



TUGAS PERTEMUAN: 8

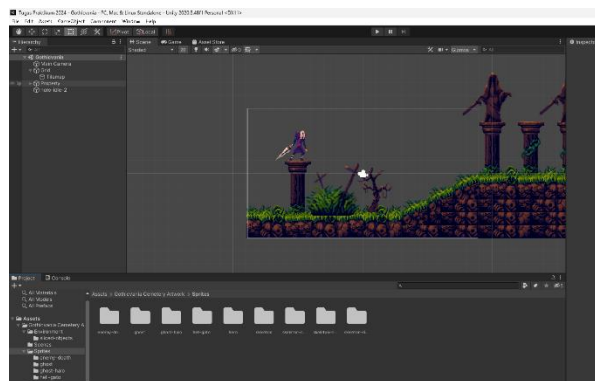
CAMERA & CHARACTER MOVEMENT

NIM	:	2118073
Nama	:	Enrico Erdhani
Kelas	:	B
Asisten Lab	:	Maria Avriliaana Surat Lelaona (2218096)

8.1 Tugas 8 : Membuat Tilemap sesuai asset

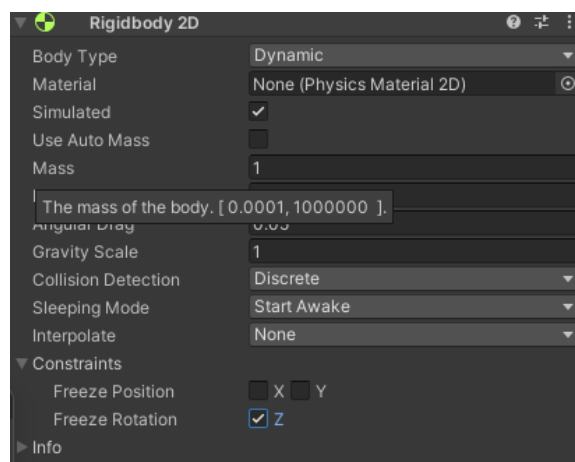
A. Pergerakan Player

1. Pertama Buka file projek Unity sebelumnya pada bab 7 untuk digunakan kembali



Gambar 8.1 Open Project

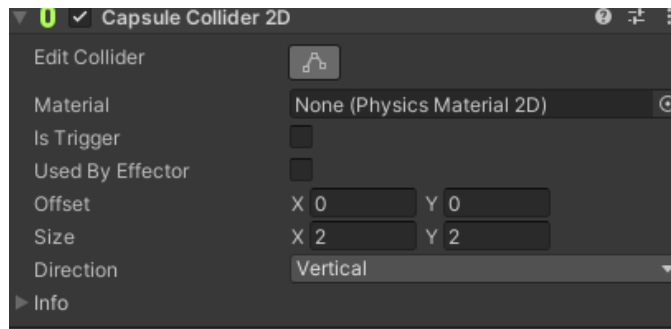
2. Lalu Pilih Player Idle tambahkan Component *Rigidbody* 2D Dan Setting seperti gambar berikut, Centang pada *Freeze Rotation Z*



Gambar 8.2 Rigidbody 2D

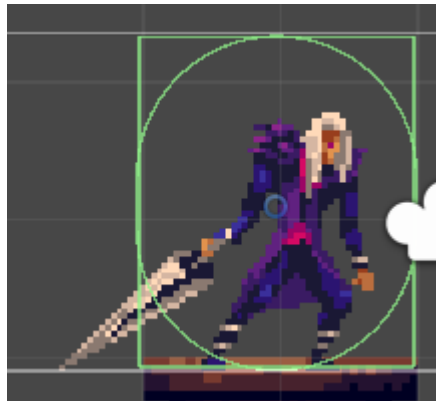


3. Kemudian tambahkan komponen *Capsule Colider* di Player Idle, lalu klik icon sebelah kanan *edit collider*.



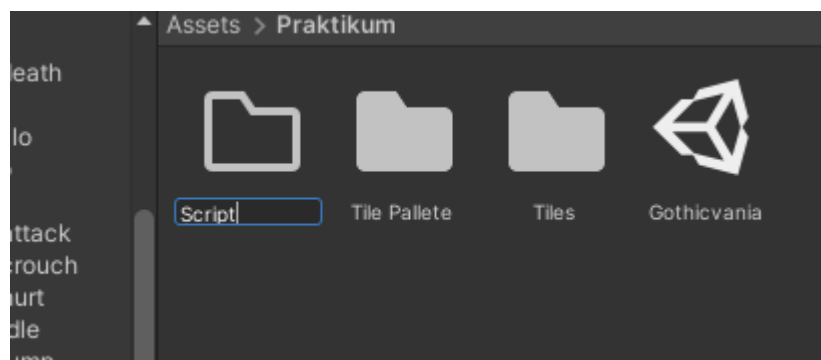
Gambar 8.3 Setting Capsule Colider

4. Kita Paskan Player Dan garis oval dengan karakternya Seperti ini.



Gambar 8.4 Capsule Colider

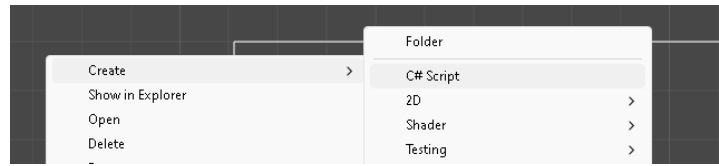
5. Jika Sudah, Kita Buat folder baru bernama Script Di Dalam Folder Pratikum.



Gambar 8.5 Membuat Folder Baru

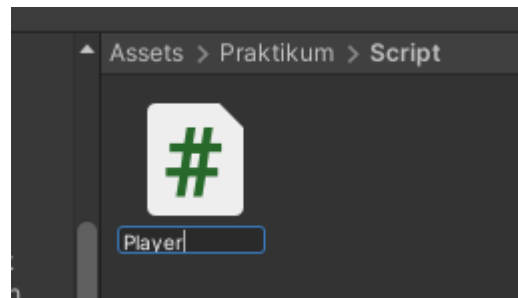


6. Kemudian Ke folder *Script*, lalu buat *C# Script*, beri nama *Player*.



Gambar 8.6 Buat C# Script

7. Lalu Drag & drop *script* player *Fall*, lalu klik 2x pada *script* player maka akan masuk kedalam *text editor* seperti ini



Gambar 8.7 Setting Script

8. Selanjutnya *source code* dibawah ini, pastikan nama *public class* harus sama dengan nama file yang dibuat.

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

public class Player : MonoBehaviour
{
    Rigidbody2D rb;

    [SerializeField] float speed = 1;
    float horizontalValue;
    bool facingRight;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update ()
    {
        horizontalValue = Input.GetAxisRaw("Horizontal");
    }

    void FixedUpdate()
    {
        Move(horizontalValue);
    }

    void Move(float dir)
    {

```



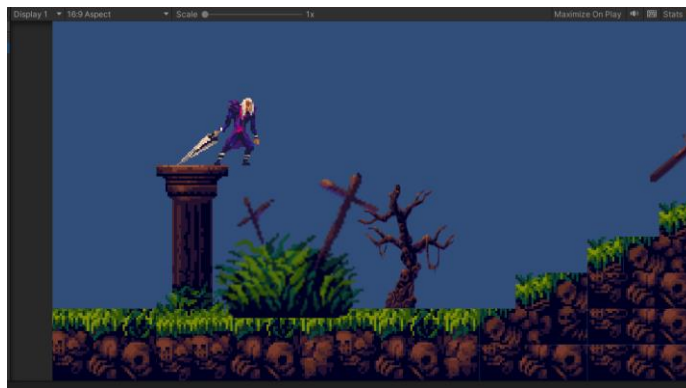
```
#region gerak kanan kiri
float xVal = dir * speed * 100 * Time.fixedDeltaTime;
Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
rb.velocity = targetVelocity;

if (facingRight && dir < 0)
{
    // ukuran player
    transform.localScale = new Vector3(-1, 1, 1);
    facingRight = false;
}

else if (!facingRight && dir > 0)
{
    // ukuran player
    transform.localScale = new Vector3(1, 1, 1);
    facingRight = true;
}

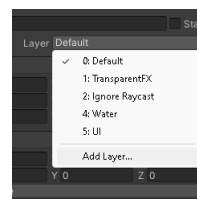
#endregion
}
```

9. Kemudian Dengan mencoba *Source code* diatas berhasil, Tekan dikeyboard “a” atau “left arrow” untuk ke arah kiri, tekan “d” atau “right arrow” untuk ke arah kanan



Gambar 8.8 Character bergerak

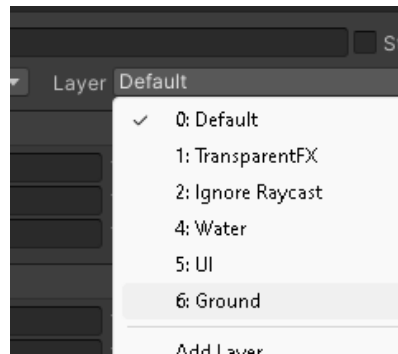
10. Kita membuat player loncat menggunakan spasi, dengan cara, klik Grid pada *Hierarchy*, pergi ke inspector, pilih *Layer*, Klik *Add Layer*



Gambar 8.9 Membuat Layer pada Grid

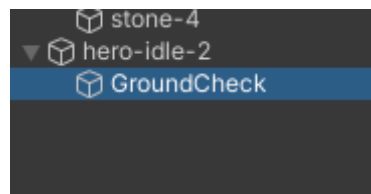


11. Lalu *Layer* menjadi *Ground*, jika muncul *pop up Change Layer*, klik *yes* saja.



Gambar 8.10 Menambahkan layer *Ground*

12. Klik kanan pada *Player Idle*, lalu *Create empty*, beri nama *Groundcheck*



Gambar 8.11 Membuat *Groundcheck*

13. Pada *Hirarki GroundCheck*, lalu gunakan *Move Tool* untuk memindahkan ke bagian bawah *Player* seperti gambar berikut.



Gambar 8.12 Menggunakan *Move Tool*



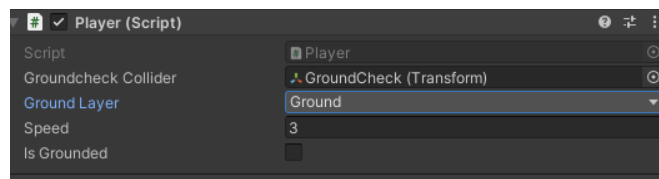
14. ke script Player tambahkan *source code* seperti ini.

```
[SerializeField] Transform groundcheckCollider;  
[SerializeField] LayerMask groundLayer;  
  
const float groundCheckRadius = 0.2f; // +  
[SerializeField] float speed = 1;  
float horizontalValue;  
  
[SerializeField] bool isGrounded; // +  
bool facingRight;
```

15. Lalu Buat void *ground check* dibawah void *fixedUpdate* & tambahkan *GorunCheck()*; pada void *fixedUpdate*

```
void FixedUpdate()  
{  
    GroundCheck();  
    Move(horizontalValue);  
}  
  
void GroundCheck()  
{  
    isGrounded = false;  
    Collider2D[] colliders =  
    Physics2D.OverlapCircleAll(groundcheckCollider.position,  
    groundCheckRadius, groundLayer);  
    if (colliders.Length > 0)  
        isGrounded = true;  
}
```

16. Pencet player Fall, lalu ke inspector ke *Grouncheck collider* tekan icon
lalu pilih yang *Grouncheck Transform*, dan pada *Ground Layer* pilih
Groundcheck



Gambar 8.13 Edit Player script



17. Lalu untuk membuat player melompat tambahkan *script* berikut

```
[SerializeField] float jumpPower = 100;

bool jump;
```

18. Tambahkan Lagi *script* berikut di bagian *void update*.

```
if (Input.GetButtonDown("Jump"))
    jump = true;
else if (Input.GetButtonUp("Jump"))
    jump = false;
```

19. Tambahkan Juga jump pada parameter Move

```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);
}
```

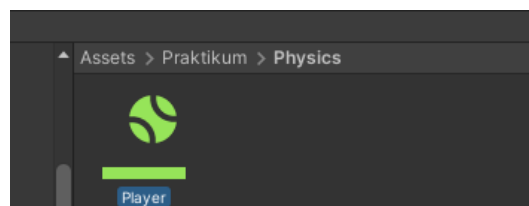
Gambar 8.14 membuat parameter move

20. Tambahkan script berikut pada *void Move*.

```
bool jumpflag

if(isGrounded && jumpflag)
{
    isGrounded = false;
    jumpflag = false;
    rb.AddForce(new Vector2(0f, jumpPower));
}
```

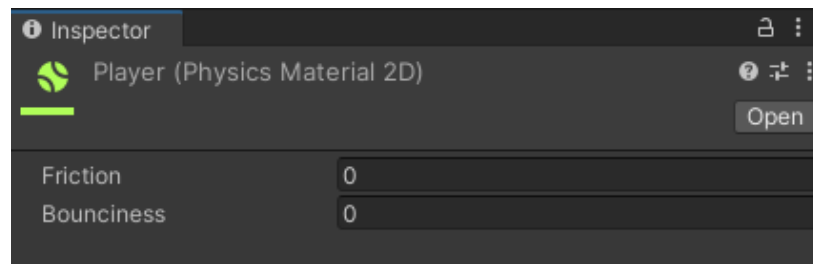
21. Pada folder Pyshics create > 2d > physical material 2d , berinama Player



Gambar 8.15 Membuat Physical material 2d

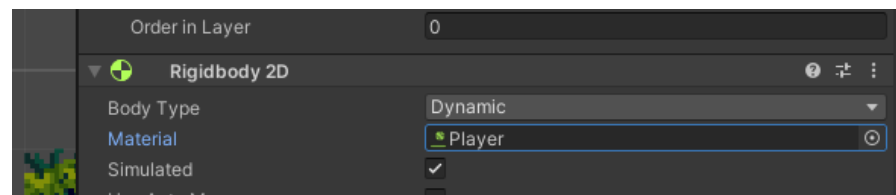


22. Pencet Player *Physics Material* 2D, dibagian menu *inspector*, *friction* & *bounces* ubah menjadi 0



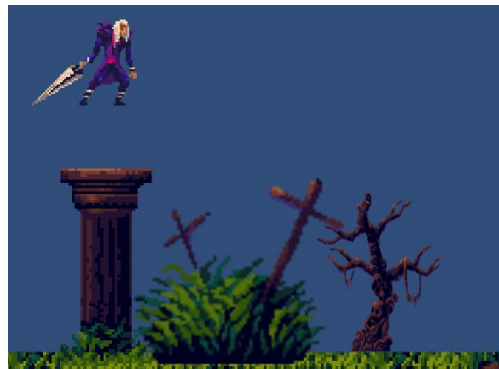
Gambar 8.16 Setting Physical material 2d

23. Menu *Hierarchy* pilih layer player Idle, pada Inspector Cari *Rigidbody* 2D lalu klik icon untuk membuka box select *physics material* 2d , lalu pilih asset Player yang sudah kita buat tadi



Gambar 8.17 Edit Body Type Rigidbody 2D

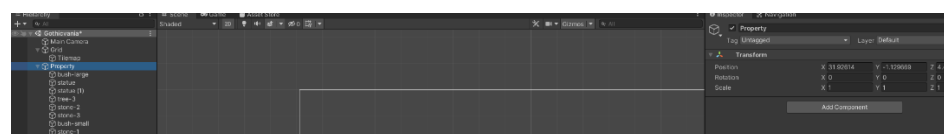
24. Kemudian play, player melompat dengan Spasi.



Gambar 8.18 Hasil akhir Gerakan Karakter

B. Camera Movement

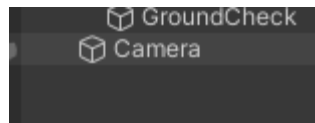
1. Pertama, Pada *Property* Ubah *Inspector* pada tag Main camera Menjadi *untaged*



Gambar 8.19 Ubah Property

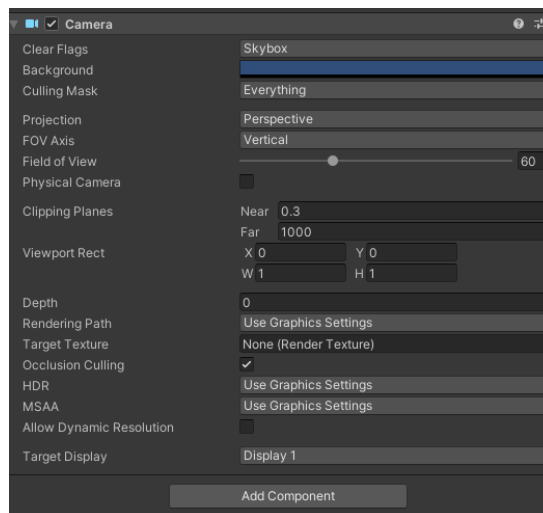


2. Lalu *Create Empty* pada Hirarki, dan Rename Menjadi Camera



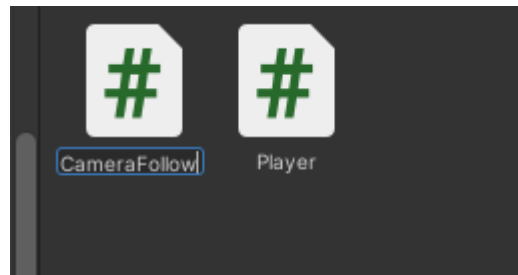
Gambar 8.20 Membuat camera

3. Setting *Layer Camera* seperti gambar dibawah ini.



Gambar 8.21 Setting layer camera

4. Kemudian Membuat File *script* baru di folder *Script* dengan nama CameraFollow.



Gambar 8.22 File Camerafollow

5. Tuliskan *script* berikut ini

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

public class CameraFollow : MonoBehaviour
{
    public float xMargin = 0.5f;
    public float yMargin = 0.5f;
    public float xSmooth = 4f;
    public float ySmooth = 4f;
    public Vector2 maxXAndY;
```



```
public Vector2 minXAndY;
private Transform player;

void Awake()
{
    player =
GameObject.FindGameObjectWithTag("Player").transform;
}

bool CheckXMargin()
{
    return Mathf.Abs(transform.position.x -
player.position.x) > xMargin;
}

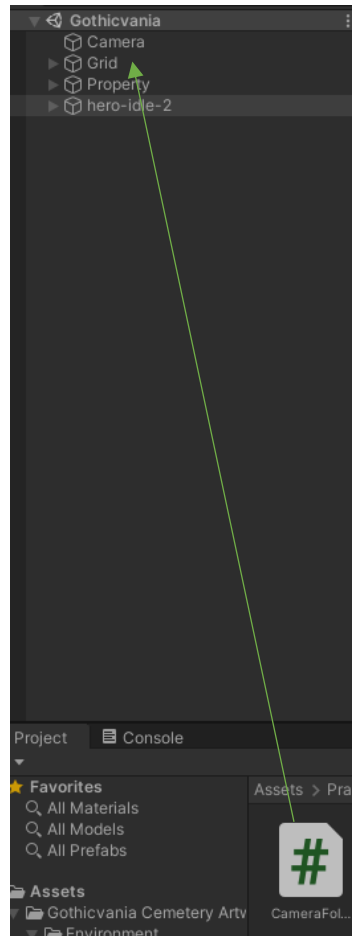
bool CheckYMargin()
{
    return Mathf.Abs(transform.position.y -
player.position.y) > yMargin;
}

void FixedUpdate()
{
    TrackPlayer();
}

void TrackPlayer()
{
    float targetX = transform.position.x;
    float targetY = transform.position.y;
    if (CheckXMargin())
        targetX = Mathf.Lerp(transform.position.x,
player.position.x,
xSmooth * Time.deltaTime);
    if (CheckYMargin())
        targetY = Mathf.Lerp(transform.position.y,
player.position.y,
ySmooth * Time.deltaTime);
    targetX = Mathf.Clamp(targetX, minXAndY.x,
maxXAndY.x); targetY =
Mathf.Clamp(targetY, minXAndY.y,
maxXAndY.y); transform.position = new
Vector3(targetX, targetY,
transform.position.z);
}
}
```

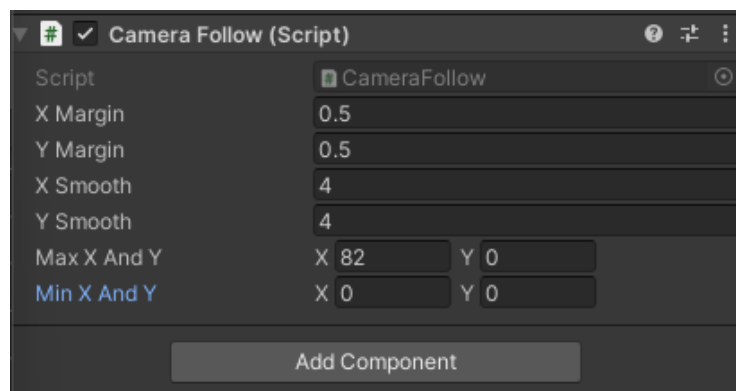


- Setelah itu, Drag & drop script *CameraFollow* ke dalam *Layer Camera*.



Gambar 8.23 Drag And Drop Script Camerafollow

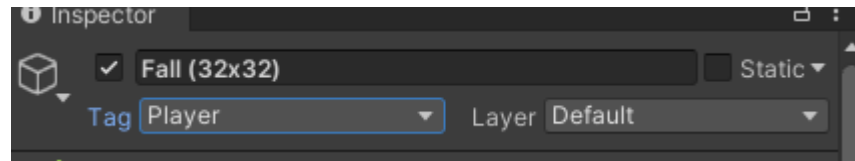
- Kemudian klik pada *camera*, buka *inspector* setting menjadi berikut.



Gambar 8.24 Inspector Camerafollow

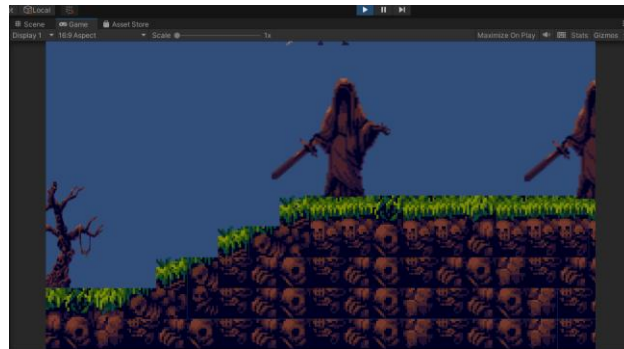


8. Kita Ubah tag di player Idle Untagged menjadi Player.



Gambar 8.25 Mengubah Tag Player

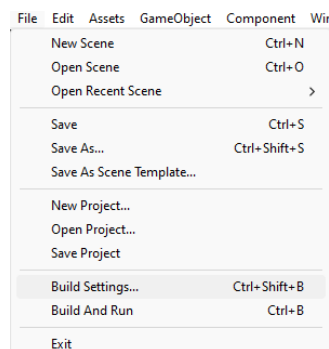
9. Pencet play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter,



Gambar 8.26 Hasil akhir Camera

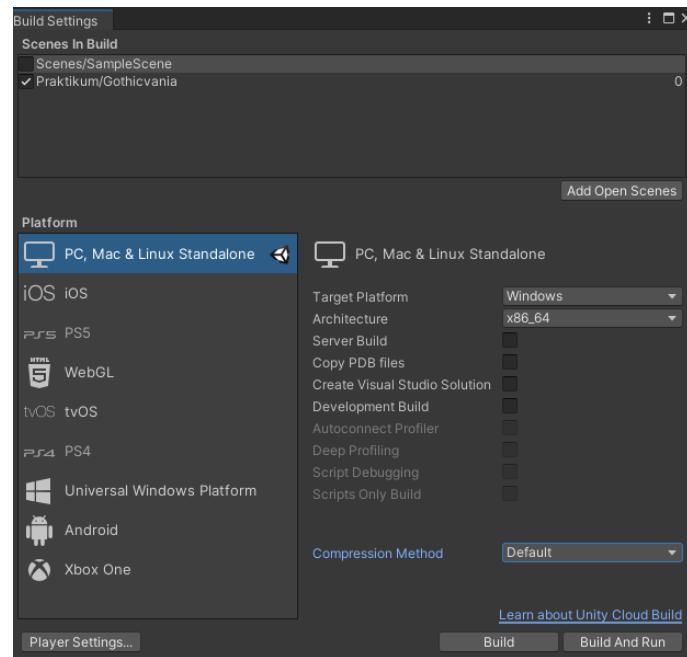
C. Render

1. Pertama, Pergi Menu File kemudian pilih *Build Setting*.



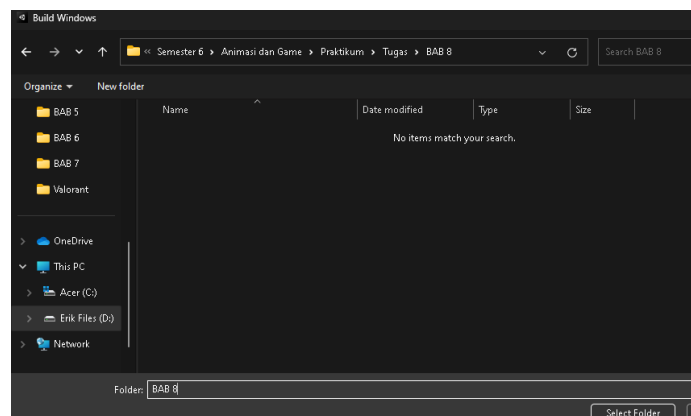
Gambar 8.27 Menu Build Setting

2. Ke *Setting Build* pilih PC, Mac & Linux, Tekan Build, pada project Tugas Kalian.



Gambar 8.28 Pilih Project Tugas

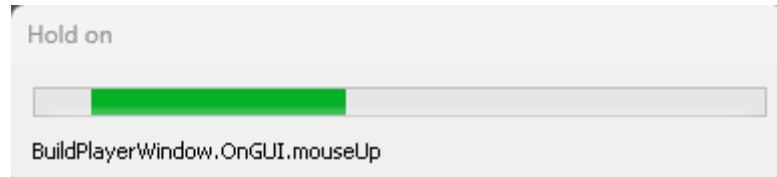
3. Kemudian Kita Pilih dimana Project disimpan.



Gambar 8.29 Menyimpan Project Game



4. Setelah Itu Kita Tunggu Hasil Render Dari Pembuatan Game Kita.



Gambar 8.30 Hasil Akhir Render

D. Kuis

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

public class CameraFollow : MonoBehaviour
{
    [SerializeField] private Transform player;

    void Update () {
        transform.position = new Vector3 (player.position.x,
        transform.position.y, transform.position.z);
    }
}
```

Penjelasan :

Source Code Di Atas Adalah *Class* bernama *CameraFollow* yang diturunkan dari *MonoBehaviour*. Variabel *Player* bertipe *Transform* yang diberi atribut [*SerializeField*]. Atribut ini memungkinkan variabel *player* untuk disetel dari editor Unity tanpa menjadikannya *public*. posisi kamera (*transform.position*) diatur agar mengikuti posisi sumbu x dari objek *player* (pemain), sedangkan posisi y dan z kamera tetap sama dengan posisi y dan z sebelumnya.

E. Link Github Pengumpulan

https://github.com/EnricoErdhani/2118073_Prak_AniGame