:24
Eva	3:09:46
Elizabeth	3:10:09

