TUGAS PERTEMUAN: 8

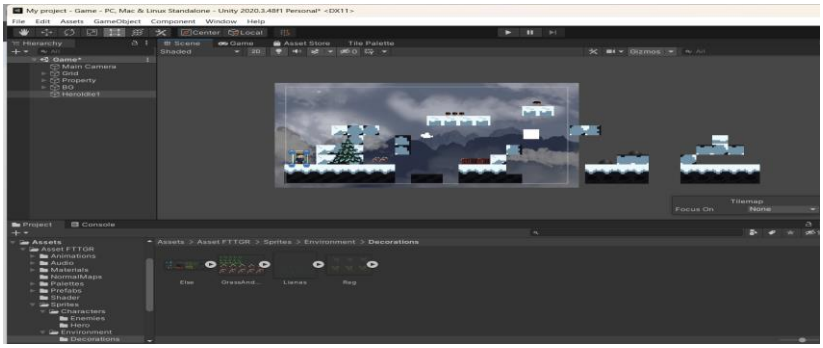
CAMERA & CHARACTER MOVEMENT

NIM	:	2118062
Nama	:	Nayla Dwi Salsabila
Kelas	:	A
Asisten Lab	:	Bagas Anardi Surya W (2118004)
Baju Adat	:	Baju Adat Batak Simalungun (Provinsi-Indonesia Barat)
Referensi	:	https://id.pngtree.com/freepng/traditional-clothes-of-batak-north-sumatera-indonesia_8129003.html

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

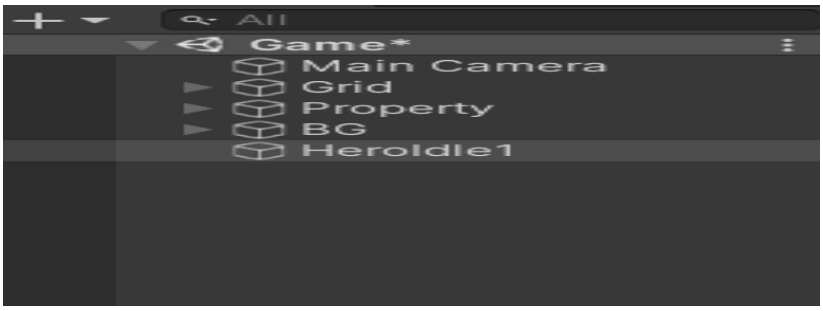
A. Membuat Pergerakan Player

1. Buka project Unity sebelumnya bab 7.



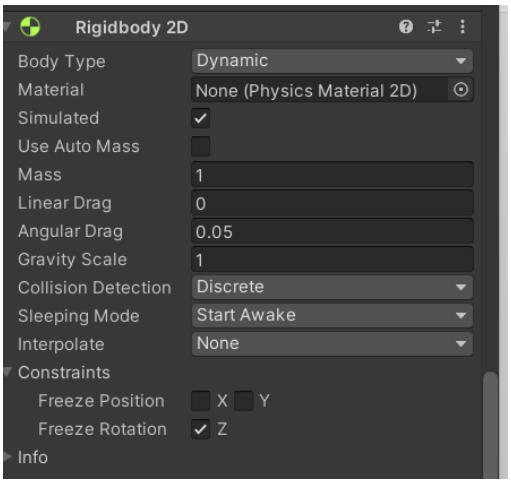
Gambar 8.1 Tampilan *Project Unity*

2. Tambahkan player bernama HeroIdle1, kemudian import ke dalam Hirarki.



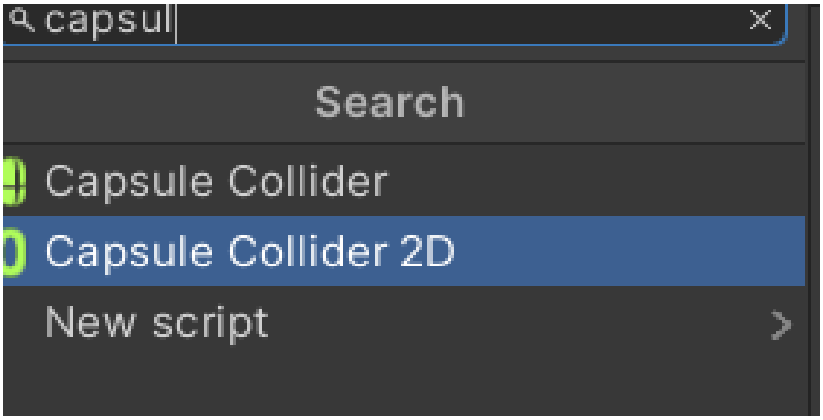
Gambar 8.2 Tampilan *Import HeroIdle1*

3. Klik HeroIdle1 tambahkan Compenet RigidBody 2D, lalu centang pada Freeze Rotation Z.



Gambar 8.3 Tampilan *Freeze Rotation Z*

4. Kemudian tambahkan komponen Capsule Colider di HeroIdle1, klik icon sebelah kanan edit colider.

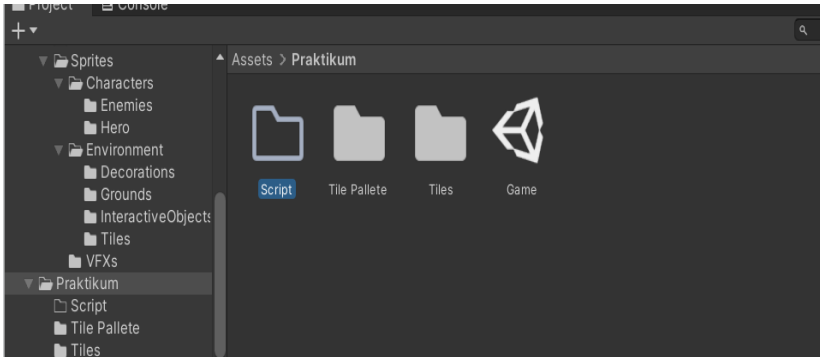


Gambar 8.4 Tampilan *Capsule Collider 2D*

5. Lalu cockan garis oval dengan karakter .

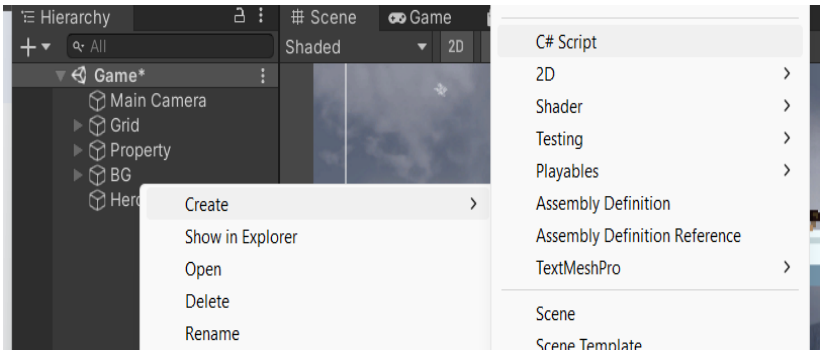
Gambar 8.5 Tampilan *Cockan garis oval*

6. Buat folder baru bernama “Script” di folder praktikum.



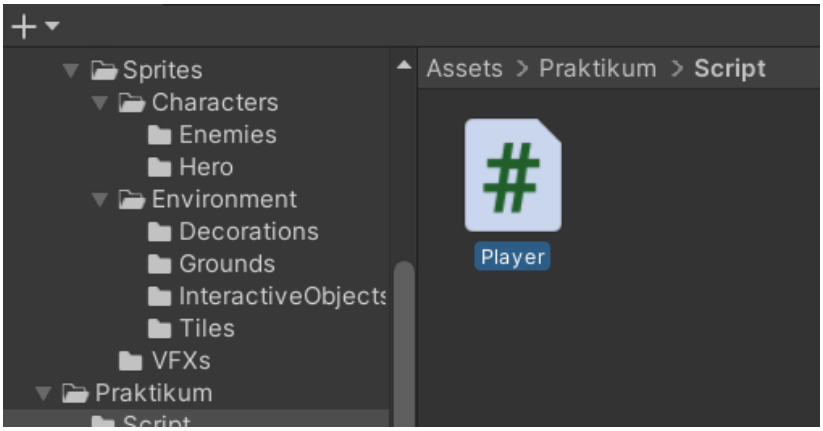
Gambar 8.6 Tampilan *Folder Script*

7. Kemudian masuk kedalam folder Script, lalu klik kanan, create > C# Script, beri nama “player”.



Gambar 8.7 Tampilan *Script Player*

8. Drag & drop script player kedalam Hirarki HideIdle1, kemudian klik 2x pada script player maka akan masuk kedalam text editor.



Gambar 8.8 Tampilan *Drag & Drop Script Player*

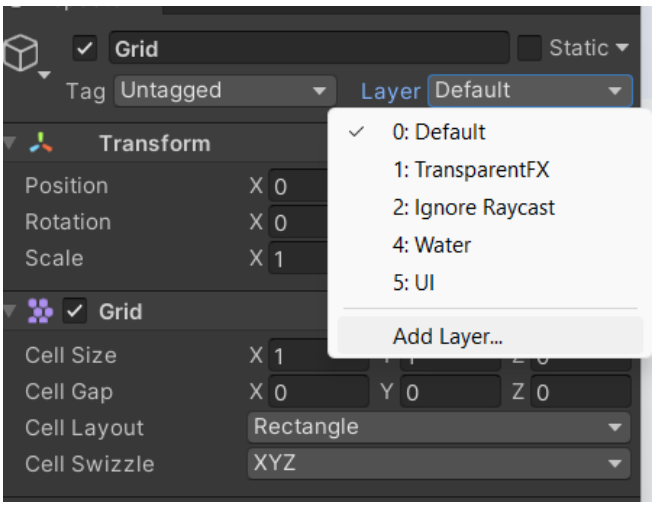
9. Lalu masukan source code dibawah ini, dan pastikan nama public class harus sama dengan nama file yang dibuat.

```
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&lt;Rigidbody2D&gt;();
}
void Update ()
{
horizontalValue                                                =
   Input.GetAxisRaw(&quot;Horizontal&quot;);
}
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 &amp;&amp; dir &lt; 0)
{
// ukuran player
transform.localScale = new Vector3(-1, 1, 1);
facingRight = false;
}
else if (!facingRight &amp;&amp; dir &gt; 0)
{
// ukuran player
transform.localScale = new Vector3(1, 1, 1);
facingRight = true;
}
#endregion
}
```

10. Kemudian play dengan menekan keybord “a” untuk arah kiri dan tekan “d” untuk arah kanan

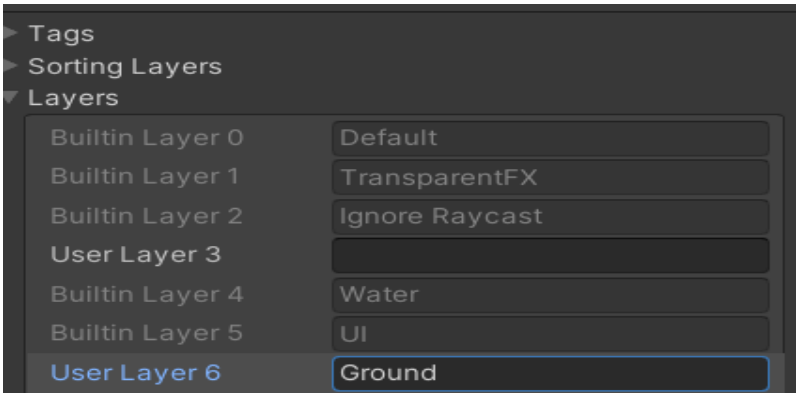
Gambar 8.9 Tampilan *left arrow dan right arrow*

11. Lalu buat player loncat menggunakan spasi, dengan cara membuat GroundCheck, klik Grid pada Hirarki, pergi ke inspector, pilih layer, Klik Add Layer



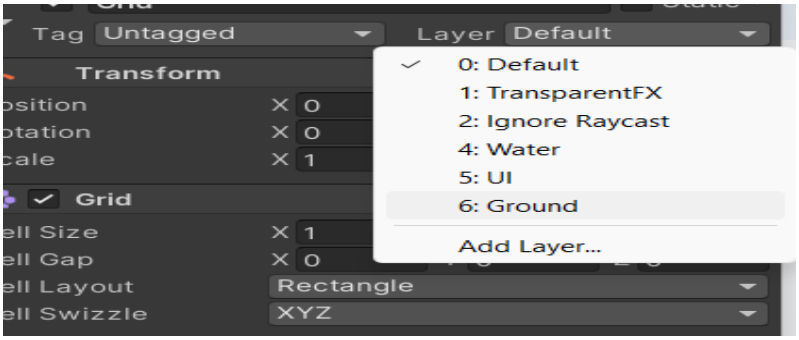
Gambar 8.10 Tampilan *Grid Add Layer*

12. Kemudian isi “Ground” pada user layer 6.



Gambar 8.11 Tampilan User layer 6

13. Lalu ubah layer menjadi Ground.

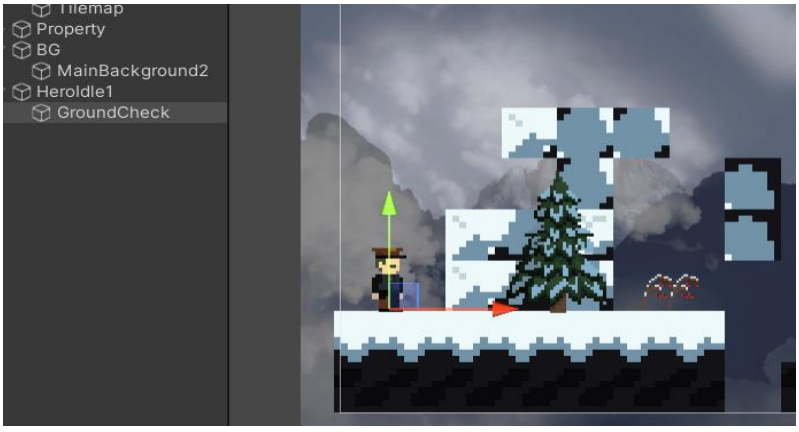


Gambar 8.12 Tampilan *Layer Ground*

14. Klik kana pada HiroIdle1 > Create Empty, beri nama GroundCheck.

Gambar 8.13 Tampilan *GroundCheck*

15. Klik pada Hirarki GroundCheck, gunakan Move Tools lalu pindahkan ke bagian bawah player.



Gambar 8.14 Tampilan Move Tools Player

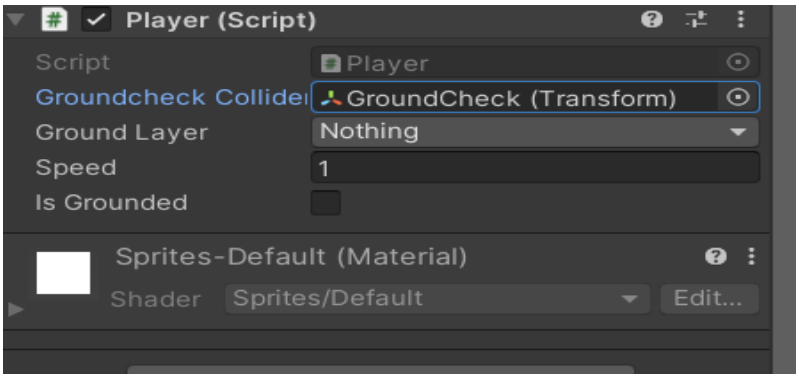
16. Kembali ke script Player tambahkan source code berikut ini

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

17. Lalu buat void groundcheck dibawah void FixedUpdate & tambahkan GroundCheck(); pada void FixedUpdate

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

18. Klik HiroIdle1, ke inpector ke effect Player script di bagian "Goruncheck Collider" tekan icon lalu pilih yang GorundCheck Transform, dan pada Ground Layer pilih Ground



Gambar 8.15 Tampilan Player (Script)

19. Kemudian buat player melompat tambahkan script berikut.

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

20. Tambahkan script berikut di bagian void update

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

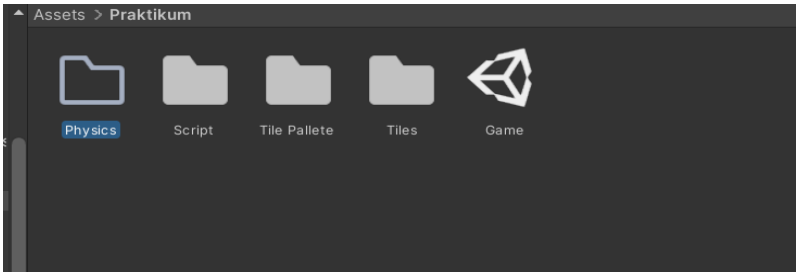
21. Tambahkan juga jump pada parameter Move.

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

22. Lalu tambahkan juga script berikut pada Void Move.

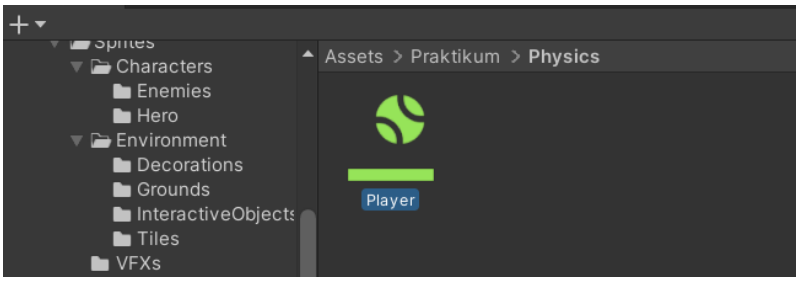
```
Void Move(float dir, bool jumpflag)
if(isGrounded &amp;&amp; jumpflag)
{
isGrounded = false;
jumpflag = false;
rb.AddForce(new Vector2(0f, jumpPower));
}
```

23. Buat folder baru di praktikum bernama “Physic”.



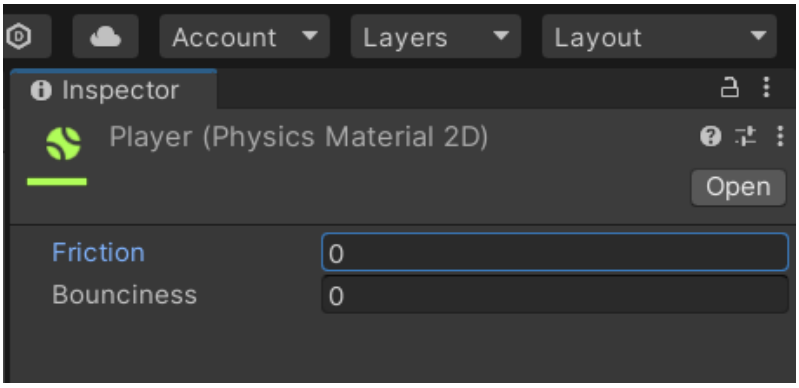
Gambar 8.16 Tampilan Folder Physic

24. Klik didalam kanan pada folder Physic, pilih create > 2D > Physical material 2d, beri nama “Player”.



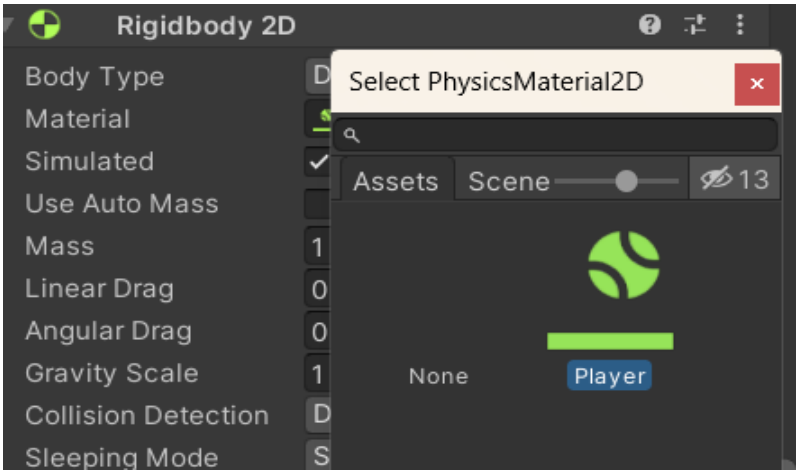
Gambar 8.17 Tampilan Physical Player

25. Kemudian klik player (physic material 2D) dibagian inpector, friction & bounces ubah mencadi 0.



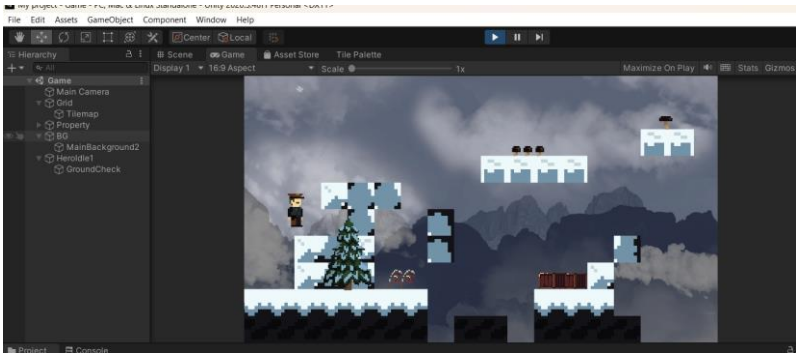
Gambar 8.18 Tampilan Player Friction

26. Klik Hirarki pilih HiroIdle1 pada inspector cari Rigidbody 2D, lalu klik icon lalu buka box select physics material 2D, pilih asset player yang sudah kita buat tadi.



Gambar 8.19 Tampilan Rigidbody 2D

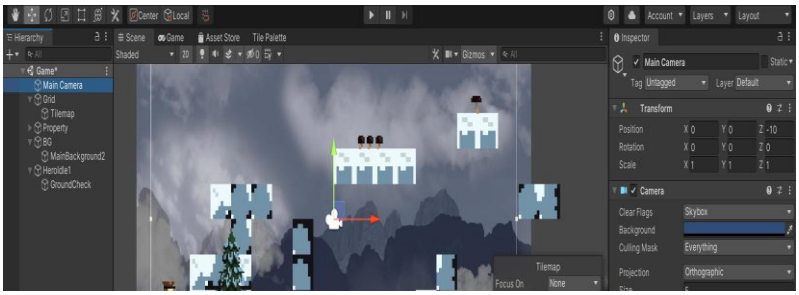
27. Tekan play, maka player bisa melompat dengan menekan spasi



Gambar 8.20 Tampilan player melompat

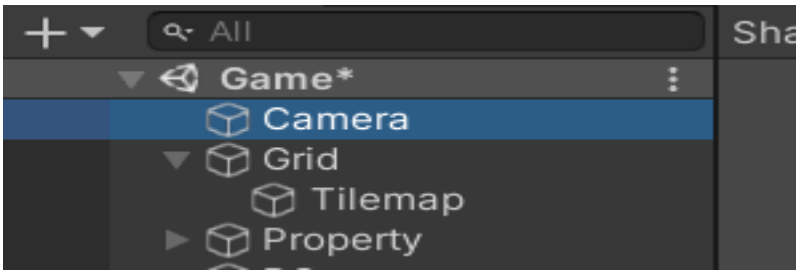
B. Camera Movement

1. Pada Hirarki Property ubah inspector pada tag main camera menjadi untaged



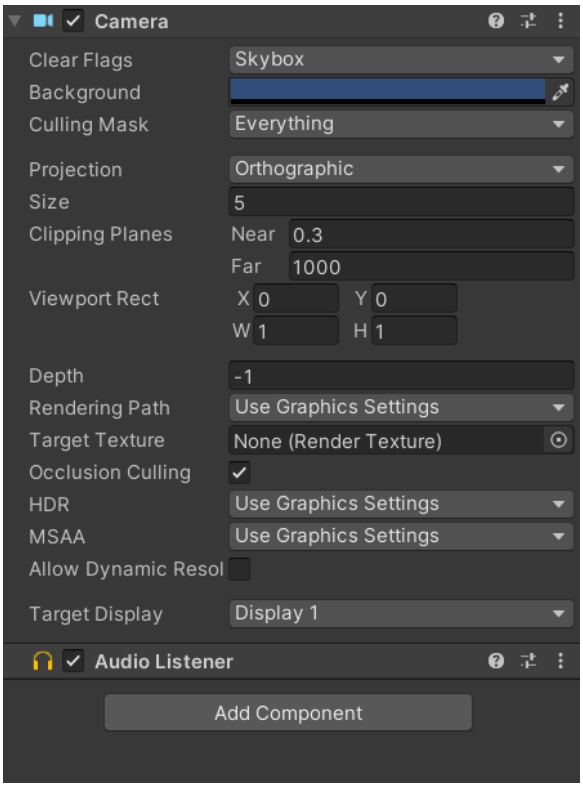
Gambar 8.21 Tampilan Property

2. Pada Effect camera pilih Remove Componet, dan klik kanan pada hirarki > Create Empty, dan beri nama Camera



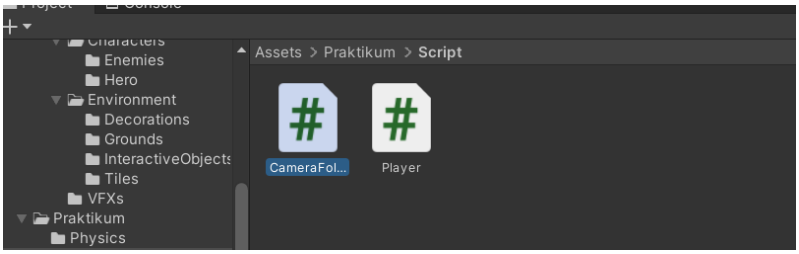
Gambar 8.22 Tampilan Camera

3. Kemudian sesuaikan setting layer camera



Gambar 8.23 Tampilan Setting Layer Camera

4. Buat file script baru di folder Script dengan nama “CameraFollow”

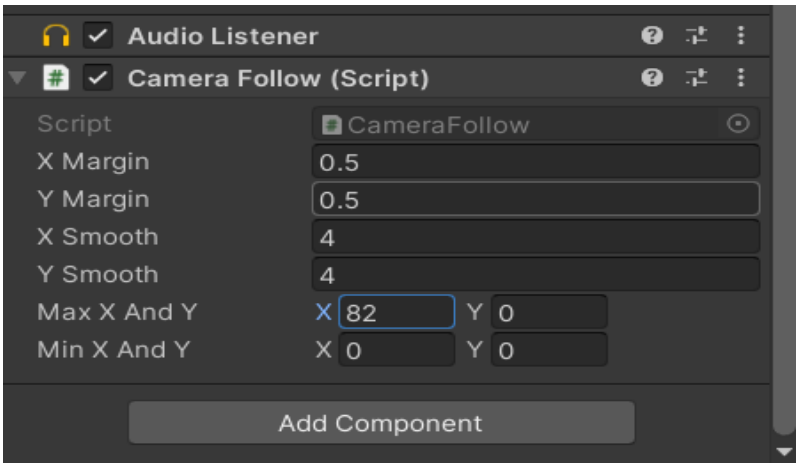


Gambar 8.24 Tampilan Script CameraFollow

5. Kemudian klik 2x pada CameraFollow tuliskan script berikut ini

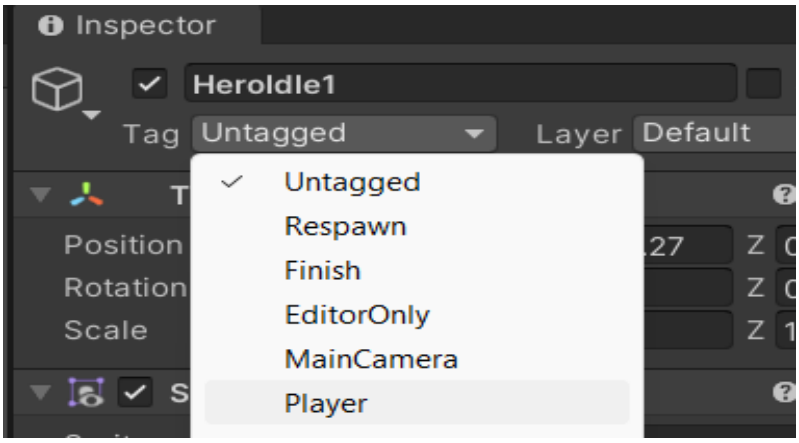
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class CameraFollow : MonoBehaviour
{
public float xMargin = 0.5f;
public float yMargin = 0.5f;
public float xSmooth = 4f;
public float ySmooth = 4f;
public Vector2 maxXAndY;
public Vector2 minXAndY;
private Transform player;
void Awake()
{
player =
GameObject.FindGameObjectWithTag(&quot;Player&quot;).t
      ransform;
}
bool CheckXMargin()
{
return Mathf.Abs(transform.position.x -
player.position.x) &gt; xMargin;
}
bool CheckYMargin()
{
return Mathf.Abs(transform.position.y -
player.position.y) &gt; yMargin;
}
void FixedUpdate()
{
TrackPlayer();
}
void TrackPlayer()
{
float targetX = transform.position.x;
float targetY = transform.position.y;
if (CheckXMargin())
targetX = Mathf.Lerp(transform.position.x,
er.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);
}
```

6. Lalu drag & drop script CameraFollow kedalam Layer Camera, kemudian klik camera, buka inspector setting menjadi berikut



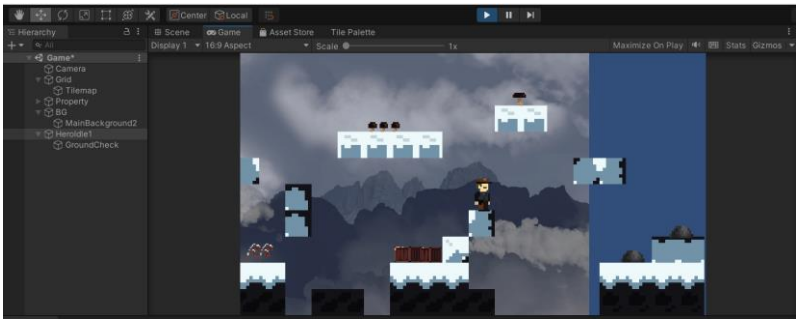
Gambar 8.25 Tampilan Setting CameraFollow

7. Ubah tag di HiroIdle1, Untagged menjadi “Player”



Gambar 8.26 Tampilan Untagged Player

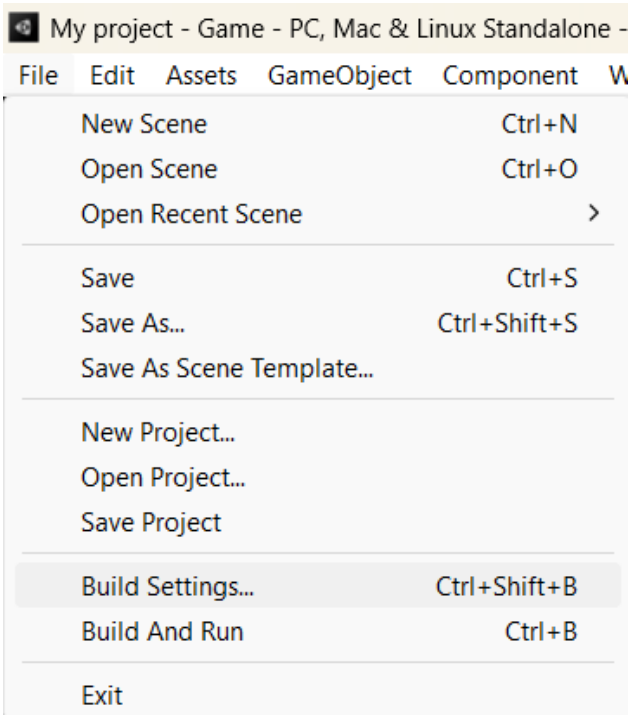
8. Lalu tekan play untuk menjalankan, maka camera akan mengikuti pergerakan karakter.



Gambar 8.27 Tampilan play pergerakan camera

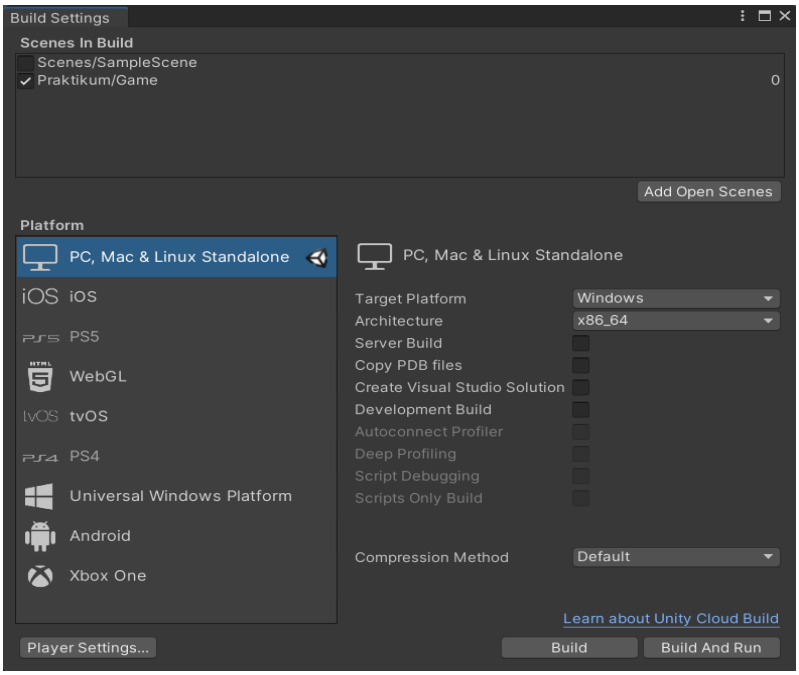
C. Render

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



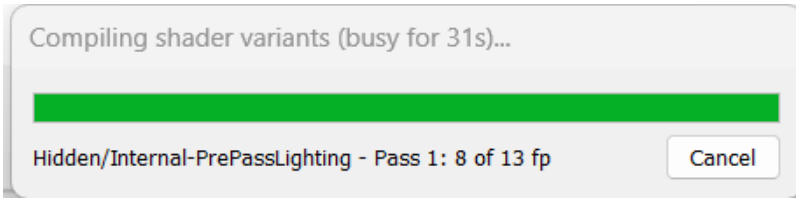
Gambar 8.28 Tampilan File Build Setting

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



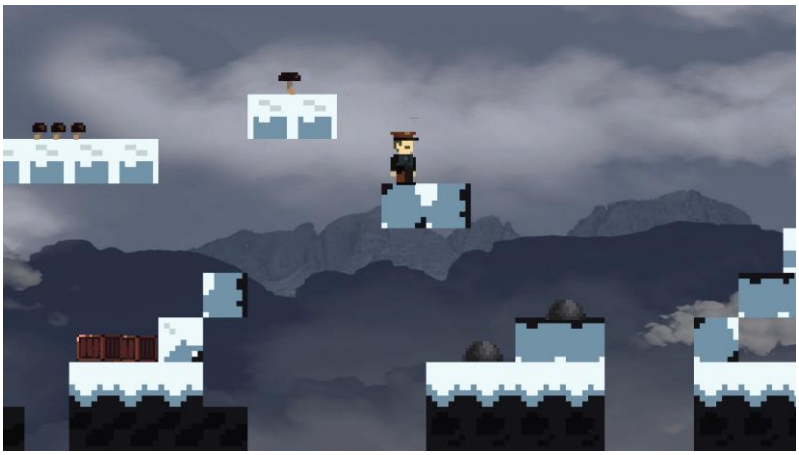
Gambar 8.29 Tampilan Setting Build

3. Pilih dimana Project disimpan, dan tunggu hasilnya



Gambar 8.30 Tampilan Tunggu Proses Simpan

4. Lalu pilih poject yang sudah di render klik 2x untuk melihat hasilnya



Gambar 8.31 Tampilan Hasil

D. Link Pengumpulan Github

Link :

KUIS

Menjelaskan Source Code dibawah ini:

Soal kuis Bab 8

```
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 :

Pada source code diatas pertama import namaspace System.Collections, System.Collections.Generic, dan UnityEngine, kemudian pada kelas camerafollow kita deklarasikan sebagai publik dan mawarisi MonoBehaviour, kemudian pada variabel player dideklarasikan sebagai private, lalu pada method update dipanggil sekali untuk per frame agar meng-update posisi kamera, lalu atur ulang nilai x diambil dari posisi pemain (player.position.x), sementara nilai y dan z tetap pada posisi kamera saat ini (transform.position.y dan transform.position.z).

Kuis CameraFollow

```
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 :

Pada source code diatas impor terlebih dahulu namespace dan mendefinisikan sebuah class CameraFollow yang mewarisi dari MonoBehaviour. Kemuadian declarasikan variabel player bertipe transform yang dapat diatur agar bersifat private , kemudian pada method update dipanggil sekali untuk per frame.