



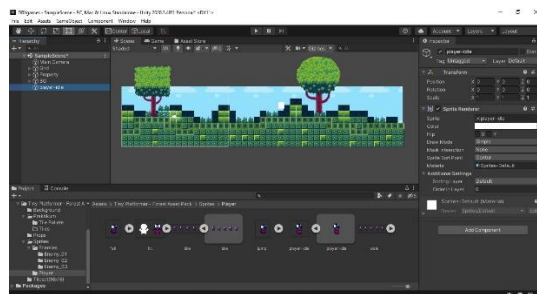
TUGAS PERTEMUAN: 8 CAMERA & CHARACTER MOVEMENT

NIM	:	2118090
Nama	:	Derry Frediansa
Kelas	:	C
Asisten Lab	:	Nur Aria Hibnastiar (2118078)

8.1 Tugas 8 : Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

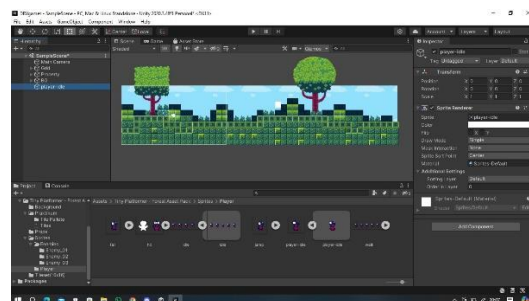
A. Membuat Pergerakan Player

1. Buka kembali file proyek Unity sebelumnya pada tugas bab 7 untuk digunakan kembali.



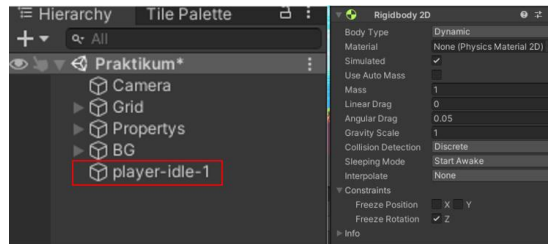
Gambar 8.1 Tampilan Awal

2. Selanjutnya tambahkan player bernama player-idle, pilih yang idle, Import kedalam Hirarki.



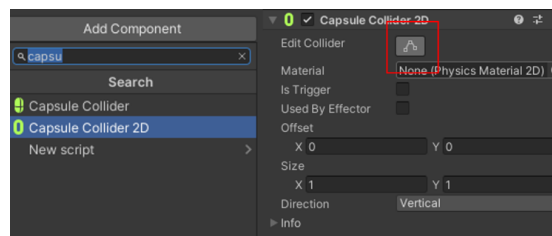
Gambar 8.2 Tampilan Player-idle

3. Kemudian klik player-idle-1 tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z.



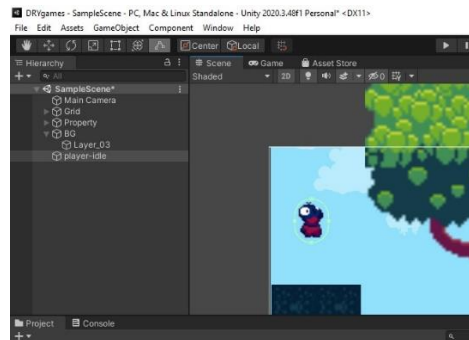
Gambar 8.3 Tampilan Freeze Rotation

4. Lalu tambahkan komponen Capsule Colider di player-idle-1, lalu klik icon sebelah kanan edit colider.



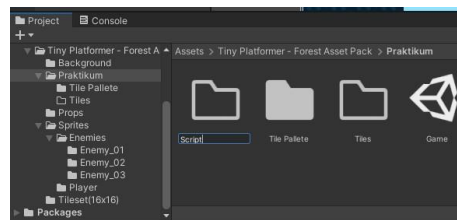
Gambar 8.4 Tampilan Edit Collider

5. Selanjutnya cocokan garis oval degan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya.



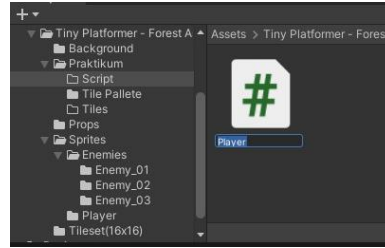
Gambar 8.5 Tampilan Setting Karakter

6. Kemudian buka Folder praktikum, lalu bikin folder baru bernama Script.



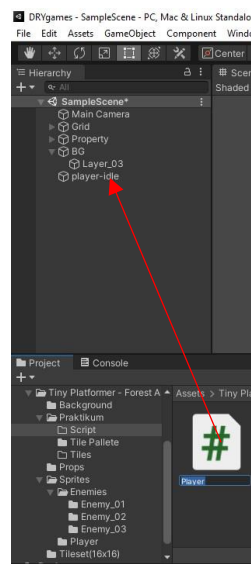
Gambar 8.6 Tampilan Folder Script

7. Lalu masuk kedalam folder Script, lalu buat C# Script, beri nama Player.



Gambar 8.7 Tampilan Script Player

8. Selanjutnya drag & drop script player kedalam Hirarki player-idle-1, lalu klik 2x pada script player maka akan masuk kedalam text editor seperti ini.



Gambar 8.8 Cara Drag Script

9. Kemudian Masukkan 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);
}

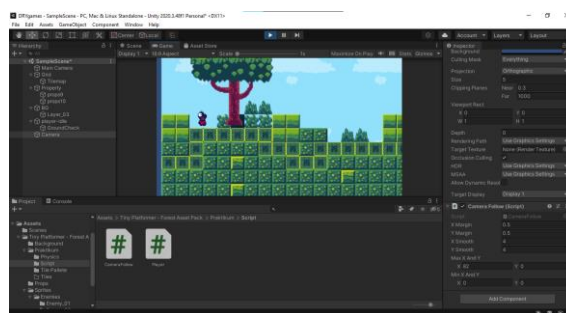
void Move(float dir)
{
    #region gerak kanan kiri
    float xVal = dir * speed * 100 * Time.fixedDeltaTime;
    Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
    rb.velocity = targetVelocity;

    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;
    }

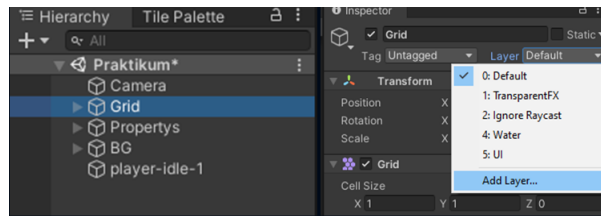
    #endregion
}
}
```

10. Lalu untuk 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 8.9 Tampilan Kanan Kiri

11. Selanjutnya Untuk membuat player loncat menggunakan spasi, kita perlu membuat GroundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer.



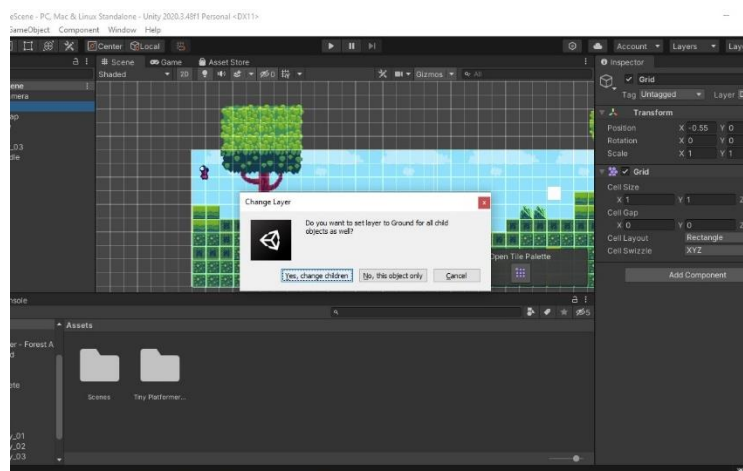
Gambar 8.10 Cara Add Layer

12. Kemudian isi “Ground” pada User Layer 6.



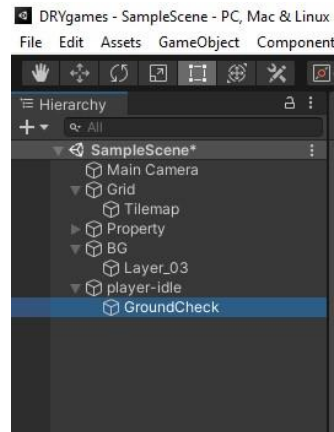
Gambar 8.11 Isi Ground Layer

13. Lalu ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja.



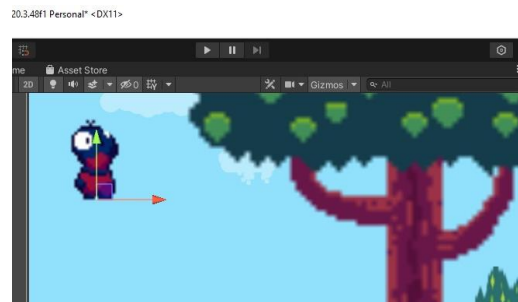
Gambar 8.12 Tampilan Change Layer

14. Selanjutnya klik kanan pada player-idle-1, lalu Create empty, beri nama GorundCheck.



Gambar 8.13 Tampilan Folder Groundcheck

15. Kemudian klik pada Hirarki GorundCheck, lalu gunakan “Move Tools” untuk memindahkan ke bagian bawah Player seperti gambar berikut.



Gambar 8.14 Tampilan Move Groundcheck

16. Lalu kembali 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;
```

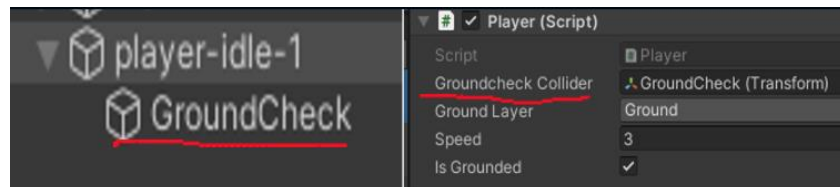
17. Selanjutnya bBuat void ground check dibawah void fixedUpdate & tambahkan GorunCheck(); pada void fixedUpdate.

```
void FixedUpdate()  
{  
    GroundCheck();  
    Move(horizontalValue);  
}  
  
void GroundCheck()  
{  
    isGrounded = false;
```



```
Collider2D[] colliders =  
Physics2D.OverlapCircleAll(groundcheckCollider.position,  
groundCheckRadius, groundLayer);  
if (colliders.Length > 0)  
isGrounded = true;  
}
```

18. Selanjutnya klik player-idle-1, lalu ke inspector ke effect Player script di bagian “Groundcheck collider” tekan icon lalu pilih yang GroundCheck Transform, dan pada Ground Layer pilih Ground.



Gambar 8.15 Tampilan Effect Player

19. Kemudian untuk membuat player melompat tambahkan script berikut

```
[SerializeField] float jumpPower = 100;  
  
bool jump;
```

20. Lalu tambahkan juga script berikut di bagian void update.

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

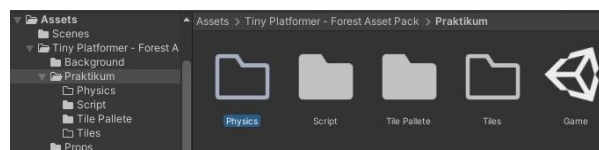
21. Selanjutnya tambahkan juga jump pada parameter Move.

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

22. Kemudian tambahkan script berikut pada void Move.

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

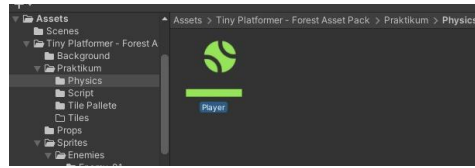
23. Lalu buat folder baru di praktikum bernama “Physics”.



Gambar 8.16 Tampilan Folder Physics

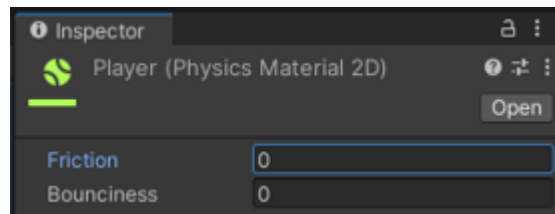


24. Selanjutnya didalam folder Pyshics create > 2d > physical material 2d ,
berinama “Player”.



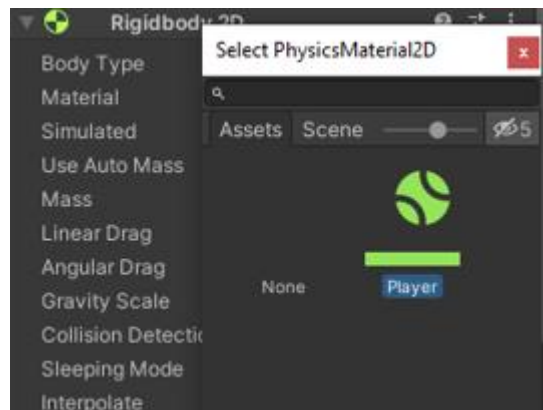
Gambar 8.17 Material Player 2D

25. Kemudian klik Player (Physics Material 2D), dibagian menu inspector,
friction & bounces ubah menjadi 0.



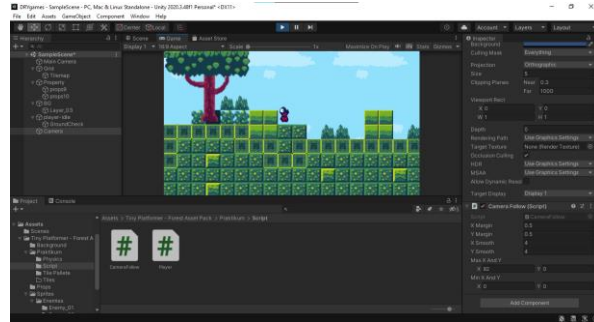
Gambar 8.18 Setting Friction

26. Lalu klik Hierarchy pilih layer player idle 1, pada Inspector Cari
Rigidbody 2D lalu klik icon untuk membuka box select physhics
material 2d , lalu pilih asset Player yang sudah kita buat tadi.



Gambar 8.19 Tampilan Pilih Material2D

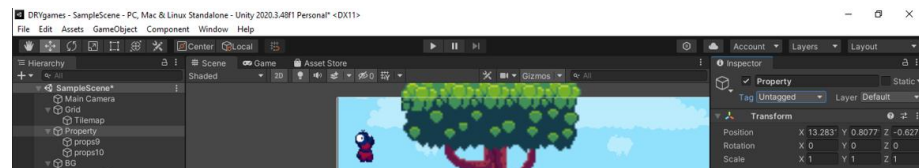
27. Selanjutnya tekan play, maka player bisa melompat dengan menekan
spasi.



Gambar 8.20 Tampilan Melompat

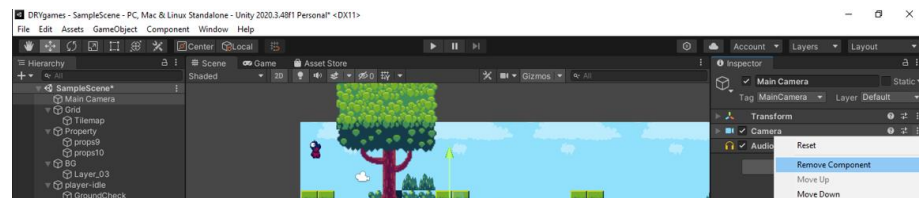
B. Camera Movement

1. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untagged.



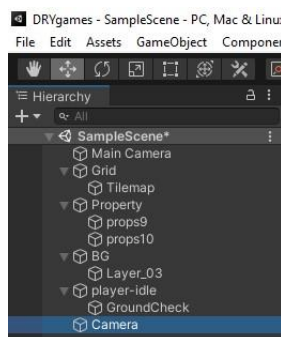
Gambar 8.21 Tampilan Untagged

2. Lalu pada Effect Camera pilih Remove Component.



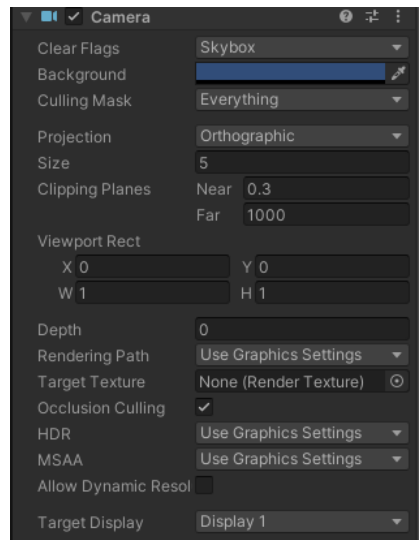
Gambar 8.22 Tampilan Remove Camera

3. Selanjutnya Create Empty pada Hirarki, dan Rename Menjadi Camera.



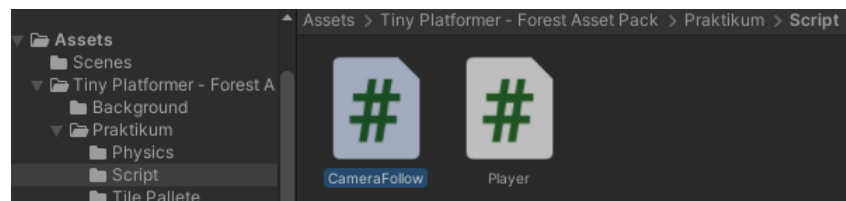
Gambar 8.23 Create Folder Camera

4. Kemudian sesuaikan Setting Layer Camera seperti gambar dibawah ini.



Gambar 8.24 Tampilan Setting Camera

5. Lalu buat file script baru di folder Script dengan nama "CameraFollow".



Gambar 8.25 Tampilan Script Camera

6. Selanjutnya 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;
    }
}
```

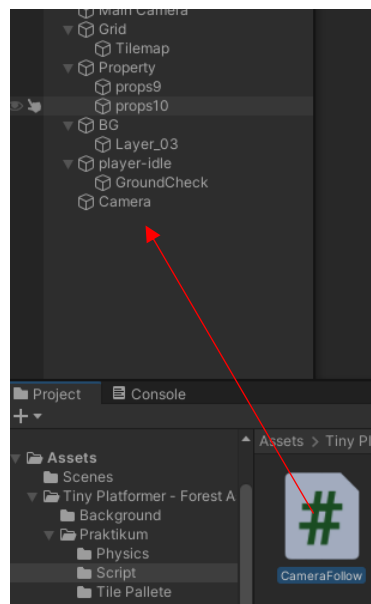


```
bool CheckYMargin()
{
    return Mathf.Abs(transform.position.y -
player.position.y) > yMargin;
}

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);
}
```

7. Kemudian drag & drop script CameraFollow Kedalam Layer Camera.



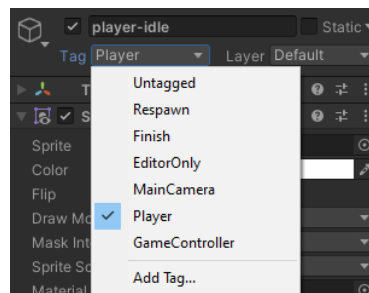
Gambar 8.26 Cara Drag Script

8. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya.



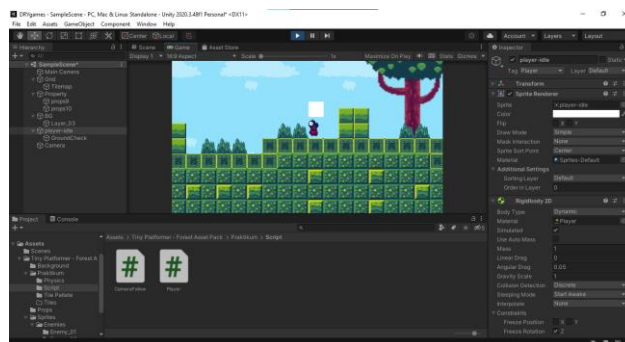
Gambar 8.27 Setting Camera Follow

9. Selanjutnya ubah tag di player-idle-1 Untagged menjadi "Player".



Gambar 8.28 Merubah Layer Player

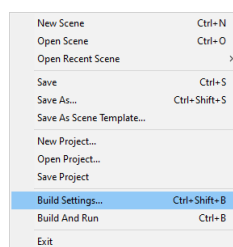
10. Kemudian tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter.



Gambar 8.29 Tampilan Camera

C. Render

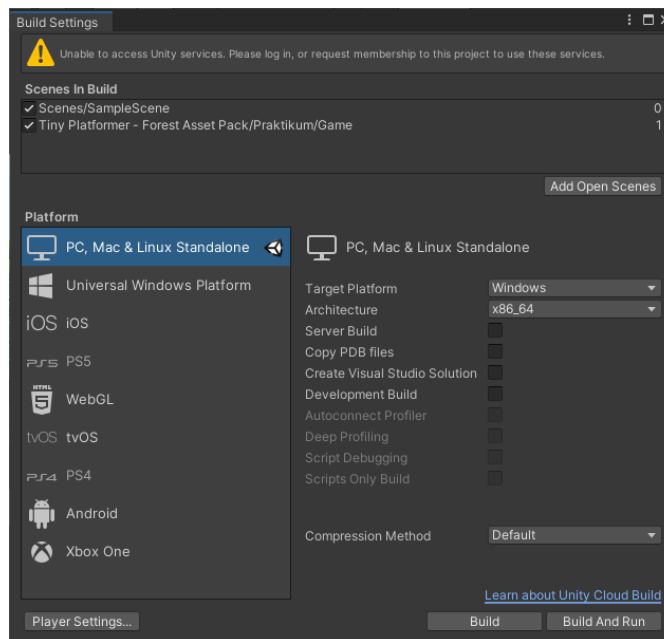
1. Pergi ke menu File kemudian pilih Build Setting (Ctrl + Shift + B).



Gambar 8.30 Tampilan Builds setting

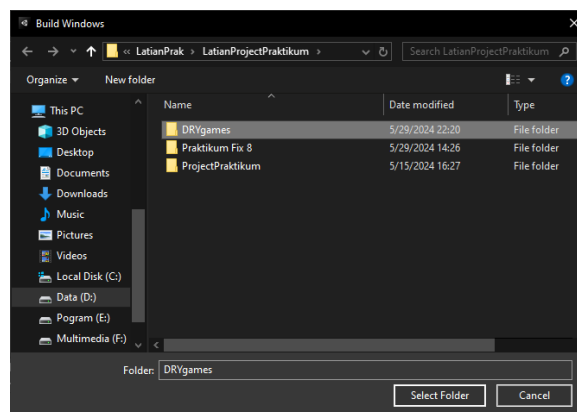


2. Lalu pada Setting Build ini pilih PC, Mac & Linux, Tekan Build, pastikan pada menu Scene in Build berada pada project Tugas Kalian.



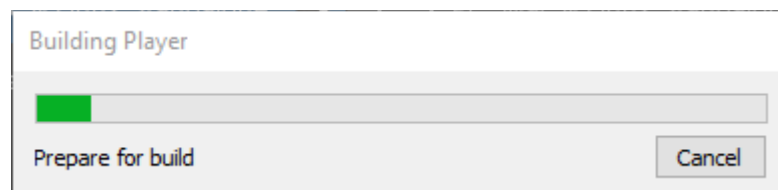
Gambar 8.31 Tampilan Setting Render

3. Selanjutnya pilih dimana Project disimpan, dan tunggu hasilnya.



Gambar 8.32 Tempat Simpan File

4. Kemudian ini hasilnya



Gambar 8.33 Tunggu Hasil Build

8.2 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);
    }
}
```

Analisa :

Kode ini adalah sebuah script C# yang digunakan dalam Unity, sebuah game engine yang sering digunakan untuk mengembangkan game 2D dan 3D. Script ini mengatur kamera agar mengikuti posisi pemain pada sumbu X, sementara posisi pada sumbu Y dan Z tetap sama. using System.Collections dan using System.Collections.Generic: Ini adalah direktif untuk menggunakan namespace yang menyediakan berbagai koleksi standar. using UnityEngine: Ini adalah direktif untuk menggunakan namespace Unity yang berisi kelas dan fungsi utama untuk pengembangan game. public class CameraFollow : MonoBehaviour: Mendeklarasikan sebuah kelas CameraFollow yang merupakan turunan dari MonoBehaviour, yang merupakan dasar untuk semua script di Unity. [SerializeField] private Transform player;: Mendeklarasikan variabel player dengan tipe Transform. SerializeField memungkinkan variabel ini dapat diatur melalui inspector di Unity editor meskipun bersifat private. void Update(): Metode Update dipanggil sekali per frame. Ini adalah tempat di mana kita meletakkan logika yang harus diperiksa atau diperbarui secara berkala. transform.position = new Vector3(player.position.x, transform.position.y, transform.position.z);: Mengatur posisi kamera (transform.position) menjadi posisi baru. Posisi baru ini memiliki koordinat X yang sama dengan posisi pemain (player.position.x), sedangkan koordinat Y dan Z tetap sama dengan posisi kamera saat ini (transform.position.y dan transform.position.z). Kode ini membuat kamera di dalam game Unity mengikuti pemain hanya pada sumbu X. Ini sering digunakan dalam game side-scroller di mana kamera perlu mengikuti pemain secara horizontal.