



TUGAS PERTEMUAN: 10

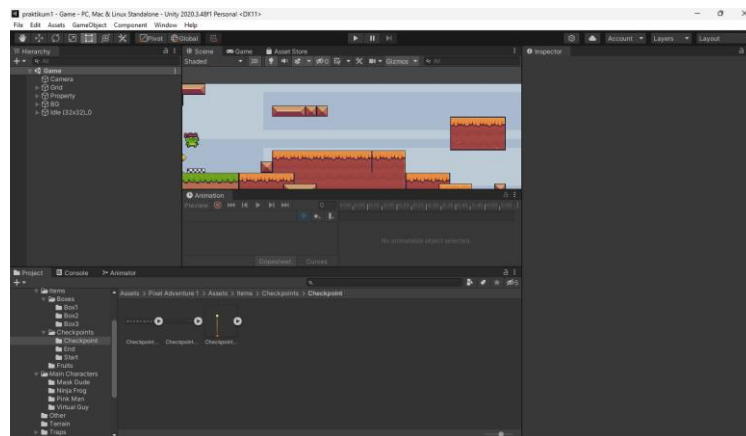
ENEMY AI AND ATTACK

NIM	:	2118108
Nama	:	Firmansyah Alafair Hermawan
Kelas	:	C
Asisten Lab	:	ZAIN ARYANATA (2118051)

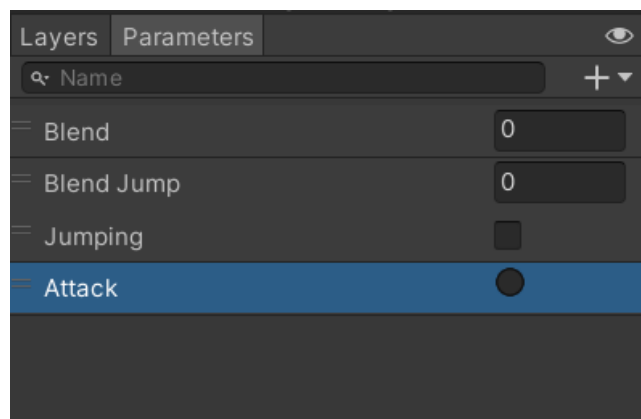
1.1 Tugas 1 : Membuat Enemy AI dan Attack

A. Membuat Mekanisme Attack

1. Bukalah Project unity sebelumnya

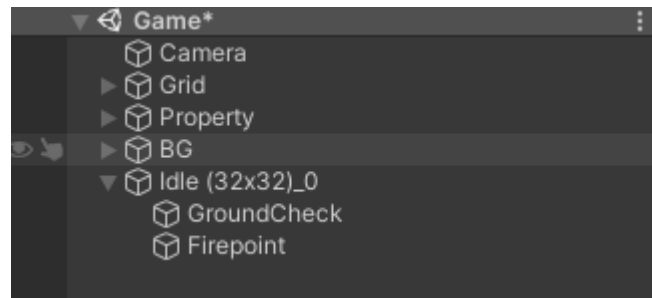


2. Tambahkan parameter baru seperti berikut ini.

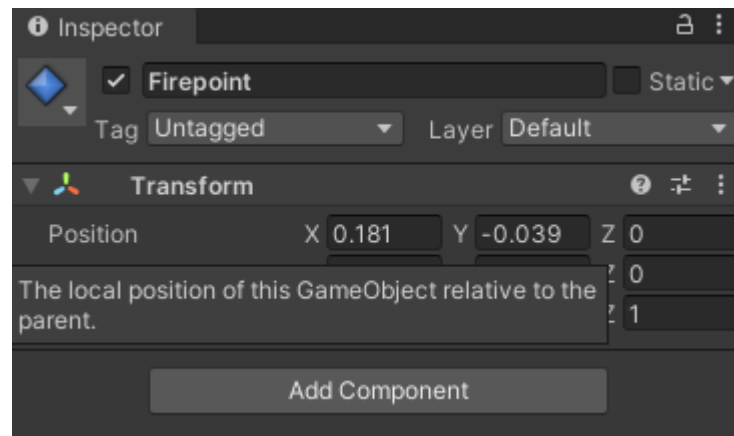




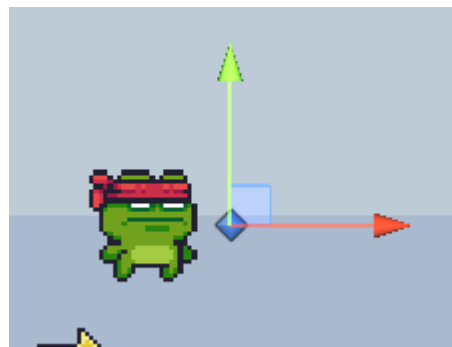
3. Buat objek baru bernama Firepoint di dalam objek player



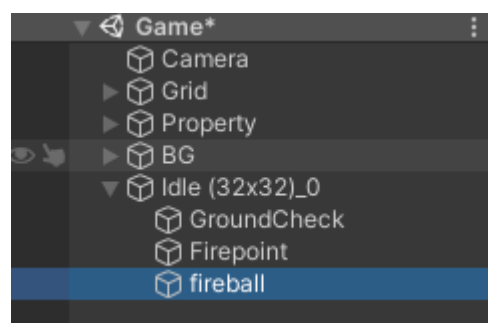
4. Masuk pada inspector dari Firepoint , lalu rubah warna dari icon berikut.



5. Atur posisi bullet seperti gambar berikut ini.

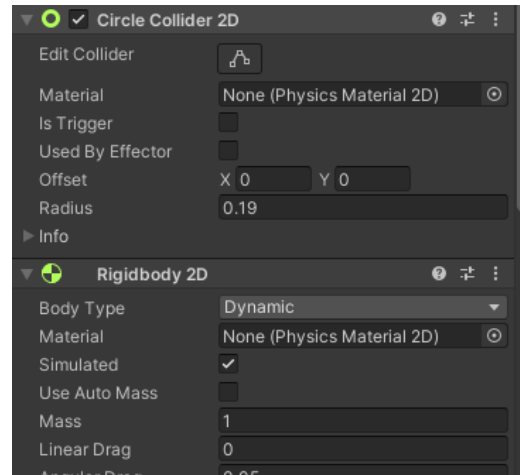


6. Drag and drop asset pedang lalu kita rename menjadi fireball





7. Pada objek fireball Add Component circle collider 2D dan rigidbody 2D..



8. Tambahkan source code Pada script Player.cs

```
if (Input.GetKeyDown(KeyCode.C)){
    StartCoroutine(Attack());
}

}

void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);

    animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
    animator.SetFloat("Blend Jump", rb.velocity.y);
    jump = false; // Reset jump flag after each FixedUpdate
}

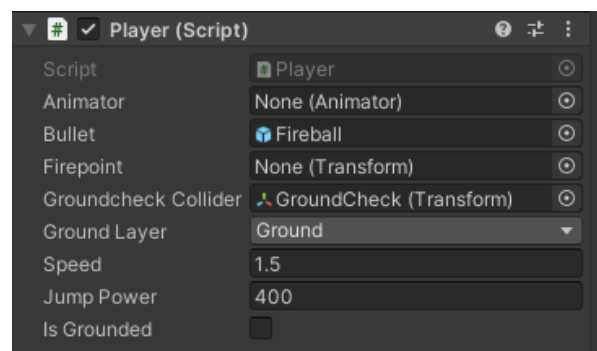
IEnumerator Attack(){
    animator.SetTrigger("Attack");
    yield return new WaitForSeconds(0.25f);

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

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

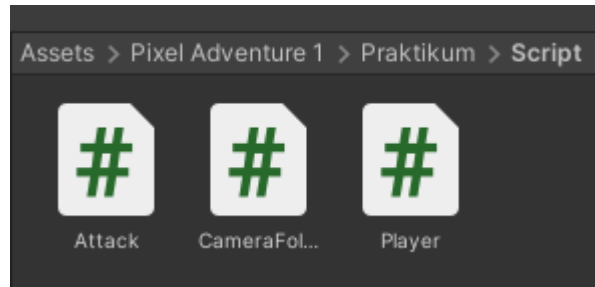
    GameObject fireball = Instantiate(bullet, Firepoint.position, rotation);
    fireball.GetComponent<Rigidbody2D>().velocity = new Vector2(direction, 0);
    Destroy(fireball, 2f);
}
```

9. Atur inspector pada player bagian player Script seperti berikut ini.





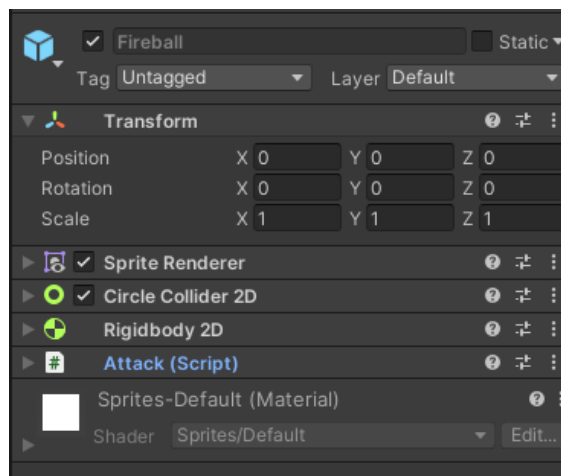
10. buat script baru dengan nama Attack



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

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Attack : MonoBehaviour
6 {
7     private void OnTriggerEnter2D(Collider2D collision)
8     {
9         if (collision.gameObject.CompareTag("Enemy"))
10        {
11            Destroy(gameObject);
12            Destroy(collision.gameObject);
13        }
14    }
15 }
16
```

12. Drag & drop file script Attack ke objek fireball

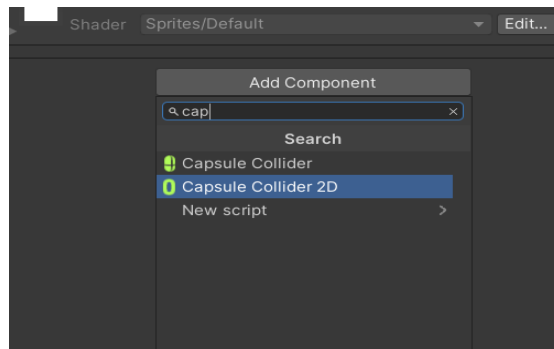


13. Tambahkan asset musuh 1 pada scene game

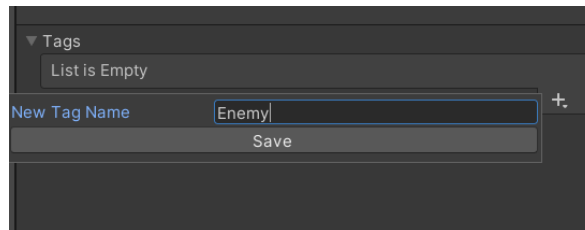




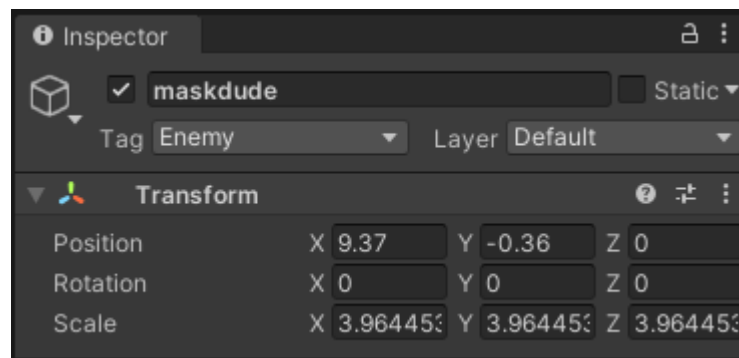
14. Pada objek musuh 1 Add Component capsule collider 2D



15. Tambahkan tag bernama enemy



16. Pada objek musuh 1 pilih tag enemy



17. Berikut tampilan mekanisme attack



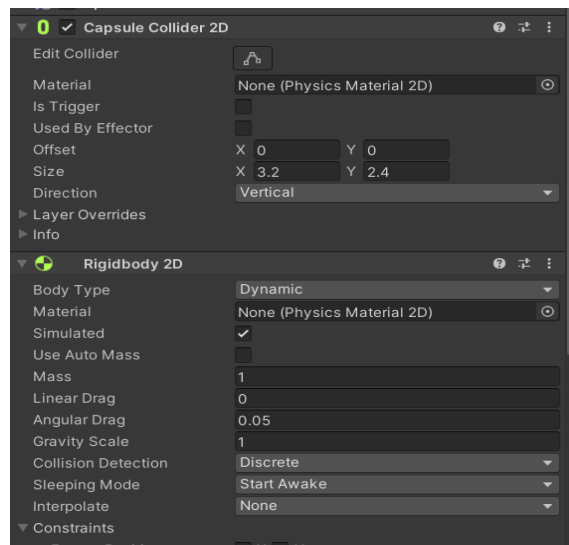


B. Membuat Enemy Behavior NPC

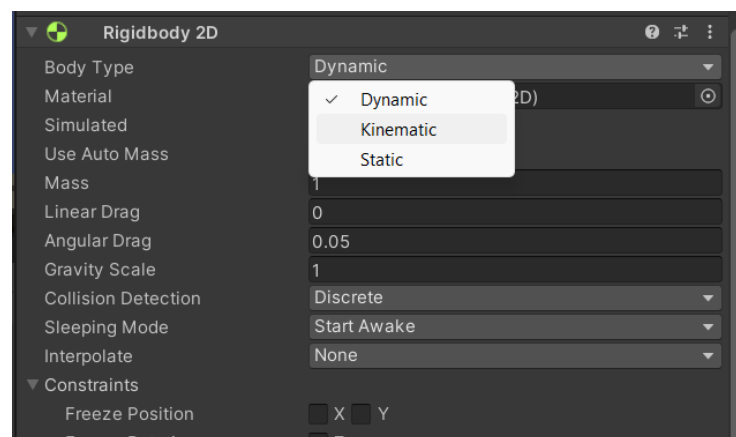
1. Tambahkan objek musuh 1 dengan cara drag and drop ke jendela hierarchy.



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

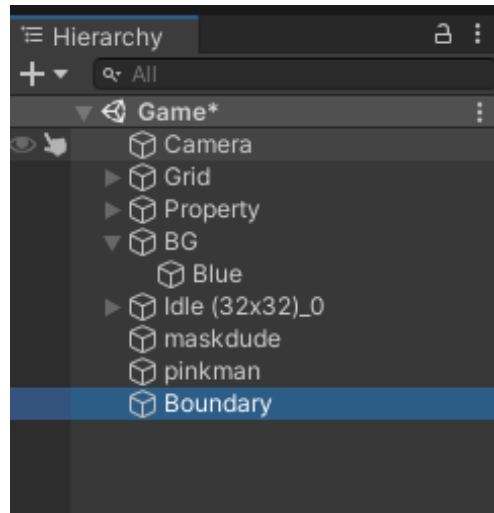


3. Pada komponen Rigidbody 2D ubah tipe body menjadi Kinematic.

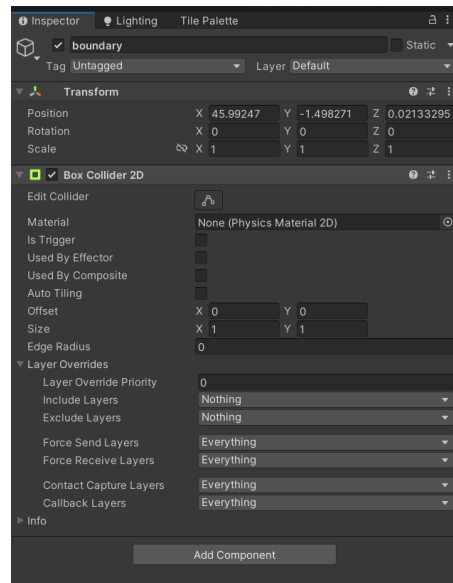




4. Tambahkan objek empty, lalu ubah nama menjadi Boundary



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



6. Buat file script baru bernama Enemy_Behavior





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

```
SEMESTER 6 > 0. PECAHAN ANIGAME > 7 > BAB7_2118123 > Assets > Tugas7 > Sc
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Enemy_Behavior : MonoBehaviour
{
    [SerializeField] float moveSpeed = 1f;
    [SerializeField] Transform player; // Referensi ke pemain
    Rigidbody2D rb;

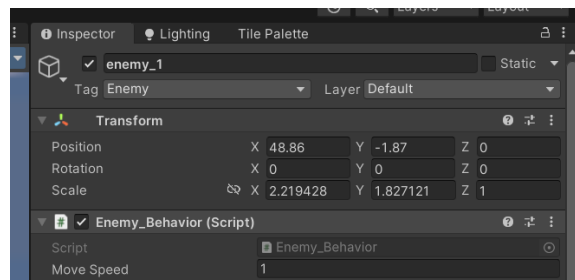
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        // Perbarui posisi musuh untuk mengikuti pemain
        FollowPlayer();
    }

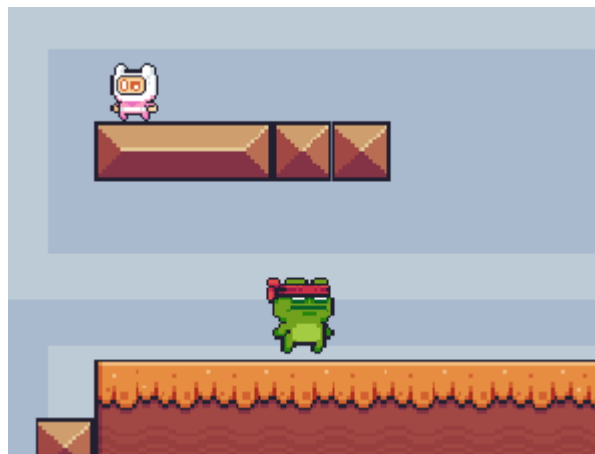
    void FollowPlayer()
    {
        if (player != null)
        {
            Vector2 direction = (player.position - transform.position).normalized;
            rb.velocity = direction * moveSpeed;
        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        // Tidak perlu menggunakan isMovingUp lagi karena mus
    }
}
```

8. Drag and drop file script Enemy_Behavior ke enemy_1.



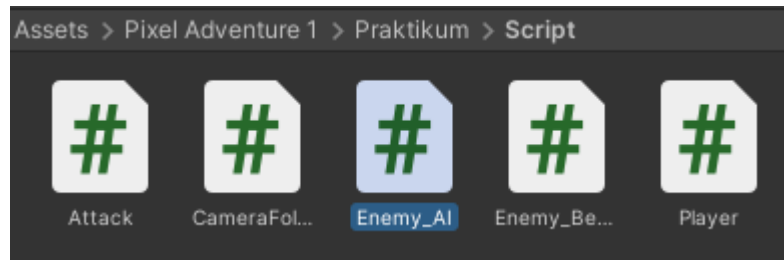
9. Jalankan Gamenya.





C. Membuat Enemy AI

1. tambahkan script baru,dengan nama Enemy_AI



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

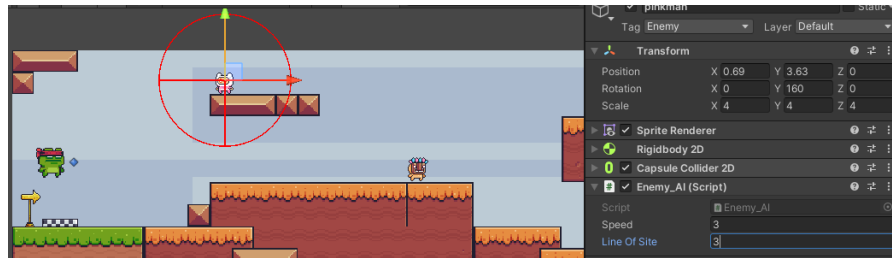
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

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; // Posisi awal musuh

    void Start() {
        player =
        GameObject.FindGameObjectWithTag("Player").transform
        ;
        // Menyimpan posisi awal musuh
        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 enemy_1, setting line of site dan speed-nya.



4. Run Gamenya, maka enemy_1 akan mengikuti pergerakan player.



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. Buat file script baru bernama EnemyAttacked



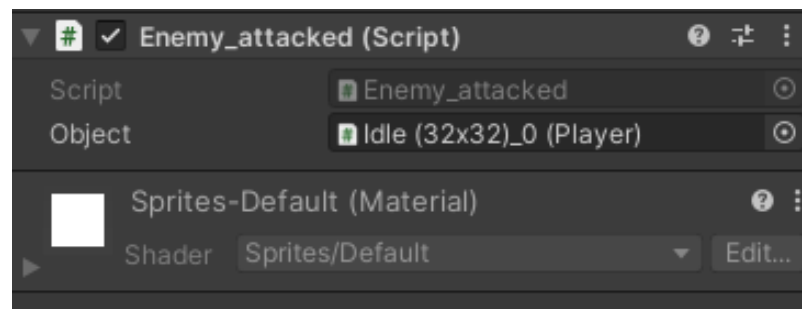
Gambar 10.30 Script EnemyAttacked

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)
            {
                playerObject.Respawn(); // Call the
                Respawn method instead of destroying the player
            }
        }
    }
}
```

7. Drag & drop pada objek enemy_1 dan ubah objek ke player





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.