



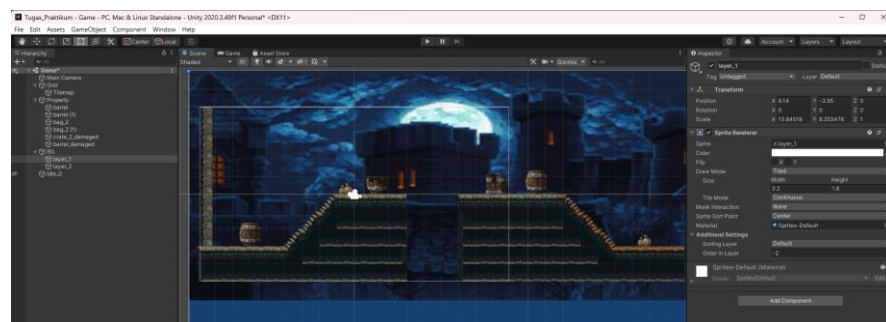
TUGAS PERTEMUAN: 8 CAMERA & CHARACTER MOVEMENT

NIM	:	2118080
Nama	:	Syahdan Izzur Rosuli
Kelas	:	B
Asisten Lab	:	M. Zainul Musyafa' (2118050)
Baju Adat	:	Cak dan Ning (Jawa Timur)
Referensi	:	https://cdnwpseller.gramedia.net/wp-content/uploads/2021/09/01210916/Baju-Cak-dan-Ning.jpg

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

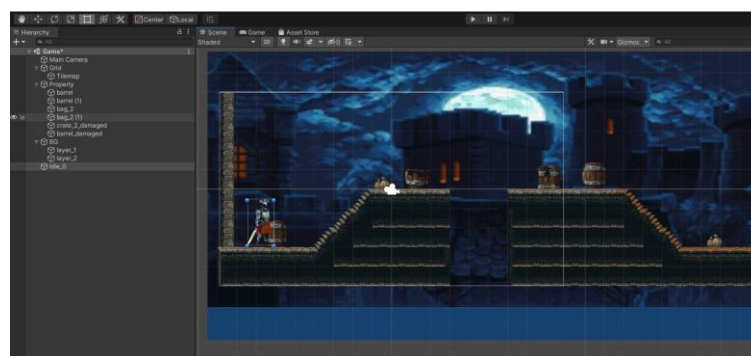
A. Membuat Pergerakan Player

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



Gambar 8.1 Buka Project

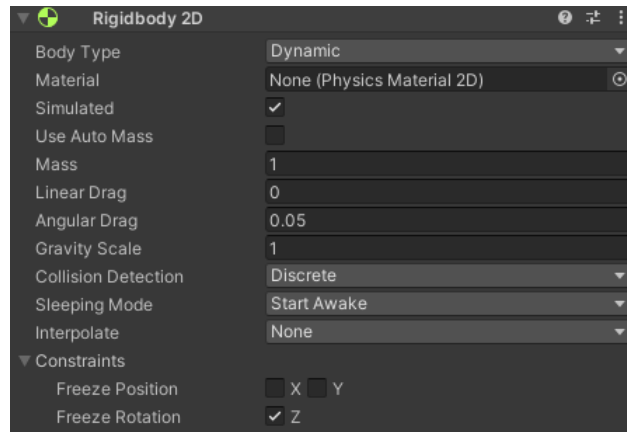
2. Selanjutnya tambahkan player, pilih yang idle, Import kedalam Hirarki.



Gambar 8.2 Import Player

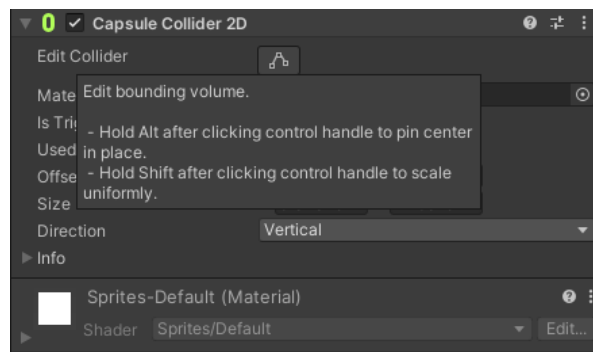


3. Kemudian klik player-idle-1 tambahkan Component Rigidbody 2D, sesuaikan settingannya seperti gambar berikut, Centang pada Freeze Rotation Z.



Gambar 8.3 Rigidbody 2D

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



Gambar 8.4 Capsule Collider

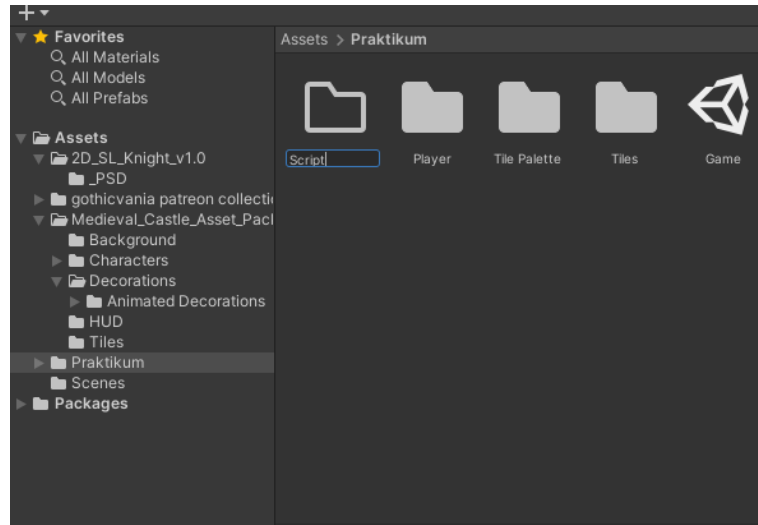
5. Selanjutnya cocokkan garis oval dengan karakternya atau bisa di inputkan Offset X, Y dan juga Size X, Y nya.



Gambar 8.5 Cocokkan ukuran

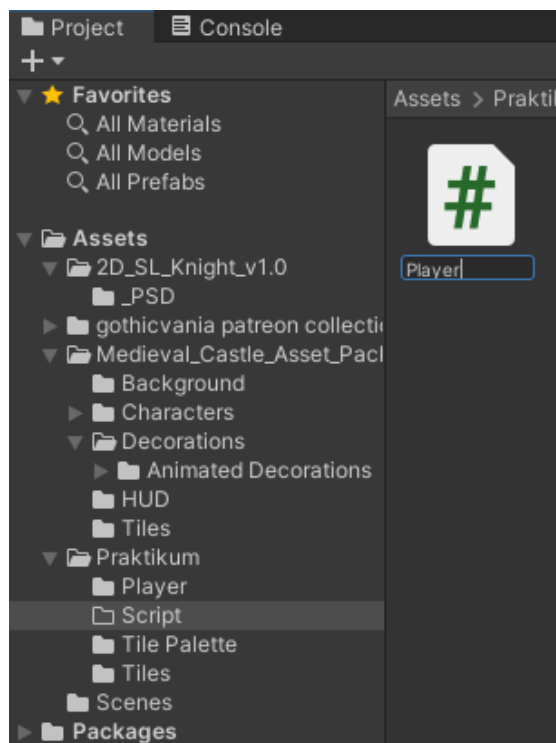


6. Kemudian buka Folder praktikum, lalu bikin folder baru bernama Script.



Gambar 8.6 Buat folder Script

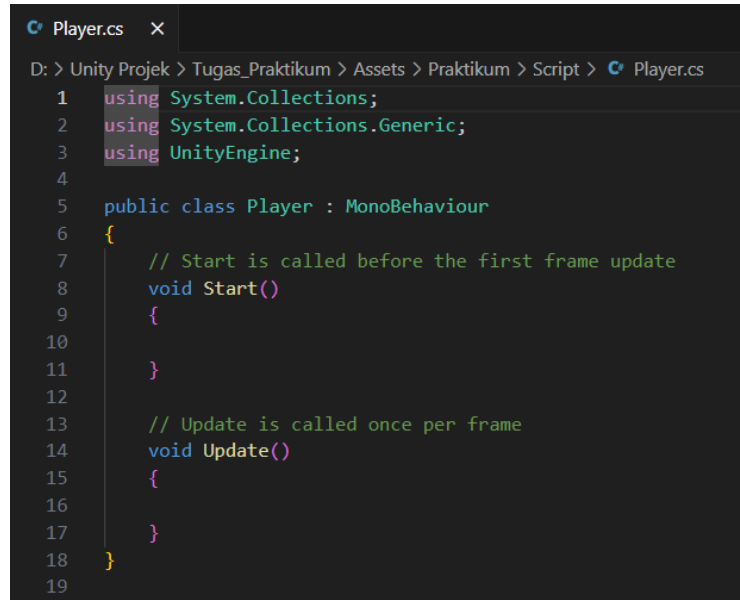
7. Lalu masuk kedalam folder Script, lalu buat C# Script, beri nama Player.



Gambar 8.7 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.



```
Player.cs x
D: > Unity Projek > Tugas_Praktikum > Assets > Praktikum > Script > Player.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Player : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16
17     }
18 }
19
```

Gambar 8.8 Script Editor

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
}
}
```

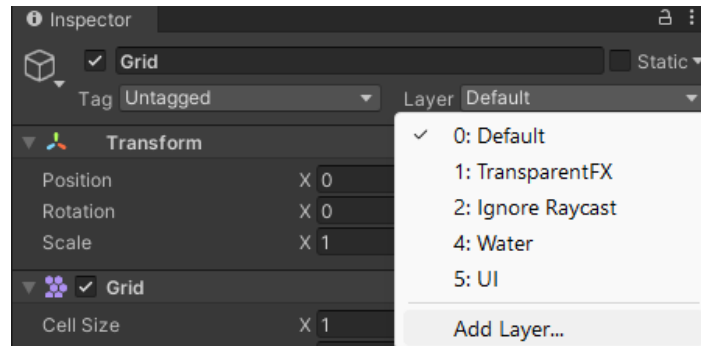
10. Lalu untuk mencoba Source code diatas berhasil, Tekan dikeyboard “a” atau “left arrow” untuk ke arah kiri, tekan “d” atau “right arrow” untuk ke arah kanan.



Gambar 8.9 Arahkan Player

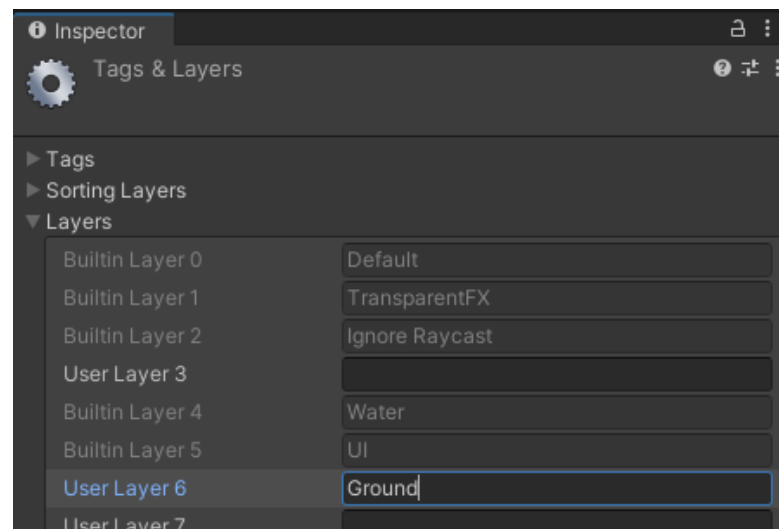


11. Selanjutnya Untuk membuat player loncat menggunakan spasi, kita perlu membuat GroundCheck dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer.



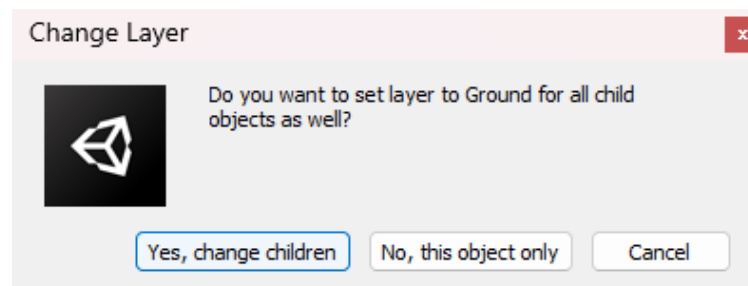
Gambar 8.10 Add Layer

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



Gambar 8.11 Layer Ground

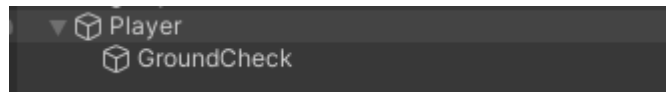
13. Lalu ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja.



Gambar 8.12 Ganti Layer



14. Selanjutnya klik kanan pada Player, lalu Create empty, beri nama GroundCheck.



Gambar 8.13 Pembuatan Ground Check

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



Gambar 8.14 Ground Check

16. Lalu kembali ke script Player tambahkan source code seperti ini.

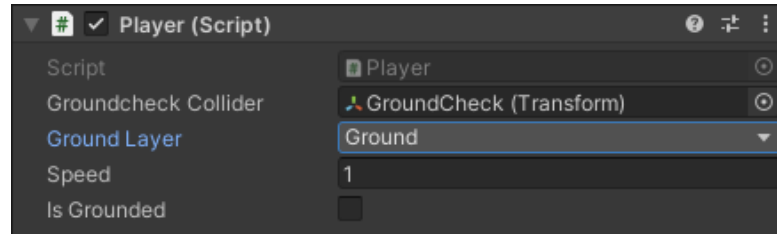
```
[SerializeField] Transform groundcheckCollider;  
[SerializeField] LayerMask groundLayer;  
  
const float groundCheckRadius = 0.2f; // +  
[SerializeField] float speed = 1;  
float horizontalValue;  
  
[SerializeField] bool isGrounded; // +  
bool facingRight;
```

17. Selanjutnya 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;  
}
```



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



Gambar 8.15 Pemberian GroundCheck

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);  
}
```

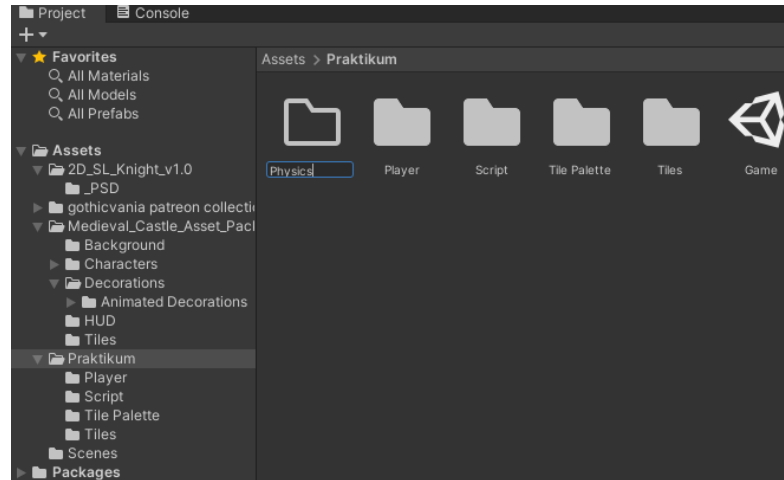
Gambar 8.16 Tambah Parameter jump

22. Kemudian tambahkan script berikut pada void Move.

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

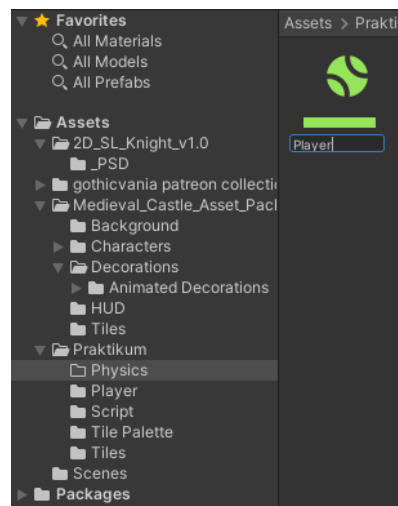



23. Lalu buat folder baru di praktikum bernama “Physics”.



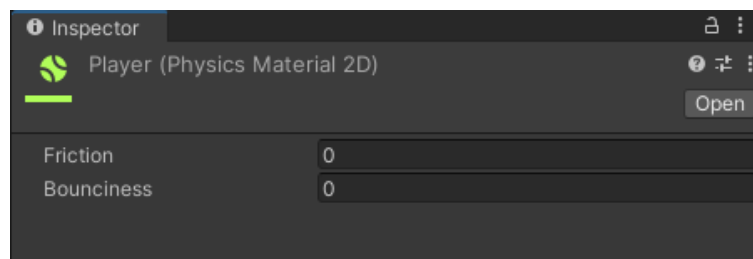
Gambar 8.17 Folder Physics

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



Gambar 8.18 Pembuatan Physical Material 2D

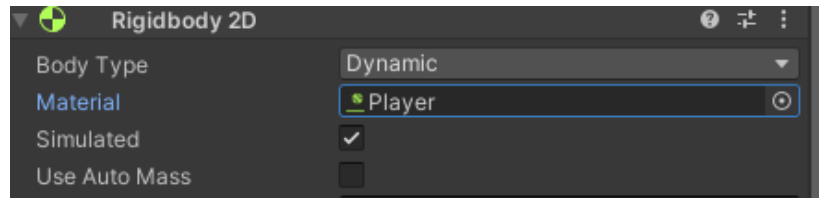
25. Kemudian klik Player (Physics Material 2D), dibagian menu inspector,
friction & bounces ubah menjadi 0.



Gambar 8.19 Inspector Physics Material

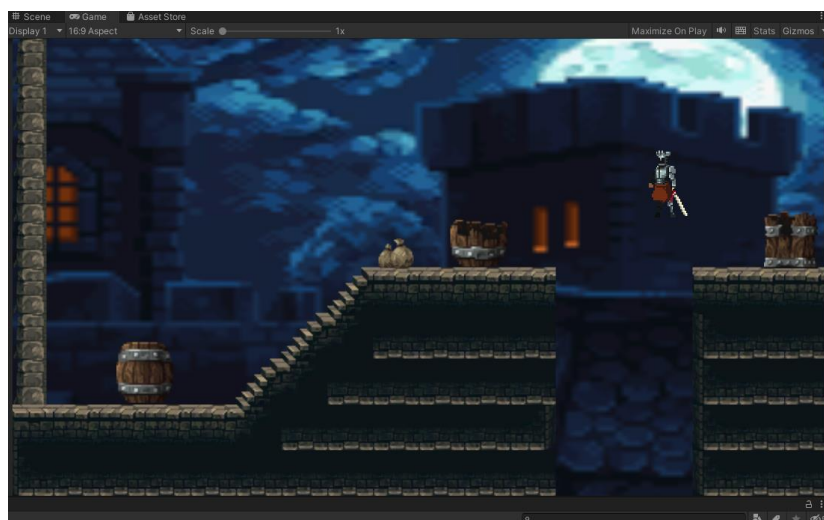


26. Lalu klik Hierarchy pilih layer player idle 1, pada Inspector Cari Rigidbody 2D lalu klik icon untuk membuka box select physics material 2d , lalu pilih asset Player yang sudah kita buat tadi.



Gambar 8.20 Penggunaan Physics Material

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



Gambar 8.21 Pengaplikasian Jump

B. Camera Movement

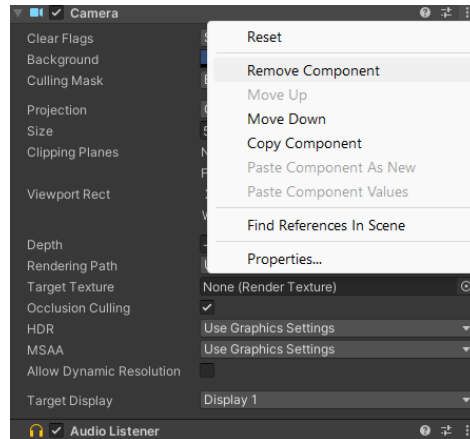
1. Pada Hirarki Property Ubah Inspector pada tag Main camera Menjadi untaged.



Gambar 8.22 Ubah Hirarki Property

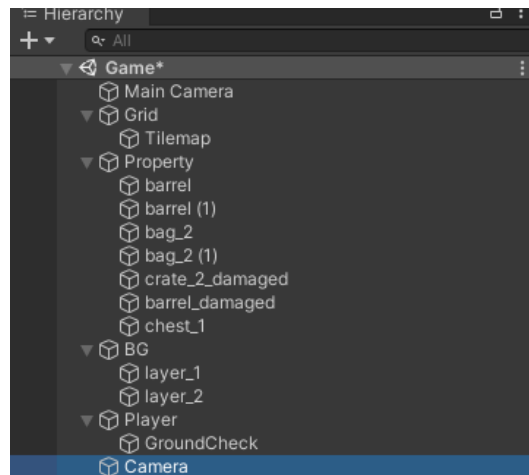


2. Lalu pada Effect Camera pilih Remove Component.



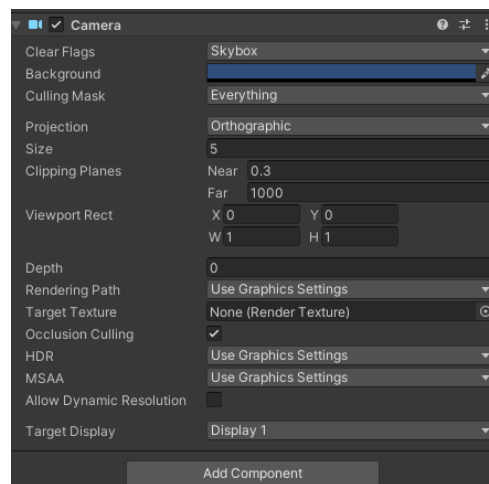
Gambar 8.23 Remove Camera

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



Gambar 8.24 Hirarki Camera

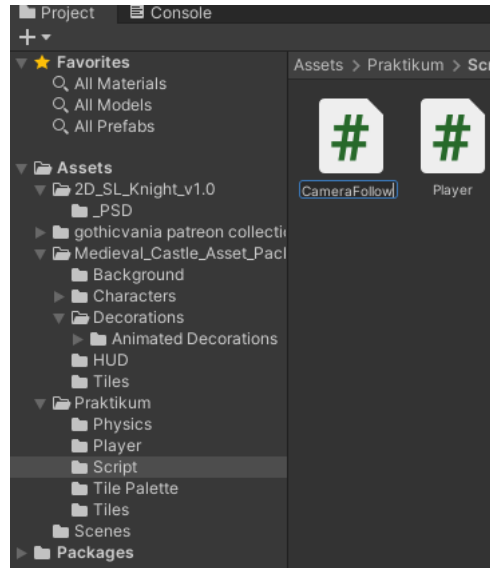
4. Kemudian sesuaikan Setting Layer Camera seperti gambar dibawah ini.



Gambar 8.25 Setting Camera



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



Gambar 8.26 Script CameraFollow

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;
    }

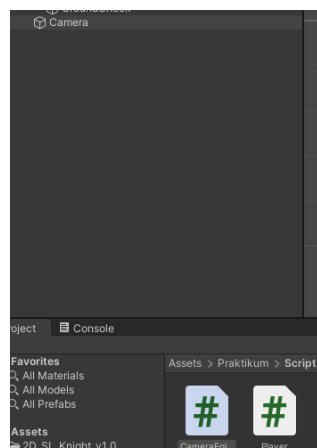
    void FixedUpdate()
    {

```



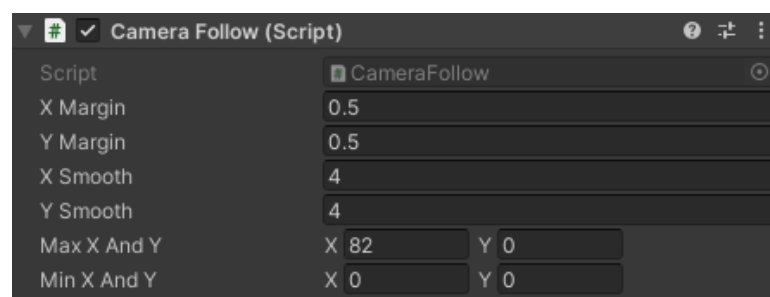
```
    TrackPlayer();  
}  
  
void TrackPlayer()  
{  
    float targetX = transform.position.x;  
    float targetY = transform.position.y;  
    if (CheckXMargin())  
        targetX = Mathf.Lerp(transform.position.x,  
player.position.x,  
        xSmooth * Time.deltaTime);  
    if (CheckYMargin())  
        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 8.27 Tambah CameraFollow

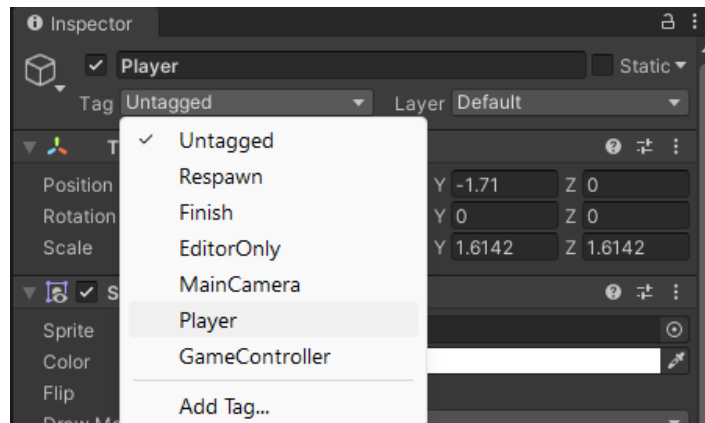
8. Lalu klik pada camera, buka inspector Pada bagian Camera Follow (Script) Ubah Bagian Max X dan Max Y nya.



Gambar 8.28 Ubah Script CameraFollow



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



Gambar 8.29 Ubah tag player

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

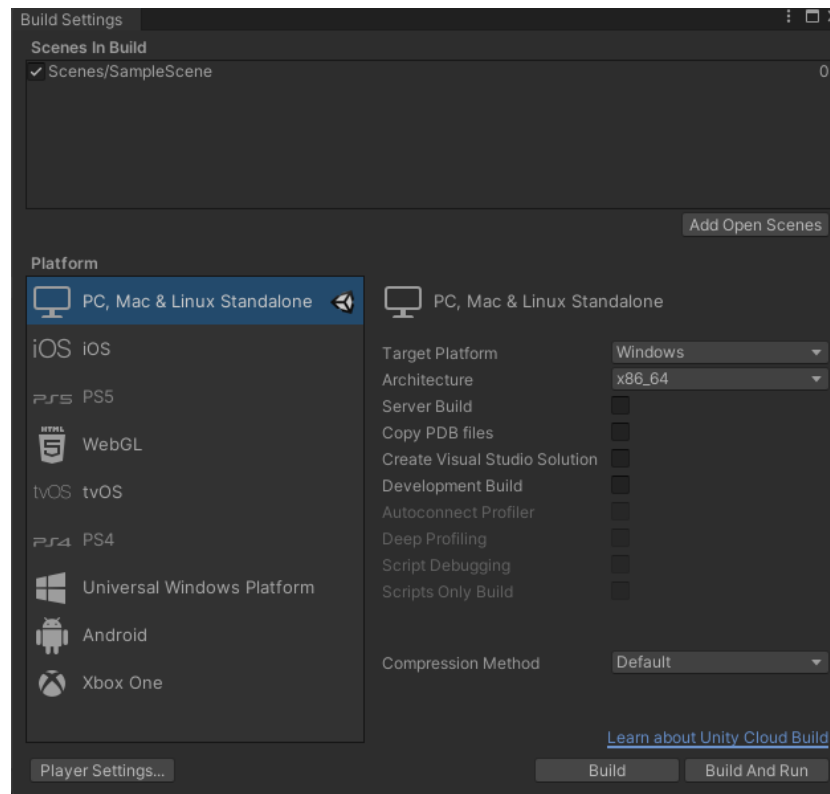


Gambar 8.30 Jalankan Projek



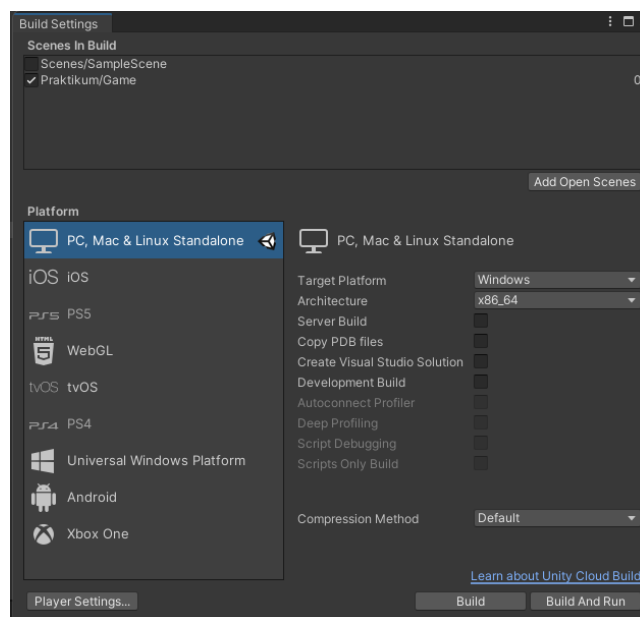
C. Render

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



Gambar 8.31 Build Setting

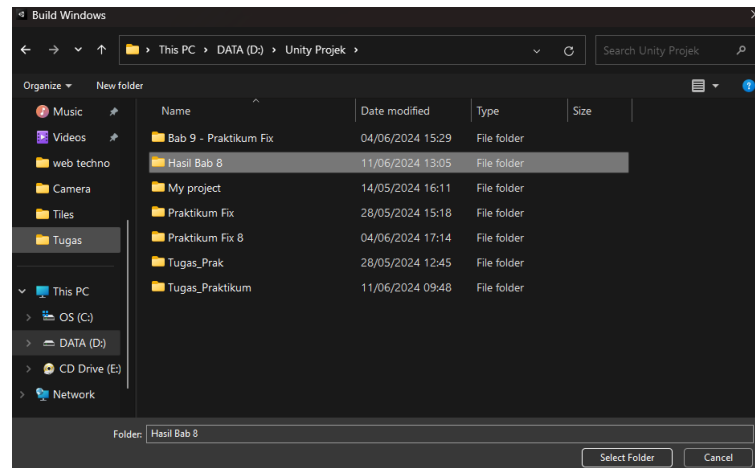
2. Lalu pada Setting Build ini pilih PC, Mac & Linux, Tekan Build, pastikan pada menu Scene in Build berada pada project Tugas Kalian.



Gambar 8.32 Ubah platform



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



Gambar 8.33 Pilih menyimpan Projek

4. Kemudian Mainkan untuk melihat hasilnya.



Gambar 8.34 Hasil Akhir



8.2 KUIS

Menjelaskan Source Code dibawah ini

```
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 diatas merupakan source code yang digunakan untuk mengubah posisi dari kamera agar mengikuti player. Yang dimana pada void update terdapat transform.position yang berisi logika untuk mengatur posisi kamera setiap frame agar selalu sejajar dengan posisi horizontal (x) dari pemain.