TUGAS PERTEMUAN: 8

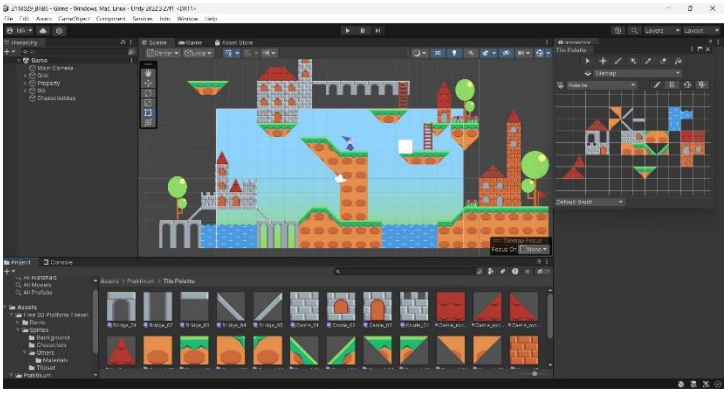
Camera & Character Movement

NIM	:	2118029
Nama	:	Mohammad Arvin Manda Aradhana
Kelas	:	A
Asisten Lab	:	APRILLIA DWI DYAH S. (2118143)

1.1 Tugas 1 :

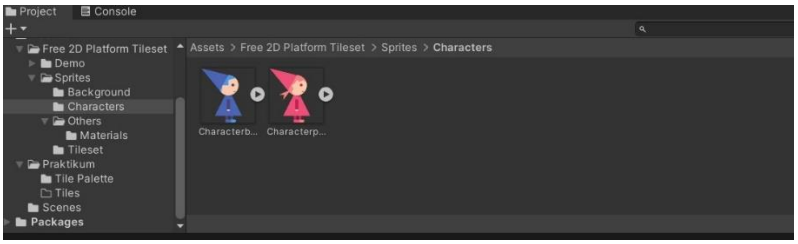
A. Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

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



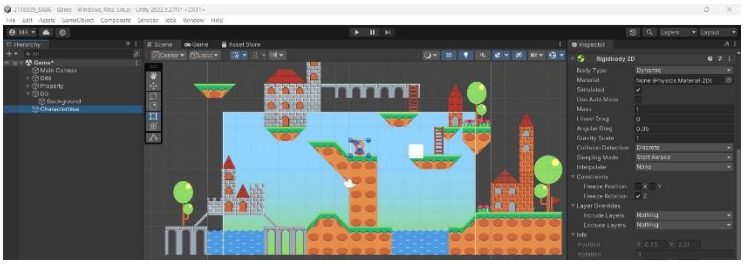
Gambar 8.1 Membuka file project file sebelumnya

2. Tambahkan pemain bernama character, impor ke dalam Hirarki



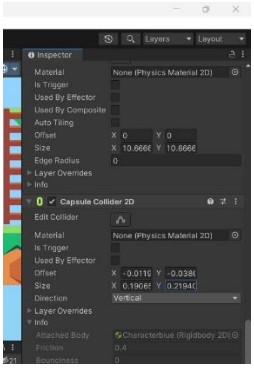
Gambar 8.2 Menambahkan character

3. Klik character, tambahkan Komponen Rigidbody 2D, sesuaikan pengaturannya seperti gambar berikut, dan centang opsi Freeze Rotation Z.



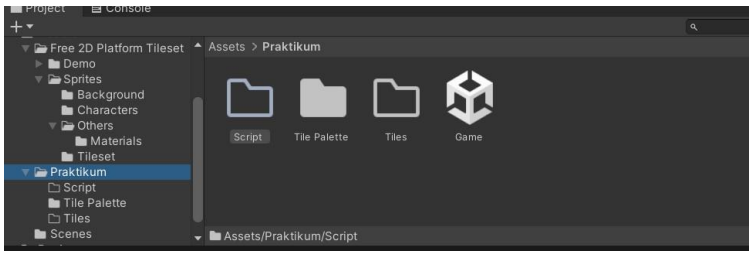
Gambar 8.3 Menambahkan komponen *Rigibody 2D*

4. Kemudian sesuaikan bentuk oval dengan karakter tersebut atau masukkan nilai Offset X, Y serta Size X, Y sesuai kebutuhan.



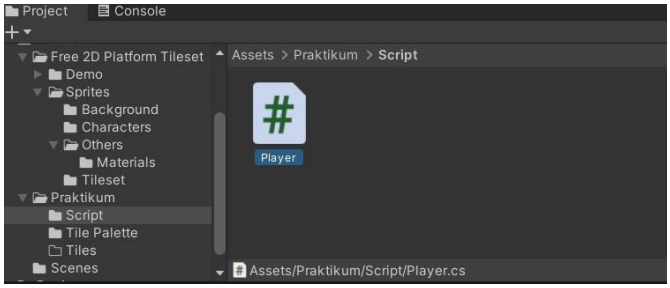
Gambar 8.4 Menambahkan *Capsule Collider*

5. Buka Folder praktikum, kemudian buat folder baru dengan nama Script.



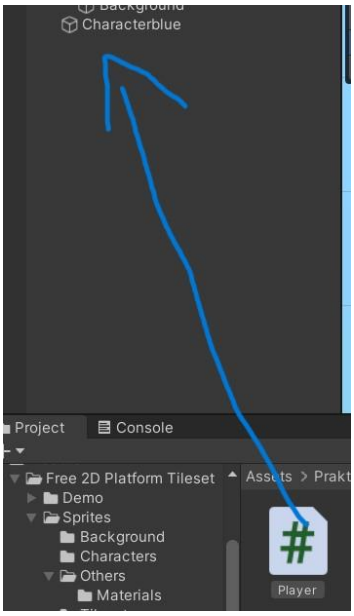
Gambar 6.5 Membuat folder script

6. Masuk ke dalam folder Script, lalu buat skrip C# dan beri nama Player.



Gambar 8.6 Membuat file skrip C#

7. Seret dan lepas skrip Player ke dalam Hirarki pada *Character*, lalu klik dua kali pada skrip Player untuk membukanya di text editor seperti ini.



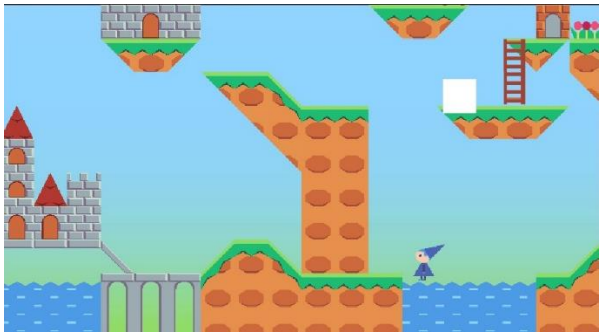
Gambar 8.7 Memasukkan file skrip ke caharcter

8. Masukkan source code berikut, dan pastikan nama public class sesuai dengan nama file yang dibuat.

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

public class Player : MonoBehaviour
{
    Rigidbody2D rb;

    bool jump;

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

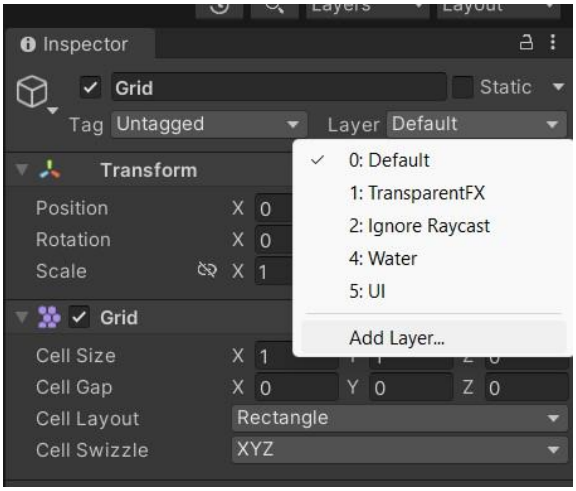
```
void Move(float dir, bool jump)
{
    #region gerak kanan kiri
    float xVal = dir * speed * 100 * Time.fixedDeltaTime;
        Vector2   targetVelocity   =   new   Vector2(xVal,
rb.velocity.y);
    rb.velocity = targetVelocity;

    // Flip the character's sprite based on direction
    if (facingRight && dir < 0)
    {
            transform.localScale   =   new   Vector3(-1   *
Mathf.Abs(transform.localScale.x), transform.localScale.y,
transform.localScale.z);
        facingRight = false;
    }
    else if (!facingRight && dir > 0)
    {
                        transform.localScale     =     new
Vector3(Mathf.Abs(transform.localScale.x),
transform.localScale.y, transform.localScale.z);
        facingRight = true;
    }
    #endregion

    // Jump logic
    if (jump && isGrounded)
    {
         rb.velocity = new Vector2(rb.velocity.x, 0f); //
Reset vertical velocity before jumping
        rb.AddForce(new Vector2(0f, jumpPower));
    }
}
}
```

9. Untuk mencoba apakah source code di atas berhasil, tekan "a" atau "left arrow" pada keyboard untuk bergerak ke kiri, dan tekan "d" atau "right arrow" untuk bergerak ke kanan.



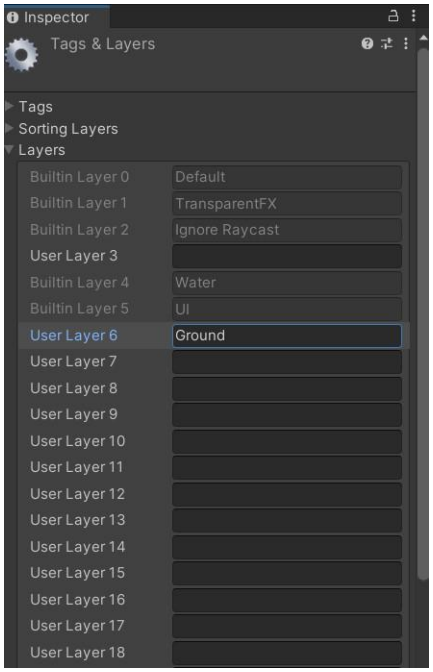
Gambar 8.8 Pergerakan karakter ketika ditekan pada keyboard

10. Untuk membuat pemain dapat melompat menggunakan tombol spasi, kita perlu membuat GroundCheck. Caranya adalah dengan mengklik Grid pada Hirarki, lalu pergi ke inspektor, pilih Layer, dan klik Tambahkan Layer.



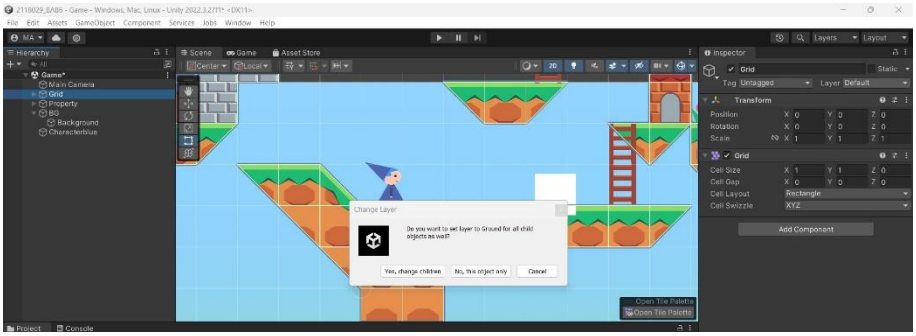
Gambar 8.10 inspector pada *grid*

11. Lalu isi “Ground” pada User Layer 6



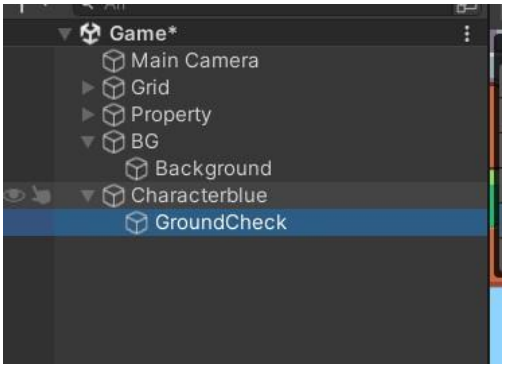
Gambar 8.11 Mengisi layer 6 menjadi *ground*

12. Ubah Layer menjadi Ground, jika muncul pop up Change Layer, klik yes saja



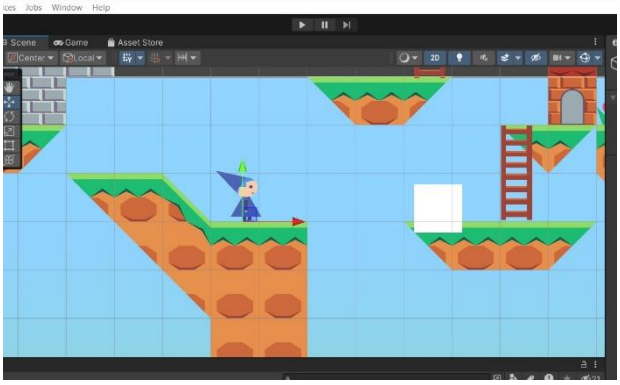
Gambar 8.12 Mengubah layer menjadi ground

13. Klik kanan pada character, lalu Create empty, beri nama GorundCheck



Gambar 8.13 Membuat GroundCheck pada hierarchy

14. Klik pada GroundCheck di Hirarki, kemudian gunakan "Move Tools" untuk memindahkannya ke bagian bawah Pemain seperti yang ditunjukkan dalam gambar berikut.



Gambar 8.14 Memindahkan character

15. Kembali ke script Player tambahkan source code seperti ini

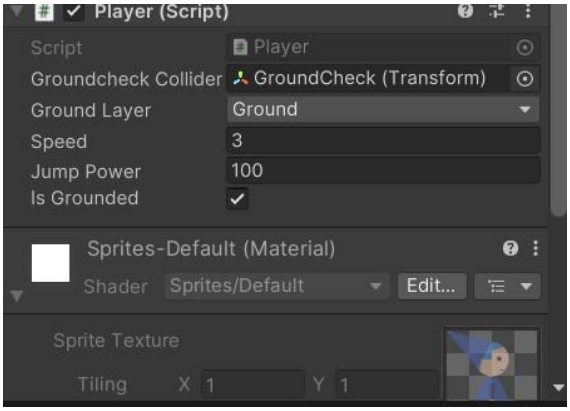
```
[SerializeField] Transform groundcheckCollider;
[SerializeField] LayerMask groundLayer;
const float groundCheckRadius = 0.2f;
[SerializeField] float speed = 1;
```

```
[SerializeField] float jumpPower = 100;
float horizontalValue;

bool isGrounded;
 bool facingRight = true;  // Set default facing direction
 to right
```

16. Buat void ground check dibawah void fixedUpdate & tambahkan GorunCheck(); pada void fixedUpdate

```
void FixedUpdate()
    {
       GroundCheck();
       Move(horizontalValue, jump);
       jump = false; // Reset jump after applying it in Move
    }
void GroundCheck()
{
    isGrounded = false;
                    Collider2D[]         colliders         =
Physics2D.OverlapCircleAll(groundcheckCollider.position,
groundCheckRadius, groundLayer);
    foreach (Collider2D collider in colliders)
    {
                                  if       (collider       !=
gameObject.GetComponent<Collider2D>())
        {
            isGrounded = true;
            break;
        }
    }
}
```

17. Klik pada player-idle-1, lalu di inspektor pergi ke efek skrip Pemain pada bagian "Collider GroundCheck", tekan ikon lalu pilih Transform GroundCheck, dan pada Layer Ground pilih Ground.



Gambar 8.15 Mengubah *GroundCheck Collider*

18. Lalu untuk membuat player melompat tambahkan script berikut

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

```
bool jump;
```

19. Tambahkan juga script berikut di bagian void update

```
horizontalValue = Input.GetAxisRaw("Horizontal");

        // Set jump to true only if the player is grounded
and the jump button is pressed
        if (Input.GetButtonDown("Jump") && isGrounded)
        {
            jump = true;
        }
```

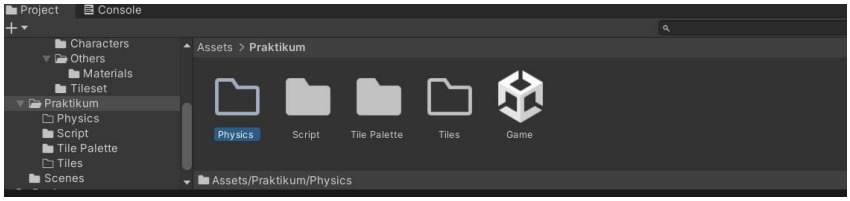
20. Tambahkan juga jump pada parameter Move

```
void FixedUpdate()
    {
       GroundCheck();
       Move(horizontalValue, jump);
        jump = false; // Reset jump after applying it in
Move
    }
```

21. Tambahkan script berikut pada void Move

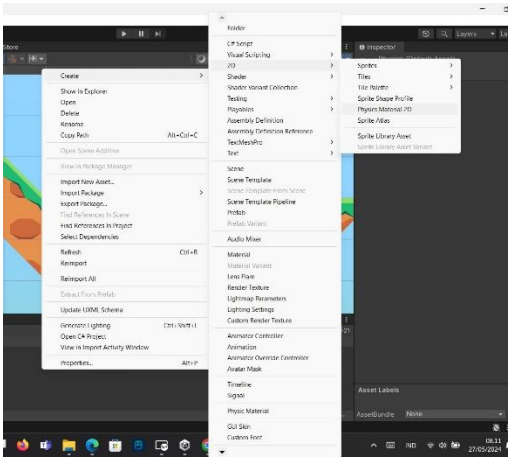
```
void Move(float dir, bool jump)
{
    #region gerak kanan kiri
    float xVal = dir * speed * 100 * Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
    rb.velocity = targetVelocity;

    // Flip the character's sprite based on direction
    if (facingRight && dir < 0)
    {
            transform.localScale = new Vector3(-1 *
Mathf.Abs(transform.localScale.x), transform.localScale.y,
transform.localScale.z);
        facingRight = false;
    }
    else if (!facingRight && dir > 0)
    {
                        transform.localScale = new
Vector3(Mathf.Abs(transform.localScale.x),
transform.localScale.y, transform.localScale.z);
        facingRight = true;
    }
    #endregion
```

22. Buat folder baru di praktikum bernama “Physics”



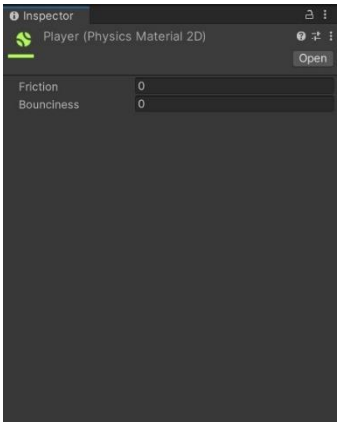
Gambar 8.16 Mengubah *GroundCheck Collider*

23. Di dalam folder Physics, buatlah folder baru bernama 2D, kemudian tambahkan bahan material fisik 2D dan beri nama "Player".



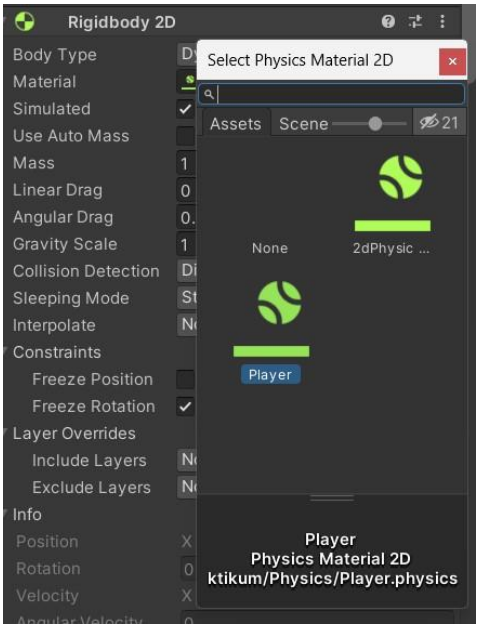
Gambar 8.17 Membuat material fisik 2D

24. Klik Player (Physics Material 2D), di bagian inspektor, atur gesekan (friction) dan pantulan (bounces) menjadi 0.



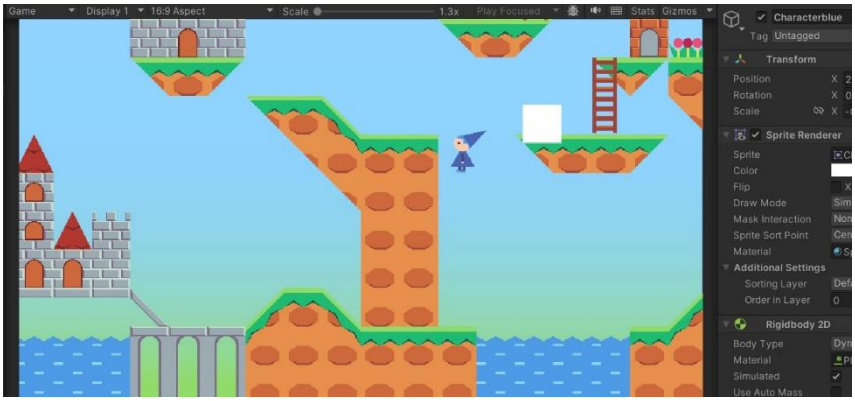
Gambar 8.18 Mengubah friction dan bounces

25. Klik Hierarchy, pilih layer player idle 1, di Inspector cari Rigidbody 2D, kemudian klik ikon untuk membuka pilihan material fisika 2D, dan pilih asset Player yang sudah dibuat sebelumnya.



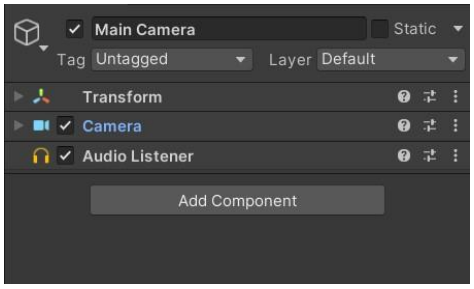
Gambar 8.19 Mengubah friction dan bounces

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



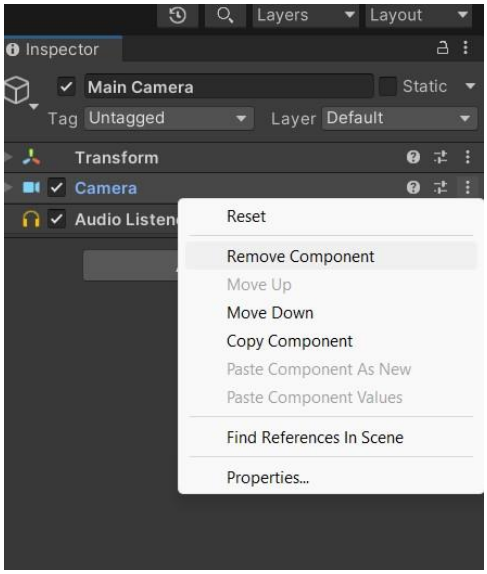
Gambar 8.20 Tampilan player saat melompat

27. Di Hirarki, ubah Inspector pada properti Main Camera menjadi "untagged".



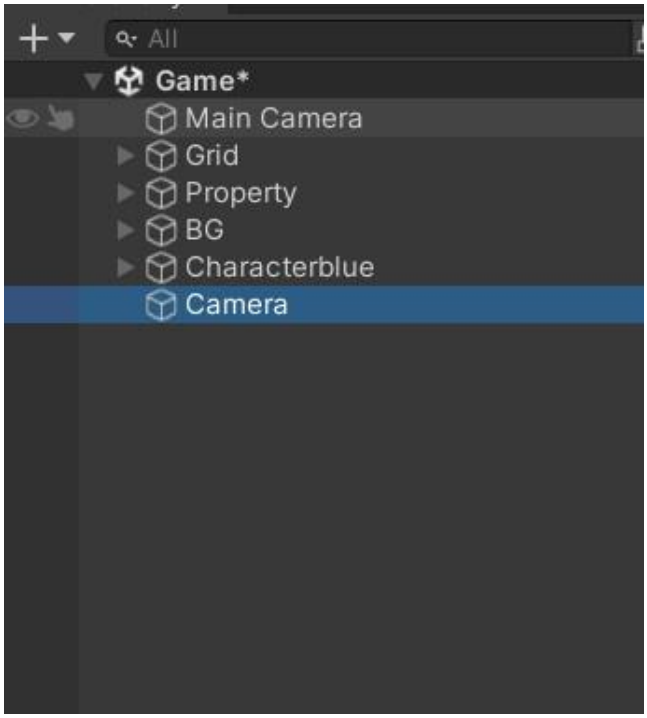
Gambar 8.21 Mengubah tag *untagged*

28. Pada Effect Camera pilih Remove Component



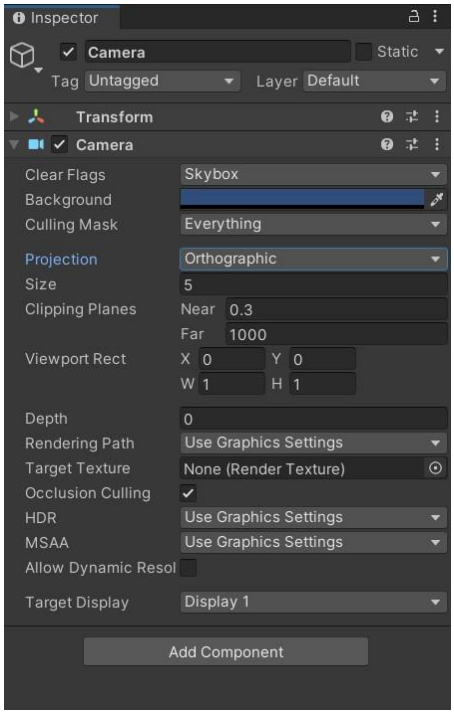
Gambar 8.22 Meremove *effect camera*

29. Buatlah objek kosong pada Hirarki, dan ganti namanya menjadi "Camera".



Gambar 8.23 Mengubah object hirarki menjadi *Camera*

30. Sesuaikan Setting Layer Camera seperti gambar dibawah ini



Gambar 8.24 Menyesuaikan settingan layer camera

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

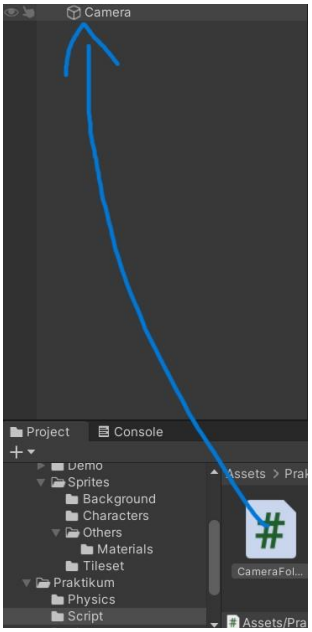
Gambar 8.25 Membuat file script *CameraFollow*

32. Lalu 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);
    }
}
```

33. Drag & drop script CameraFollow Kedalam Layer Camera

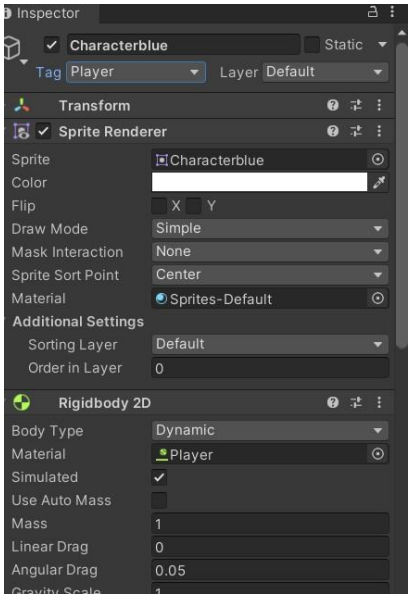


Gambar 8.26 Mendrag and drop script camerafollow

34. Setelah itu, klik pada kamera, buka inspector, dan di bagian "Camera Follow (Script)", ubah nilai Max X dan Max Y-nya.

Gambar 8.27 Mengubah nilai max s dan y

35. Ubah tag di character Untagged menjadi ”Player”

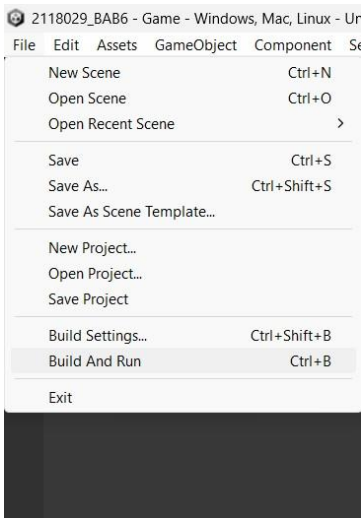


Gambar 8.28 Mengubah tag menjadi Player

36. Tekan play untuk menjalankan, maka sekarang kamera akan mengikuti pergerakan karakter

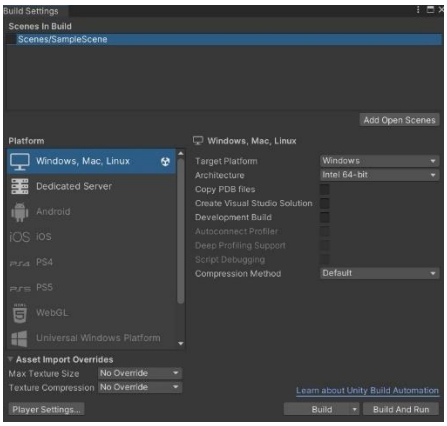
Gambar 8.29 Tampilan kamera akan mengikuti pergerakan karakter

37. Buka menu File, lalu pilih Build Settings (Ctrl + Shift + B).



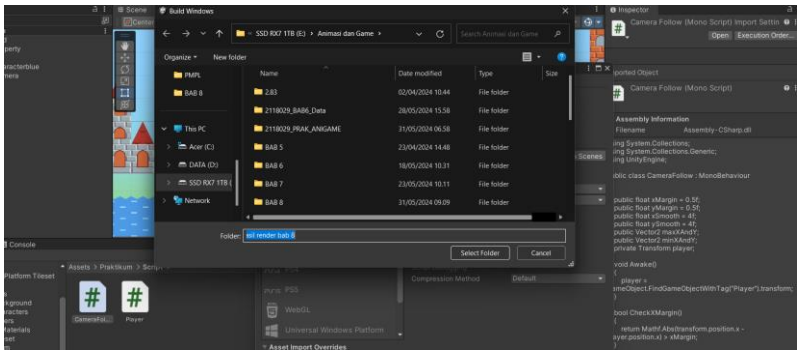
Gambar 8.30 Build and Run

38. Di pengaturan build, pilih opsi PC, Mac & Linux, lalu tekan tombol Build, dan pastikan bahwa scene di Build berada dalam proyek.



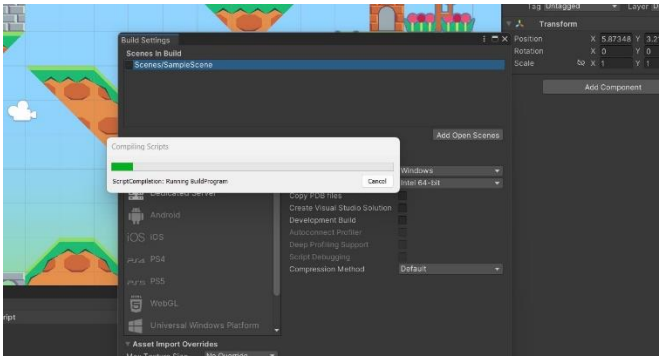
Gambar 8.30 Mensetting build

39. Pilih dimana Project disimpan, dan tunggu hasilnya



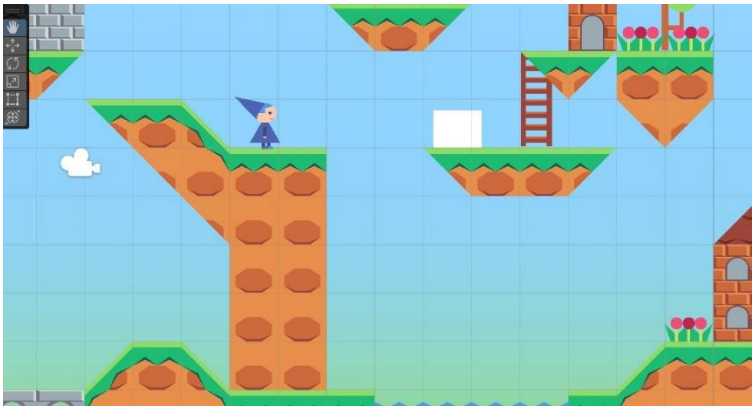
Gambar 8.31 Menyimpan project

40. Menunggu running



Gambar 8.31 Menunggu hasil running

41. Tampilan hasil build and run



Gambar 8.32 Hasil build and run

B. Kuis : Menjelaskan Source Code 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 :

Program di atas adalah skrip untuk membuat kamera mengikuti pergerakan pemain dalam permainan. Dengan menggunakan Unity dan bahasa pemrograman C#, skrip ini mengatur agar kamera bergerak horizontal mengikuti posisi pemain. Setiap frame yang diperbarui, posisi kamera diatur untuk memiliki nilai X yang sama dengan posisi X pemain,

tetapi nilai Y dan Z kamera tetap tidak berubah. Ini membuat kamera hanya mengikuti gerakan pemain secara horizontal tanpa bergerak vertikal atau kedalam. Dengan ini, kamera akan terus mengikuti pemain saat bergerak ke kanan atau kiri dalam permainan, memberikan tampilan yang konsisten dan menangkap seluruh aksi horizontal dalam permainan.