TUGAS PERTEMUAN: 8

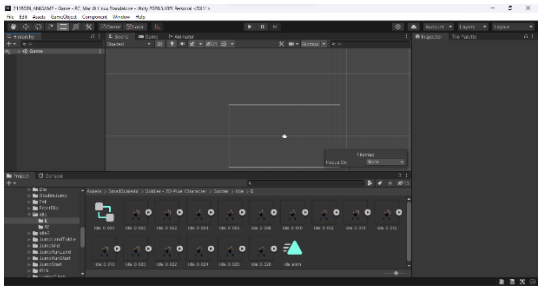
CAMERA & CHARACTER MOVEMENT

NIM	:	2118069
Nama	:	Michael Kevin Adinata
Kelas	:	A
Asisten Lab	:	Devina Dorkas Manuela (2218108)
Baju Adat	:	Baju Kain Rumput (Provinsi Indonesia Timur)
Referensi	:	https://i0.wp.com/www.romadecade.org/wp-content/uploads/2021/11/Gambar-Baju-Kain-Rumput-Papua-Barat.jpg?w=765&ssl=1

8.1 Tugas 8 : Membuat Pergerakan Player dan Camera

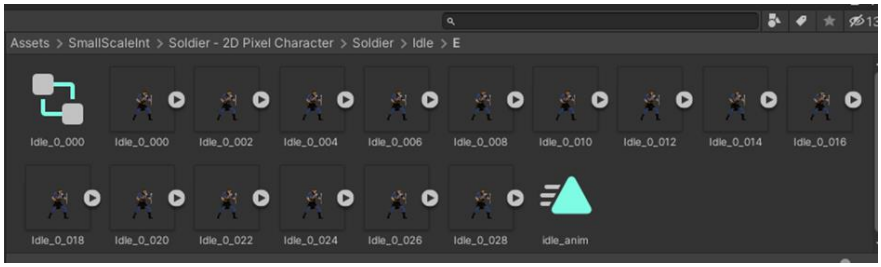
A. Membuat Pergerakan Player

1. Buka project unity yang sudah dibuat sebelumnya.



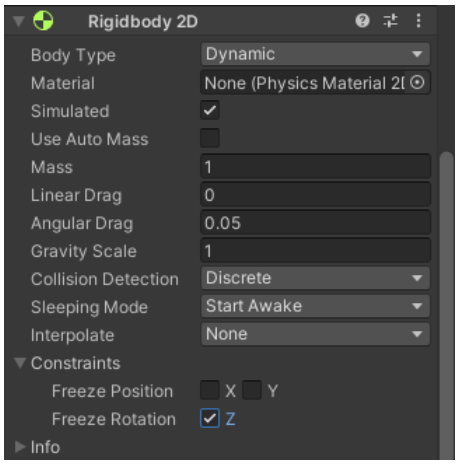
Gambar 1.1 Tampilan Unity

2. Tambahkan player, cari yang idle, lalu import ke dalam hirarki.



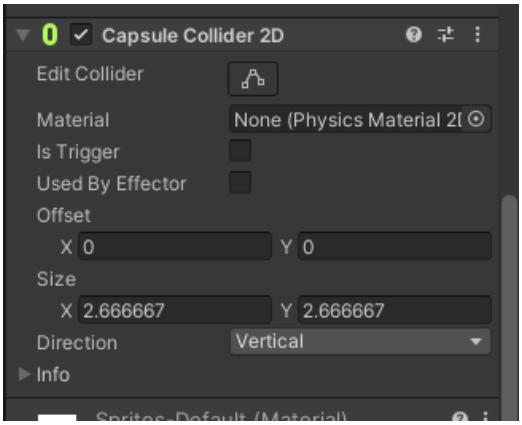
Gambar 1.2 Tampilan Import Karakter

3. Klik player pada hirarki, lalu pergi ke Inspector cari component Rigidbody 2D, lalu centang Freeze Rotation Z.



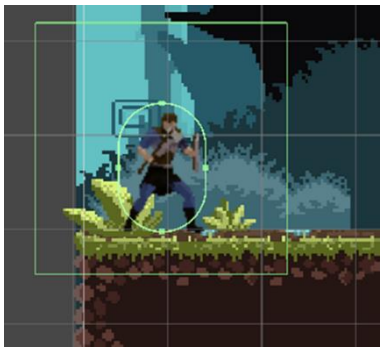
Gambar 1.3 Tampilan Freeze Rotation

4. Lalu tambahkan component Capsule Collider, lalu edit Collider.



Gambar 1.4 Tampilan Tambah Component

5. Lalu samakan garis oval degan karakternya.

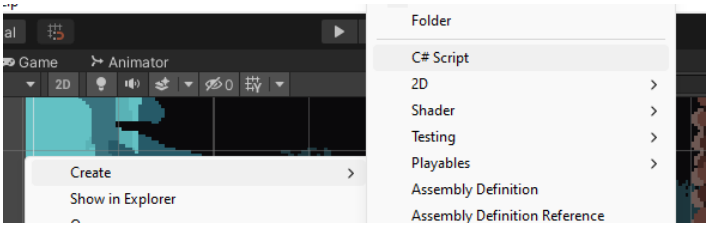


Gambar 1.5 Tampilan Mencocokkan Garis

6. Buka folder praktikum, lalu buat folder baru bernama Script.

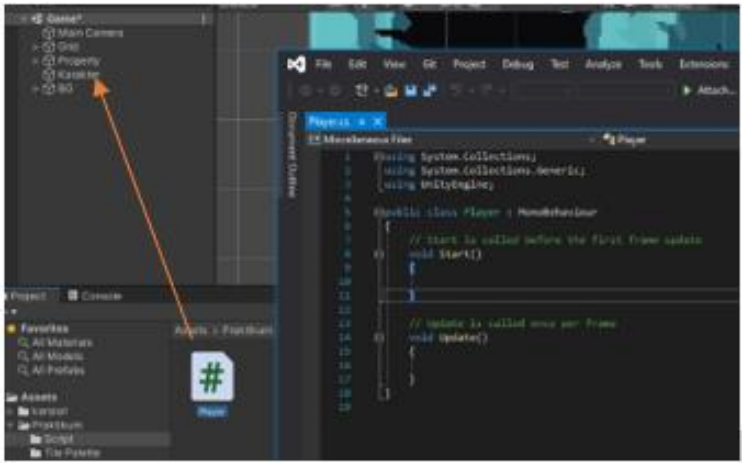
Gambar 1.6 Tampilan Folder Script

7. Masuk ke folder Script, lalu buat C# Script dan beri nama Player



Gambar 1.7 Tampilan Buat C# Script

8. Drag n drop Script kedalam hirarki player, lalu klik dua kali pada script maka akan membuka Visual Studio.

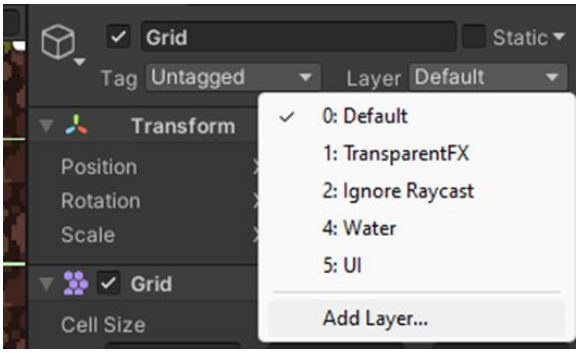


Gambar 1.8 Tampilan Script Player

9. Lalu pada text editor, masukkan source code dibawah ini.

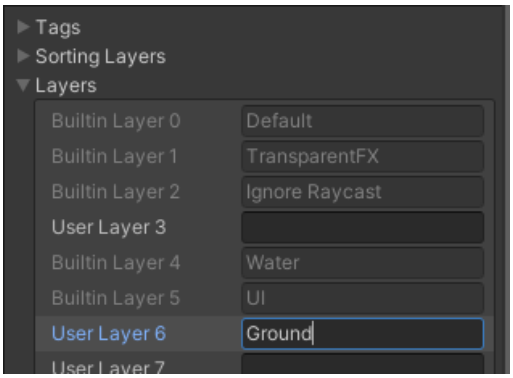
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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. Setelah itu, membuat GroundCheck untuk membuat karakter lompat, dengan cara, klik Grid pada Hierarchy, pergi ke inspector, pilih Layer, Klik Add Layer.



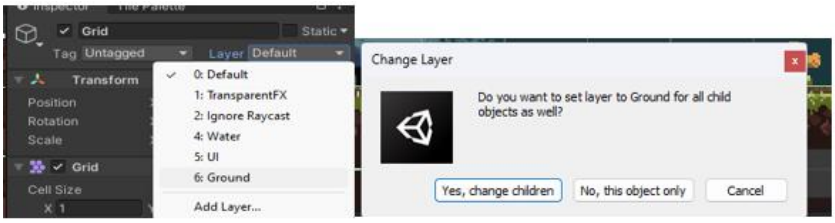
Gambar 1.9 Tampilan Add Layer

11. Lalu isikan “Ground” pada User Layer 6.



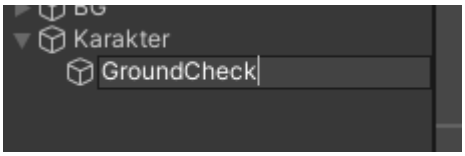
Gambar 1.10 Tampilan User Layer

12. Kemudian ubah layer menjadi Ground, jika ada peringatan, klik Yes



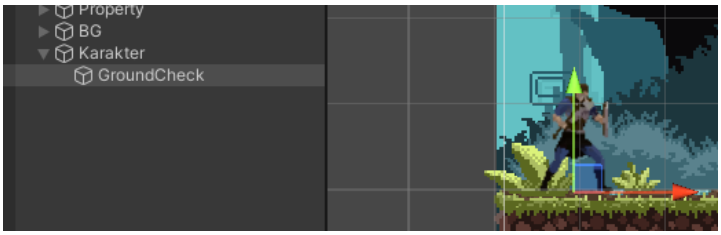
Gambar 1.11 Tampilan Ubah Layer

13. Kemudian klik kanan pada Karakter, lalu pilih Create Empty, beri nama GroundCheck.



Gambar 1.12 Tampilan Create Empty

14. Klik GroundCheck, lalu gunakan Move Tools untuk memindahkan ke bagian bawah Karakter.



Gambar 1.13 Tampilan Geser GroundCheck

15. Kemudian kembali ke Script Player, tambahkan kode dibawah ini.

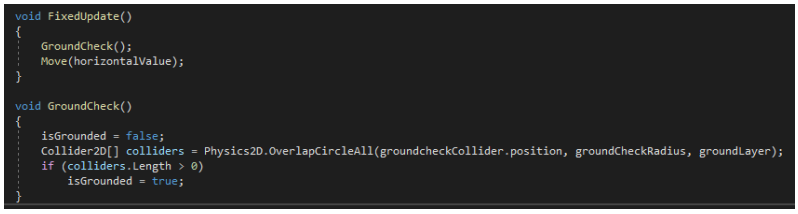
```
[SerializeField] Transform groundcheckCollider;
[SerializeField] LayerMask groundLayer;

const float groundCheckRadius = 0.2f; // +
[SerializeField] float speed = 1;
float horizontalValue;

[SerializeField] bool isGrounded; // +
bool facingRight;
```

```
Rigidbody2D rb;
[SerializeField] Transform groundcheckCollider;
[SerializeField] LayerMask groundLayer;

const float groundCheckRadius = 0.2f; // +
[SerializeField] float speed = 1;
float horizontalValue;

[SerializeField] bool isGrounded; // +
bool facingRight;
```

Gambar 1.14 Tampilan Script Player

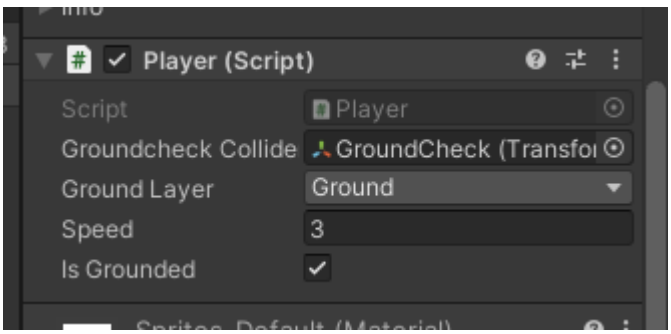
16. Setelah itu buat void ground check dibawah void fixedUpdate & tambahkan Ground Check(); pada void fixedUpdate

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

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



Gambar 1.15 Tampilan Script GroundCheck

17. Klik Karakter, lalu pergi ke inspector lalu cari Player script, setelah itu di bagian “Groundcheck Collider” tekan icon plus, lalu pilih Object GroundCheck Transform, dan pada Ground Layer pilih Ground



Gambar 1.16 Tampilan Atur Properti

18. Kembali ke script, tambahkan script dibawah untuk membuat karakter melompat.

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

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

Gambar 1.17 Tampilan Script Lompat

19. Tambahkan juga script dibawah pada fungsi void update.

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

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

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

Gambar 1.18 Tampilan Tambahan Script

20. Tambahkan parameter jump pada fungsi Move.

```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);
}
    void Player.Move(float dir)
```

Gambar 1.19 Tampilan Tambah Parameter

21. Setelah itu, tambahkan script dibawah pada void Move

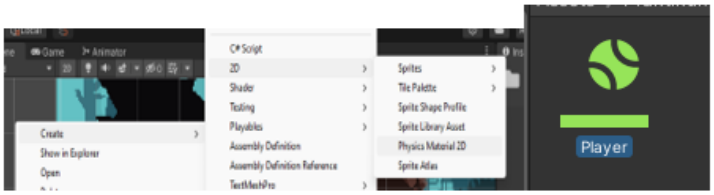
```
void Move(float dir, bool jumpflag)
{
    if (isGrounded && jumpflag)
    {
        isGrounded = false;
        jumpflag = false;
        rb.AddForce(new Vector2(0f, jumpPower));
    }
```

Gambar 1.20 Tampilan Script Move

22. Lalu buat folder baru, beri nama Physics

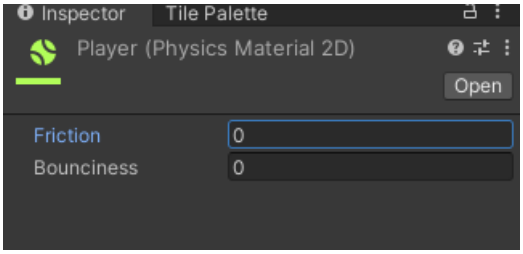
Gambar 1.21 Tampilan Buat Folder

23. Buat Physical Material 2D dan beri nama Player di dalam folder Physics



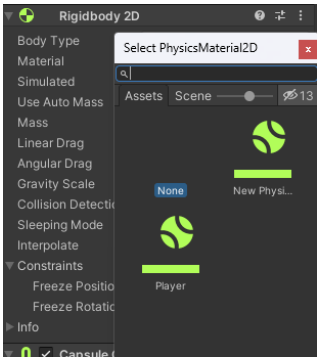
Gambar 1.22 Tampilan Buat Physics

24. Klik Player, lalu pada bagian Inspector, ubah Friction dan Bounciness menjadi 0



Gambar 1.23 Tampilan Buat Folder

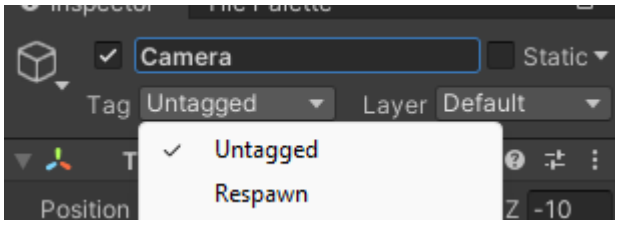
25. Klik Karakter pada hirarki, lalu pada Inspector cari Rigidbody 2D setelah itu klik icon plus untuk membuka box select physhics material 2D, lalu pilih asset Player yang sudah dibuat



Gambar 1.24 Tampilan Pilih Physics

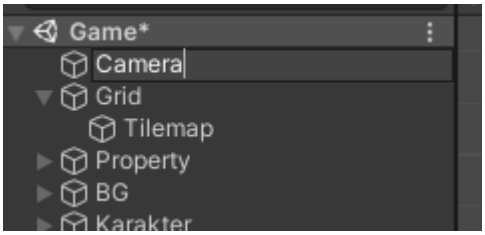
B. Membuat Pergerakan Kamera

1. Pada hirarki, pilih Main Camera, lalu ubah tag menjadi Untagged



Gambar 1.25 Tampilan Ubah Tag

2. Rename Main Camera menjadi Camera



Gambar 1.26 Tampilan Rename Object

3. Masuk pada folder script, buat C# Script baru dan beri nama CameraFollow

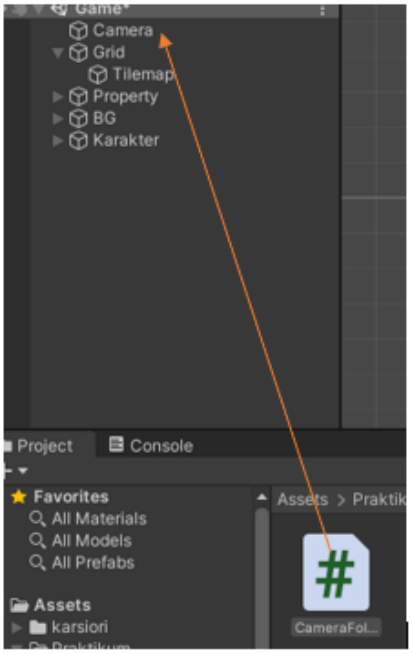
Gambar 1.27 Tampilan Buat Script

4. Klik dua kali, lalu masukkan source code dibawah 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()
```

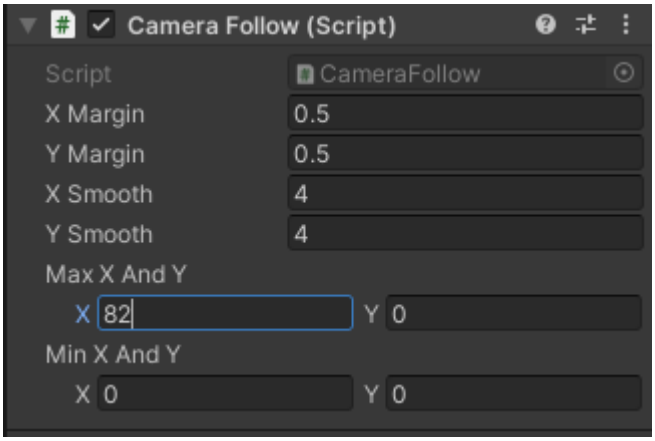
```
    {
        player =
GameObject.FindGameObjectWithTag("Player").transform;
    }

    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);
    }
}
```

5. Drag n drop Script CameraFollow ke dalam Object Camera.



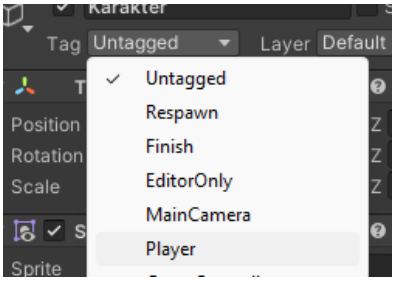
Gambar 1.28 Tampilan Import Script

6. Lalu klik Camera, pergi ke Inspector lalu cari Script, ubah pada bagian Max X And Y seperti gambar.



Gambar 1.29 Tampilan Ubah Properti

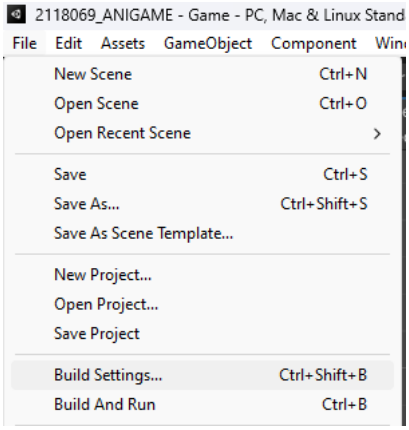
7. Ubah tag pada Karakter menjadi Player.



Gambar 1.30 Tampilan Ubah Tag

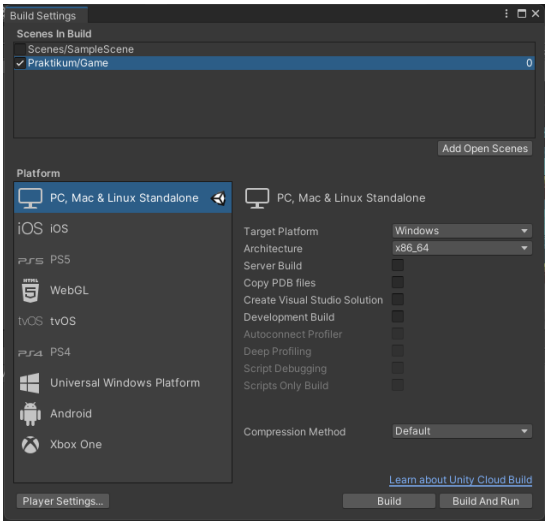
C. Render

1. Pergi ke menu File lalu cari Build Settings.



Gambar 1.31 Tampilan Menu File

2. Pada Build Setting ini pilih PC, Mac & Linux, lalu centang hanya Scene Game yang sudah dibuat, setelah itu klik Build, tunggu hingga selesai



Gambar 1.32 Tampilan Build Setting

D. 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);
    }

}
```

Penjelasan :

Source code diatas adalah source code CameraFollow, pertama membuat variable player dengan tipe data Transform yang dideklarasikan sebagai private dengan atribut SerializeField, yang berarti variable player ini dapat diatur pada Inspector di Unity. Lalu tedapat void Update, method ini dipanggil setiap frame, didalam method ini posisi kamera diatur ulang setiap framenya. Posisi X kamera diatur sesuai dengan posisi X dari Player, lalu Posisi Y dan Z kamera akan tetap sama seperti sebelum nya.

E. Link Github Pengumpulan

https://github.com/MichaelKevv/2118069_PRAK_ANIGAME