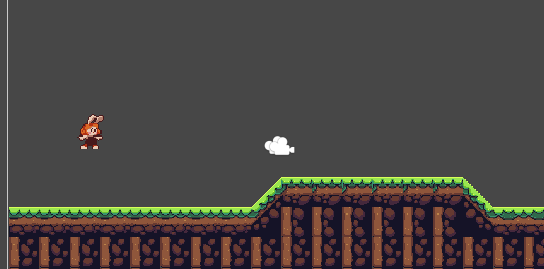
# 10 ENEMY AI AND ATTACK

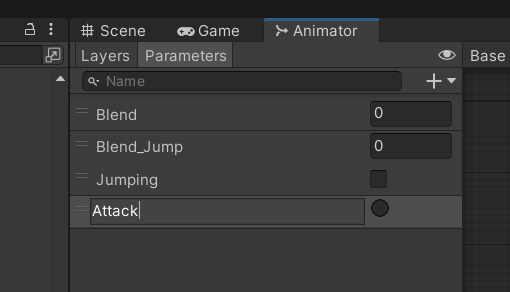
|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118116 |
| **Nama** | : | Ivan Kristianto Santoso |
| **Kelas** | : | D |
| **Asisten Lab** | : |  |

## 1.1 Tugas 1 : Membuat Enemy AI dan Attack

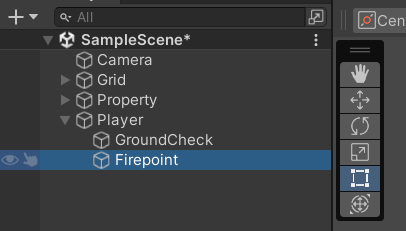
1. **Membuat Mekanisme Attack**
2. Bukalah Project unity sebelumnya



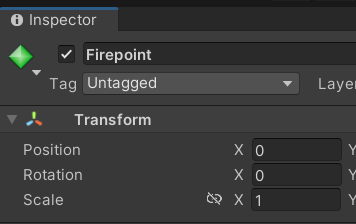
1. Tambahkan parameter baru seperti berikut ini.



1. Buat objek baru bernama Firepoint di dalam objek player



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



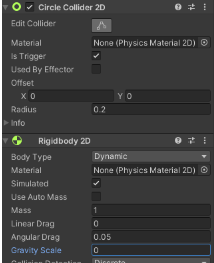
1. Atur posisi bullet seperti gambar berikut ini.



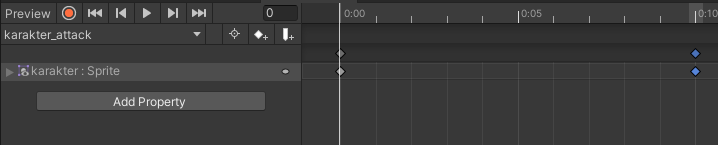
1. Drag and drop asset pedang lalu kita rename menjadi fireball



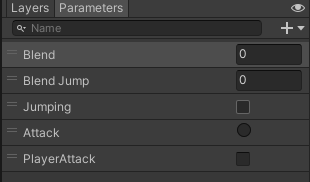
1. Pada objek fireball Add Component circle collider 2D dan rigidbody 2D..



1. Klik karakter, pada tab Animation tambahkan clip karakter\_attack, lalu pilih animasi menyerang karakter dan sesuaikan juga durasinya.



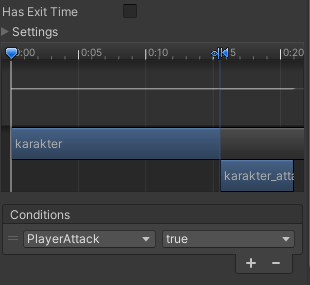
1. Pada Tab Animator tambahkan parameter PlayerAttack dengan tipe Bool.



1. Buat Transisi bolak balik dari karakter ke karakter\_attack



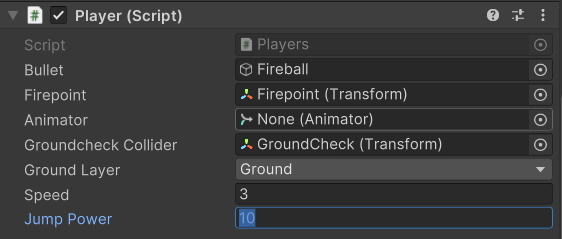
1. Klik transisi dari karakter ke karakter\_attack tambahkan conditions PlayerAttack dan atur menjadi true, atur juga hal yang serupa dengan transisi karakter\_attack ke karakter hanya saja nilai dari kondisi menjadi false. Jangan lupa hilangkan has exit time dan transisi duration menjadi 0.



1. Tambahkan source code Pada script Player.cs

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class player : MonoBehaviour  {    public Animator animator;    public GameObject bullet;    public Transform firePoint;    Rigidbody2D rb;    [SerializeField] Transform groundcheckCollider;    [SerializeField] LayerMask groundLayer;    const float groundCheckRadius = 0.2f; // +    [SerializeField] float speed = 1;    [SerializeField] float jumpPower = 500;    float horizontalValue;    [SerializeField] bool isGrounded; // +    bool facingRight;    bool jump;    void FixedUpdate()    {    GroundCheck();    Move(horizontalValue, jump);    animator.SetFloat("Blend",Mathf.Abs(rb.velocity.x));    animator.SetFloat("Blend Jump",rb.velocity.y);    jump = false;    }    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, Quaternion.identity);       fireball.GetComponent<Rigidbody2D>().velocity = new Vector2(direction \* 10f, 0);         Destroy(fireball, 2f);  }    void GroundCheck()    {      isGrounded = false;      Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);      if (colliders.Length > 0){          isGrounded = true;      }      animator.SetBool("Jumping",!isGrounded);    }    private void Awake()    {      rb = GetComponent<Rigidbody2D>();      animator = GetComponent<Animator>();    }    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());          animator.SetBool("PlayerAttack", true);      }      else if (Input.GetKeyUp(KeyCode.C))      {          animator.SetBool("PlayerAttack", false);      }  }    void Move(float dir, bool jumpflag)    {      #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)      {        // ukuran player        transform.localScale = new Vector3(4, 4, 1);        facingRight = false;      }      else if (!facingRight && dir > 0)      {        // ukuran player        transform.localScale = new Vector3(4, 4, 1);        facingRight = true;      }      if(isGrounded && jumpflag)      {          isGrounded = false;          jumpflag = false;          rb.AddForce(new Vector2(2f, jumpPower));      }      #endregion    }  } |

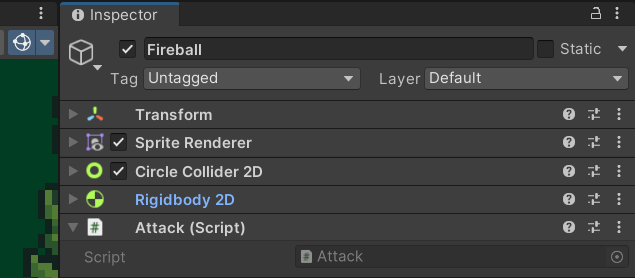
1. Atur inpector pada player bagian player Script seperti berikut ini.



1. buat script baru dengan nama Attack, 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);          }      }  } |

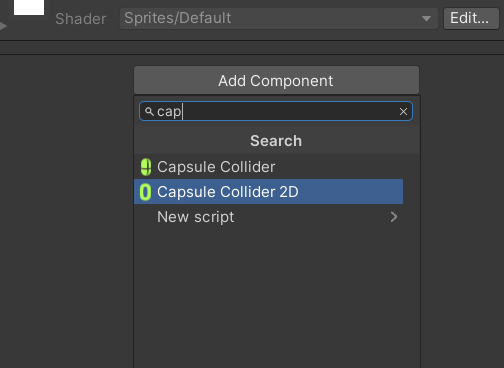
1. Drag & drop file script Attack ke objek fireball



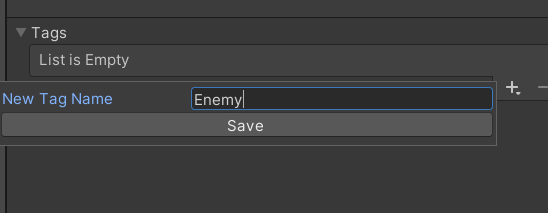
1. Tambahkan asset musuh 1 pada scene game



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



1. Tambahkan tag bernama enemy



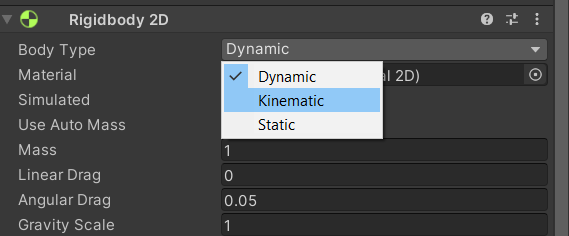
1. Pada objek musuh 1 pilih tag enemy



1. Hasil Attack



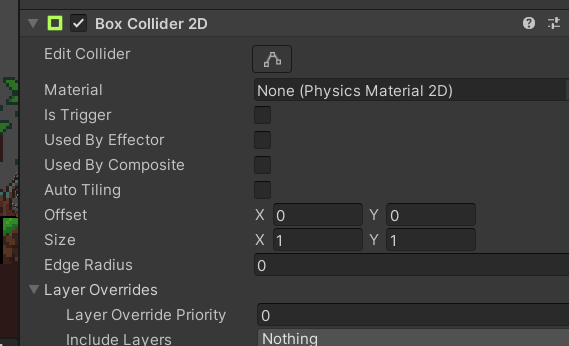
1. **Membuat Enemy Behavior NPC**
2. Pada objek musuh 1 tambahkan komponen Capsule Collider 2D dan RigidBody 2DPada komponen RigidBody 2D ubah tipe body menjadi Kinematic.

****

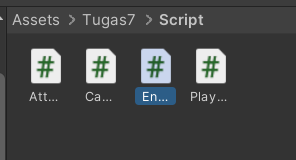
1. Tambahkan objek empty, lalu ubah nama menjadi Boundary



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



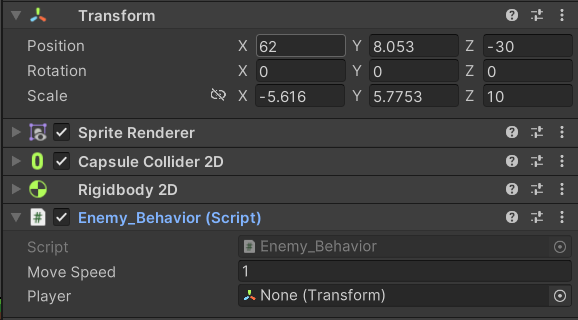
1. Buat file script baru bernama Enemy\_Behavior



1. 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;      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);      }  } |

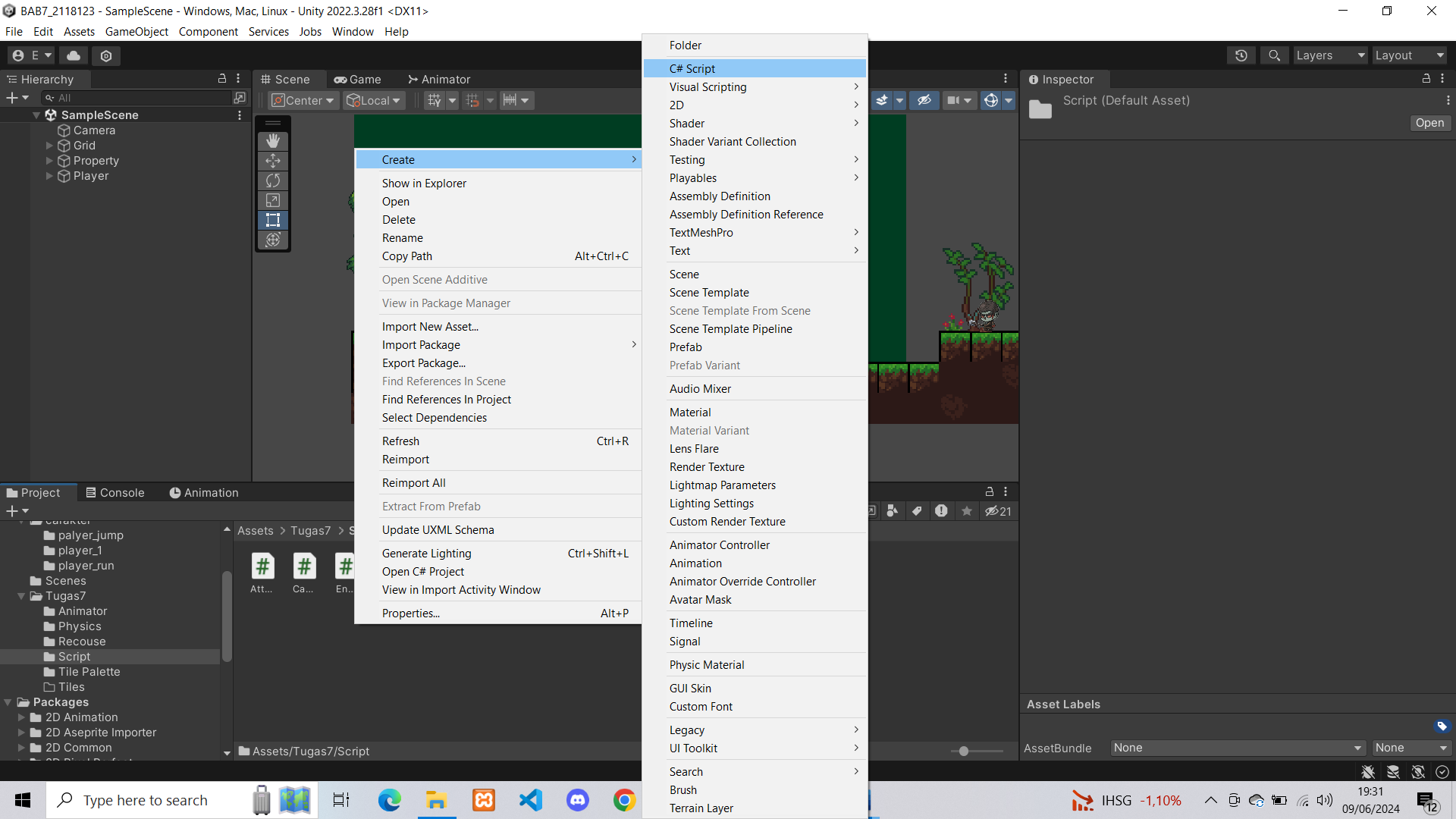
1. Drag and drop file script Enemy\_Behavior ke Musuh\_1.



1. **Membuat Enemy AI**
2. Tambahkan musuh baru pada hierarcy



1. tambahkan script baru,dengan nama Enemy\_AI



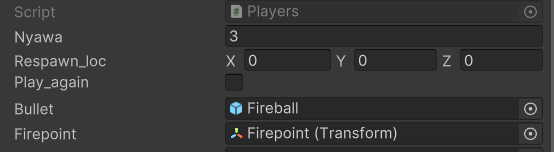
1. Masukkan source code brikut pada file script Enemy\_AI.cs

|  |
| --- |
| 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);  }  } |

1. **Respawn**
2. Buat Source code player menjadi seperti berikut

|  |
| --- |
| 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 = 500;    float horizontalValue;    [SerializeField] bool isGrounded; // +    bool facingRight;    bool jump;    void FixedUpdate()    {    GroundCheck();    Move(horizontalValue, jump);    animator.SetFloat("Blend",Mathf.Abs(rb.velocity.x));    animator.SetFloat("Blend Jump",rb.velocity.y);    jump = false;    }    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, Quaternion.identity);       fireball.GetComponent<Rigidbody2D>().velocity = new Vector2(direction \* 10f, 0);         Destroy(fireball, 2f);  }    void GroundCheck()    {      isGrounded = false;      Collider2D[] colliders = Physics2D.OverlapCircleAll(groundcheckCollider.position, groundCheckRadius, groundLayer);      if (colliders.Length > 0){          isGrounded = true;      }      animator.SetBool("Jumping",!isGrounded);    }    void playagain(){        if(play\_again == true){          nyawa = 3;          transform.position = respawn\_loc;          play\_again = false;        }    }    private void Awake()    {      rb = GetComponent<Rigidbody2D>();      animator = GetComponent<Animator>();      respawn\_loc = transform.position;    }    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());          animator.SetBool("PlayerAttack", true);      }      else if (Input.GetKeyUp(KeyCode.C))      {          animator.SetBool("PlayerAttack", false);      }      if (nyawa < 0){        playagain();      }      if(transform.position.y<-10){        play\_again = true;        playagain();      }  }    void Move(float dir, bool jumpflag)    {      #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)      {        // ukuran player        transform.localScale = new Vector3(4, 4, 1);        facingRight = false;      }      else if (!facingRight && dir > 0)      {        // ukuran player        transform.localScale = new Vector3(4, 4, 1);        facingRight = true;      }      if(isGrounded && jumpflag)      {          isGrounded = false;          jumpflag = false;          rb.AddForce(new Vector2(2f, jumpPower));      }      #endregion    }  } |

1. Pada Player ubah nyawa menjadi 3



1. **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.