



TUGAS PERTEMUAN: 10

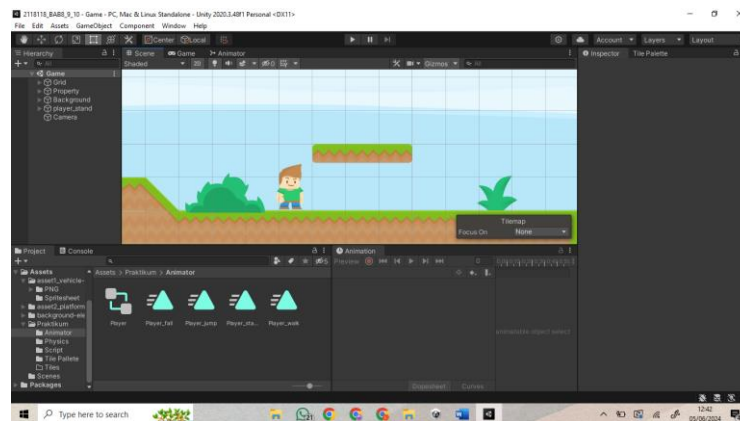
ENEMY AI AND ATTACK

NIM	:	2118118
Nama	:	Legming Dwi Anggraini
Kelas	:	C
Asisten Lab	:	Nayaka Apta Nayottama (2218102)

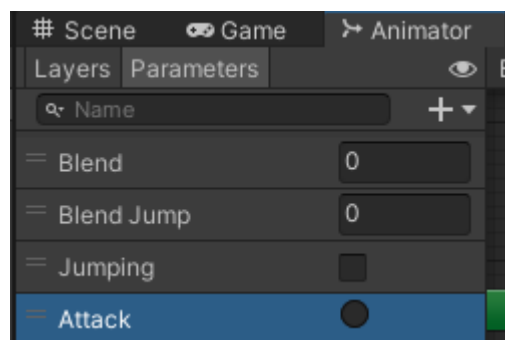
1.1 Tugas 1 : Membuat Enemy AI dan Attack

A. Membuat Mekanisme Attack

1. Buka *project* sebelumnya pada Unity

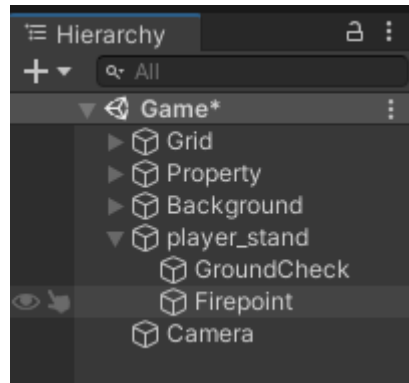


2. Tambahkan parameter baru dengan tipe data Trigger dan beri nama Attack

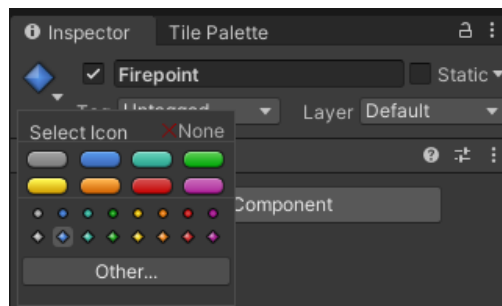




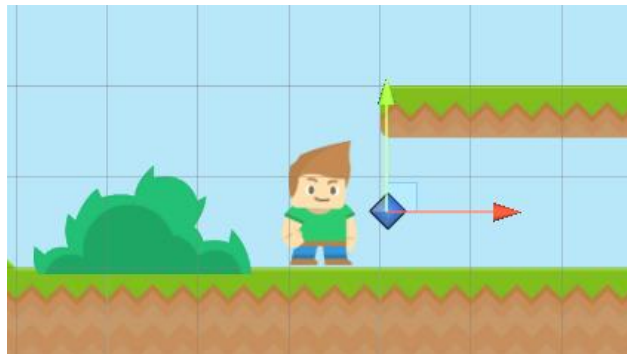
3. Buat objek baru bernama Firepoint di dalam objek player



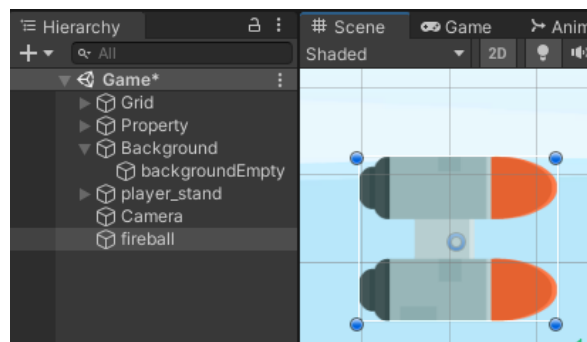
4. Pada jendela inspector ubah icon objek firepoint menjadi bullet berwarna biru seperti berikut



5. Letakkan bullet tepat berada di depan player

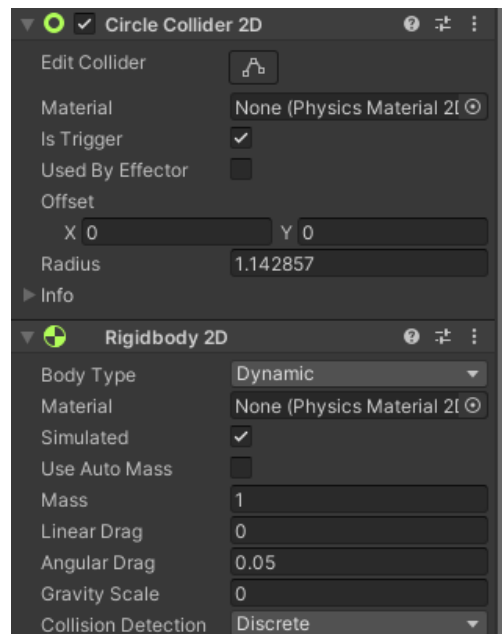


6. Drag and drop asset jetpack dan ubah nama menjadi fireball





7. Pada objek fireball tambahkan komponen circle collider 2D dan rigidbody 2D, serta sesuaikan seperti gambar berikut.



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

```
public GameObject bullet;
public Transform firePoint;

//letakkan di bawah void FixedUpdate()
IEnumerator Attack(){
    animator.SetTrigger("Attack");
    yield return new WaitForSeconds(0.25f);

    float direction = facingRight ? 1f : -1f;
    float rotationAngle = facingRight ? -90f : 90f;

    Quaternion rotation = Quaternion.Euler(0, 0,
rotationAngle);

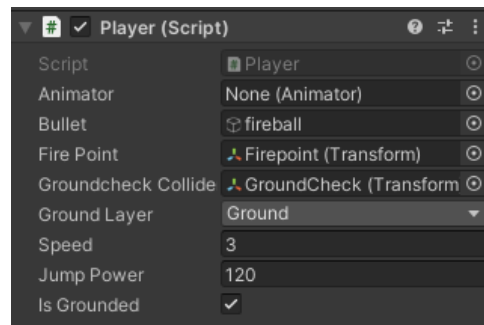
    GameObject fireball = Instantiate(bullet,
firePoint.position, rotation);
    fireball.GetComponent<Rigidbody2D>().velocity = new
Vector2(direction * 10f, 0);

    Destroy(fireball, 2f);}

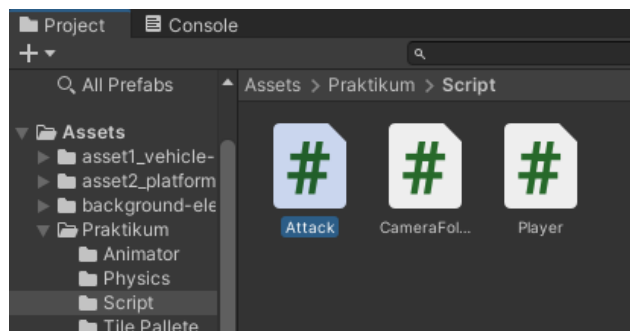
//tambahkan pada void Update()
if (Input.GetKeyDown(KeyCode.C)){
    StartCoroutine(Attack());
}
```



9. Pada objek player bagian script sesuaikan bagian bullet dan firepoint-nya



10. Membuat file script baru dengan nama Attack

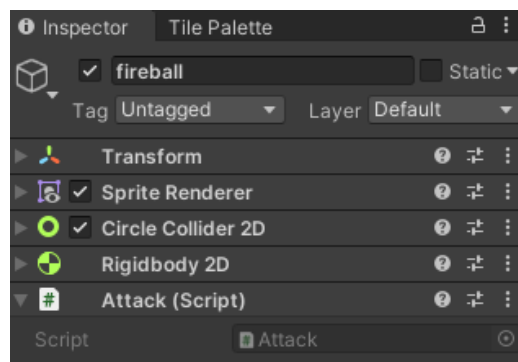


11. Tambahkan source code berikut ke dalam file Attack.cs

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

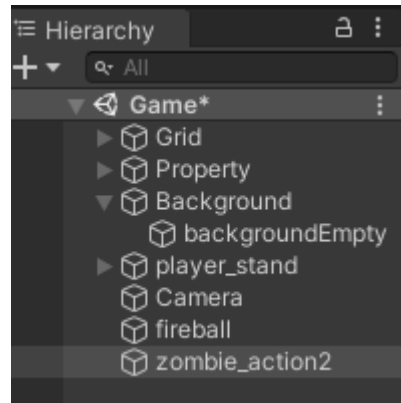
public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
            Destroy(collision.gameObject);
        }
    }
}
```

12. Drag and drop file script Attack ke objek fireball

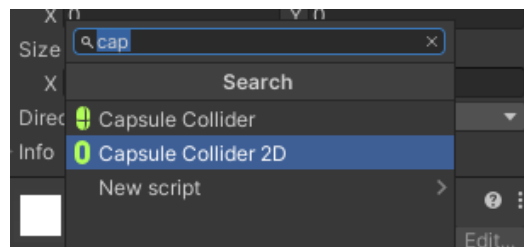




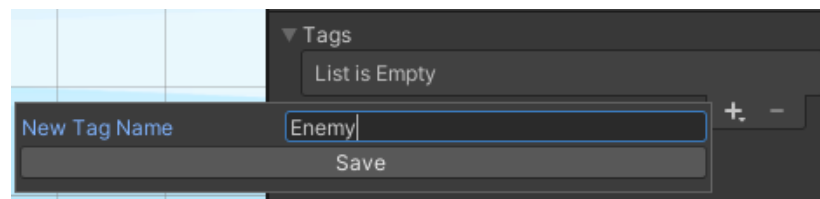
13. Tambahkan asset zombie pada scene game



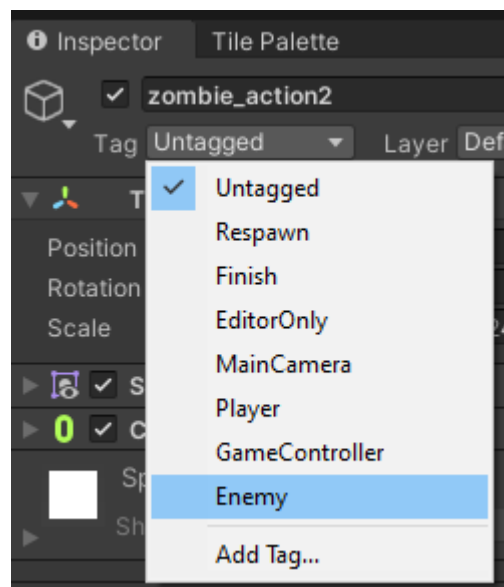
14. Pada objek zombie tambahkan komponen capsule collider 2D



15. Tambahkan tag bernama enemy

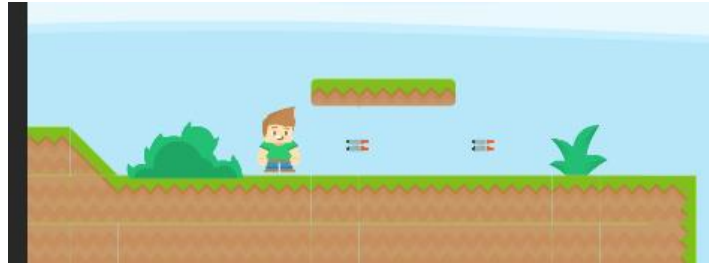


16. Pada objek zombie pilih tag enemy



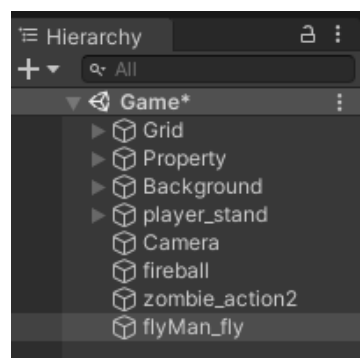


17. Berikut tampilan mekanisme attack

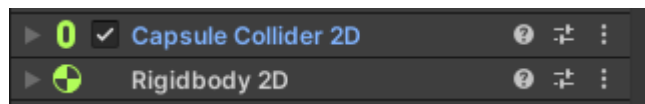


B. Membuat Enemy Behavior NPC

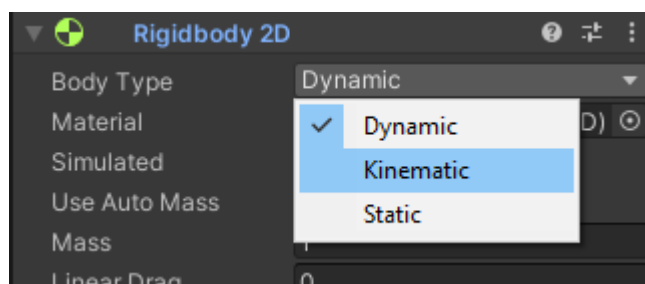
1. Tambahkan objek flyMan_fly dengan cara drag and drop ke jendela hierarchy.



2. Pada objek tersebut tambahkan komponen Capsule Collider 2D dan Rigidbody 2D

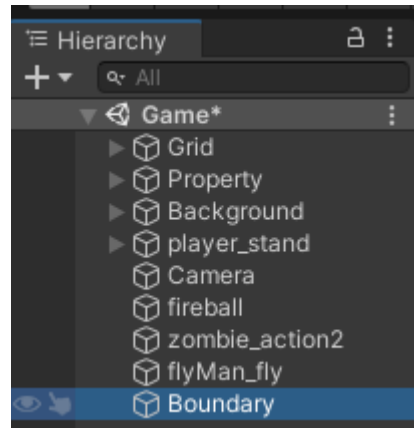


3. Pada komponen Rigidbody 2D ubah tipe body menjadi Kinematic.

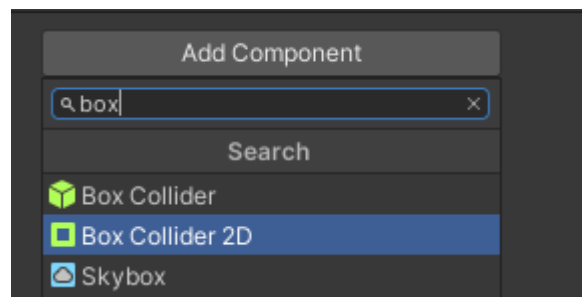




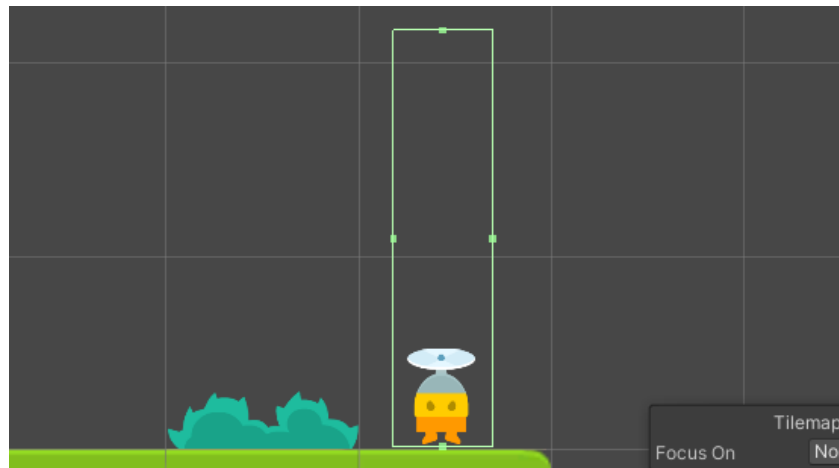
4. Tambahkan objek empty, lalu ubah nama menjadi Boundary



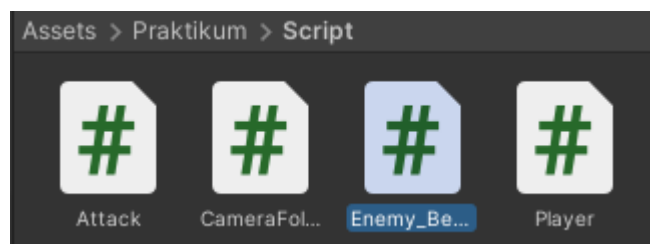
5. Pada objek baru tersebut, tambahkan komponen Box Collider 2D



6. Atur ukuran collider seperti gambar berikut



7. Buat file script baru bernama Enemy_Behavior





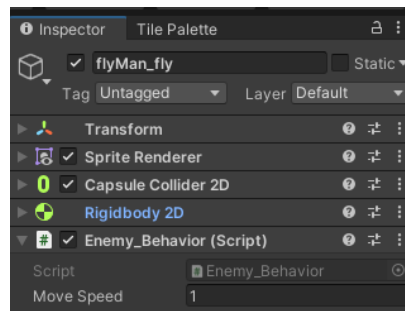
8. Tambahkan source code berikut pada file script Enemy_Behavior.cs

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

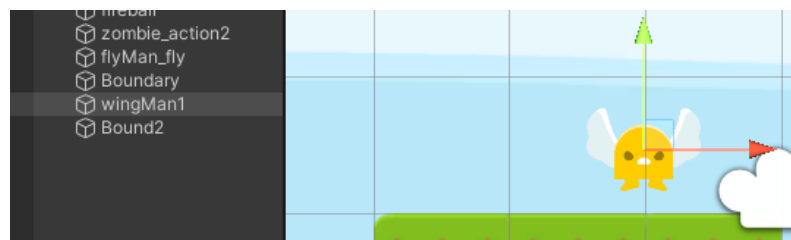
public class Enemy_Behavior : MonoBehaviour
{
    [SerializeField] float moveSpeed = 1f;
    Rigidbody2D rb;
    bool isMovingUp = true;

    void Start() {
        rb = GetComponent<Rigidbody2D>();
    }
    void Update() {
        if (isMovingUp) {
            rb.velocity = new Vector2(0f, moveSpeed);
        } else {
            rb.velocity = new Vector2(0f, -moveSpeed);
        }
    }
    private void OnTriggerExit2D(Collider2D collision) {
        isMovingUp = !isMovingUp;
    }
}
```

9. Drag and drop file script Enemy_Behavior ke objek flyMan.



10. Ulangi langkah ke-1 hingga ke-9 untuk objek yang lain yaitu “wingMan1”



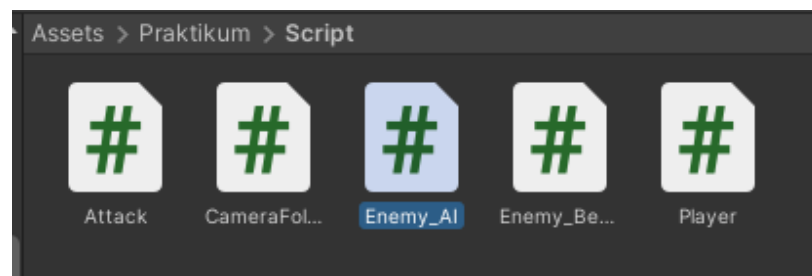


11. Jalankan hasilnya sebagai berikut



C. Membuat Enemy AI

1. Membuat file script dengan nama Enemy_AI



2. Masukkan source code berikut 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; // Posisi awal musuh

    // Use this for initialization
    void Start()
    {
        // Mencari pemain berdasarkan tag
        player =
        GameObject.FindGameObjectWithTag("Player").transform;
        // Menyimpan posisi awal musuh
        initialPosition =
        GetComponent<Transform>().position;
    }

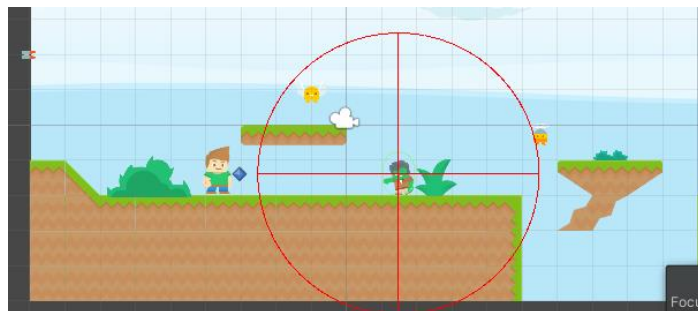
    // Update is called once per frame
    void Update()
    {
        // Menghitung jarak antara musuh dan pemain
        float distanceToPlayer =
        Vector2.Distance(player.position, transform.position);
    }
}
```



```
// Jika pemain berada dalam jarak penglihatan musuh
if (distanceToPlayer < lineOfSite)
{
    // Musuh bergerak menuju pemain
    transform.position =
Vector2.MoveTowards(this.transform.position,
player.position, speed * Time.deltaTime);
}
else
{
    // Musuh kembali ke posisi awal
    transform.position =
Vector2.MoveTowards(transform.position,    initialPosition,
speed * Time.deltaTime);
}

private void OnDrawGizmosSelected()
{
    Gizmos.color = Color.red;
    Gizmos.DrawWireSphere(transform.position,
lineOfSite);
}
}
```

3. Drag and drop file script ke objek zombie, lalu atur line of site dan juga speed-nya.



4. Tampilkan hasilnya seperti berikut, zombie akan mengikuti pergerakan player.





5. Tambahkan objek spikeMan dan ulangi langkah ke-3. Jika ingin mengetahui hasilnya jalankan seperti langkah ke-4.



D. Respawn

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

```
public int nyawa;  
[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()

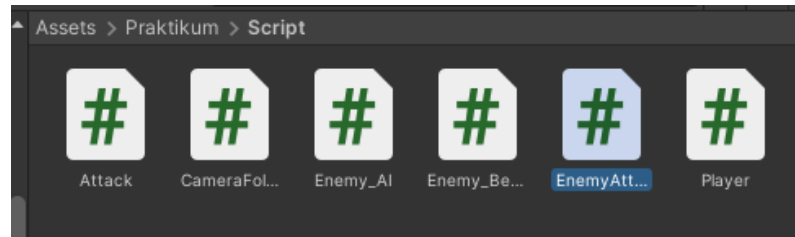
```
void playagain(){  
    if (play_again == true){  
        nyawa = 3;  
        transform.position = respawn_loc;  
        play_again = false;  
    }  
}
```

4. Tambahkan source code berikut pada void Update()

```
if (nyawa < 0){  
    playagain();  
}  
  
if (transform.position.y < -10){  
    play_again = true;  
    playagain();  
}
```



5. Buat file script baru bernama EnemyAttacked



6. Tambahkan source code berikut pada script EnemyAttacked.cs

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

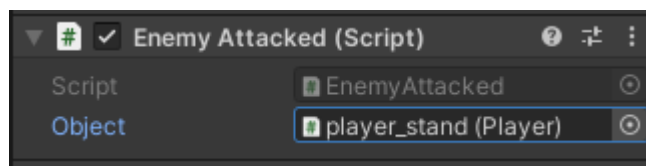
public class EnemyAttacked : MonoBehaviour
{
    [SerializeField] private Player Object;

    void Start()
    {
        if (Object == null)
        {
            Object =
GameObject.FindWithTag("Player").GetComponent<Player>();
        }
    }

    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player"))
        {
            Object.nyawa--;

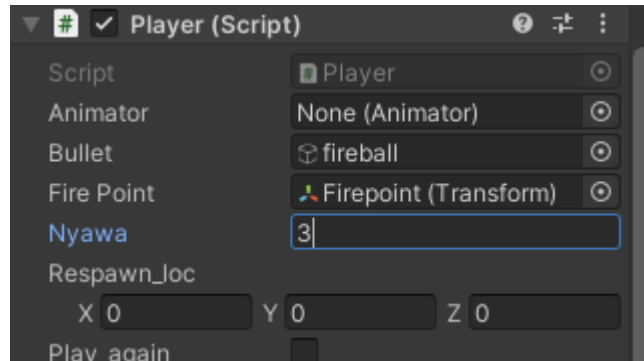
            if (Object.nyawa < 0)
            {
                Object.play_again = true;
            }
        }
    }
}
```

7. Drag and drop pada objek zombie dan ubah objek ke player





8. Pada objek Player ubah nyawa menjadi 3



9. Jalankan hasilnya seperti berikut



A. Kuis Melengkapi Script

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f;
    public int attackDamage = 10;
    public string enemyTag = "Enemy"; // Tag untuk mengidentifikasi musuh

    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))
            {
                // Damage logic here
            }
        }
    }
}
```



```
// Musuh terkena serangan, kurangi health
mereka
    Health healthComponent =
hit.collider.GetComponent<Health>();
    if (healthComponent != null)
    {
healthComponent.TakeDamage(attackDamage);
    }
}
}
```

Penjelasan :

Source code di atas diperbaiki pada void PerformMeleeAttack(). Pertama, tipe variabel attackRange diubah dari int menjadi float untuk mencerminkan penggunaannya sebagai nilai jarak serangan. Kedua, typo pada InputGetButtonDown diperbaiki menjadi Input.GetButtonDown, dan attacDamage diperbaiki menjadi attackDamage. Kemudian, 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. Dengan perubahan ini, script mampu mengurangi health musuh tanpa memerlukan script tambahan terpisah untuk EnemyHealth.

B. Link Github Pengumpulan

https://github.com/Legming-DA/2118118_PRAK_ANIGAME.git