



TUGAS PERTEMUAN: 10

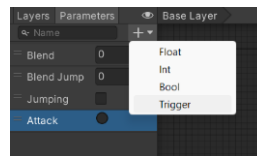
Respawn and AI Enemy Attack

NIM	:	2118126
Nama	:	Prita Patricia Lakzmi
Kelas	:	D
Asisten Lab	:	Wisando Berlian P. (2218095)

10.1 Tugas 10 : Membuat Mekanisme Attack

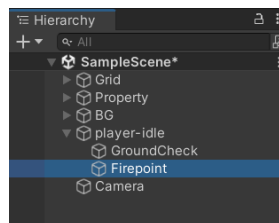
A. Mekanisme Attack

1. Buka *Project Unity* pada 2118126_BAB9. Dan buat 1 buah parameter dengan tipe data *Trigger* dan rename menjadi *Attack*.



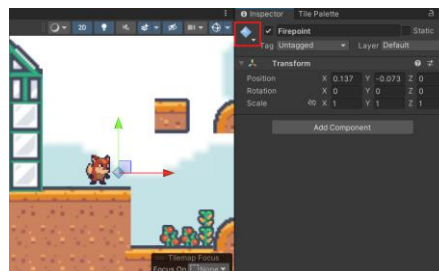
Gambar 10.1 Tambah Parameter *Attack*

2. Pada *Hierarchy* buat *GameObject* dan di dalam *player_idle* dan beri nama *Firepoint*.



Gambar 10.2 *GameObject* Dengan Nama *Firepoint*

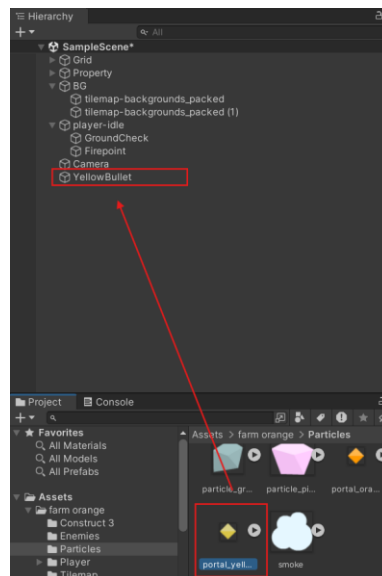
3. *Inspector* pada *Firepoint*, ubah *Icon* menjadi seperti pada gambar dibawah ini. Dan posisikan tepat di depan *player*.



Gambar 10.3 Ubah *Icon Firepoint*

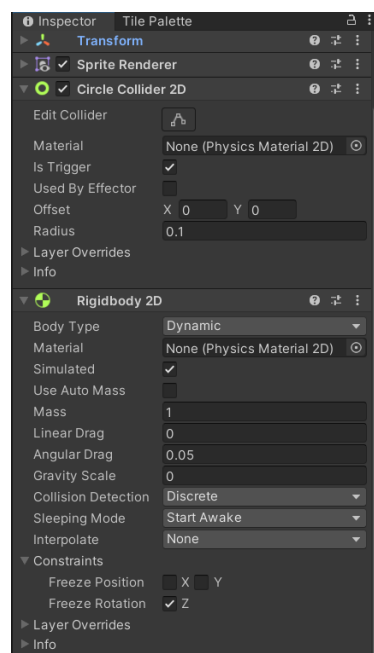


4. *Drag and drop* 'portal_yellow' ke dalam *Hierarchy YellowBullet*. Portal_yellow ini akan digunakan sebagai peluru.



Gambar 10.4 *Drag and Drop Bullet* ke *YellowBullet*

5. *Add Component Circle Collider 2D* dan *Rigidbody 2D* pada *Hierarchy YellowBullet*. Dan atur sesuai dengan gambar dibawah ini.



Gambar 10.5 *Inspector YellowBullet*

6. Buka *script Player* dan tambahkan *source code* berikut di dalam kelas *Player* paling atas.

```
public Animator animator;  
public GameObject bullet;
```



```
public Transform firePoint;
```

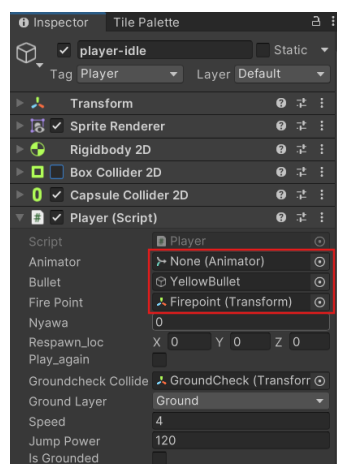
7. Tambahkan *script* untuk peluru dan letakkan dibawah fungsi *fixedUpdate*.

```
IEnumerator Attack() {  
    animator.SetTrigger("Attack");  
    yield return new WaitForSeconds(0.25f);  
    float direction = 1f;  
    GameObject yellowbullet = Instantiate(bullet,  
    firePoint.position, Quaternion.identity);  
  
    yellowbullet.GetComponent<Rigidbody2D>().velocity =  
    new Vector2(direction * 10f, 0);  
    Destroy(yellowbullet, 2f);  
}
```

8. Dan tambahkan *code* berikut pada *Function Void Update*.

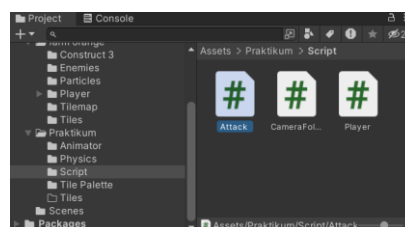
```
if (Input.GetKeyDown(KeyCode.C)) {  
    StartCoroutine(Attack());  
}
```

9. Pada *inspector* player-idle atur *component* *Player (Script)*, ubah *Bullet* menjadi *YellowBullet*. Dan *Firepoint* menjadi *Firepoint (Transform)*.



Gambar 10.6 Mengatur *Inspector* *Player-idle*

10. Tambahkan file *script* dengan nama *Attack* di dalam folder *script*.



Gambar 10.7 Membuat File *Script* *Attack*

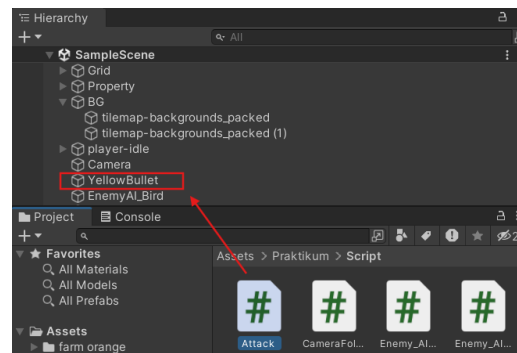


11. Tambahkan *script Attack* dibawah ini.

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

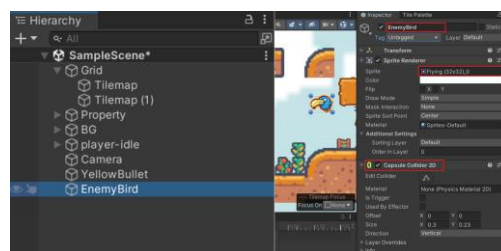
public class Attack : MonoBehaviour
{
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            Destroy(gameObject);
            Destroy(collision.gameObject);
        }
    }
}
```

12. Masukkan file *script Attack* ke dalam *Hierarchy YellowBullet*.



Gambar 10.8 Drag and Drop Script Attack ke YellowBullet

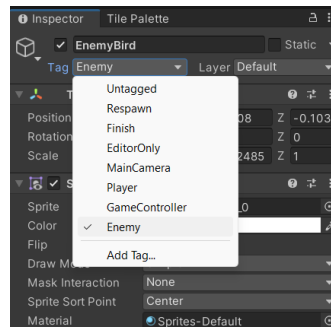
13. Kemudian tambahkan *component Capsule Collider 2D* pada *inspector* 'EnemyAI_Bird'. Dan pada component *Sprite Renderer* pastikan *sprite* sudah di isi dengan *enemy* yang akan digunakan.



Gambar 10.9 Add Component Capsule Collider 2D



14. Add tag dengan nama *Enemy* pada *inspector* ‘EnemyAI_Bird’. Dan gunakan *tag Enemy* pada ‘EnemyAI_Bird’.

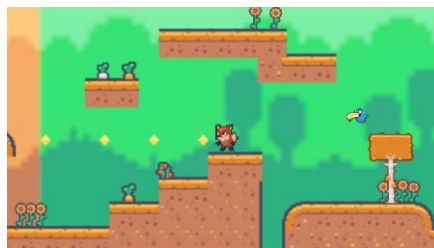


Gambar 10.10 Ubah Tag Pada EnemyAI_ Bird

15. Ubah *script* pada *script* peluru dengan *code* dibawah ini untuk membuat peluru dapat ditembakkan sesuai posisi dari *player*.

```
float direction = facingRight ? 1f : -1f;
```

16. Tampilan pada saat di *play*.

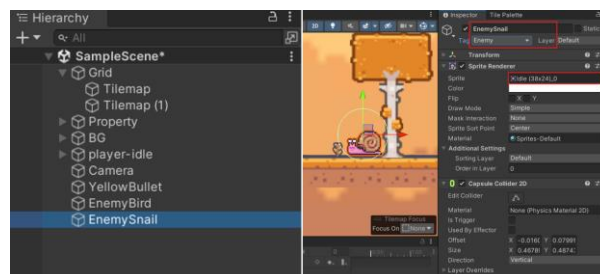


Gambar 10.11 Tampilan Saat Di Play

10.2 Membuat Enemy AI

A. Enemy Behavior NPC Snail

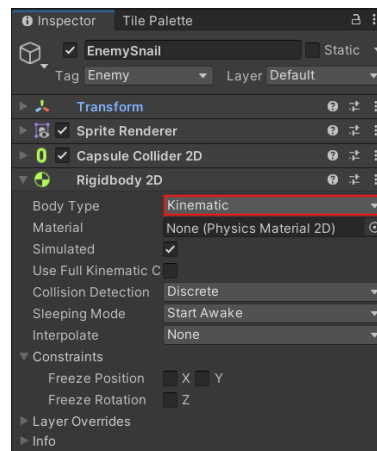
1. *Create Empty* pada *Hierarchy* dengan nama “EnemySnail”. Dan berikan *component* Sprite Renderer dan Capsule Collider 2D. Lalu atur tag menjadi *Enemy*.



Gambar 10.12 *Create Empty* Dengan Nama *EnemySnail*

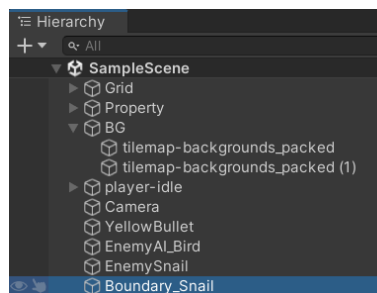


2. Tambahkan *component* Rigidbody 2D pada dan atur *Body Type* nya menjadi 'Kinematic'.



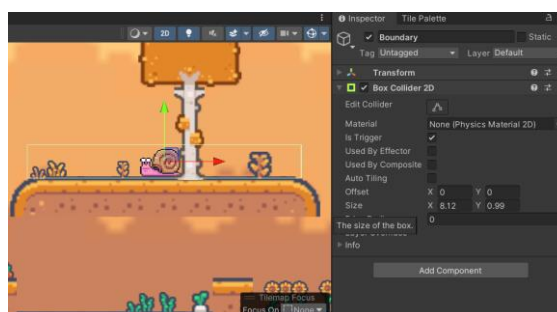
Gambar 10.13 Atur *Component* Pada EnemySnail

3. *Create Empty* pada *Hierarchy* dan *rename* menjadi "Boundary_Snail".



Gambar 10.14 *Create Empty* Dengan Nama Boundary_Snail

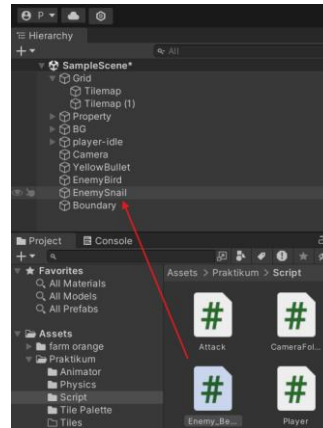
4. *Add Component* Box Collider 2D pada Boundary_Snail, atur posisi *boundary* dan centang *Is Trigger*.



Gambar 10.15 Atur *Boundary* Pada Boundary_Snail



5. Buat sebuah file *script* didalam folder *Script* beri nama “Enemy_Snail”. Kemudian *drag and drop* ke dalam *Hierarchy* EnemySnail



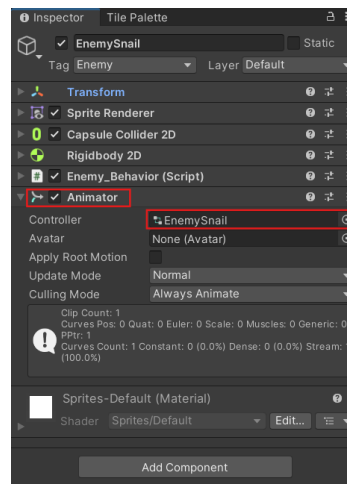
Gambar 10.16 *Drag and Drop Script Enemy_Snail ke EnemySnail*

6. Tambahkan *script* Enemy_Snail dibawah ini.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Enemy_Snail : 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);
    }
}
```

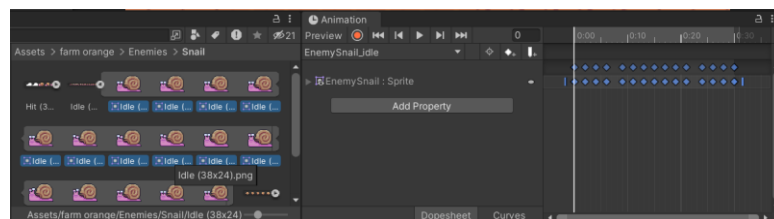


7. *Add Component Animator* pada EnemySnail dan atur *Controller* nya menjadi EnemySnail.



Gambar 10.17 *Add Component Animator* di EnemySnail

8. Buat File *Animator Controller* di dalam folder *Animator* dan beri nama “EnemySnail”. Kemudian buat animasi pada EnemySnail. Kemudian *drag and drop enemy* ke dalam *timeline frame* dari frame 1-30.



Gambar 10.18 Membuat Animasi di EnemySnail

9. Tampilan pada saat di *play*.

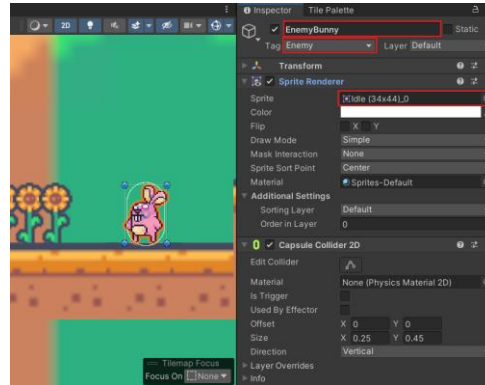


Gambar 10.19 Tampilan Saat Di *Play*



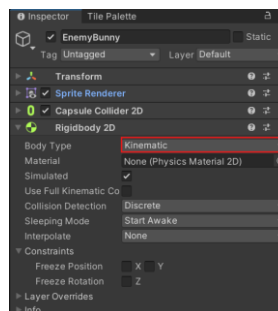
B. Enemy Behavior NPC Bunny

1. *Create Empty* pada *Hierarchy* dengan nama “EnemyBunny”. Dan berikan *component* *Sprite Renderer* dan *Capsule Collider 2D*. Lalu atur tag menjadi *Enemy*.



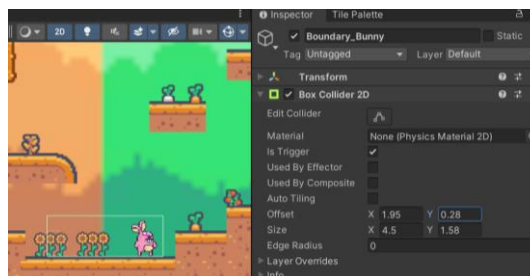
Gambar 10.20 *Create Empty* Dengan Nama EnemyBunny

2. Tambahkan *component* *Rigidbody 2D* pada dan atur *Body Type* nya menjadi ‘Kinematic’.



Gambar 10.21 Atur *Component* Pada EnemySnail

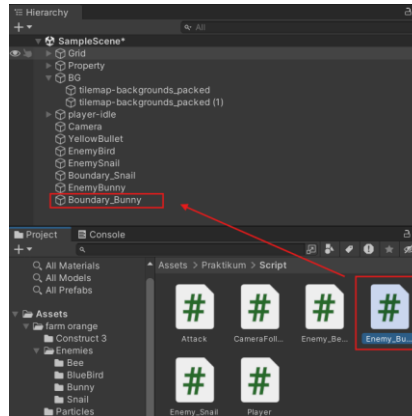
3. *Create Empty* pada *Hierarchy* dan *rename* menjadi “Boundary_Bunny”. *Add Component* *Box Collider 2D* pada *Boundary_Bunny*, atur posisi *boundary* dan centang *Is Trigger*.



Gambar 10.22 Atur *Boundary* Pada Boundary_Snail

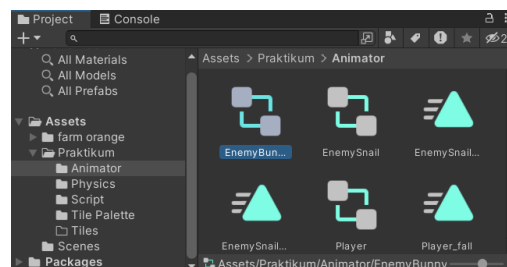


4. Buat sebuah file script didalam folder *script* beri nama “Enemy_Bunny”. Kemudian *drag and drop* ke dalam *Hierarchy* Boundary_Bunny.



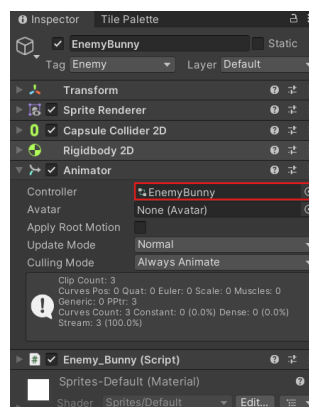
Gambar 10.23 *Drag and Drop Script Enemy_Bunny* ke *Enemy_Bunny*

5. Buat File *Animator Controller* di dalam folder *Animator* dan beri nama “EnemyBunny”.



Gambar 10.24 *Animator Controller* Bernama *EnemyBunny*

6. *Add Component Animator* pada *EnemyBunny* dan atur *Controller* nya menjadi *EnemyBunny*. Dan berikan animasi pada *EnemyBunny*.



Gambar 10.25 *Add Component Animator* di *EnemySnail*



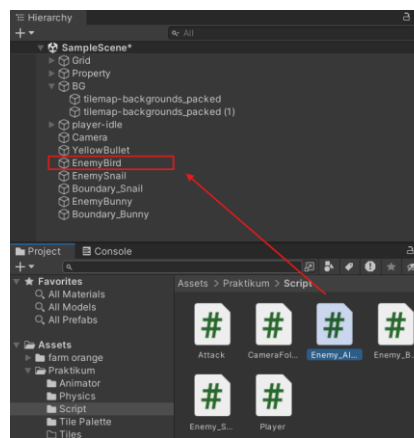
7. Tampilan pada saat di *play*.



Gambar 10.26 Tampilan Saat di *Play*

C. Enemy AI Bird

1. Masukkan *script* *Enemy_AI_Bird* ke dalam *Hierarchy* *EnemyAI_Bird*.



Gambar 10.27 *Drag and Drop Script* ke *EnemyAI_Bird*

2. Tambahkan *Script* *Enemy_AI_Bird* dibawah ini.

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

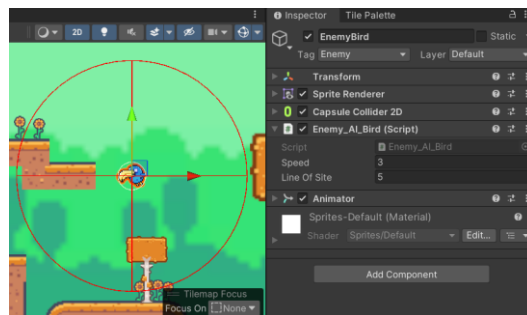
public class Enemy_AI_Bird : MonoBehaviour
{
    public float speed; // Kecepatan gerakan musuh
    public float lineOfSite; // Jarak penglihatan musuh
    private Transform player; // Transform dari pemain
    private Vector2 initialPosition; // Posisi awal
    musuh
    void Start()
    {
        player =
        GameObject.FindGameObjectWithTag("Player").transform;
        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. Buka *Enemy_AI_Bird (Script)* pada *inspector* *EnemyAI_Bird* dan atur seperti pada gambar dibawah ini.



Gambar 10.28 Atur *Inspector* *EnemyAI_Bird*

4. Tampilan pada saat di *play*.

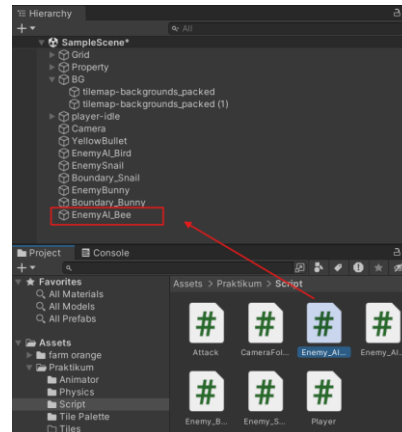


Gambar 10.29 Tampilan Saat di *Play*



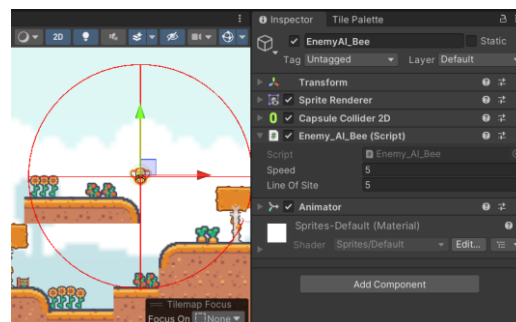
D. Enemy AI Bee

1. Masukkan *script* Enemy_AI_Bee ke dalam *Hierarchy* EnemyAI_Bee.



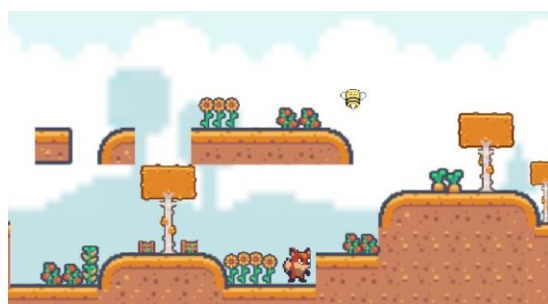
Gambar 10.30 Drag and Drop Script ke EnemyAI_Bee

2. Buka Enemy_AI_Bee (Script) pada *inspector* EnemyAI_Bee dan atur seperti pada gambar dibawah ini.



Gambar 10.31 Atur *Inspector* EnemyAI_Bee

3. Tampilan pada saat di *play*.



Gambar 10.32 Tampilan Saat di *Play*



10.3 Respawn

1. Buka file *script* (Player.cs) dan tambahkan variabel nyawa seperti dibawah ini. Letakkan di dalam kelas *player*.

```
public int nyawa;  
[SerializeField] Vector3 respawn_loc;  
public bool play_again;
```

2. Tambah *code* dibawah ini untuk mengatur posisi *respawn* agar sesuai dengan posisi awal bermain dimulai.

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

3. Kemudian tambahkan kode dibawah ini di dalam *void update* Player.cs agar ketika nyawa *player* dibawah 0 maka akan melakukan respawn

```
if (nyawa < 0) {  
    playagain();  
}
```

4. Selanjutnya tambahkan kode berikut dibawah *code* sebelumnya agar ketika *player* jatuh dibawah *platform* akan melakukan *respawn*

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

5. Tambahkan fungsi *playagain()* dalam script Player.cs

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

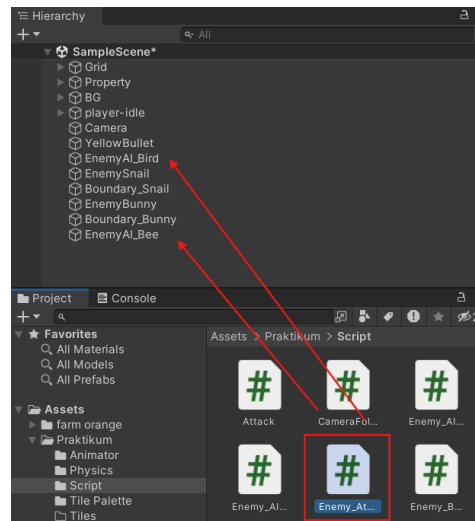
6. Tambah file *script* (Enemy_Attacked.cs) dan isikan *source code* dibawah ini.

```
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>()  
            ;  
        }  
    }  
}
```



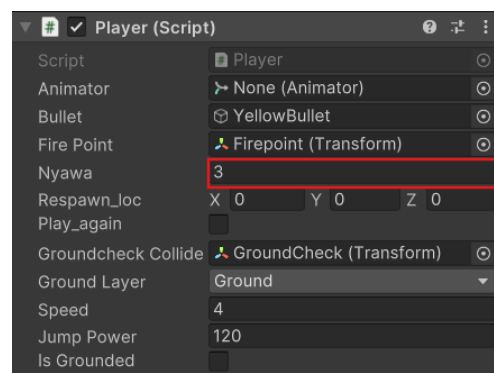
```
    }  
}  
void OnTriggerEnter2D(Collider2D other)  
{  
    if (other.CompareTag("Player"))  
    {  
        Object.nyawa--;  
        if (Object.nyawa < 0)  
        {  
            Object.play_again = true;  
        }  
    }  
}
```

7. *Drag and drop script* Enemy_Attack ke dalam *Hierarchy* 'EnemyAI_Bee' dan 'EnemyAI_Bird'.



Gambar 10.33 Tampilan Saat di Play

8. Pada *inspector* player-idle di dalam *Player (Script)* berikan nilai pada Nyawa menjadi 3.





Gambar 10.34 Tampilan Saat di *Play*

10.4 KUIS

Source Code Salah :

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public int attackRange = 2.0f;
    public int attacDamage = 10;
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            PerformMeleeAttack();
        }
    }
    void PerformMeleeAttack()
    {
        RaycastHit hit;
        if (Physics.Raycast(transform.position,
transform.forward, out hit, attackRange))
        {
            // Lengkapi kode di sini untuk mengenai musuh dan
mengurangi health mereka
        }
    }
}
```

Source Code Benar :

```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    public int attackRange = 2.0f;
    public int attacDamage = 10;
    void Update()
    {
        if (Input.GetButtonDown("Fire1"))
        {
            PerformMeleeAttack();
        }
    }
    void PerformMeleeAttack()
    {
        RaycastHit hit;
        if (Physics.Raycast(transform.position,
transform.forward, out hit, attackRange))
        {
            EnemyHealth enemyHealth =
hit.transform.GetComponent<EnemyHealth>();

            if (enemyHealth != null)
            {
                enemyHealth.TakeDamage(attacDamage);
            }
        }
    }
}
```




```
}  
    }  
}
```

Analisa :

Source code PlayerAttack diatas adalah source code untuk memungkinkan player menyerang musuh dan mengurangi kesehatan enemy berdasarkan input deteksi tabrakan dengan enemy. Apabila player menekan button Fire1 maka PerformMeleeAttack akan dipanggil. Dan source code ini juga memberikan jangkauan bagi player untuk mendeteksi jika ada serangan dan tabrakan dengan enemy di depan yang masuk kedalam jangkauannya. Dan jika player terkena serangan player memiliki komponen EnemyHealth dan TakeDamage dipanggil untuk mengurangi kesehatan musuh sebesar attacDamage.