



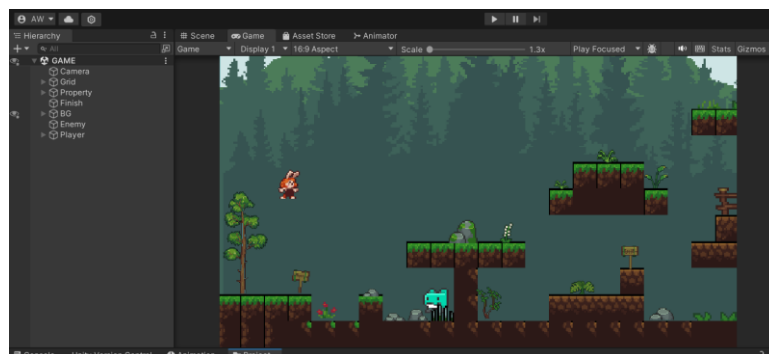
## TUGAS PERTEMUAN: 10 RESPAWN AND AI ENEMY ATTACK

NIM	:	2118077
Nama	:	Abdul Wahid
Kelas	:	B
Asisten Lab	:	MARIA AVRILIANA SURAT LELAONA (2218096)
Baju Adat	:	Pangasi Baju Adat sunda (jawa barat)
Referensi	:	<a href="https://akcdn.detik.net.id/community/media/visual/2022/04/20/pakaian-adat-jawa-barat-5_43.jpeg?w=700&amp;q=90">https://akcdn.detik.net.id/community/media/visual/2022/04/20/pakaian-adat-jawa-barat-5_43.jpeg?w=700&amp;q=90</a>

### 10.1 Tugas 10 : Respawn and AI Enemy Attack

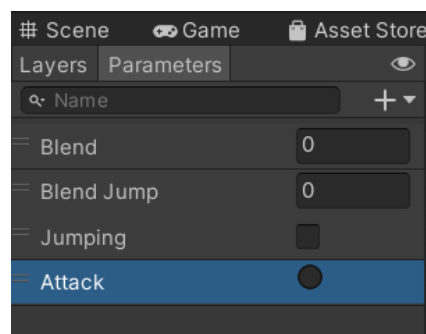
#### A. Membuat Mekanisme Attack

1. Buka Project Bab 9 Untuk melanjutkannya



Gambar 10.1 Project Bab 9

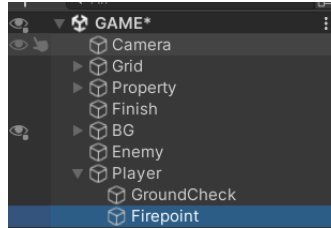
2. Kemudian pada menu Tab Animator Tambahkan Parameter Trigger, Rename Menjadi *Attack*



Gambar 10.2 Attack

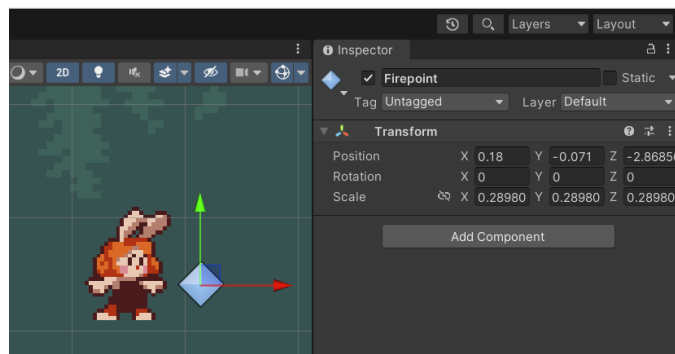


3. Setelah menambahkan parameter Attack, Langkah selanjutnya adalah membuat Layer *Game object* baru didalam *player-idle-1*, Klik kanan pilih *Create Empty* lalu Rename menjadi *Firepoint*



Gambar 10.3 Firepoint

4. Pada menu *Hierarchy* klik Firepoint untuk setting pada Inspector, Ubah *Icon* Menjadi titik, atur letak titik didepan player



Gambar 10.4 Inspector Firepoint

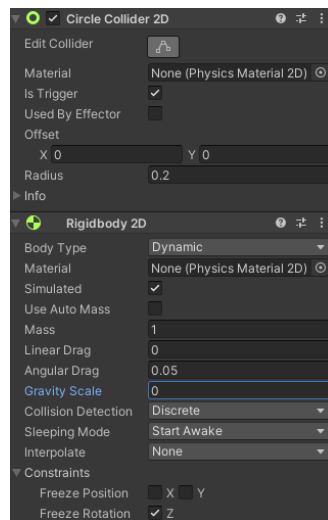
5. Pada menu Hierarchy Tambahkan star-1, *rename* menjadi *star*



Gambar 10.5 Star

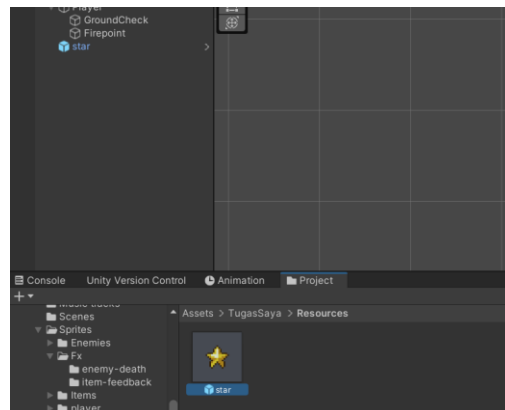


6. Klik star-1 untuk menambahkan Component Circle Collider 2d, dan Riggidbody 2D, Setting sesuai gambar dibawah ini



Gambar 10.6 Component

7. Buat Folder baru *Resources* di menu Project, kemudian drag and drop star kedalam folder Resources, dan hapus *star* pada *Hierarchy*



Gambar 10.7 Resources

8. Pada Script Player Tambahkan Script dibawah ini

```
#Tambahkan dibawah fungsi fixedUpdate
IEnumerator Attack()
{

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

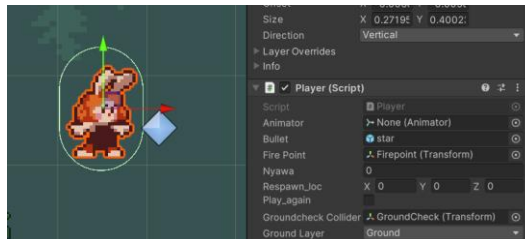
    float direction = 1f;

    GameObject      star      =      Instantiate(bullet,
firePoint.position, Quaternion.identity);
        star.GetComponent<Rigidbody2D>().velocity      =
        new Vector2(direction * 10f, 0);
```



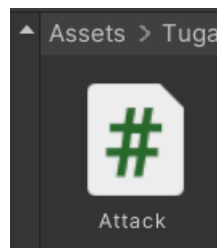
```
        Destroy(star, 2f);
    }
    #Tambahkan pada Function Void Update
    if (Input.GetKeyDown(KeyCode.C))
    {
        StartCoroutine(Attack());
    }
}
```

9. Pada Inspector Player, Ubah seperti dibawah ini, Dimana Bullet berisi object yang akan ditembak sedangkan fire point adalah titik tembak pertama



Gambar 10.8 Player script setting

10. Buat Script Attack pada folder Script



Gambar 10.9 Script attack

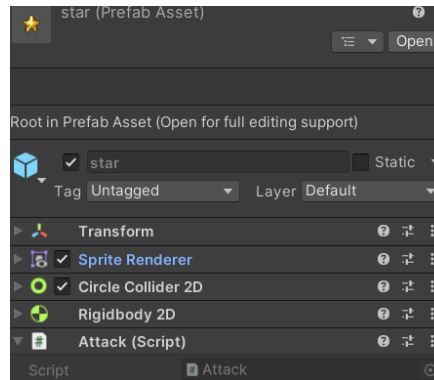
11. Tambahkan Script Attack dibawah ini

```
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);
        }
    }
}
```

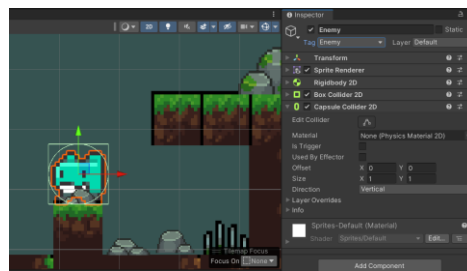


12. Didalam folder resource Tambahkan Script Attack di Prefab star, dengan cara Klik star kemudian pada menu Inspector arahkan Script Attack kedalam Inspector



Gambar 10.10 Star attack script

13. Tambahkan Tag Enemy dengan cara Pilih Add Tag, kemudian add tag to the list, Tuliskan Enemy



Gambar 10.11 Tag Enemy

14. Tembak Enemy dengan menekan Tombol C untuk menghancurkan musuh

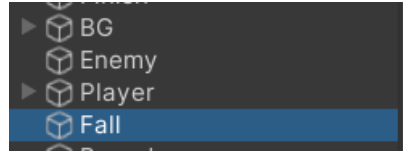


Gambar 10.12 Tembak Enemy

## 10.2 Membuat Enemy AI

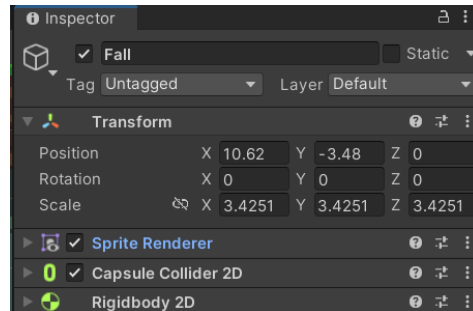
### A. Enemy Behavior NPC

1. Tambahkan fall enemy ke Hierarchy.



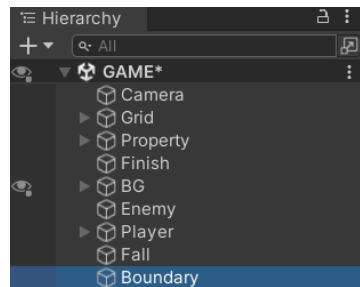
Gambar 10.13 fall

2. Tambahkan sebuah komponen bernama Capsule Colider 2D dan Rigidbody dalam inspector game objek fall



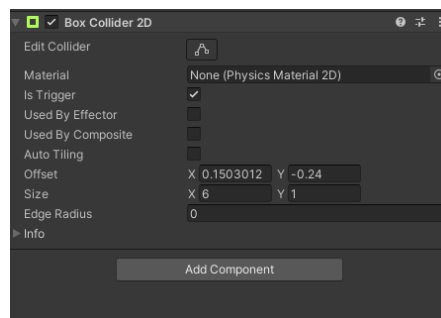
Gambar 10.14 Component Fall

3. Create Empty object pada Hierarchy, Rename Menjadi *Boundary*



Gambar 10.15 Boundary

4. Tambahkan Box Collider 2d pada Boundary, centang pada Is Trigger lalu atur sesuai keinginan pada size dan offside



Gambar 10.16 Boundary Component



5. Buat sebuah file script didalam folder Script beri nama “Enemy\_Behavior”, kemudian drag dan masukkan ke dalam game object.



Gambar 10.17 Enemy\_Behavior

6. Tambahkan Script dibawah ini

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

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

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

    void Update()
    {
        if (isFacingRight())
        {
            rb.velocity = new Vector2(moveSpeed, 0f);
        }
        else
        {
            rb.velocity = new Vector2(-moveSpeed, 0f);
        }
    }

    private bool isFacingRight()
    {
        return transform.localScale.x > Mathf.Epsilon;
    }

    private void OnTriggerExit2D(Collider2D collision)
    {
        transform.localScale = new Vector2(-transform.localScale.x, transform.localScale.y);
    }
}
```

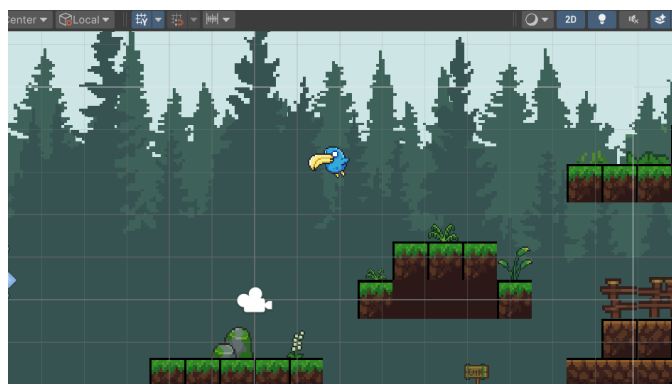
7. Jalankan Program



Gambar 10.18 Tampilan Enemy Behavior NPC

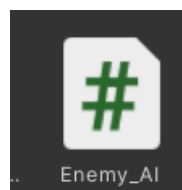
## B. Enemy AI

1. Cari sebuah sprite pack bernama 'enemy' dan buka folder bernama flying. Tambahkan flying pada Hierarchy



Gambar 10.19 Flying

2. Buat Script Enemy\_AI pada folder Praktikum - Script



Gambar 10.20 Enemy\_AI

3. Tambahkan Script dibawah ini

```
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
```





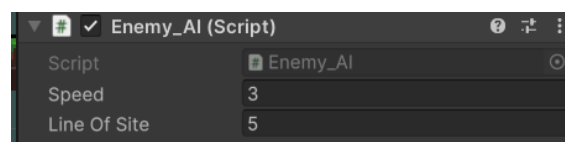
```
void Start()
{
    // Mencari pemain berdasarkan tag
    player =
GameObject.FindGameObjectWithTag("Player").transform;
    // Menyimpan posisi awal musuh
    initialPosition =
GetComponent<Transform>().position;
}

// 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);
    }
}

// Untuk menggambar jarak penglihatan musuh di
editor
private void OnDrawGizmosSelected()
{
    Gizmos.color = Color.red;
    Gizmos.DrawWireSphere(transform.position,
lineOfSite);
}
}
```

4. Pada Inspector Enemy\_Ai, Atur Speed juga Line of Site untuk menentukan jarak dan speed pada enemy



Gambar 10.21 Enemy\_AI Script



## 5. Running Game, maka eagle akan mengikuti Gerakan Player



Gambar 10.22 Running Enemy AI

### C. Respawn

1. Buka file script (Player.cs) ubah jadi seperti dibawah ini

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

public class Player : MonoBehaviour
{
    public Animator animator;
    public GameObject bullet;
    public Transform firePoint;
    public int nyawa;
    [SerializeField] Vector3 respawn_loc;
    public bool play_again;

    Rigidbody2D rb;
    [SerializeField] Transform groundcheckCollider; // +
    [SerializeField] LayerMask groundLayer; // +

    const float groundCheckRadius = 0.2f; // +
    [SerializeField] float speed = 1;
    [SerializeField] float jumpPower = 200; // ++

    float horizontalValue;

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

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();
        rb.freezeRotation = true;

        respawn_loc = transform.position;
    }

    void playagain() {
```



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

    // Update is called once per frame
    void Update ()
    {
        horizontalValue = Input.GetAxisRaw("Horizontal");
        if (Input.GetButtonDown("Jump")){
            animator.SetBool("Jumping", true);
            jump = true;
        }
        else if (Input.GetButtonUp("Jump")){
            jump = false;
        }
        if (Input.GetKeyDown(KeyCode.C))
        {
            StartCoroutine(Attack());
        }
        if (nyawa < 0){
            playagain();
        }
        if (transform.position.y < -10){
            play_again = true;
            playagain();
        }
    }

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

        animator.SetFloat("Blend", Mathf.Abs
(rb.velocity.x));
        animator.SetFloat("Blend Jump", rb.velocity.y);
    }

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

        float direction = 1f;

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

        Destroy(star, 2f);
    }

    void GroundCheck()
```



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

    animator.SetBool("Jumping", !isGrounded);
}

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

    #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)
    {
        facingRight = false;
    }
    else if (!facingRight && dir > 0)
    {
        facingRight = true;
    }

    // Menyesuaikan arah karakter tanpa mengubah skala
    if (facingRight)
    {
        transform.localScale = new
Vector3(Mathf.Abs(transform.localScale.x),
transform.localScale.y, transform.localScale.z);
    }
    else
    {
        transform.localScale = new Vector3(-
Mathf.Abs(transform.localScale.x),
transform.localScale.y, transform.localScale.z);
    }
    #endregion
}
}
```

2. Tambahkan file script (Enemy\_Attacked.cs) dan isikan source code dibawah ini



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

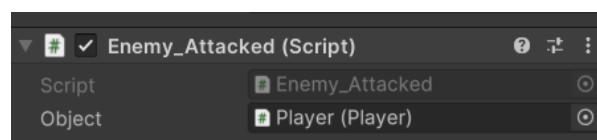
public class Enemy_attacked : MonoBehaviour
{
    [SerializeField] private Player Object;

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

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

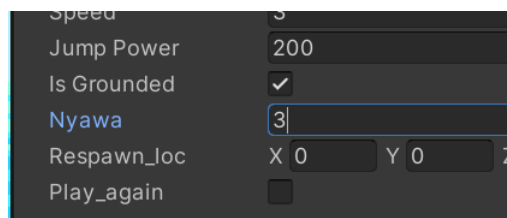
            if (Object.nyawa < 0)
            {
                Object.play_again = true;
            }
        }
    }
}
```

3. Pada hierarchy fliying Tambahkan Script enemy attack, arahkan object pada player-idle-1



Gambar 10.23 Enemy Attacked script

4. Klik game object Player, pergi ke Inspector dan ubah nilai Nyawa menjadi 3 pada Player(Script)



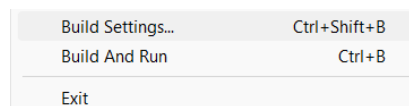
5. Jika di play, Player mengenai atau menyentuh oposum-1 sebanyak 3 kali maka nyawa akan berkurang 1 dan jika nyawa kurang dari 0 maka akan respawn ke titik awal



Gambar 10.24 Running Respawn

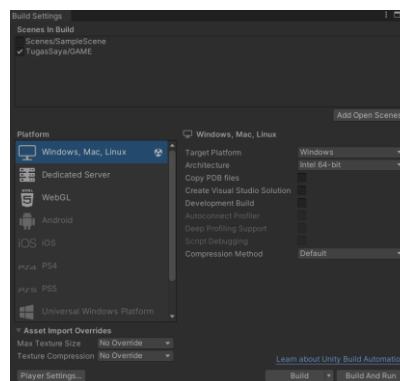
## D. Render

1. Pergi ke menu File kemudian pilih Build Setting (Ctrl + Shift + B)



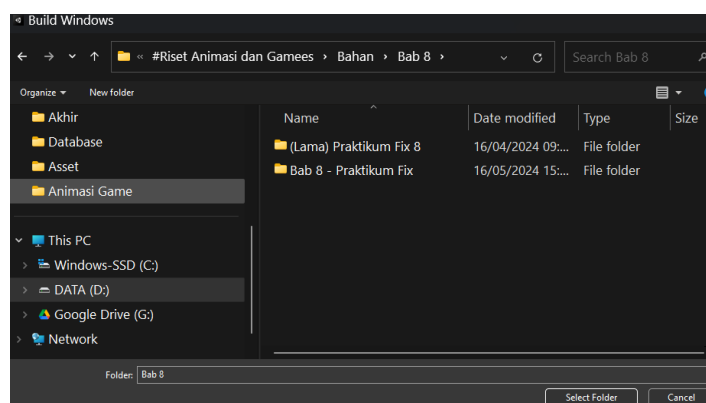
Gambar 10.25 Window Build setting

2. Pada Setting Build ini pilih PC, Mac & Linux, Tekan Build, pastikan pada menu Scene in Build berada pada project Tugas Kalian



Gambar 10.26 Build Setting

3. Pilih dimana Project disimpan, dan tunggu hasilnya



Gambar 10.27 Lokasi Proyek



#### 4. Hasil Render



Gambar 10.28 Hasil Render

### 10.3 Kuis

#### Soal kuis Bab 10

```
using UnityEngine;
public class PlayerAttack : MonoBehaviour
{
    public float attackRange = 2.0f; // Ganti int dengan float
    untuk jarak serangan
    public int attackDamage = 10;
    // Perbaiki nama variabel
    dari "attacDamage" ke "attackDamage"
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            PerformMeleeAttack();
        }
    }
    void PerformMeleeAttack()
    {
        RaycastHit hit;
        if
        (Physics.Raycast(transform.position,
        transform.forward, out hit, attackRange))
        {
            // Memeriksa apakah objek yang terkena memiliki
            komponen EnemyHealth
            EnemyHealth
            enemyHealth
            hit.transform.GetComponent<EnemyHealth>();
            if (enemyHealth != null)
            {
                // Mengurangi health musuh
                enemyHealth.TakeDamage(attackDamage);
            }
        }
    }
}
```

Analisa Source code :



Pada kode yang telah diperbaiki, fungsi utama dari PlayerAttack adalah untuk mengelola serangan melee pemain dalam game Unity. Ketika pemain menekan tombol yang ditentukan (Fire1), metode PerformMeleeAttack dipanggil. Metode ini menggunakan Physics.Raycast untuk memancarkan sinar dari posisi pemain ke arah depan. Jika sinar tersebut mengenai objek dalam jarak yang ditentukan (attackRange), skrip memeriksa apakah objek yang terkena memiliki komponen EnemyHealth. Jika iya, maka metode TakeDamage dari EnemyHealth dipanggil dengan parameter attackDamage, yang mengurangi kesehatan (currentHealth) dari musuh tersebut. Jika kesehatan musuh mencapai nol atau kurang, metode Die dipanggil untuk memusnahkan atau menonaktifkan musuh.

berikan tanda merah yang menyebabkan source code error

```
void HandleJumpInput()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        animator.SetBool("isJumping", );
        rb.AddForce(Vector2.up
        *
        ForceMode2D.Impulse);
    }
    else if (Input.GetKey(KeyCode.Space))
    {
        animator.SetBool("isJumping",);
    }
}

void HandleMovementInput()
{
    float move = Input.GetAxis("Horizontal");
    jumpForce,
    if (move != 1)
    {
        animator.SetBool("isIdle", true);
        transform.Translate(Vector3.left
        *
        move
        Time.deltaTime);
    }
    else
    {
        animator.SetBool("isWalking", false);
    }
    if (move != 0)
    {
        transform.localScale = new Vector3(-4, 1, 1);
    }
    else if (move > 0)
    {
        transform.localScale = new Vector3(1, 2, 1);
    }
}
```





```
}  
}
```

Analisa :

Pada source code diatas yang menyebabkan error pertama pada “`animator.SetBool("isJumping", );`” ini karena seharusnya terdapat nilai boolean (True atau False) yang mengakibatkan error krn sintaknya tidak lengkap, kedua “`if (move != 1)`” seharusnya bernilai 0 agar player bisa bergerak dan tidak hanya diam, ketiga “`transform.Translate(Vector3.left * move * Time.deltaTime);`” seharusnya script tersebut di taruk dibagian else agar vektor bisa bergerak ke arah lainnya, keempat “`animator.SetBool("isWalking", false);`” karena seharusnya set walking ke true agar bisa bergerak dan tidak diam, kelima “`(-4, 1, 1);` dan `(1, 2, 1);`” karena seharusnya bernilai `(-1,1,1)` agar negative x for left agar tidak menimpa kondisi else if yang akan dijalankan lalu seharusnya bernilai `(1,1,1)` agar positive x for right