TUGAS PERTEMUAN: 10

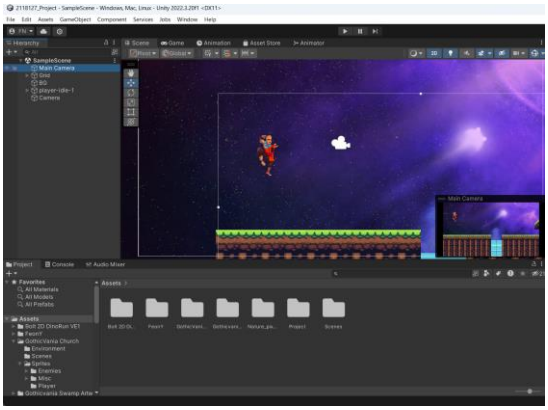
ENEMY AI AND ATTACK

NIM	:	2118127
Nama	:	Fathur Satya Nugroho
Kelas	:	D
Asisten Lab	:	WISANDO BERLIAN PANDENSOLANG (2218095)

1.1 Tugas 1 : Membuat Enemy AI dan Attack

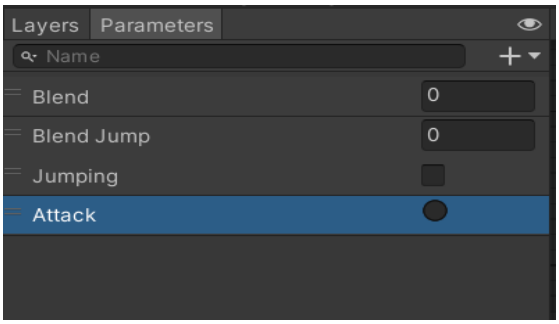
A. Membuat Mekanisme Attack

1. Bukalah Project unity bab 9 sebelumnya



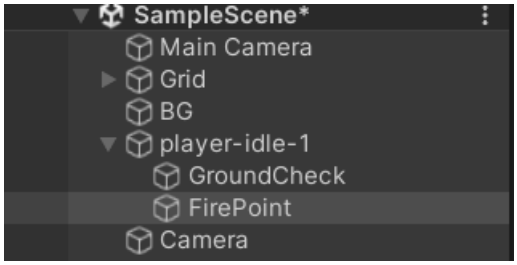
Gambar 10.1 Project Bab 9

2. Tambahkan parameter baru attack



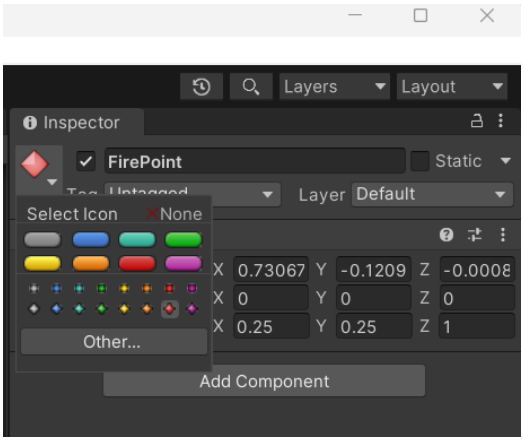
Gambar 10.2 Animator Attack

3. Buat objek baru bernama Firepoint di karakter



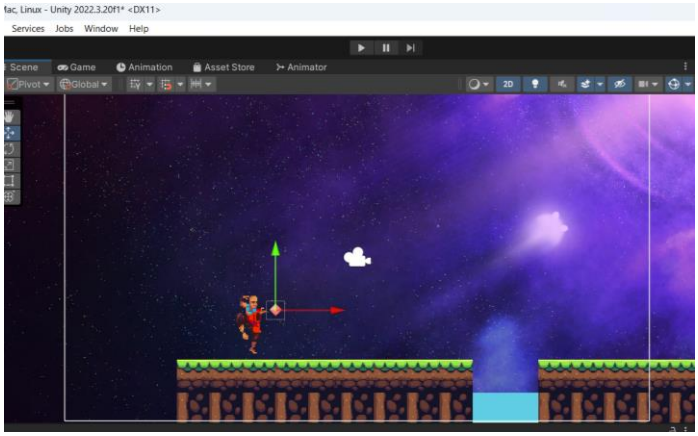
Gambar 10.3 Object Firepoint

4. Masuk pada inspector dari Firepoint , lalu rubah warna dari iconnya menjadi warna Hijau



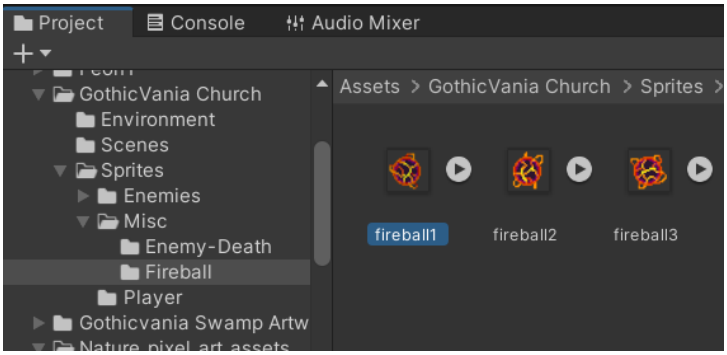
Gambar 10.4 icon Firepoint

5. setting posisi bullet seperti gambar dibawah ini



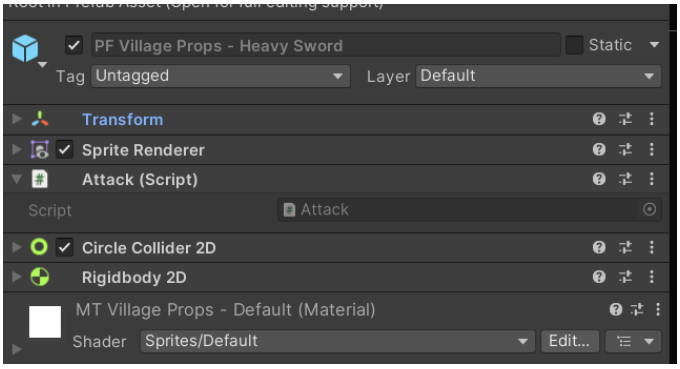
Gambar 10.5 setting Bullet

6. Drag and drop asset pedang lalu kita rename menjadi fireball



Gambar 10.6 rename Fireball

7. Masuk fireball, Add Component circle collider 2D dan rigidbody 2D.

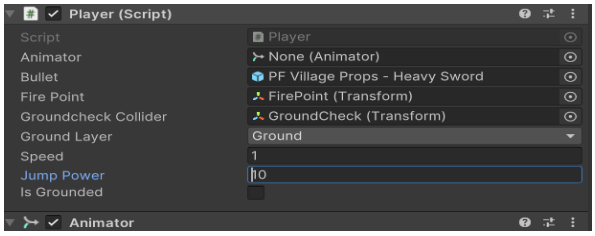


Gambar 10.7 circle collider 2D & rigidbody

8. Tambahkan source code Pada script Player.cs

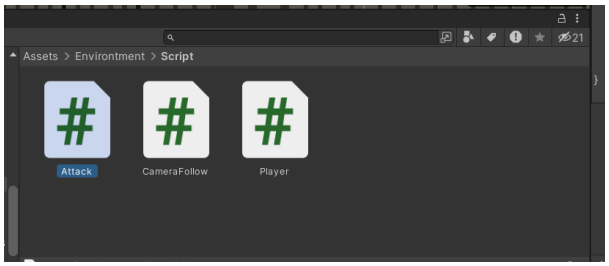
```
IEnumerator Attack()
    {animator.SetTrigger("Attack");
        yield return new WaitForSeconds(0.25f);

float direction = facingRight ? 1f : -1f;
float rotationAngle = facingRight ? 0f : 180f;
        Quaternion rotation = Quaternion.Euler(0, 0,
rotationAngle);

        GameObject Fireball = Instantiate(bullet,
Firepoint.position, rotation);
        Fireball.transform.localScale = new
Vector3(0.5f, 0.5f, 1f); // Sesuaikan ukuran sesuai
kebutuhan
       Fireball.GetComponent<Rigidbody2D>().velocity
= new Vector2(direction * 5f, 0);

        Destroy(Fireball, 2f);
   }
```

9. setting inpector pada player bagian player Script



Gambar 10.8 Inspector Players

10. buat script baru dengan nama Attack.

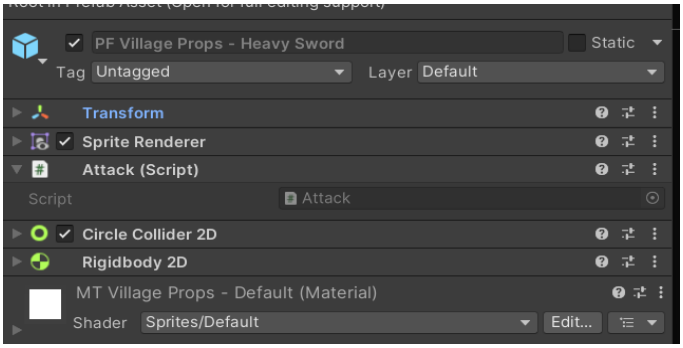


Gambar 10.9 new Script

11. Tambahkan source code berikut ke dalam file Attack.cs

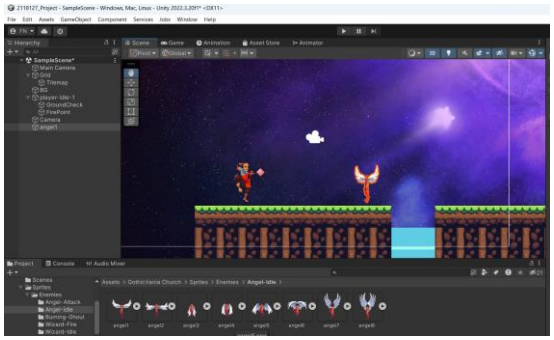
```
        using System.Collections;
        using System.Collections.Generic;
        using UnityEngine;
        public class Attack : MonoBehaviour
        {private     void     OnTriggerEnter2D(Collider2D
        collision)
            {if
        (collision.gameObject.CompareTag("Enemy"))
                {
                    Destroy(gameObject);
                    Destroy(collision.gameObject);
                }
            }
}
```

12. Drag & drop file script Attack ke objek fireball



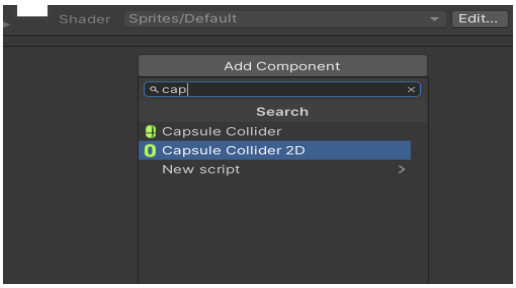
Gambar 10.10 Inspector Fireball

13. Tambahkan asset musuh 1 pada scene game



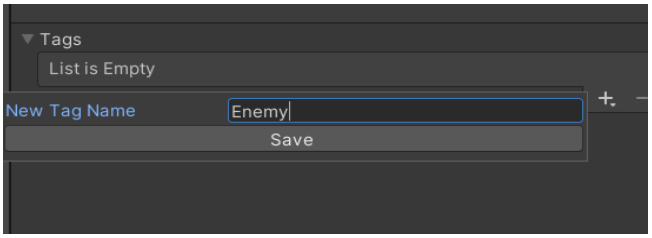
Gambar 10.11 Add musuh

14. Pada objek musuh 1 Add Component capsule collider 2D



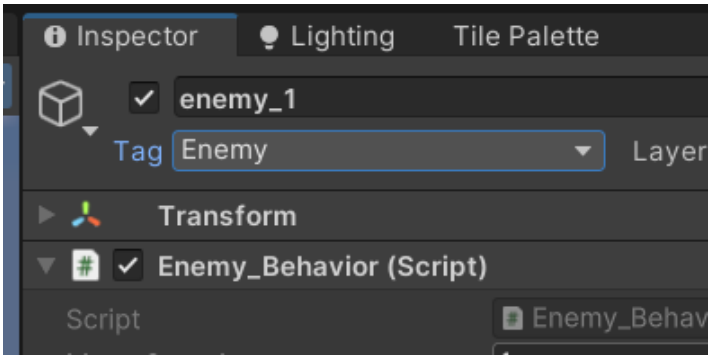
Gambar 10.12 Add Capsule Colider 2D

15. Tambahkan tag bernama enemy



Gambar 10.13 Tag Enemy

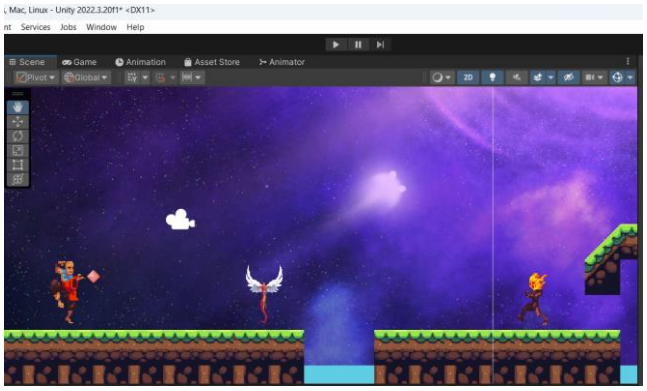
16. Berikut tampilan mekanisme attack



Gambar 10.14 Karakter Enemy

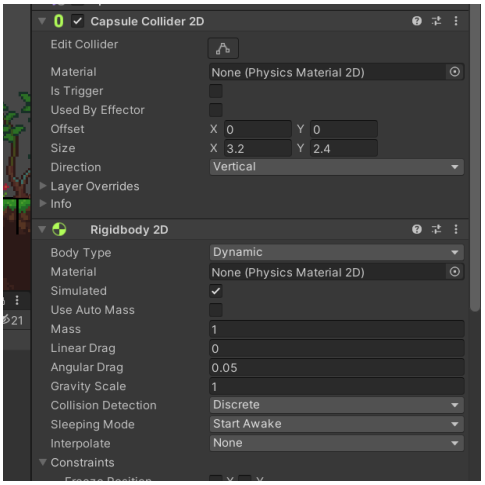
B. Membuat Enemy Behavior NPC

1. Tambahkan objek musuh 1 dengan cara drag and drop ke jendela hierarchy.



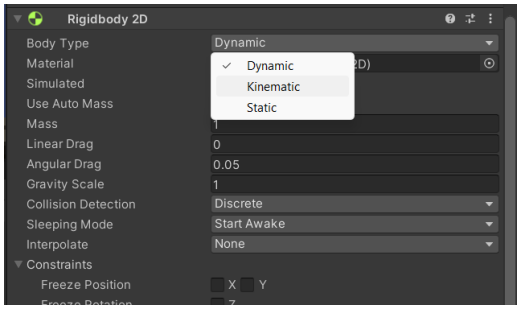
Gambar 10.15 Add musuh

2. Pada objek musuh 1 tambahkan komponen Capsule Collider 2D dan RigidBody 2D



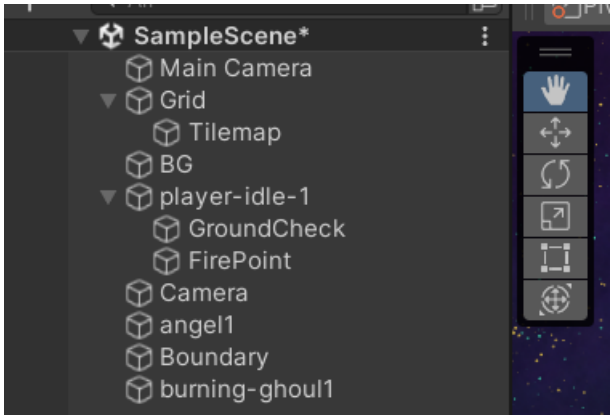
Gambar 10.16 Add Component baru

3. Pada komponen RigidBody 2D ubah tipe body menjadi Kinematic



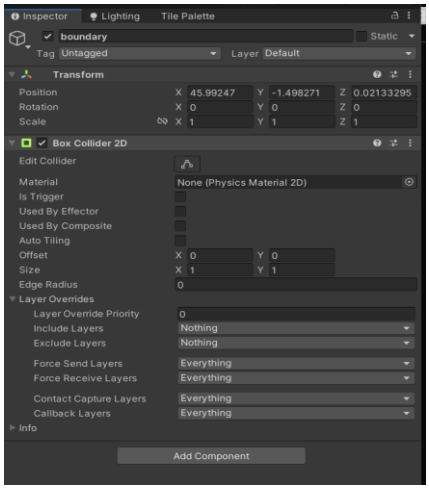
Gambar 10.17 new Kinematic

4. Tambahkan objek empty, lalu ubah nama menjadi Boundary



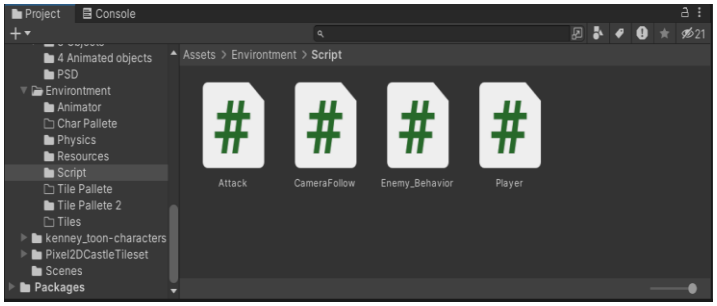
Gambar 10.18 Add Boundary

5. Pada objek baru tersebut, tambahkan komponen Box Collider 2D



Gambar 10.19 Add new Box Collider 2D

6. Buat file script baru bernama Enemy_Behavior



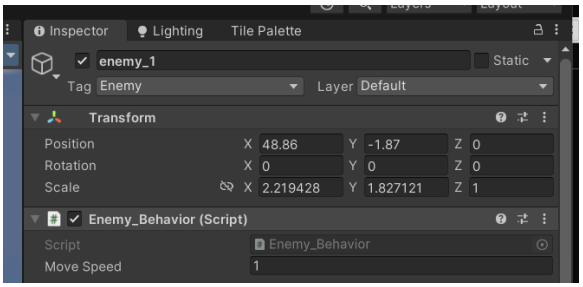
Gambar 10.20 Add new Script

7. Tambahkan source code berikut pada file script Enemy_Behavior.cs

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

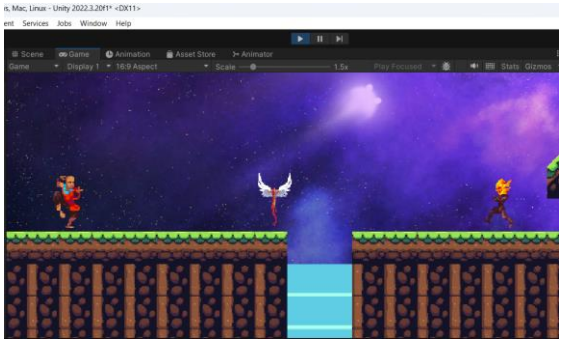
```
        using UnityEngine;
        public class Enemy_Behavior : MonoBehaviour
        { [SerializeField] float moveSpeed = 1f;
            [SerializeField] Transform player;
            Rigidbody2D rb;
            void Start(){
                rb = GetComponent<Rigidbody2D>();
            }void Update(){
                //   Perbarui   posisi   musuh   untuk
        mengikuti pemain
                FollowPlayer();
            }void FollowPlayer(){
                if (player != null){
                    Vector2         direction         =
        (player.position                              -
        transform.position).normalized;
                    rb.velocity    =    direction    *
        moveSpeed;
                }
            }private  void  OnTriggerExit2D(Collider2D
        collision){
            }
}
```

8. Drag and drop file script Enemy_Behavior ke enemy_1



Gambar 10.21 Inspector Enemy_Behavior

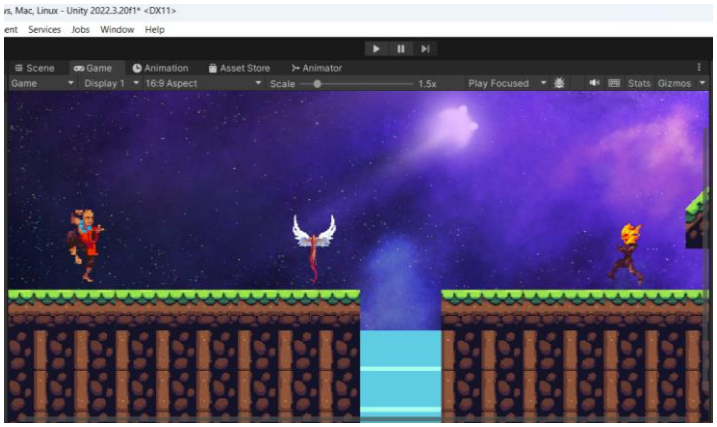
9. Jalankan Gamenya



Gambar 10.22 Run Game

C. Membuat Enemy AI

1. Tambahkan script baru,dengan nama Enemy_AI

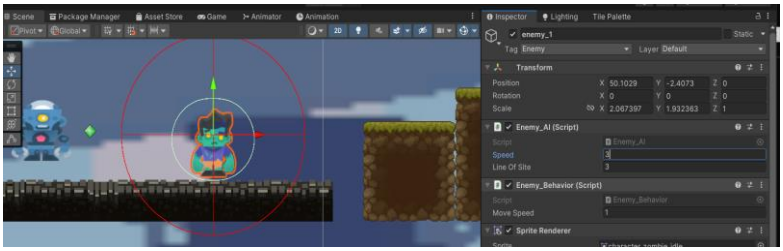


Gambar 10.23 Add new Script

2. Masukkan source code brikut pada file script Enemy_AI.cs

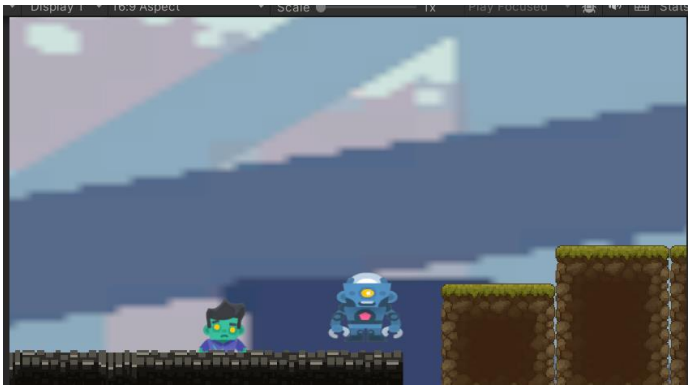
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy_AI : MonoBehaviour
{public float speed; // Kecepatan gerakan musuh
    public float lineOfSite; // Jarak penglihatan
musuh
    private Transform player; // Transform dari
pemain
    private Vector2 initialPosition;
    void Start(){
        player                                          =
GameObject.FindGameObjectWithTag("Player").transform
initialPosition                                         =
GetComponent<Transform>().position;
    }void Update(){
        float           distanceToPlayer            =
Vector2.Distance(player.position,
transform.position);
        if (distanceToPlayer < lineOfSite){
            transform.position                      =
Vector2.MoveTowards(this.transform.position,
player.position, speed * Time.deltaTime);
        }else{
            transform.position                      =
Vector2.MoveTowards(transform.position,
initialPosition, speed * Time.deltaTime);
        }
    }private void OnDrawGizmosSelected()
    {Gizmos.color = Color.red;
        Gizmos.DrawWireSphere(transform.position,
lineOfSite);
    }
}
```

3. Drag & drop file script ke objek musuh 1, setting line of site dan speed-nya



Gambar 10.24 setting line of site

4. Run Gamenya,maka enemy_1 akan mengikuti pergerakan player



Gambar 10.25 coba Game

D. Respawn

1. Tambahkan source code berikut pada file script Player.cs

```
public int nyawa = 3; // Inisialisasi nyawa dengan 3
[SerializeField] Vector3 respawn_loc;
    public bool play_again;
```

2. Tambahkan source code berikut pada void Awake()

```
respawn_loc = transform.position;
```

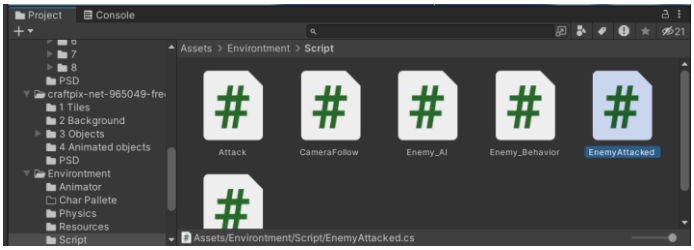
3. Buat void playagain() dan letakkan di bawah void Awake()

```
      void playagain()
          {if (play_again == true)
              {nyawa = 3;
                  transform.position = respawn_loc;
                  play_again = false;
        }
```

4. Tambahkan source code berikut pada void Update()

```
      if (nyawa <= 0)
              {play_again = true;
                  playagain();
        }
```

5. Add file script baru bernama EnemyAttacked

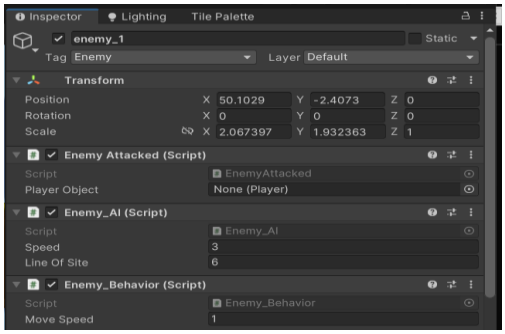


Gambar 10.26 Script EnemyAttacked

6. Tambahkan source code berikut pada script EnemyAttacked.cs

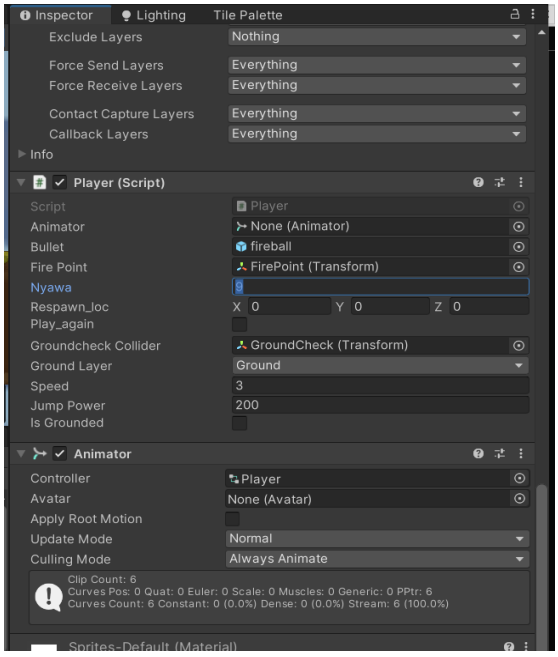
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class EnemyAttacked : MonoBehaviour
{
    [SerializeField] private Player playerObject; //
Mengubah nama variabel untuk lebih deskriptif
    void Start()
    {if (playerObject == null)
        {playerObject                                =
GameObject.FindWithTag("Player").GetComponent<Player
>();
        }
    }void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            playerObject.nyawa--;

            if (playerObject.nyawa <= 0)
            {
Destroy(playerObject.gameObject);    //    Menghapus
player dari game
                playerObject.play_again = true;
            }
        }
    }
}
```

7. Drag & drop pada objek musuh_1 dan ubah objek ke player



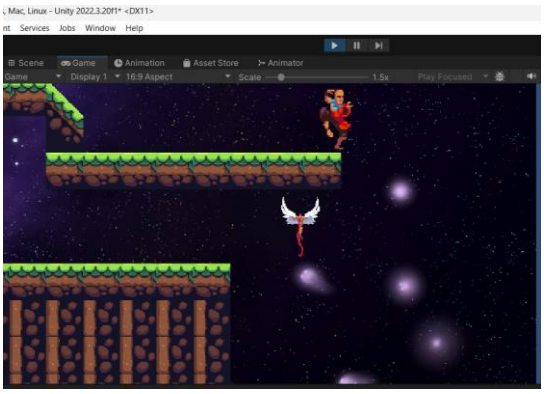
Gambar 10.27 Inspector Musuh_1

8. Setting nyawa Player menjadi 3



Gambar 10.28 Setting Nyawa 3

9. Jalankan Gamenya



Gambar 10.29 play Game

A. Melengkapi Script

```
using UnityEngine;
public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f;
    public int attackDamage = 10;
    public string enemyTag = "Enemy";
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            PerformMeleeAttack();
        }
    }
    void PerformMeleeAttack()
    {
        RaycastHit hit;
        if            (Physics.Raycast(transform.position,
transform.forward, out hit, attackRange))
        {
            if (hit.collider.CompareTag(enemyTag))
            {Health            healthComponent            =
hit.collider.GetComponent<Health>();
                if (healthComponent != null)
                {
                 healthComponent.TakeDamage(attackDamage);
                }
            }
        }
    }
}
```

Penjelasan :

Source code di atas telah diperbaiki pada metode `PerformMeleeAttack()`. Pertama, tipe variabel `attackRange` diubah dari `int` menjadi `float` untuk mencerminkan penggunaannya sebagai nilai jarak serangan. Kedua, kesalahan ketik pada `InputGetButtonDown` diperbaiki menjadi `Input.GetButtonDown`, dan `attacDamage` diperbaiki menjadi `attackDamage`. Penambahan tag `enemyTag` memungkinkan identifikasi musuh melalui tag, memastikan hanya musuh yang terkena serangan. Dalam metode `PerformMeleeAttack`, ditambahkan pemeriksaan untuk memastikan bahwa objek yang terkena raycast memiliki komponen `Health`, yang bertanggung jawab untuk mengurangi health musuh dan menangani kematian mereka.