

**LAPORAN KUIS**  
**KOMPUTASI MULTIMEDIA**  
**“GHOST FIND SKELETON”**



**Oleh Kelompok 6:**

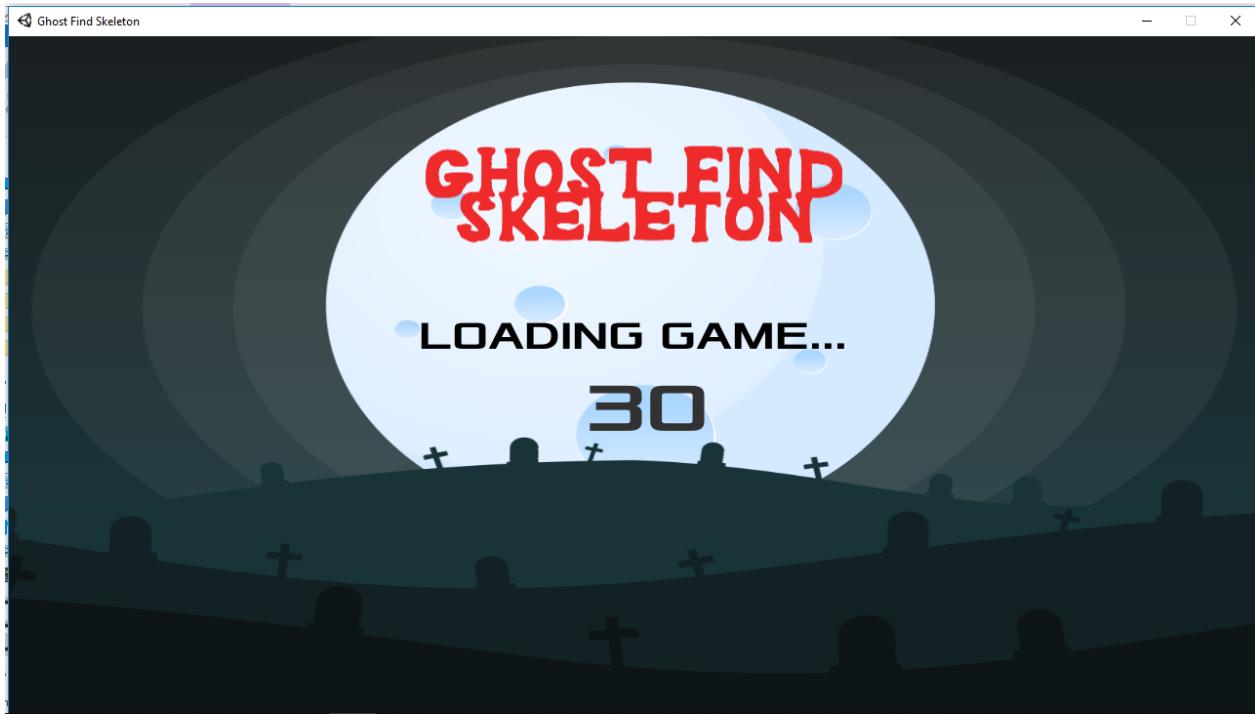
<b>Bayu Fajariyanto</b>	<b>1731710033</b>
<b>Ramdaning Puri Pradani</b>	<b>1731710102</b>
<b>Silva Aisyah Maharnis</b>	<b>1731710072</b>

**MI 2E**

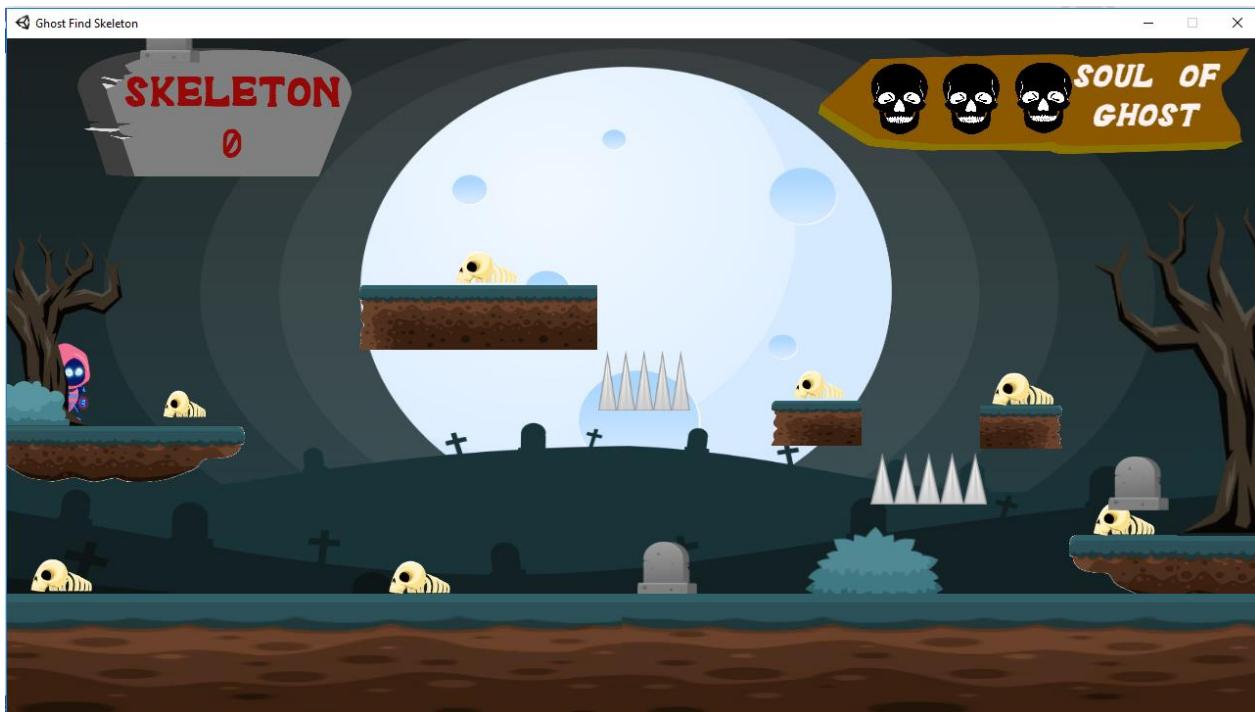
**PROGRAM STUDI MANAJEMEN INFORMATIKA**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2019**

## Tampilan Game “Ghost Find Skeleton”

**Scene Loading** adalah scene yang digunakan untuk memulai game dengan hitungan mundur. Dimulai dari 30, 29,... sampai 2,1 setelah itu game akan mulai.



**Scene SampleScene** adalah scene yang berisi game dari Game Ghost Find Skeleton.



Jika menang akan tampil seperti berikut :



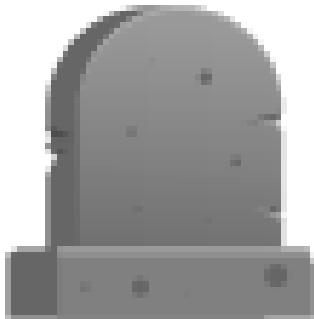
## Deskripsi Game “Ghost Find Skeleton”

Dalam Game ini terdapat beberapa object yaitu :

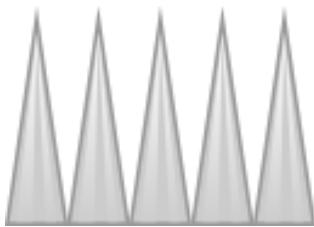
1. Ghost sebagai Player



2. Nisan sebagai Enemy



3. Spike sebagai jebakan/trap



4. Skeleton sebagai nilai.



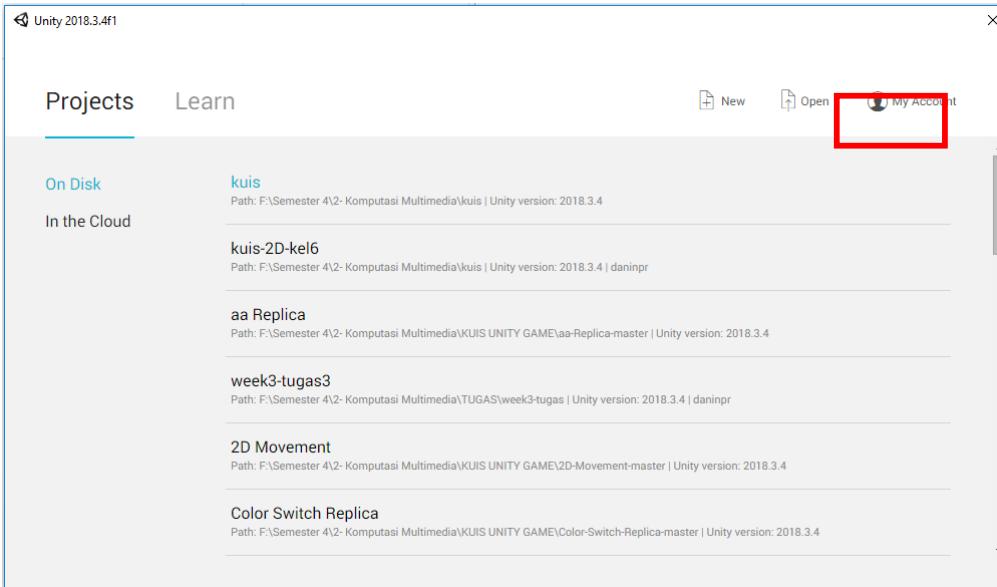
Ghost ini mencari Skeleton sebanyak-banyaknya agar point bertambah. Lalu, Nisan disini bertugas agar sebagai Enemy dari Ghost jika Ghost terkena Nisan maka nyawa dari Ghost akan berkurang (Ghost diberi nyawa 3). Ada juga Spike yaitu jebakan, jika Ghost terkena jebakan/trap Game ini akan mengulang dari awal otomatis nyawa Ghost kembali menjadi 3 dan Skeleton yang sudah di dapatkan kembali menjadi 0.

## Langkah-Langkah Pembuatan :

Laporan ini dibuat ketika Game sudah jadi dan bisa di Build.

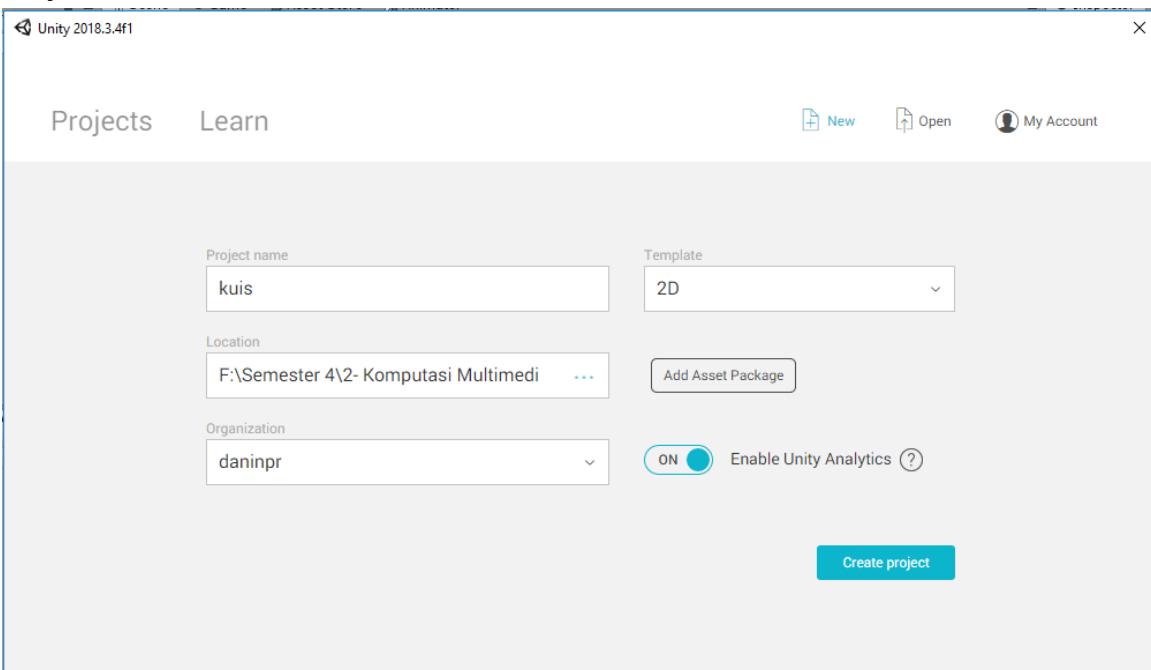
### Langkah 1 : Membuat Enviroment pada Game “Ghost Find Skeleton”

1. Membuka software unity, untuk kelompok kami menggunakan unity versi 2018.3.4f1 dan buat project baru.

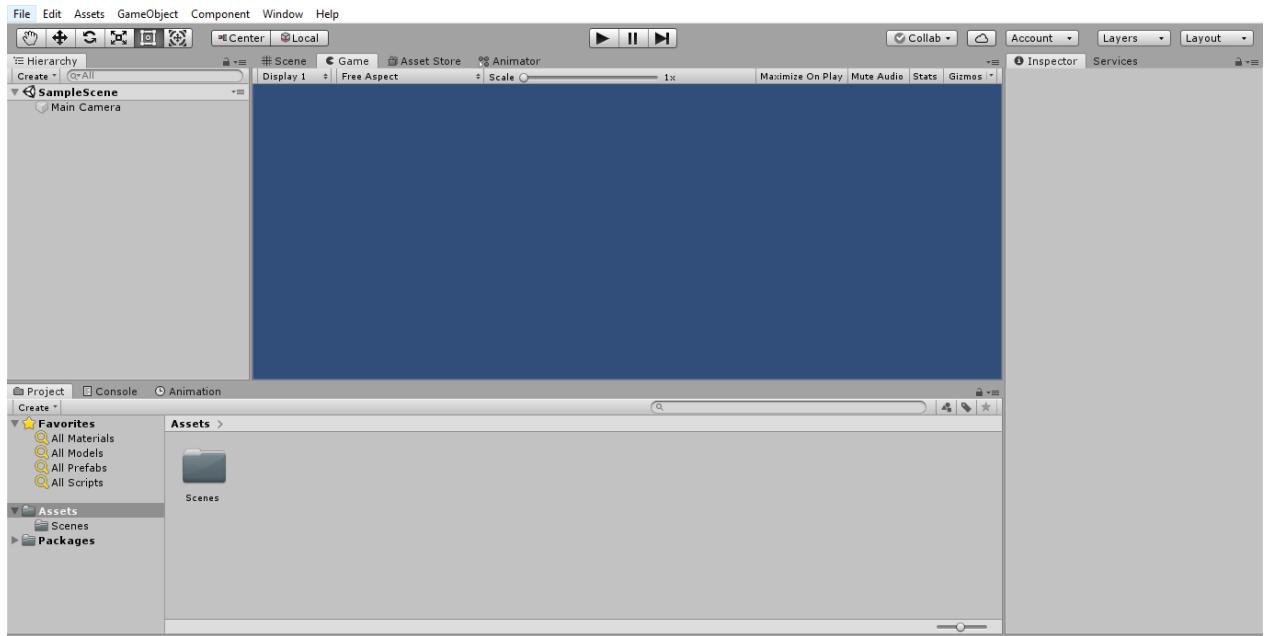


Untuk membuat project baru pada bagian **New**.

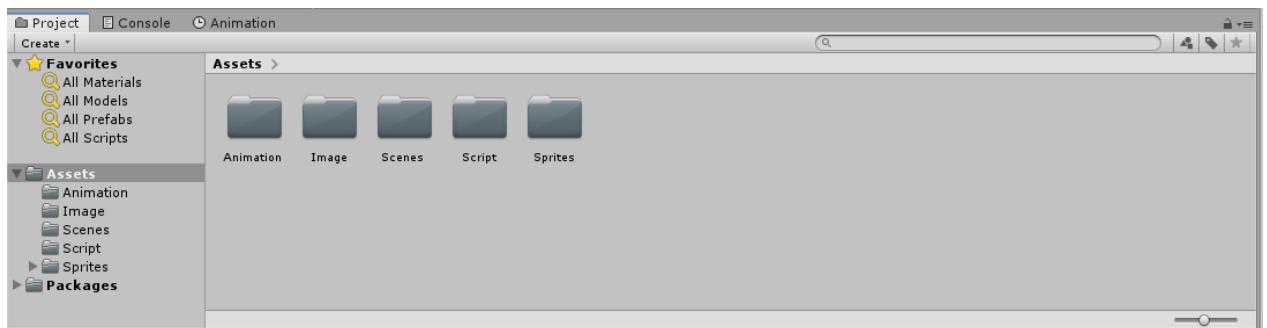
2. Untuk dapat membuat project dengan cara isi kan nama “**kuis**” yang akan dibuat pada kolom **Project Name** dan tentukan **Location** setelah itu pilihlah yang akan di buat projek **2D** pada bagian **Template**, jangan lupa setting nama **Organization**. Lalu Klik **Create Project** untuk memulai.



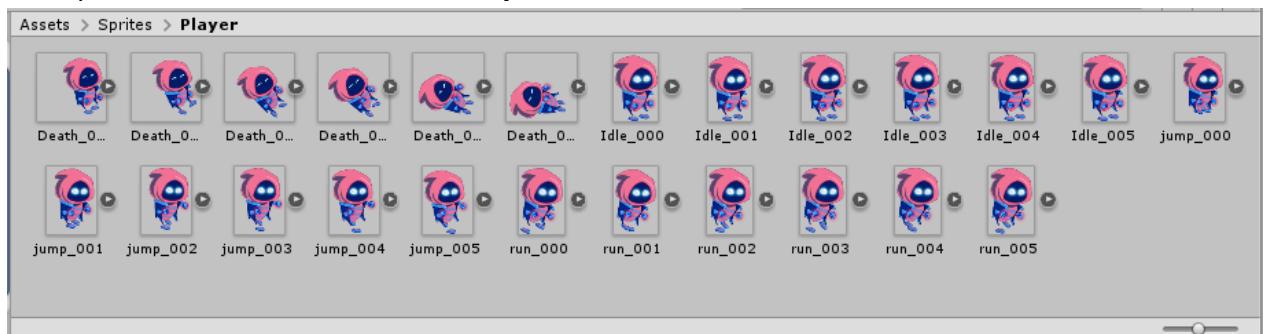
3. Jika project sudah dibuat maka akan muncul tampilan seperti berikut ini.



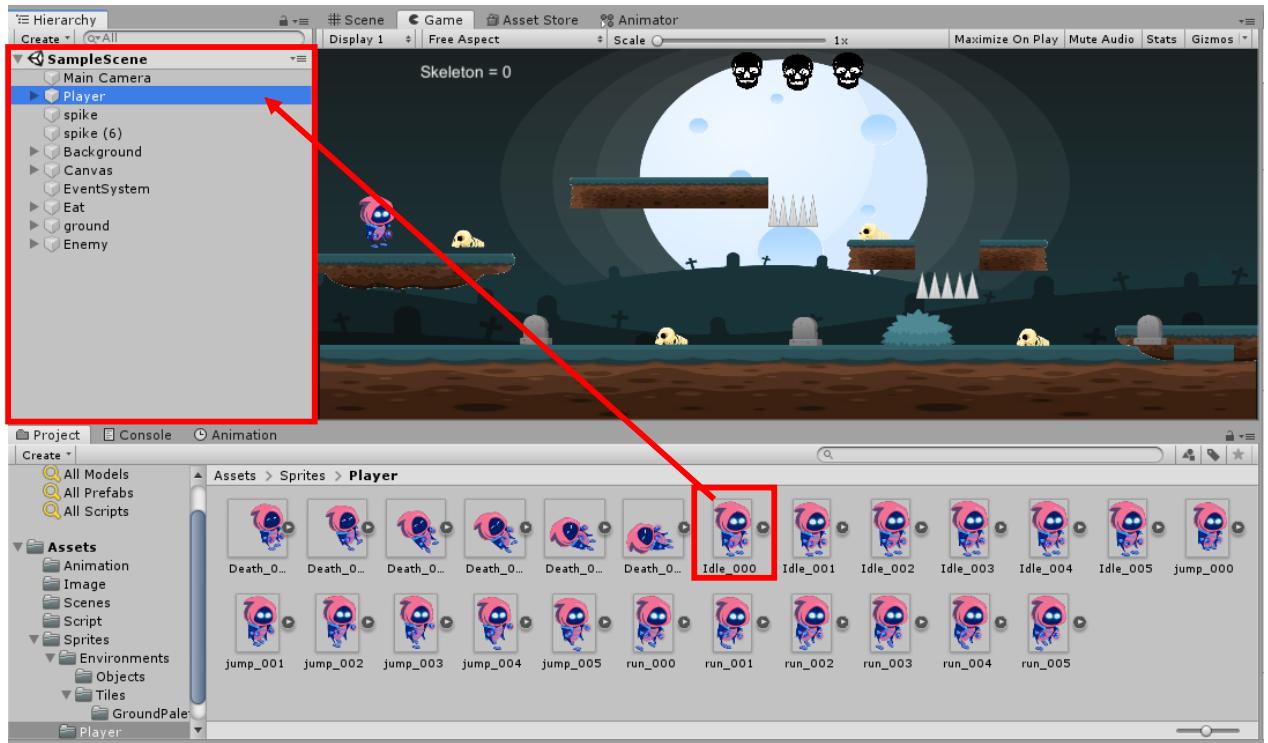
4. Pada **Assets** kelompok kami sudah menyiapkan beberapa folder untuk membuat game "Ghost Find Skeleton".



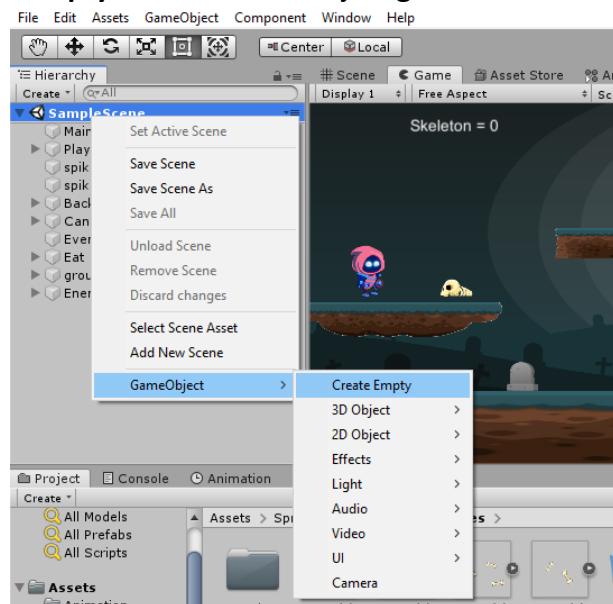
5. Langkah pertama membuat **Player** atau **Ghost**. Pada **Assets** folder **Sprites > Player** terdapat karakter untuk membuat **Player** atau **Ghost**.



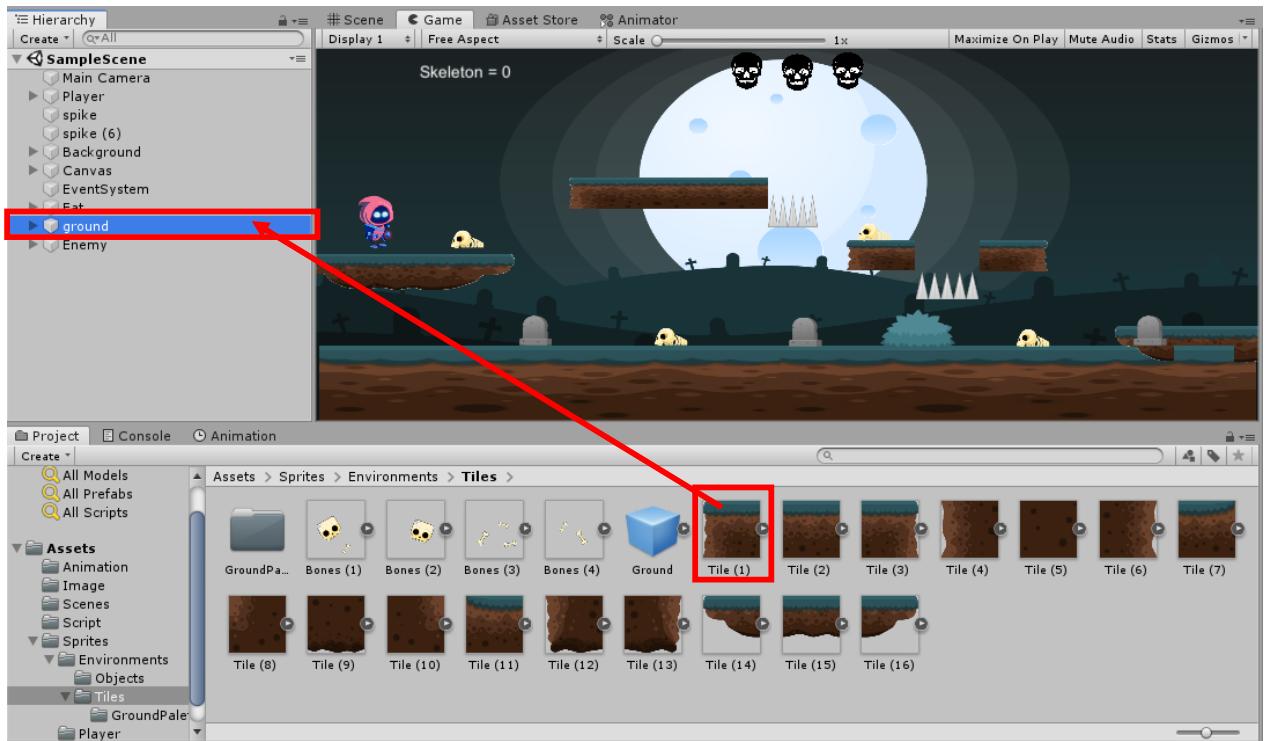
6. Pilih karakter **Idle\_000** lalu DRAG dan DROP ke **Hierarchy** atau **Sample Scene**. Setelah itu, ubah nama **Idle\_000** menjadi **Player**.



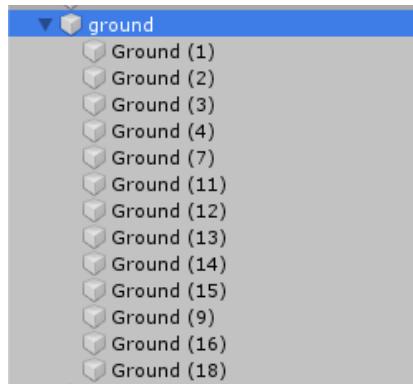
7. Membuat **GameObject Empty** dengan cara pada **Sample Scene** klik kanan lalu pilih **GameObject > Create Empty**. Ubah nama menjadi **ground**.



8. Sama seperti langkah sebelumnya, **DRAG** dan **DROP** object **Tile\_1** ( pada folder **Sprites > Enviroment > Tiles** ) ke **Hierarchy** atau **Sample Scene** di **GameObject ground**. Ubah nama **Tile\_1** menjadi **Ground**.



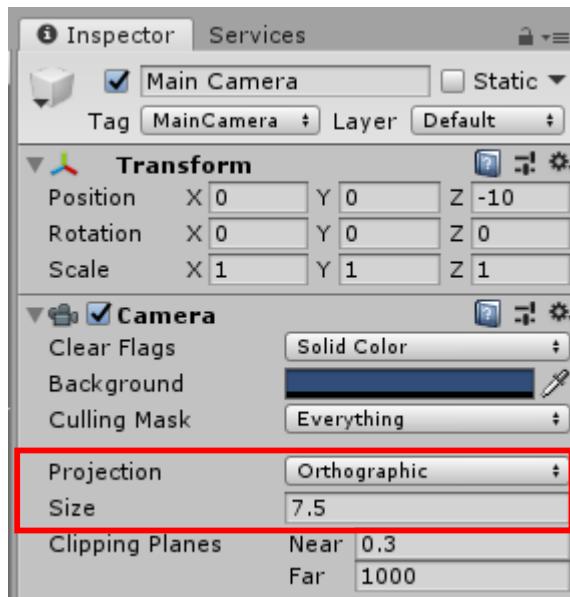
Untuk membuat lebih banyak Ground duplicate object dengan cara **CTRL+D**.



Atur letak ground dimana saja. Dan hasilnya seperti berikut ini.

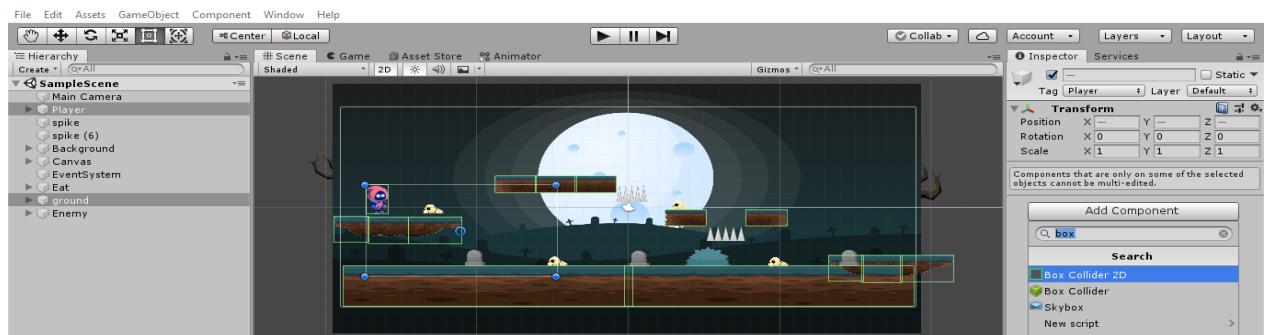


9. Pada **Main Camera** ubah ukuran kamera menjadi 7.5 (pada **Size**). Agar ukuran **Player** tidak terlihat terlalu besar.

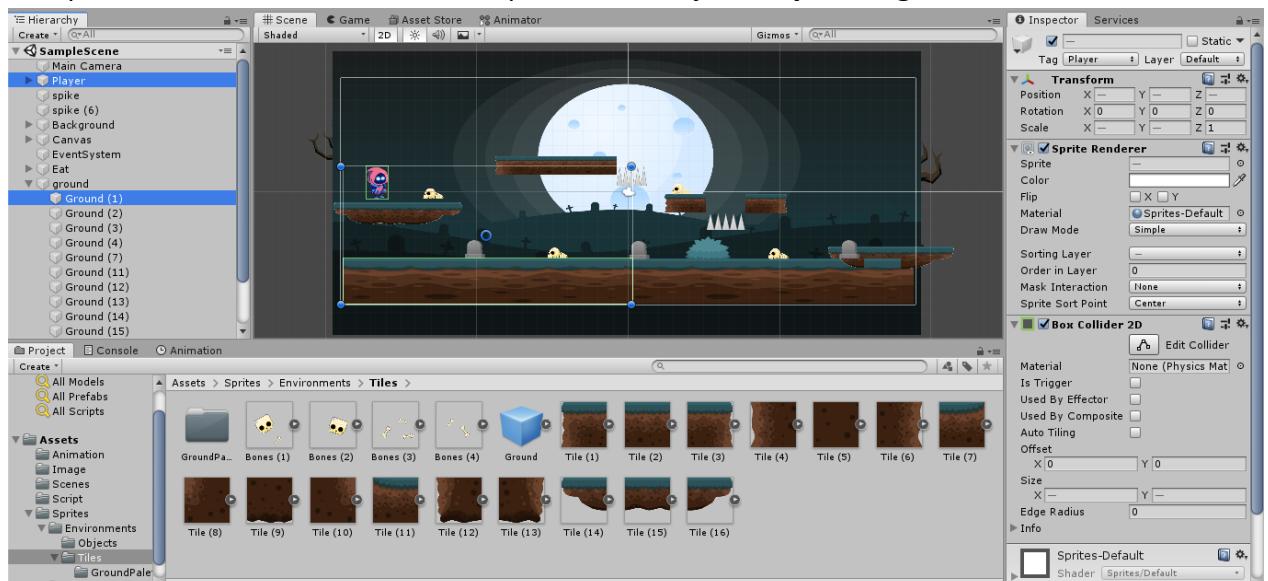


10. Tambahkan komponen **Box Collider 2D** pada **Player** dan **ground**.

Caranya, seleksi GameObject **Player** dan **ground** kemudian pada **Inspector Add Component** lalu pilih **Box Collider 2D**.



Komponen **Box Collider 2D** sudah ada pada GameObject **Player** dan **ground**.

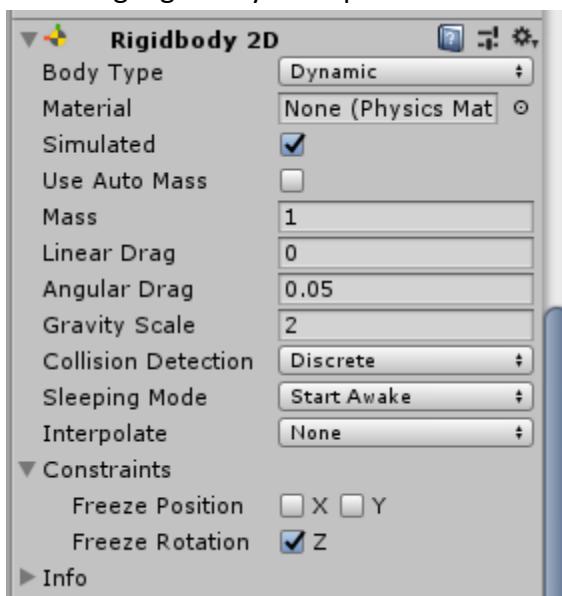


Perhatikan object **Player** dan **ground** memiliki beberapa komponen yang berbeda.

**Note :**

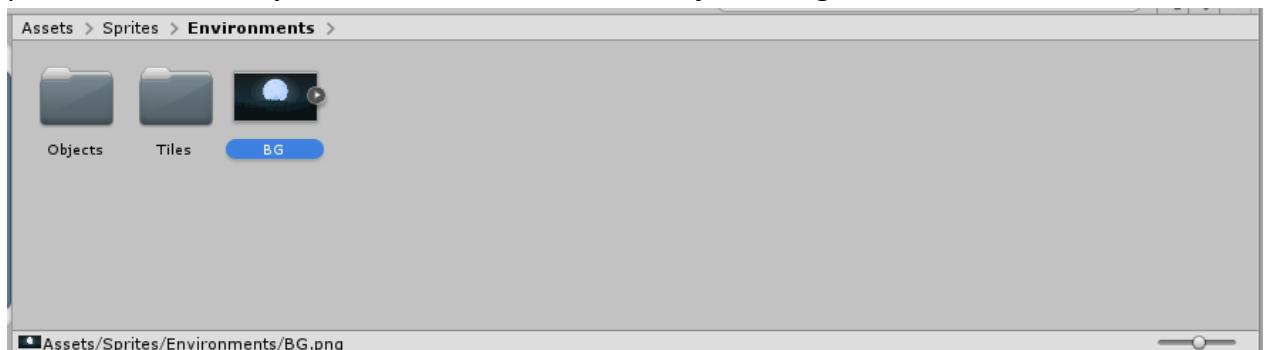
- Komponen **Transform** yaitu komponen yang dapat kita gunakan untuk memanipulasi atau memindah lokasi, rotasi dan juga scale (ukuran) ke sumbu x, y dan z.
- Komponen **Sprite Renderer** adalah komponen yang berfungsi untuk menampilkan sprite yang dapat kita gunakan untuk di dalam scene.
- Sedangkan **Box Collider 2D** yaitu komponen untuk menentukan bentuk permukaan fisik dari sebuah objek di dalam scene dan juga berguna untuk mendeteksi sebuah event saat terjadi tabrakan dengan game objek yang lain.

11. Pada GameObject Player setting Rigidbody 2D seperti di bawah ini.

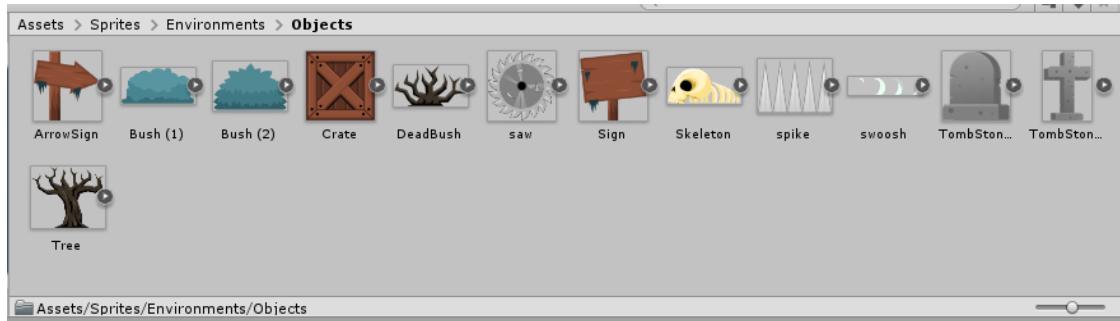


**Rigidbody 2D** ini adalah komponen yang menempatkan game objek ke dalam kendali fisika, misalnya seperti adanya pengaruh gravitasi pada game objek atau massa dari objek itu sendiri.

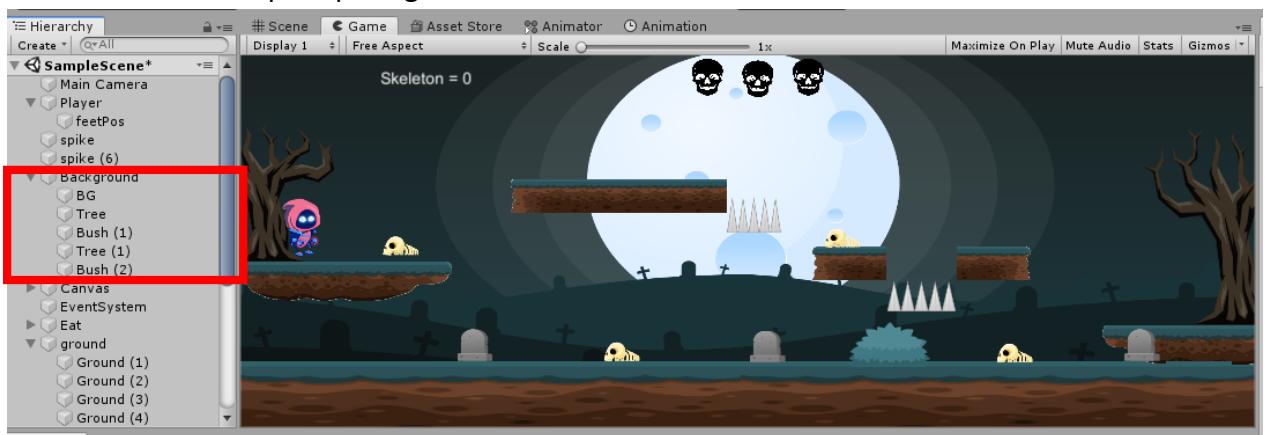
12. Membuat **GameObject Empty** dan ubah nama **Background**. **DRAG** dan **DROP** object **BG** pada **Assets folder Sprites > Enviroment** ke **GameObject Background**.



13. Tambahkan object seperti pohon dan rumput pada Background. Object ini terdapat pada **Assets folder Sprites > Enviroment > Object**, lalu **DRAG** dan **DROP** ke **GameObject Background**.



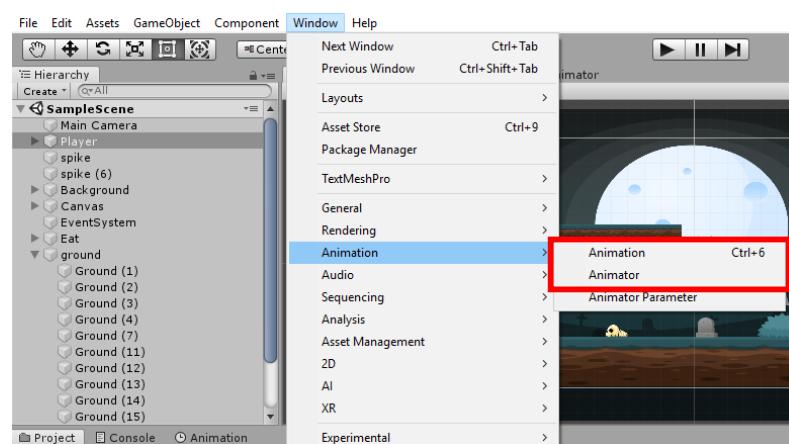
14. Jika sudah akan tampil seperti gambar di bawah ini :

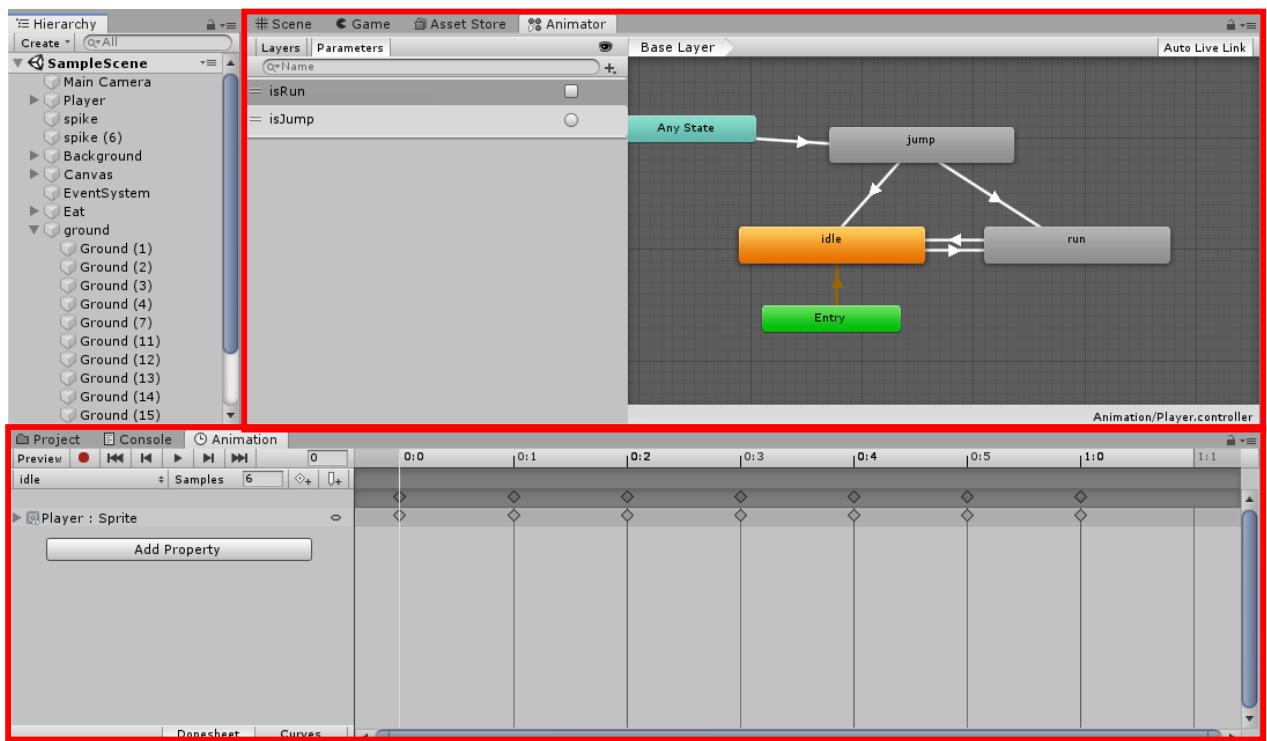


15. Coba sekarang **Play** atau **Run** terlebih dahulu, maka **Player** akan jatuh kebawah dan berhenti ketika di atas **GameObject ground**.

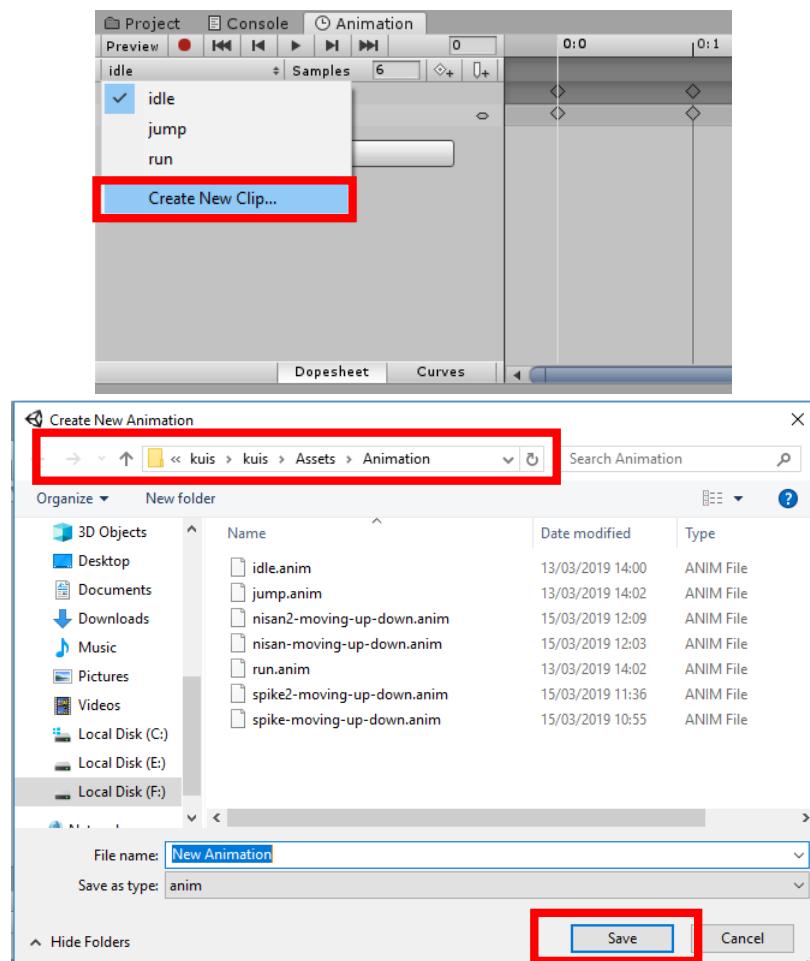
## Langkah 2 : Membuat Animasi untuk Player

- Setting transisi animasinya saat **Player** berdiam (idle), berlari (run), dan melompat (jump).
- Siapkan **Tab Animation** dan **Animator** dengan cara pilih menu **Windows** pilih **Animation** > **Animation** dan **Animation > Animator**.

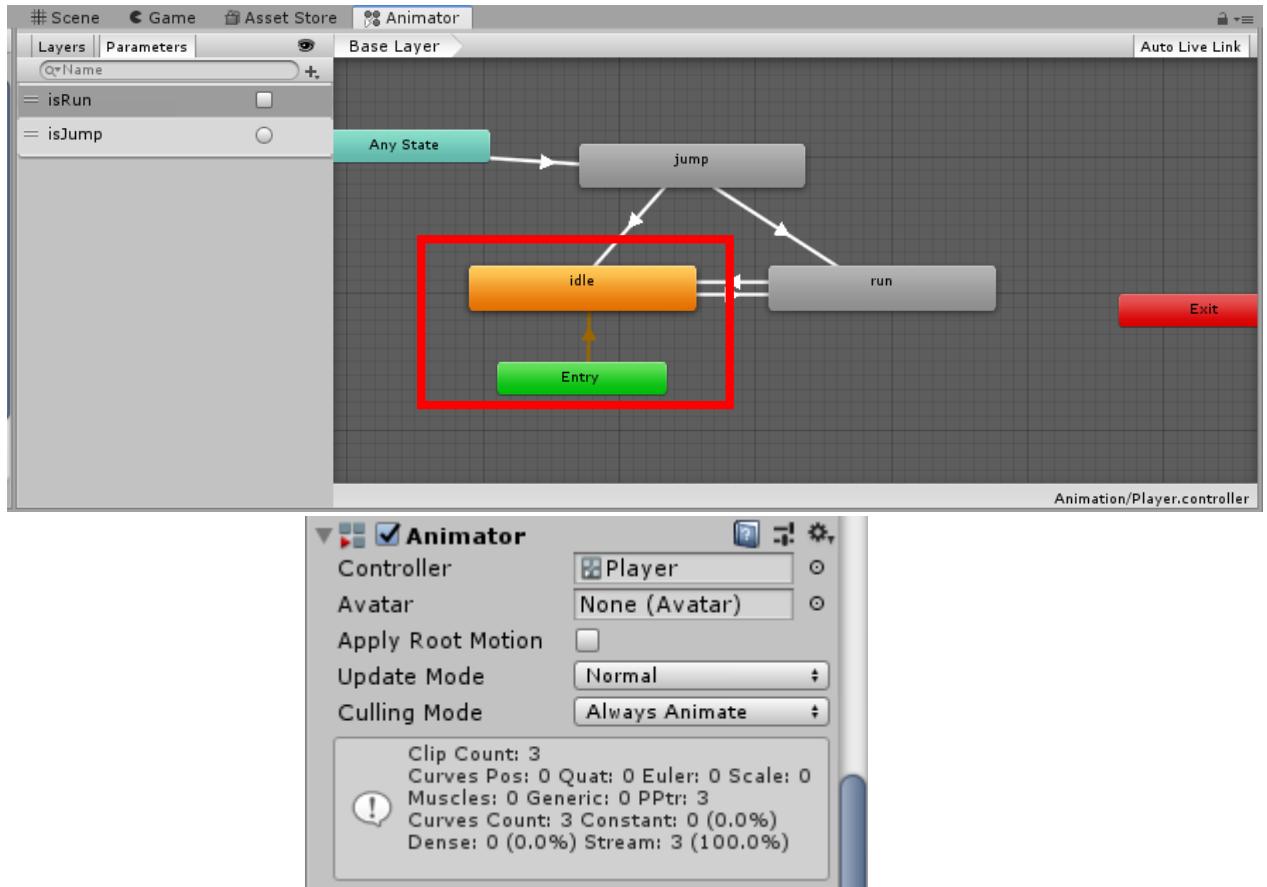




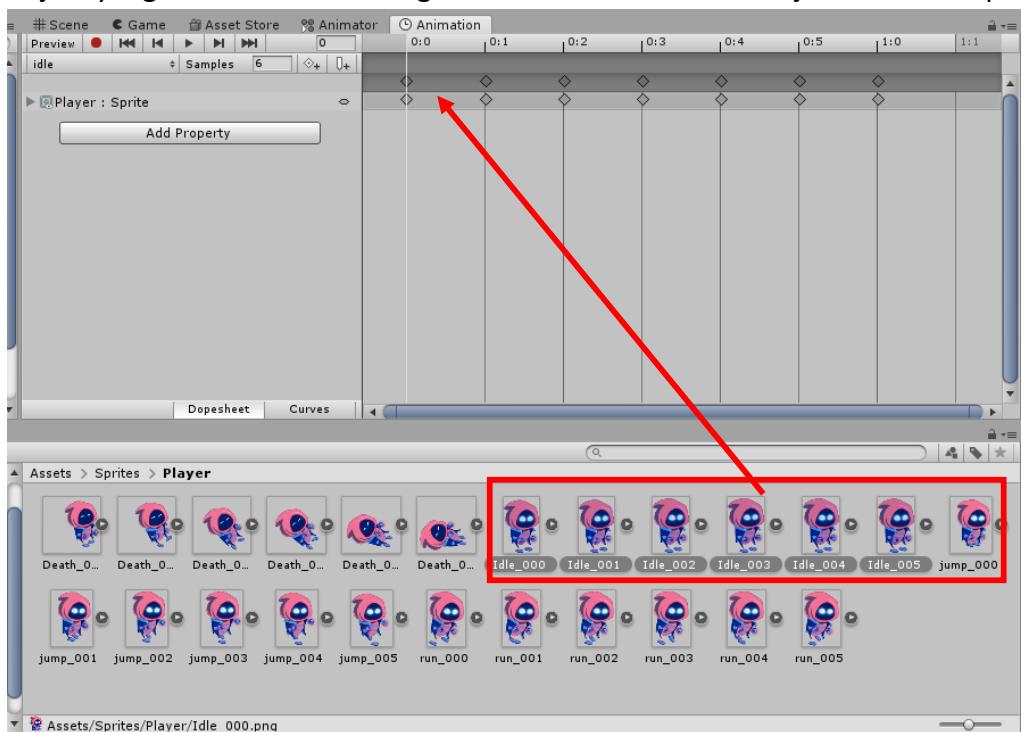
3. Lalu pilih Tab Animation dan pilih Create, simpan pada Assets folder Animation beri nama Idle, klik Save.



4. Perhatikan pada **Tab Animator** sudah terdapat **Animation Idle**.

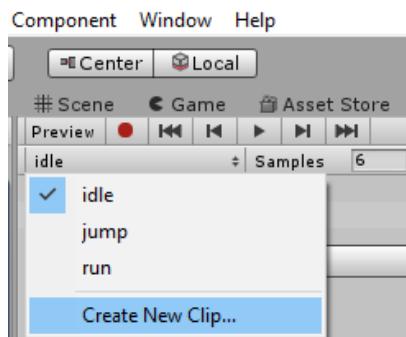


5. Lalu DRAG and DROP karakter **Idle\_000** sampai **Idle 005** ( pada **Assets** folder **Sprites > Player** ) ke dalam **Tab Animation** yang bernama **Idle**. Ganti **Sample** menjadi 6 karena object yang diambil ada 6 Idle agar animation nanti tidak berjalan terlalu cepat.



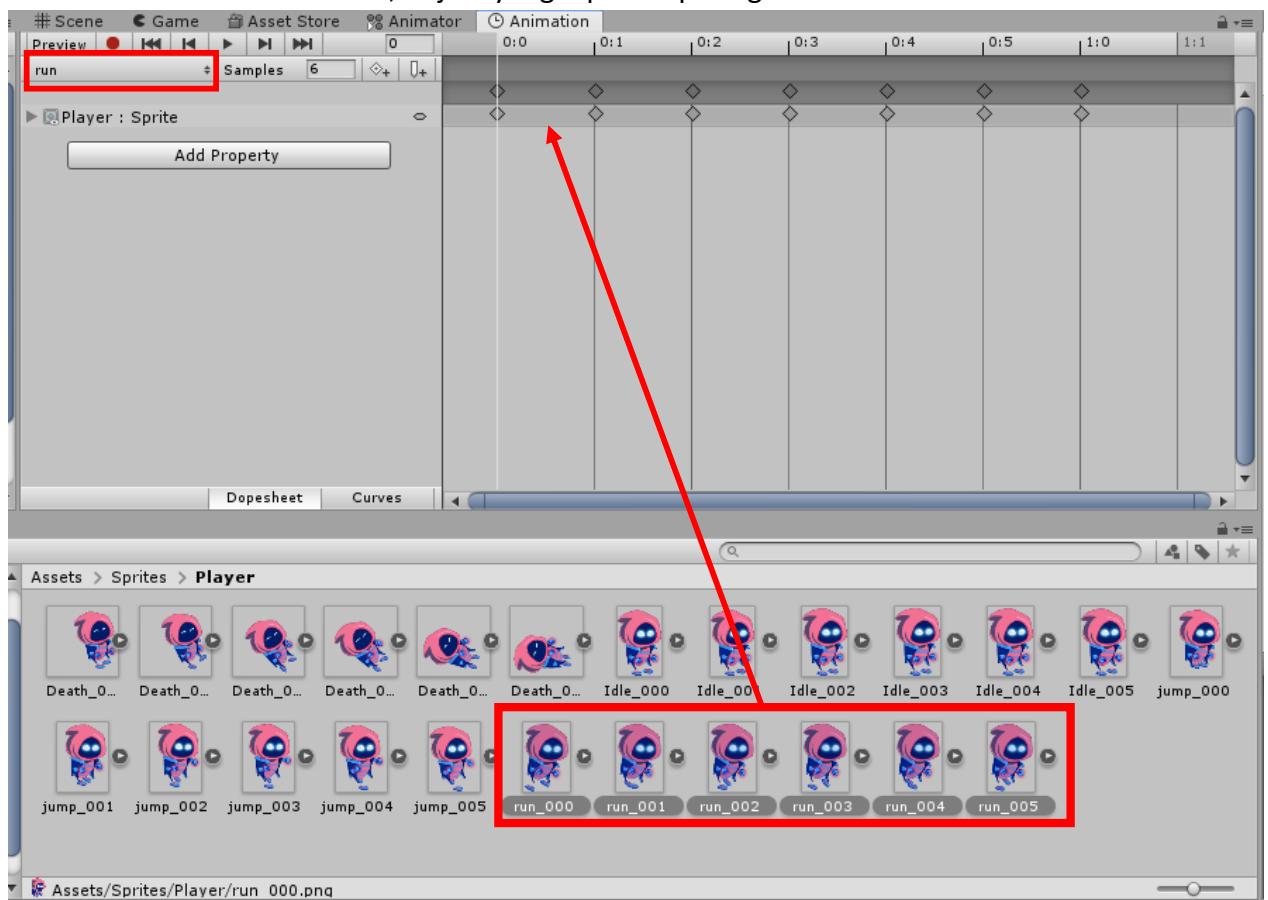
6. Membuat Animasi ketika **Player Jump** dan **Run**.

Langkahnya sama seperti membuat animasi **Idle**, pilih **Create New Clip**.

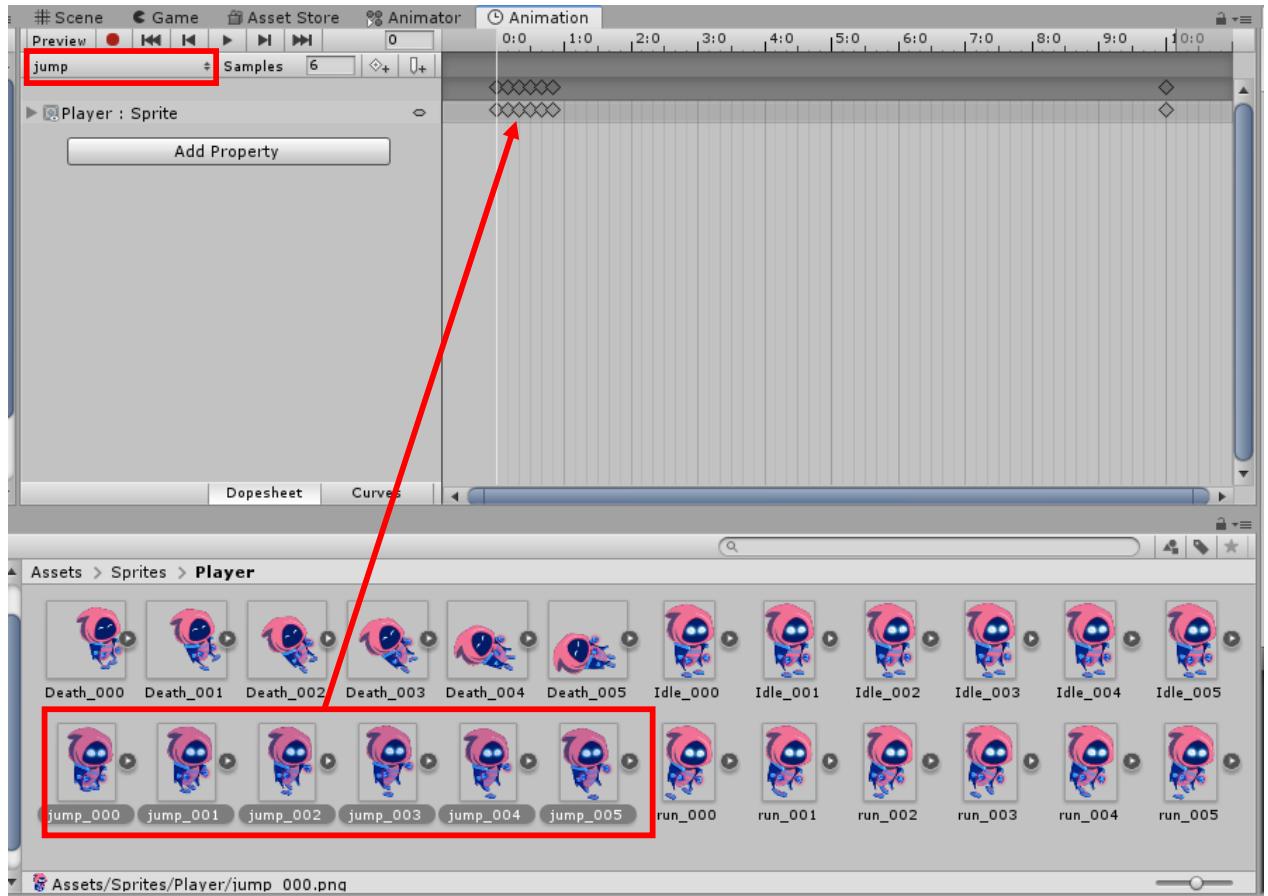


7. Ulangi langkah 3 sampai 5 untuk membuat animasi pada **Player**. Untuk Object yang di **DRAG** dan **DROP** pada langkah nomor 5 sesuai nama animasi yang dipilih, object terdapat pada **Assets folder Sprites > Player**.

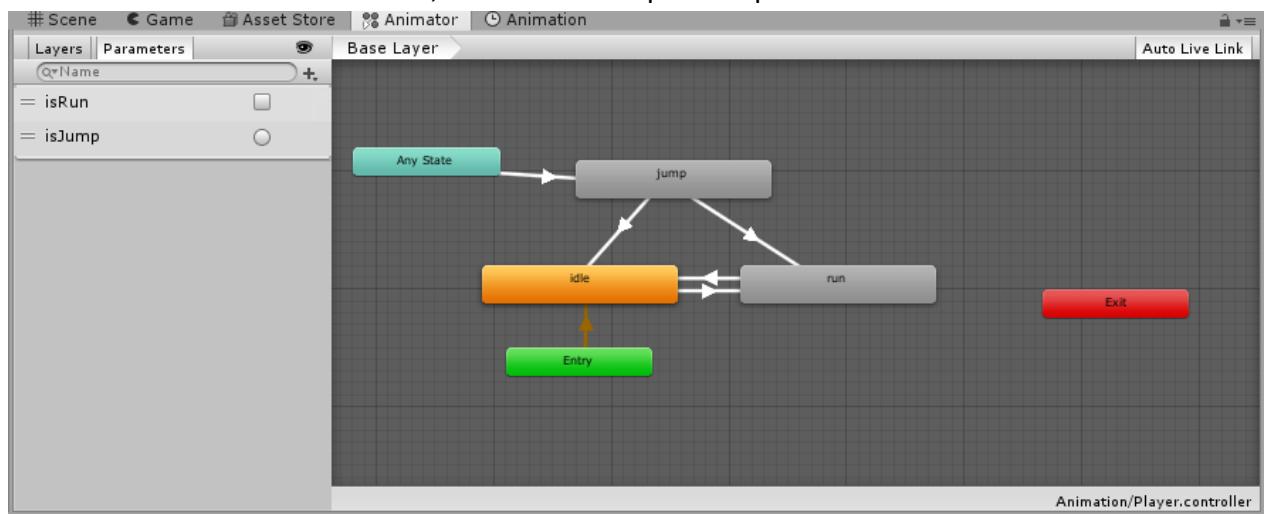
Ketika membuat animasi **Run**, object yang dipilih seperti gambar dibawah ini :



Ketika membuat animasi **Jump**, object yang dipilih seperti gambar dibawah ini :

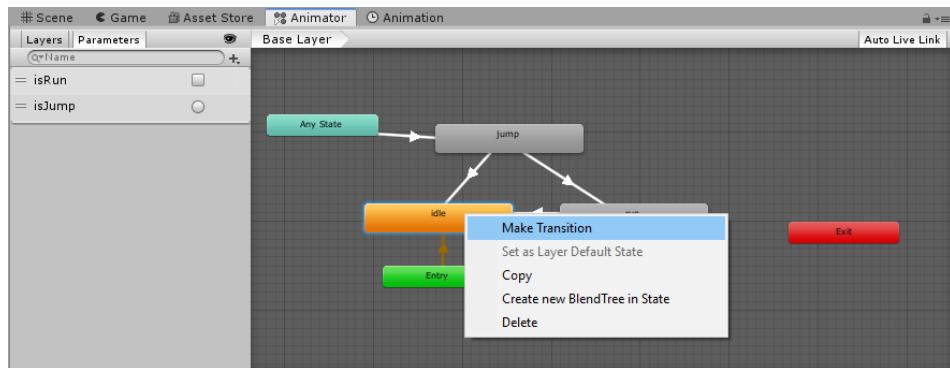


8. Setelah membuat animasi diatas, akan terlihat seperti ini pada **Tab Animator** :

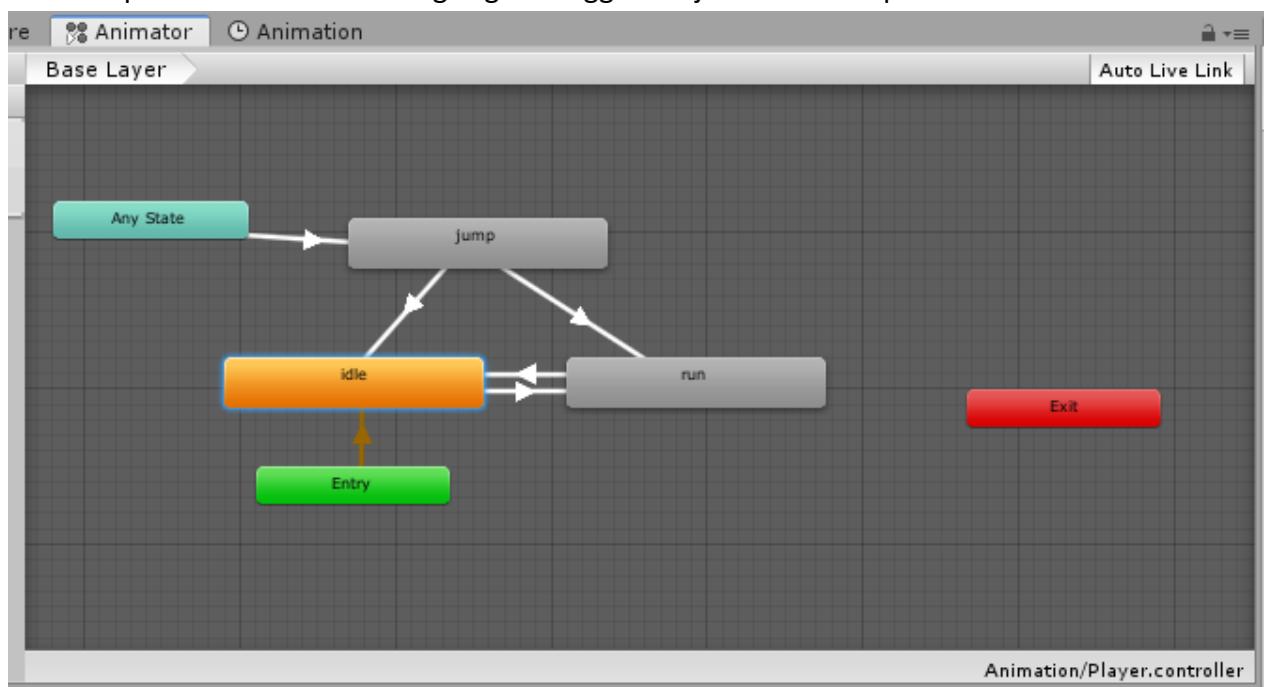


Sudah terdapat animasi **Jump** dan **Run**.

9. Membuat Transisi untuk animasi dengan cara pilih **Tab Animator**. Lalu pada **Idle** klik kanan pilih **Make Transition**.

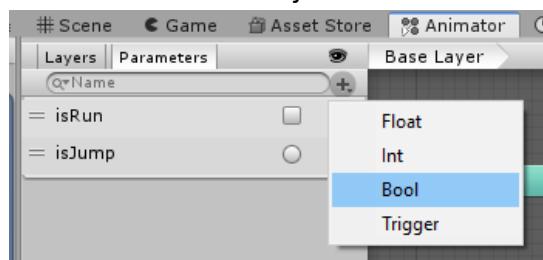


Arahkan panah ke **Run**. Dan ulangi lagi sehingga menjadi Transisi seperti di bawah ini:

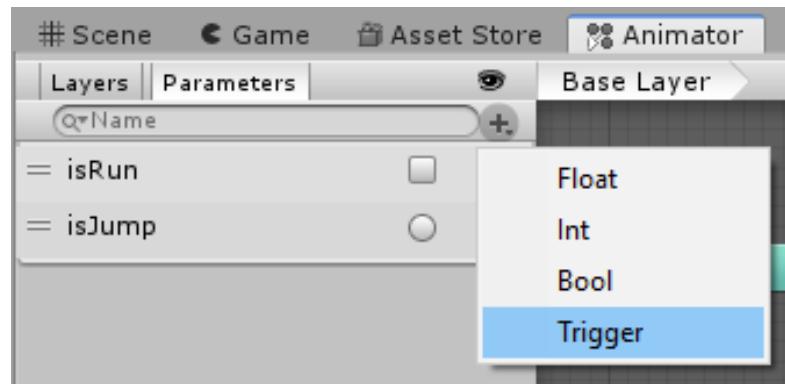


10. Langkah selanjutnya menambahkan parameters untuk mengkondisikan kapan animasi akan bergerak **Idle**, **Run** dan **Jump**.

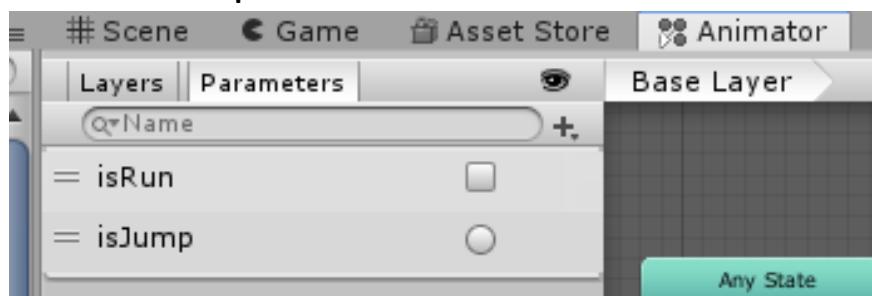
11. Pilih **Tab Animator**, disana terdapat **Layers** dan **Parameters**. Pilih Parameter dan **klik tanda +** di ujung kanan lalu pilih tipe **Bool**. Maka akan muncul Parameters baru dengan nama **NewBool** dan ubah nama tersebut menjadi **isRun**.



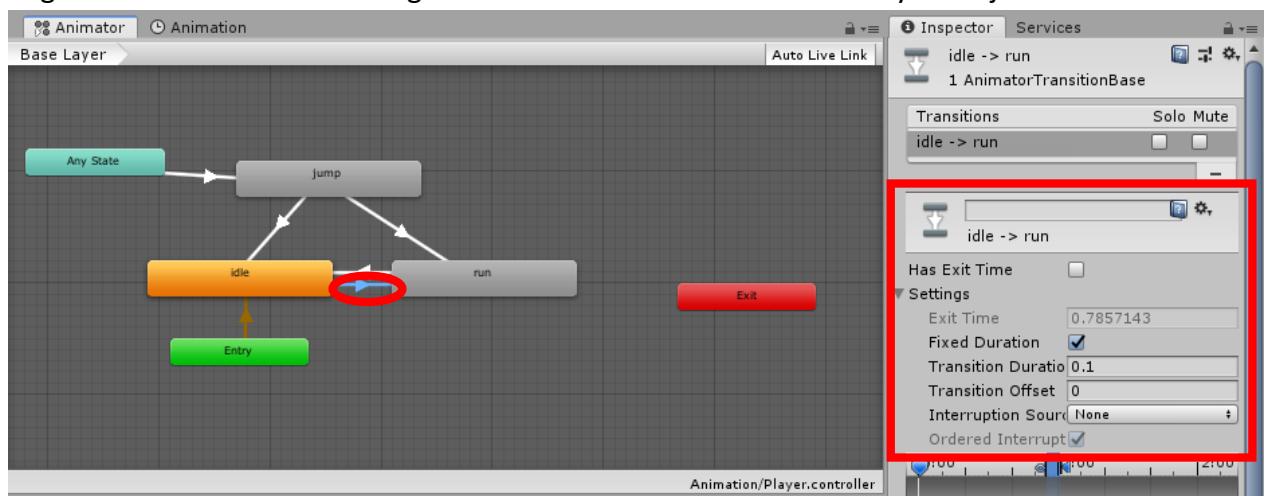
Tambahkan satu **Parameters** dengan tipe **Trigger** dan ubah nama dengan **isJump**.



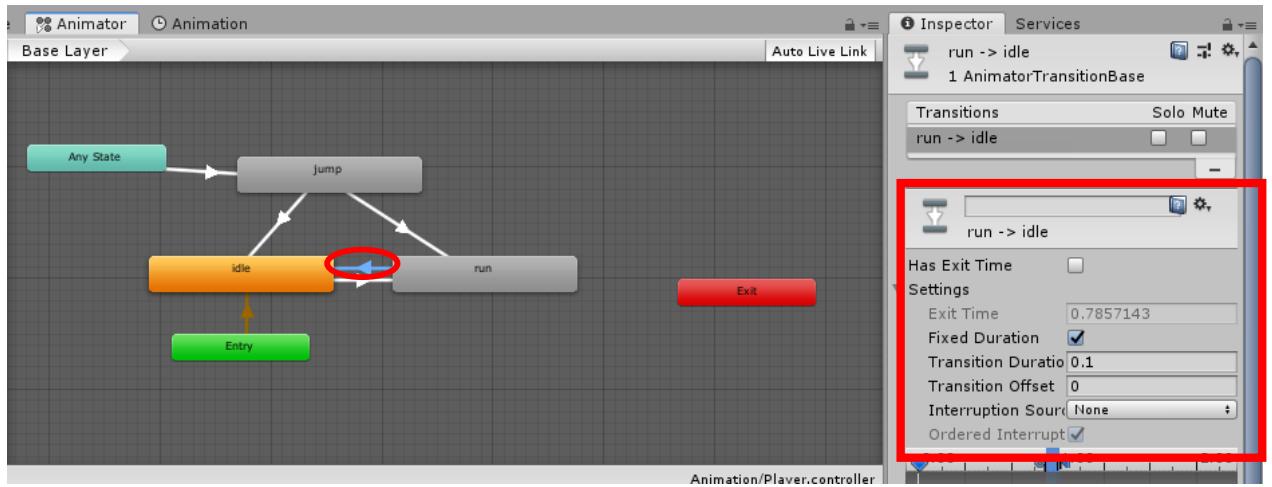
Parameters **isRun** dan **isJump** sudah terbuat.



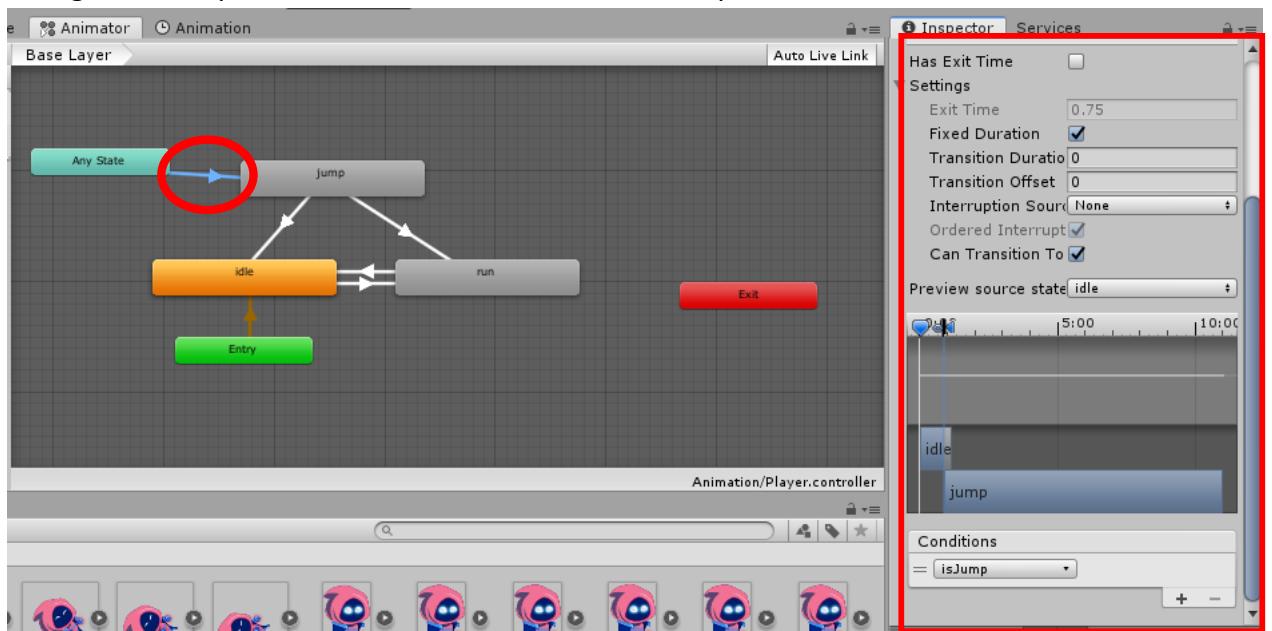
16. Selanjutnya seleksi arah panah yang mengarah dari state idle ke state run dan setting **Conditions** menjadi **isRun** dan valuenya menjadi true, lalu hilangkan tanda ceklis di bagian **Has Exit Time** dan di bagian **Transition Duration** ubah valuenya menjadi **0.1**.



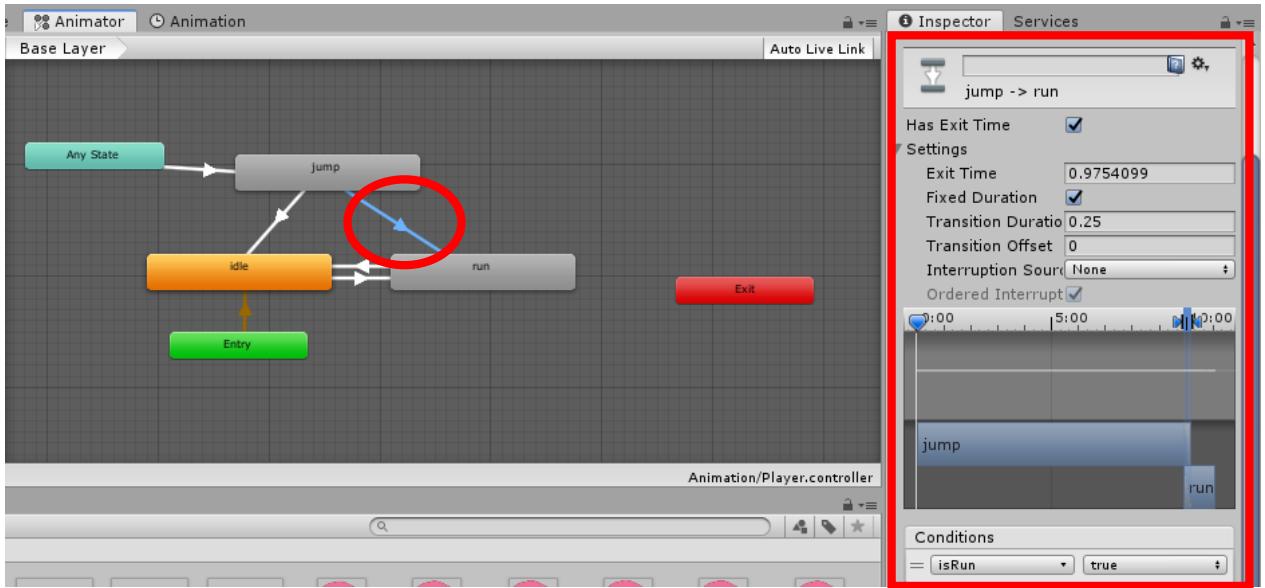
17. Ulangi lagi langkah di atas untuk **arah panah** yang mengarah dari state run ke state idle, tapi untuk bagian **Conditions** kita ubah menjadi **isRun** dan valuenya kita beri nilai false, jangan lupa hilangkan ceklis di **Has Exit Time** dan di bagian Transition Durationnya ubah nilanya menjadi **0.1**.



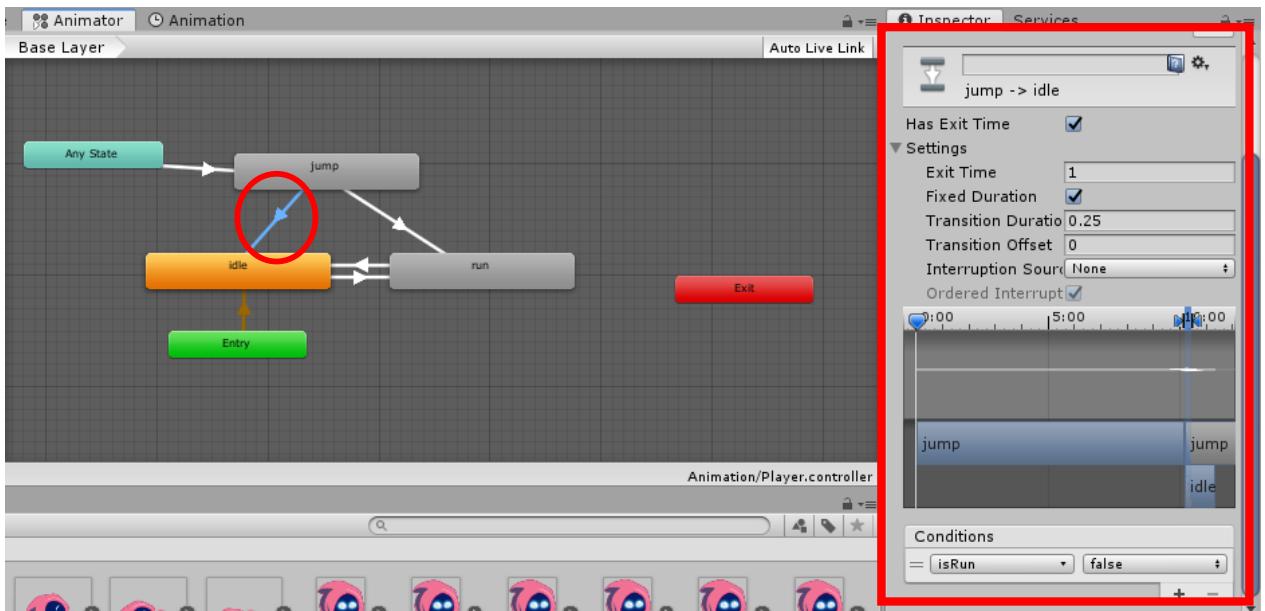
18. Lanjutkan lagi setting transisi saat **jump**, seleksi terlebih dulu arah panah dari **Any State** ke arah state **jump** dan lanjutkan dengan mengubah **Conditions** menjadi **isJump**, hilangkan ceklis pada **Has Exit Time** serta beri nilai **nol** pada **Transition Duration**.



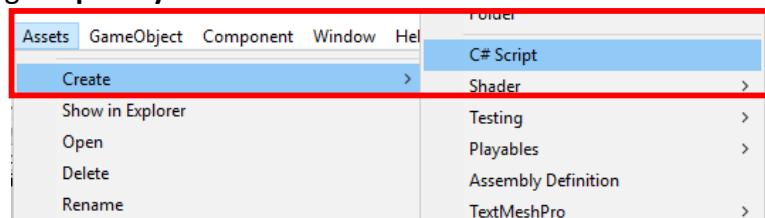
19. Atur transisi dari **jump** ke **run** dan setting seperti gambar berikut, langkahnya sama seperti yang sebelumnya yang perlu sobat perhatikan adalah pada **Conditions, Has Exit Time, Exit Time** dan **Transition Duration**.



20. Atur lagi transisi dari **jump** ke **idle** dan setting seperti gambar berikut :



21. Membuat **Script PlayerController**, sebelumnya pilih **Assets > Script** untuk menyimpan secara langsung **Script PlayerController**.



```

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

public class PlayerController : MonoBehaviour
{
    // untuk mengatur kecepatan saat Player bergerak
    [SerializeField] private float speed;
    // untuk komponen Rigidbody2D
    private Rigidbody2D rigidBody;
    // Untuk menyimpan nilai yang mengkondisikan
    // Player saat bergerak ke kanan atau ke kiri
    private float moveInput;
    // Untuk mengkondisikan benar saat Player menghadap ke kanan
    private bool facingRight;

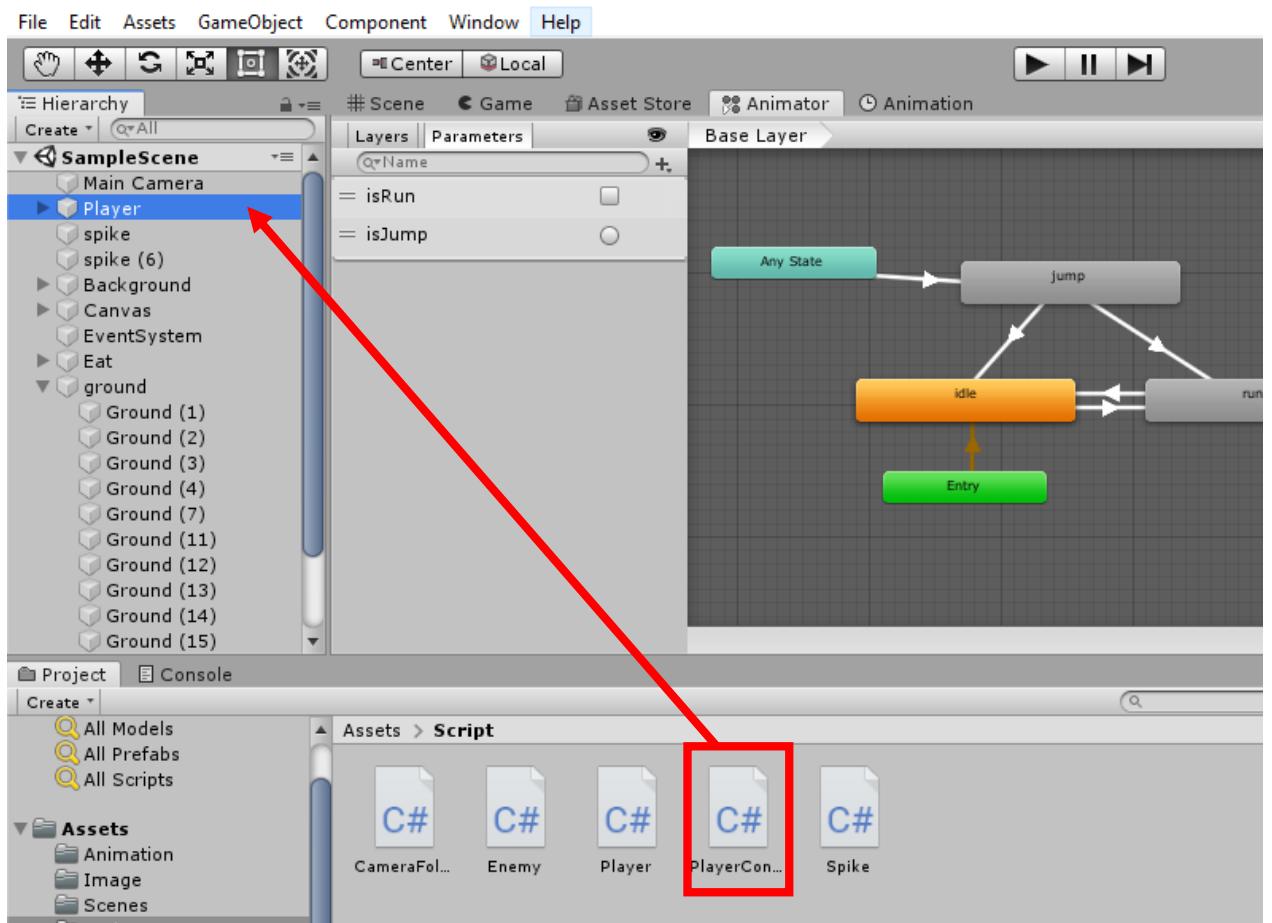
    // memberikan nilai seberapa tinggi Player dapat melompat
    [SerializeField] private float jumpForce;
    // menandakan benar jika Player menyentuh pinjakan atau ground
    [SerializeField] private bool isGrounded;
    // memastikan bahwa posisi kaki Player berada di bawah,
    // seperti telapak kaki gitu loh sobat
    [SerializeField] private Transform feetPos;
    // ini digunakan untuk mengatur seberapa besar radius kaki Player sobat
    // "Kurang lebih seperti itu :)"
    [SerializeField] private float circleRadius;
    // Ini digunakan untuk memastikan object
    // yang bertindak / kita jadikan sebagai ground
    [SerializeField] private LayerMask whatIsGround;

    // variabel ini kita panggil
    // untuk menjalankan animasi idle, run, dan jump
    private Animator anim;

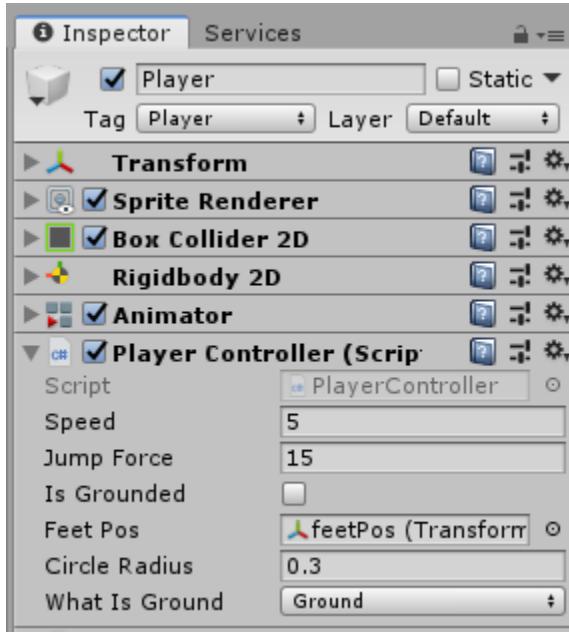
    private void Start()
    {
    }
}

```

## 22. Lalu DRAG and DROP Script pada GameObject Player.

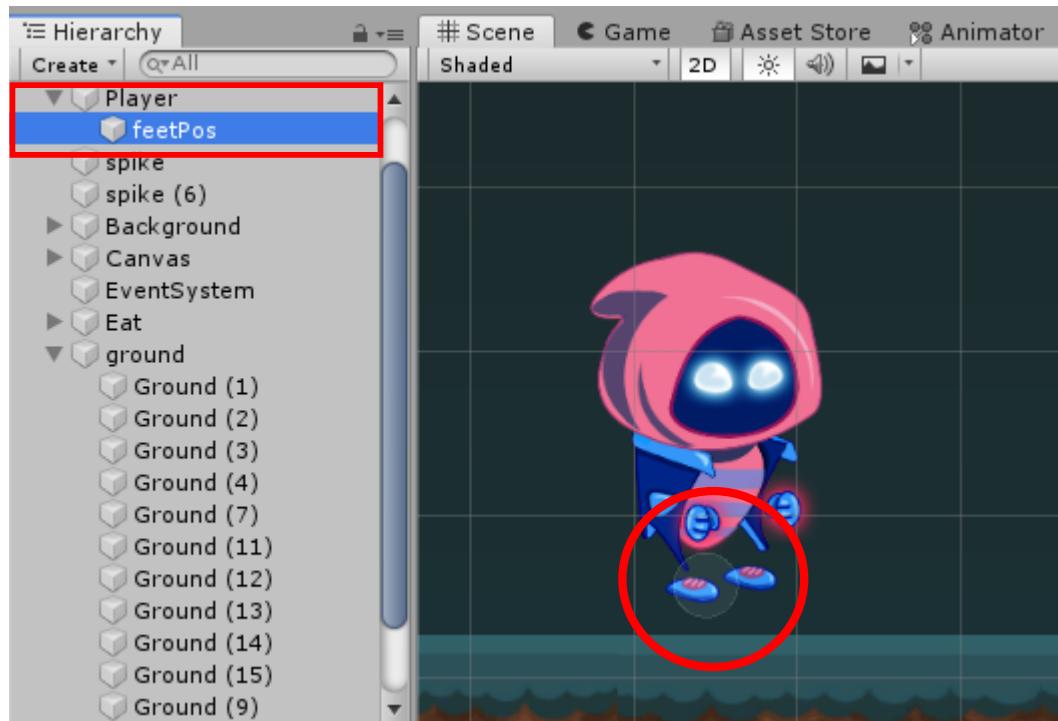


Script telah ditambah pada GameObject **Player**.



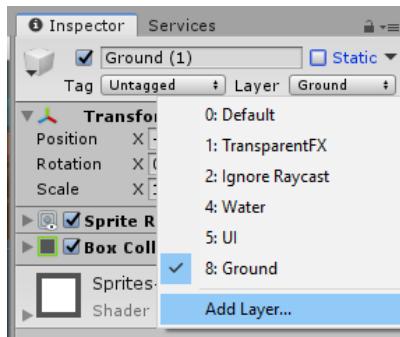
23. Setelah itu membuat objek empty untuk feetPos, ubah namanya ke jadi **feetPos**.

Posisikan feetPos ini di sekitar telapak kaki Player dan juga jadikan feetPos ini sebagai child dari objek **Player**.



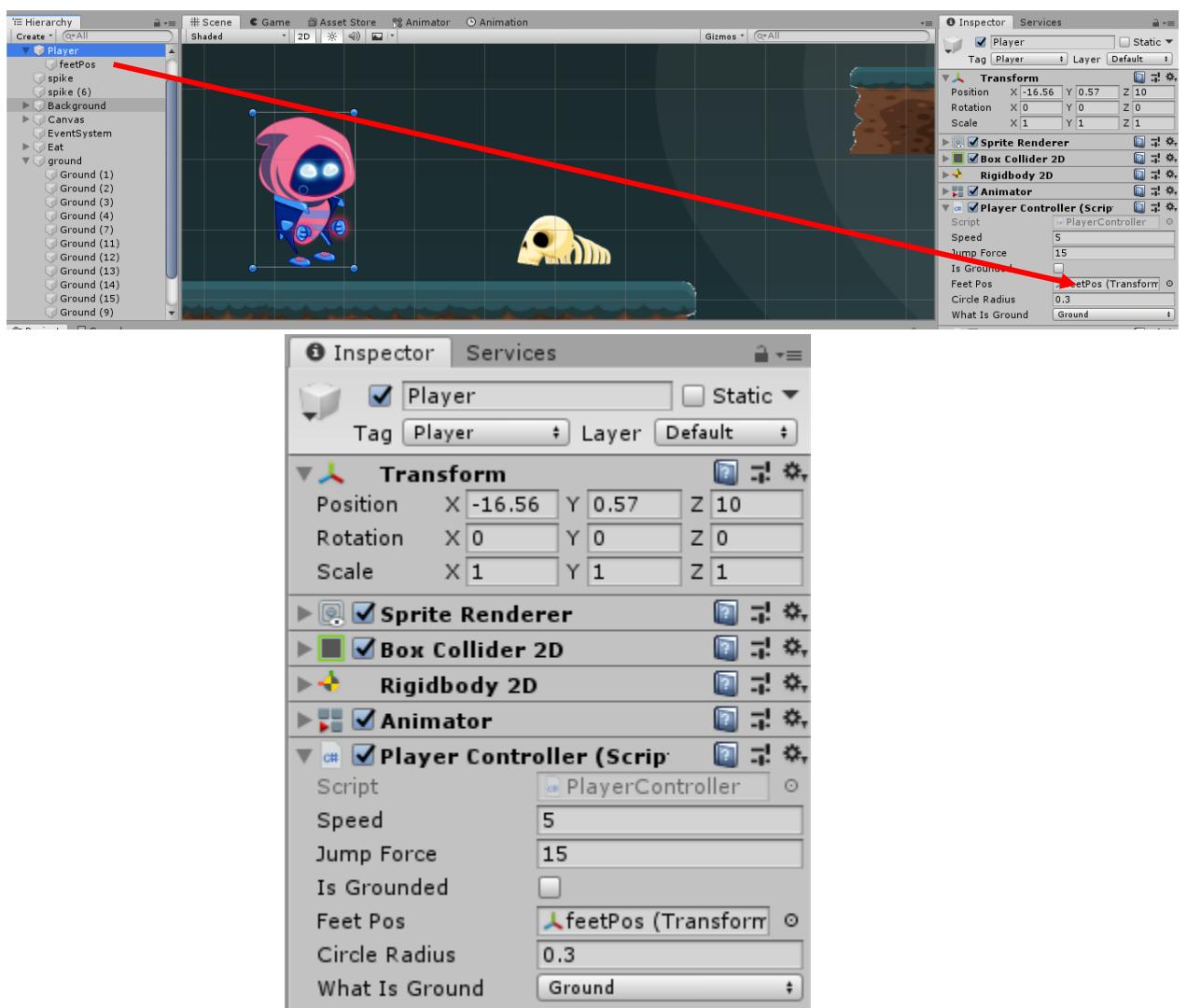
24. Setting **GameObject ground** agar object Player dapat bergerak pada ground.

Tambahkan Layer pada **GameObject ground**, dengan cara **Add Layers** lalu buat layer **Ground** setelah itu ganti **Default** dengan **Ground**.



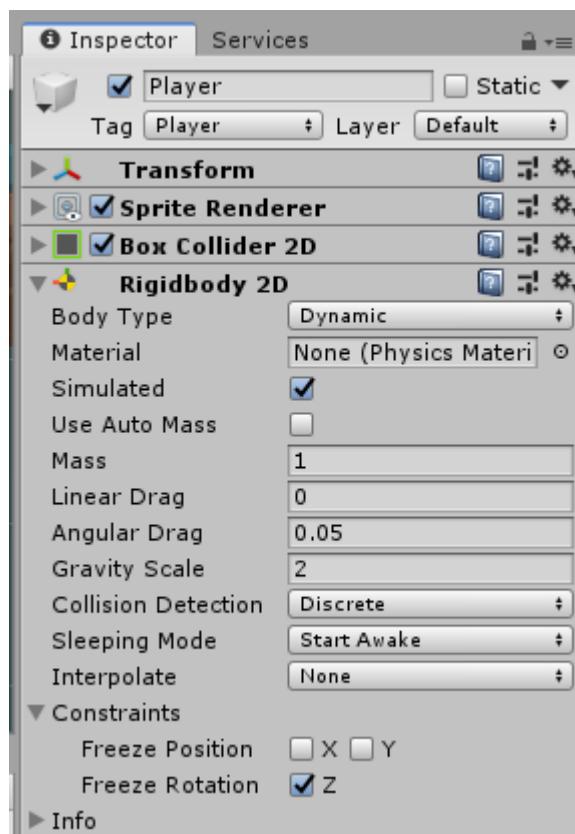
25. Pilih **GameObject Player** dan DRAG **feetPos**, ganti **Speed** menjadi 5, **Jump Force** menjadi 15.

15.



26. Coba kita **Play**, ketika menekan tombol **A** atau arah panah ke kiri maka Player akan bergerak ke kiri.

27. Pada komponen **Rigidbody 2D** ubah nilai **Gravity Scale** menjadi **2** dan ceklis **Freeze Rotation** pada sumbu z agar Player tidak berputar – putar saat di atas pijakan yang tidak seimbang.



### Langkah 3 : Membuat Player Berjalan Mengikuti Arah Kamera

1. Membuat Script CameraFollow, di simpan pada Assets folder Script.

```

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

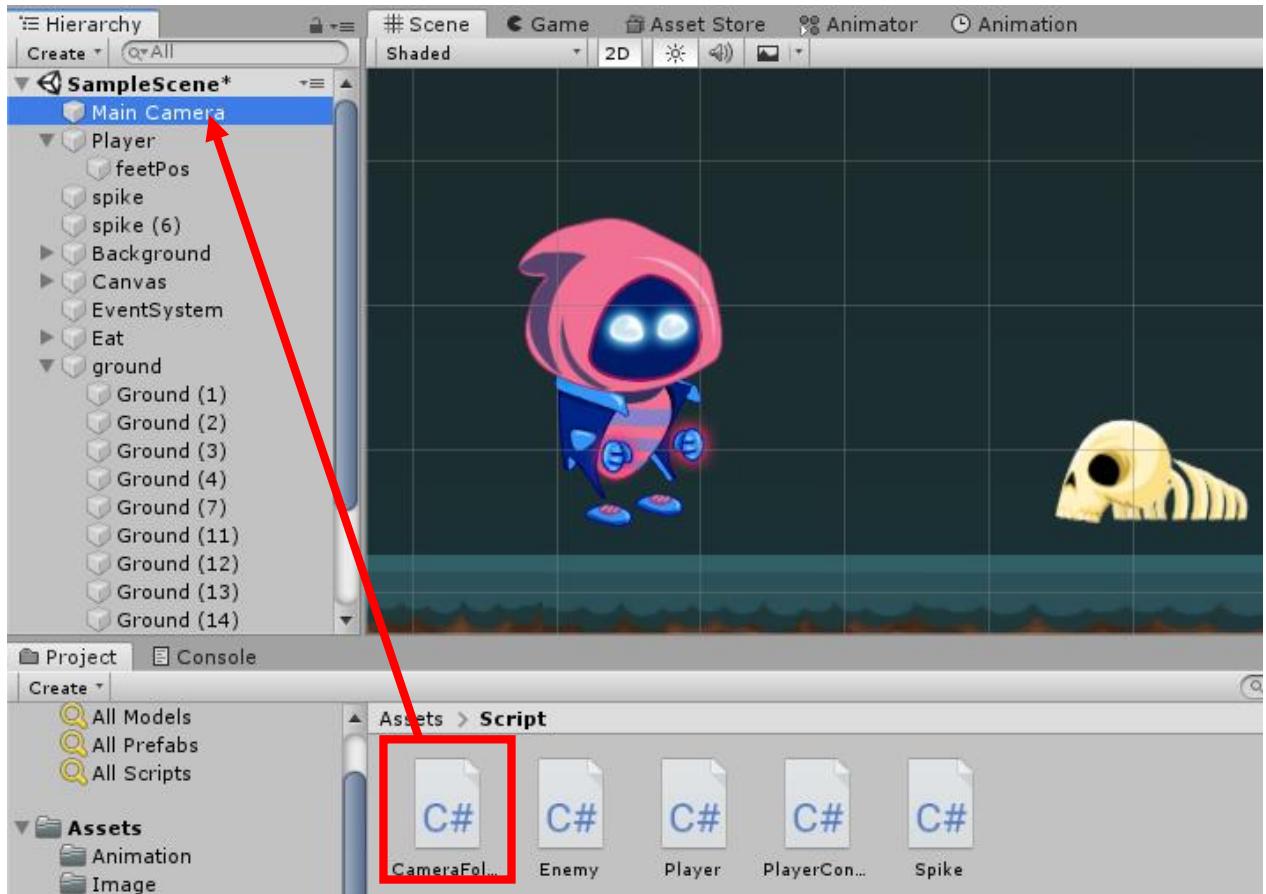
public class CameraFollow : MonoBehaviour
{
    //membuat variabel yang menjadi target
    //untuk diikuti oleh Camera
    [SerializeField] private Transform target;

    private void Update()
    {
        //this.transform.position maksutnya adalah Camera
        //terus setiap ada update maka posisi Camera persisnya
        //pada sumbu x akan bertambah atau berkurang sesuai dengan
        //sumbu x yang ada pada Player
        this.transform.position = new Vector3(target.transform.position.x, target.transform.position.y, target.transform.position.z - 10f);
    }
}

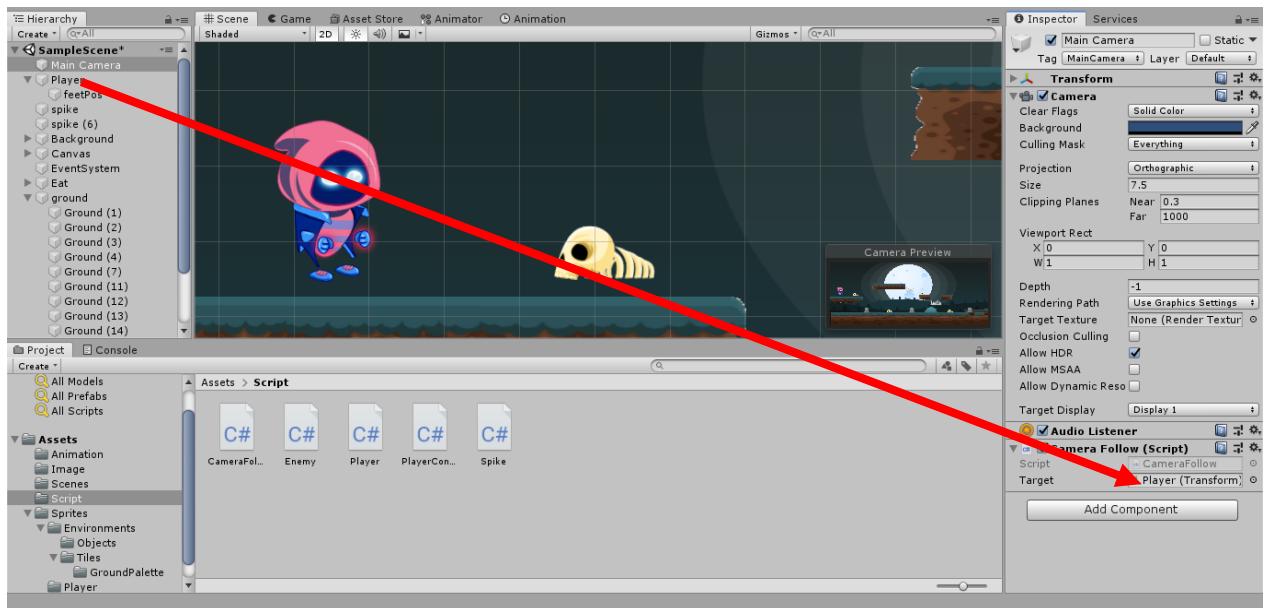
```

The screenshot shows Microsoft Visual Studio with the CameraFollow.cs script open. The script is a simple MonoBehaviour that updates its position to follow a target transform along the X-axis, while keeping its Y and Z positions constant.

2. DRAG dan DROP Script tersebut pada MainCamera.

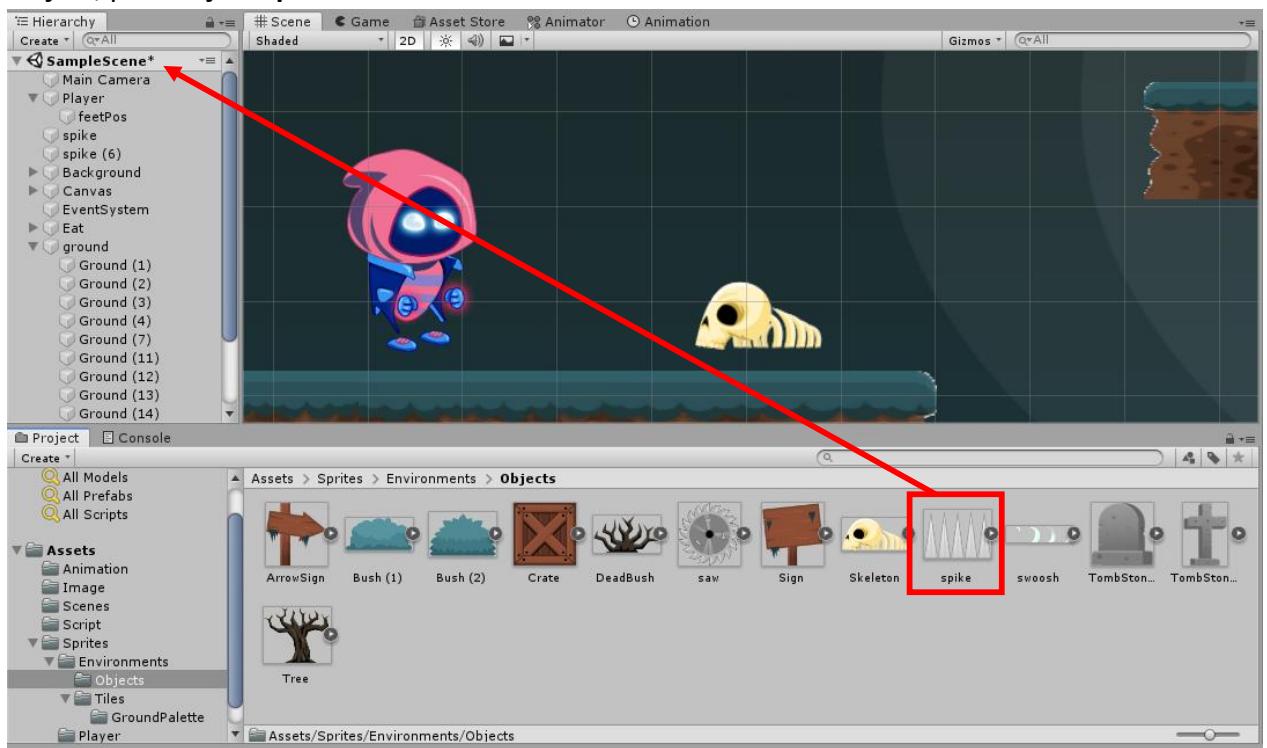


3. Kemudian DRAG and DROP GameObjet Player pada Target.

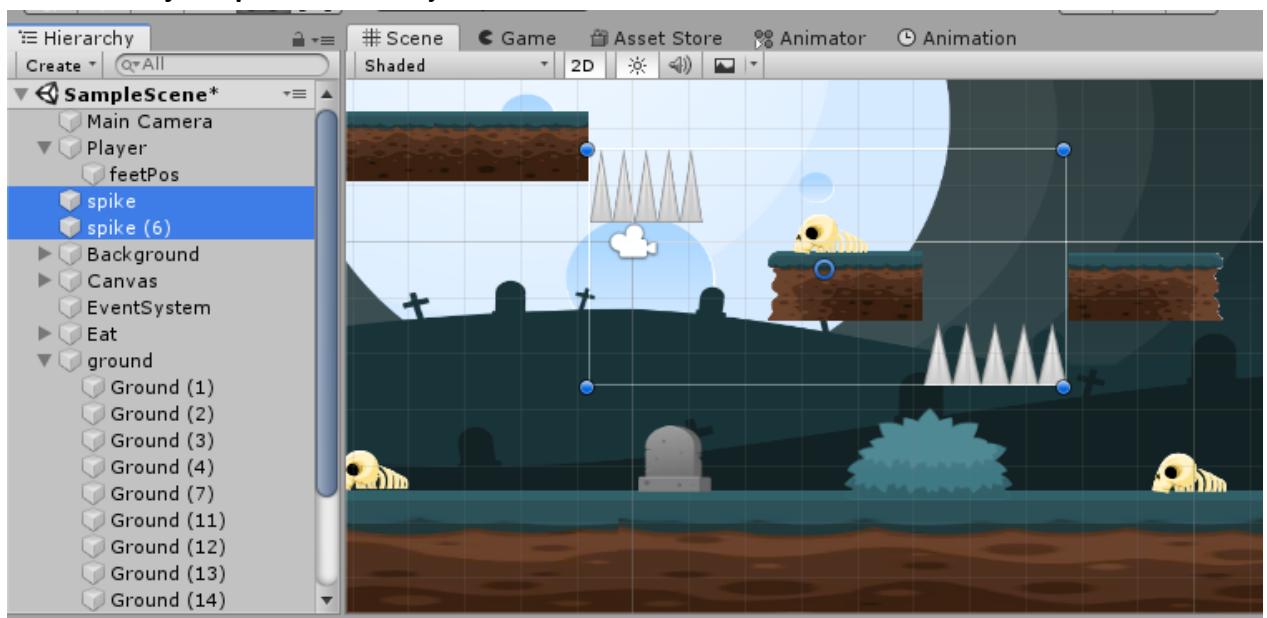


#### Langkah 4 : Membuat Jebakan atau Trap

- Pertama DRAG dan DROP object yang ada di Assets folder Sprites > Enviroment > Object, pilih object spike.



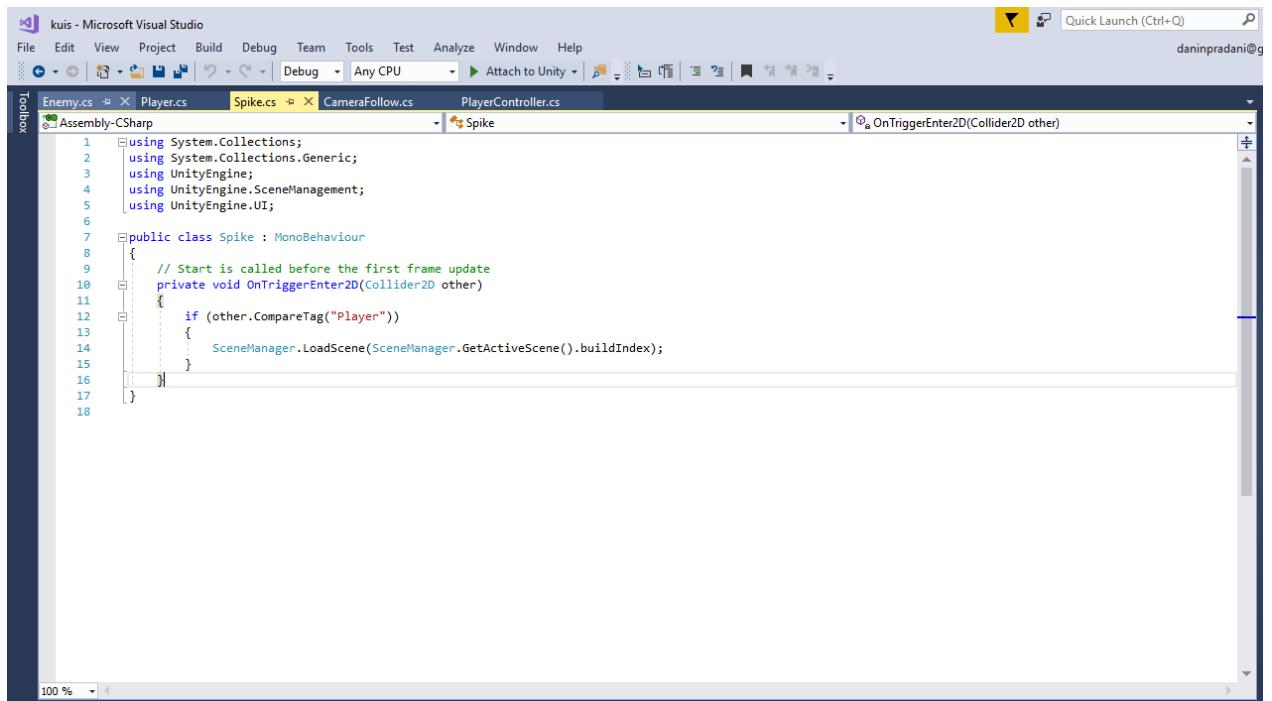
- Letakkan object spike dimana saja.



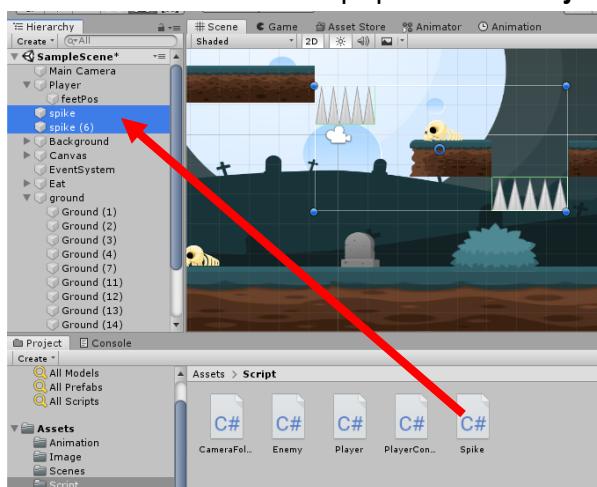
3. Tambahkan komponen Box Collider 2D pada GameObject spike, menceklis pilihan Is Trigger. Dengan cara Add Component dan pilih Box Collider 2D.



4. Membuat Script untuk GameObject spike dengan nama Spike.



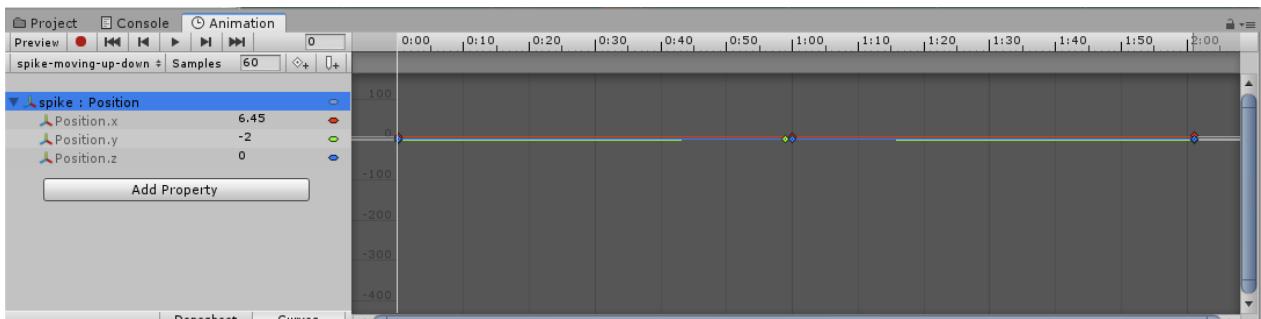
5. Lalu DRAG and DROP Script pada GameObject spike.



6. Duplicate **object spike** menjadi 2 dengan cara **CTRL+D**.
7. Tambah animasi ke atas dan ke bawah untuk **object spike**. Pada Tab **Animation** buat clip dengan nama **spike-moving-up-down** dengan cara **Create New Clip**. Lalu **Add Property > Transform > Position** pilih **+**.



8. Lalu masih pada halaman yang sama **klik Curves**, tambahkan 1 Key Frame lagi pada 2:00 dengan cara pilih Key Frame ke 2:00 lalu pilih gambar **Diamond (Add Key Frame)** dan ubah **nilai y menjadi -2** pada Key Frame ke 0:00 dan 2:00, ketika mengubah nilai y menyalakan **tombol record** yang ada di dekat **Preview**.

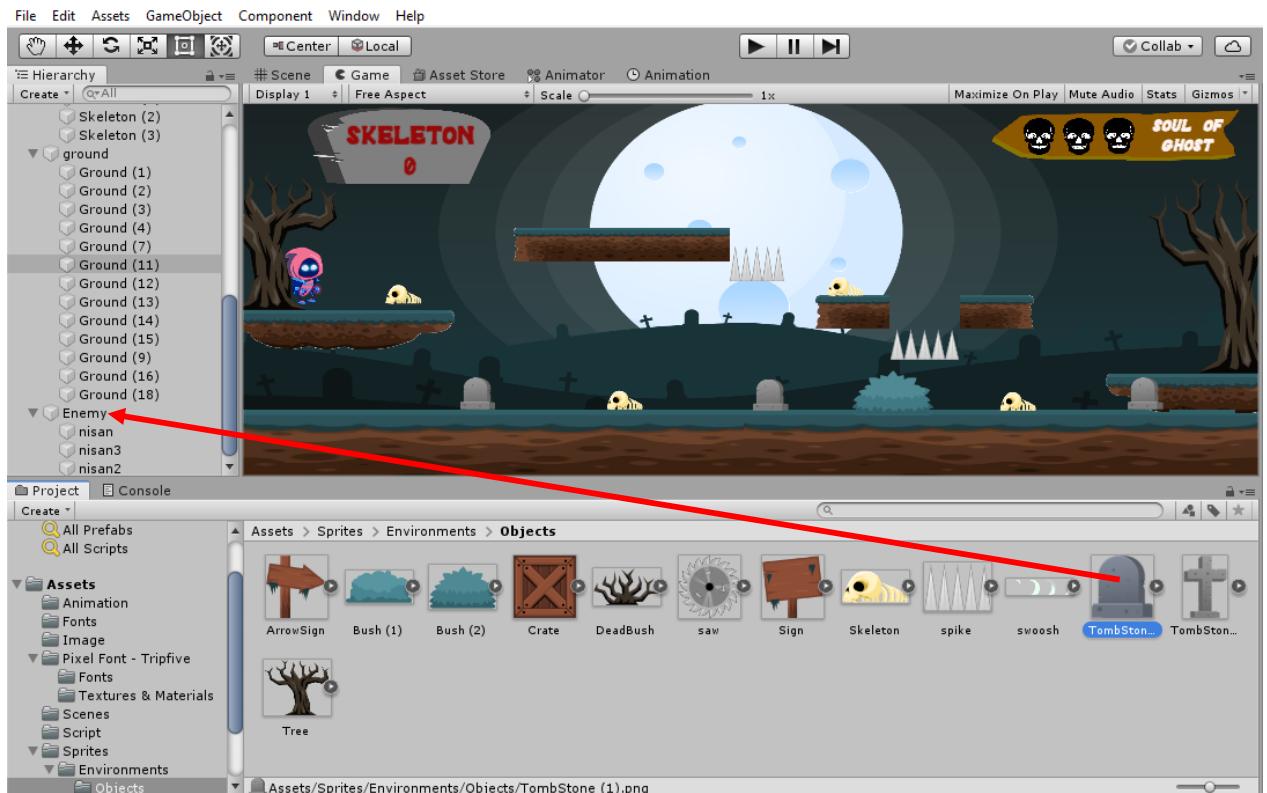


9. Untuk animasi **object spike(6)** caranya sama dengan **object spike**, hanya posisi yang beda.

## Langkah 5 : Membuat Enemy untuk Player

1. Membuat **GameObject Empty** dan ubah nama menjadi **Enemy**.

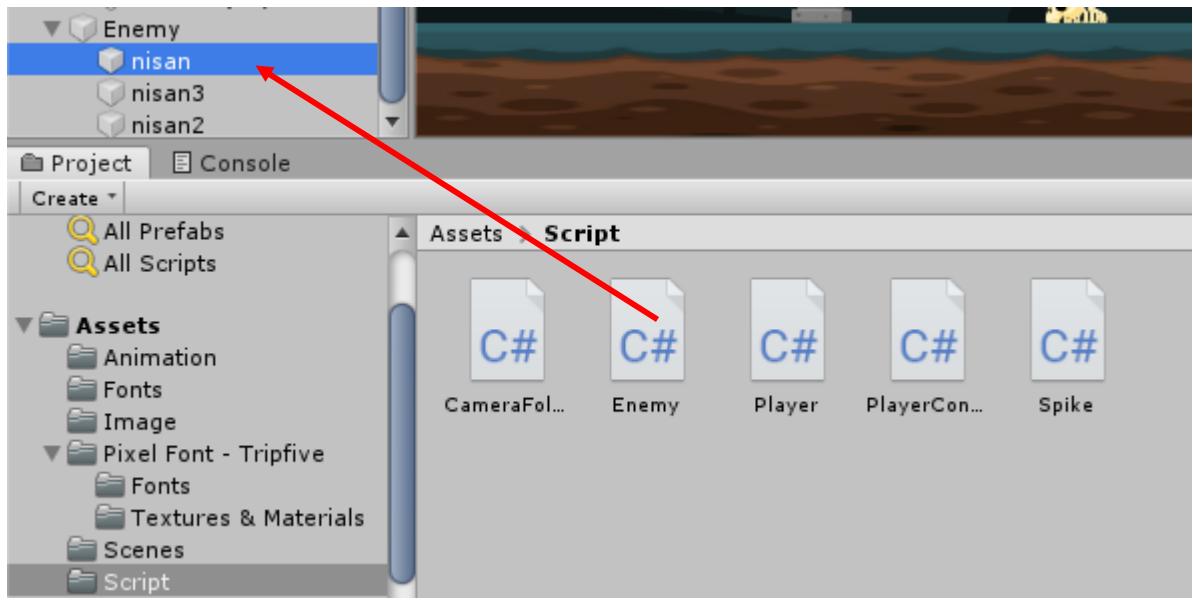
Lalu **DRAG** dan **DROP** object **Enemy** pada **Assets folder Sprites > Enviroment > Object**



2. Membuat Script **Enemy** untuk **Enemy**. Lalu **DRAG** dan **DROP** Script tersebut ke **object nisan**.

```
kuis - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug - Any CPU Attach to Unity
Assembly-Sharp
Enemy.cs
using UnityEngine;
using System.Collections;
using UnityEngine.UI;
public class Enemy : MonoBehaviour
{
    public Image nyawaImage;
    public Sprite iconSkel;
    public Sprite iconCrate;
    private bool carryingSkel = true;
    void OnTriggerEnter2D(Collider2D hit)
    {
        if (hit.CompareTag("Player"))
        {
            carryingSkel = true;
            UpdateSkelImage();
        }
    }
    private void UpdateSkelImage()
    {
        if (carryingSkel)
        {
            nyawaImage.sprite = iconSkel;
        }
        else
        {
            nyawaImage.sprite = iconCrate;
        }
    }
}
```

The screenshot shows Microsoft Visual Studio with the 'Enemy.cs' script open. The script defines a class 'Enemy' that inherits from 'MonoBehaviour'. It has three public fields: 'nyawaImage' (Image type), 'iconSkel' (Sprite type), and 'iconCrate' (Sprite type). A private field 'carryingSkel' is set to true. The 'OnTriggerEnter2D' method checks if the colliding object has the tag 'Player'. If it does, it sets 'carryingSkel' to true and calls the 'UpdateSkelImage' method. The 'UpdateSkelImage' method checks the value of 'carryingSkel' and sets the 'nyawaImage' sprite accordingly, either to 'iconSkel' or 'iconCrate'.



3. Pada Inspector object nisan, DRAG and DROP Image-nyawa pada Script Enemy di Nyawa Image, lalu teng di Icon Crate dan Crate di Icon Skel.



Untuk Image-nyawa duplicate sebanyak 3x untuk menentukan nyawa dari Ghost.

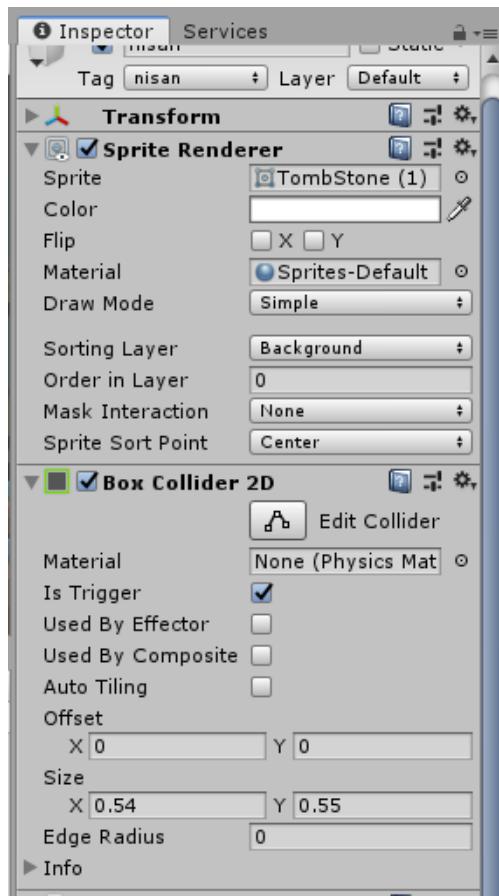


Tampilan nyawa pada Ghost. Jika nyawa berkurang akan berubah menjadi seperti ini :

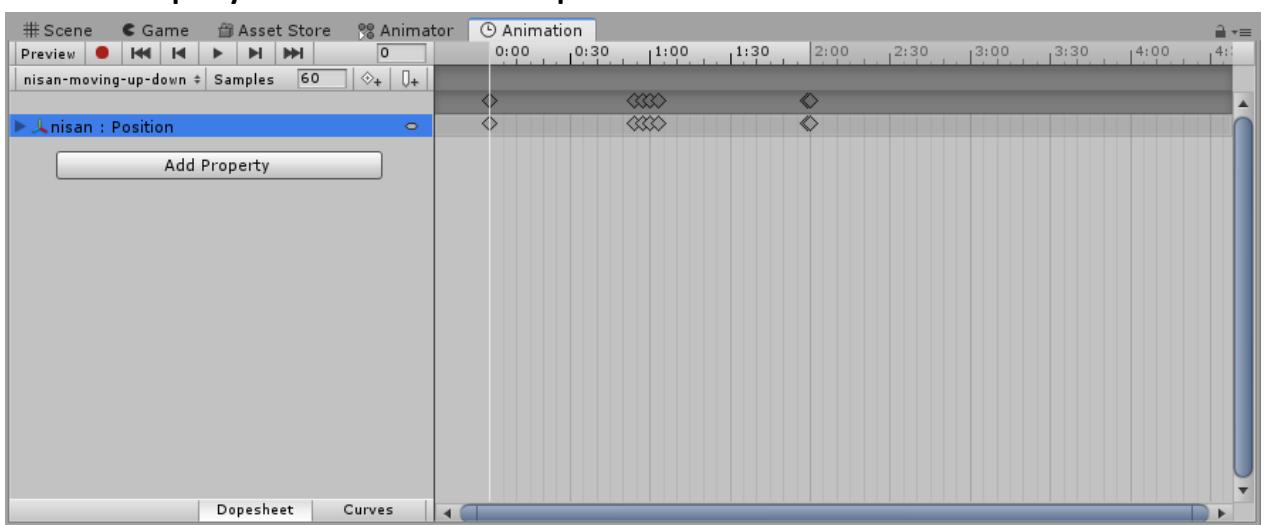


4. Duplicate object nisan menjadi 3 dengan cara **CTRL+D**.

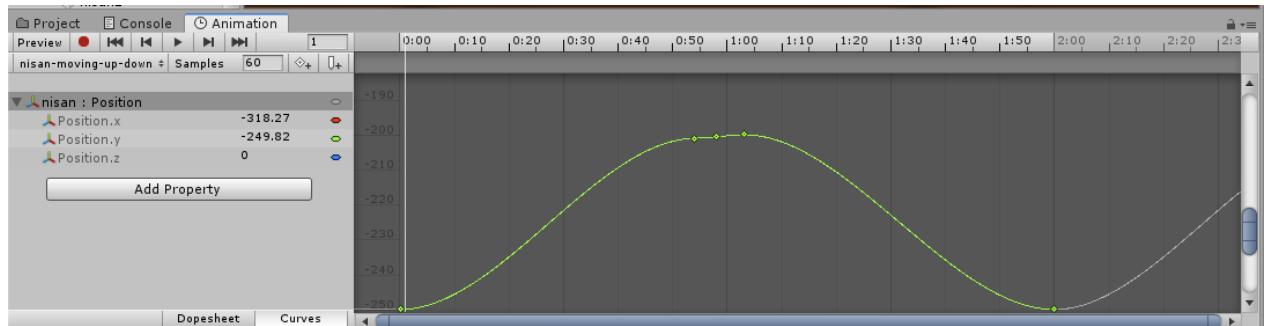
5. **Object nisan** ubah **Inspector** seperti dibawah ini dan juga menambahkan **Box Collinder 2D**.



6. Tambah animasi ke atas dan ke bawah untuk **object nisan**. Pada Tab Animation buat clip dengan nama **nisan-moving-up-down** dengan cara **Create New Clip**. Lalu **Add Property > Transform > Position** pilih **+**.



7. Lalu masih pada halaman yang sama **klik Curves**, tambahkan 1 Key Frame lagi pada 2:00 dengan cara pilih Key Frame ke 2:00 lalu pilih gambar **Diamond (Add Key Frame)** dan ubah **nilai y menjadi -249** pada Key Frame ke 0:00 dan 2:00, ketika mengubah nilai y menyalakan **tombol record** yang ada di dekat **Preview**.

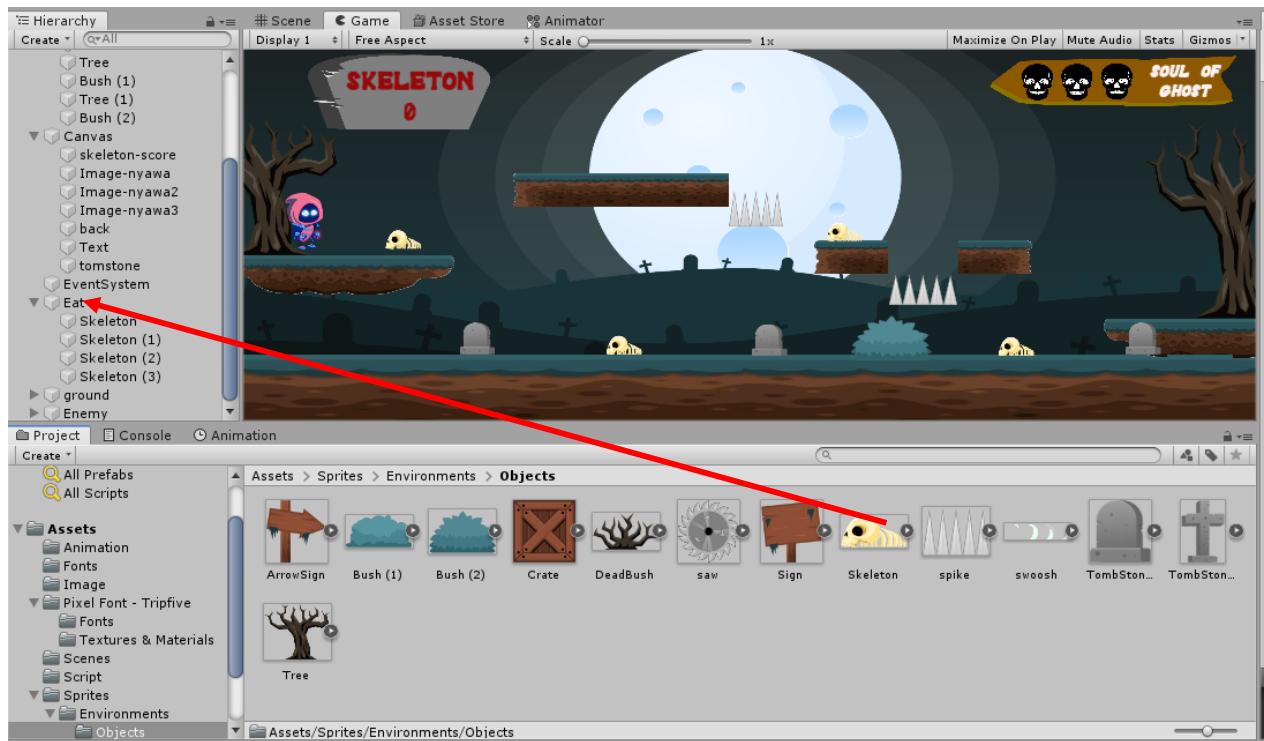


8. Untuk **object nisan3** hanya diatur posisi saja tidak menggunakan animasi.
9. Untuk **object nisan2** langkahnya sama dengan **object nisan**. Hanya yang membedakan mengatur posisi object saja.

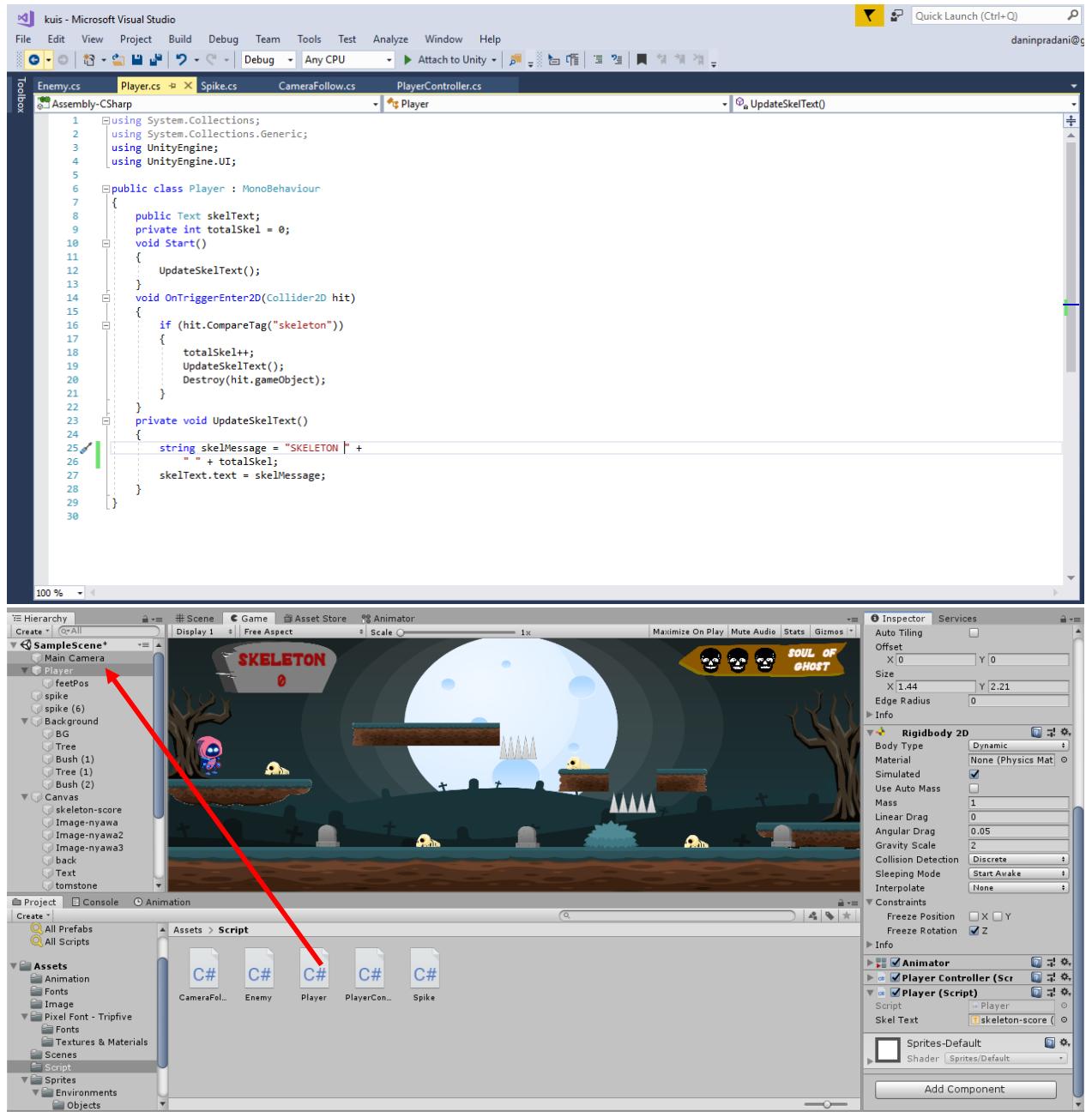
## Langkah 6 : Membuat Object Nilai/Score

1. Membuat **GameObject Empty** dan ubah nama menjadi **Eat**.

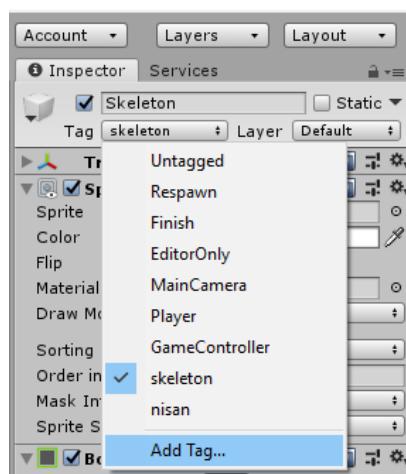
Lalu **DRAG** dan **DROP** object **Skeleton** pada **Assets folder Sprites > Enviroment > Object**



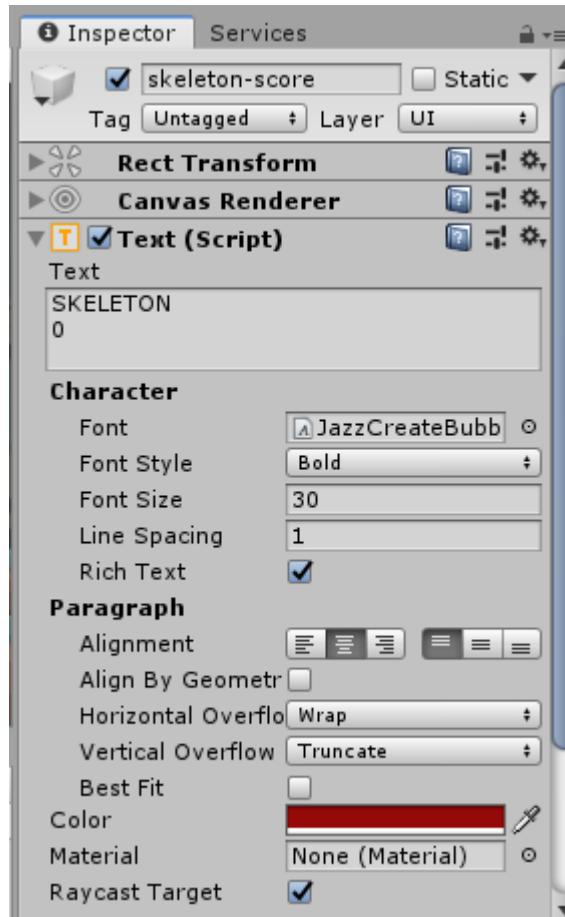
2. Membuat Script Player untuk Eat. Lalu DRAG and DROP Script tersebut ke object Player.



3. Pada object Skeleton, tambah Tag skeleton.



4. Tambahkan **UI Text** dan ubah nama dengan **skeleton-score**.



5. Lalu pada **GameObject Player DRAG** dan **DROP** **skeleton-score** ke dalam **Skel Text**.



6. Tampilan nilai/score seperti di bawah ini :



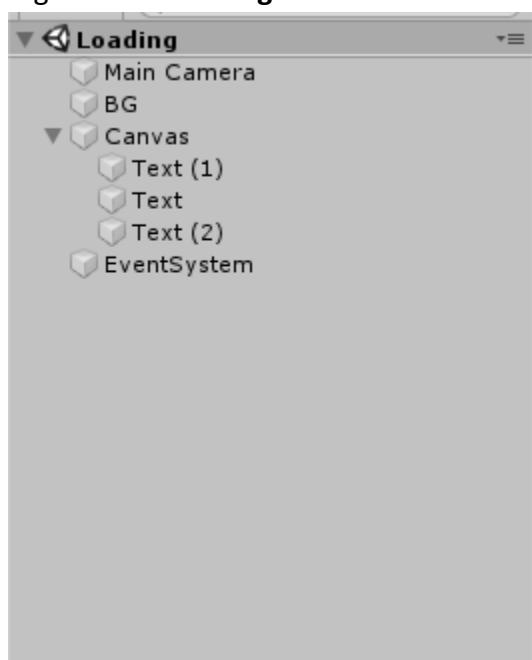


Ketika bertambah maka nilai akan **bertambah** dan **Skeleton** yang dimakan akan hilang.

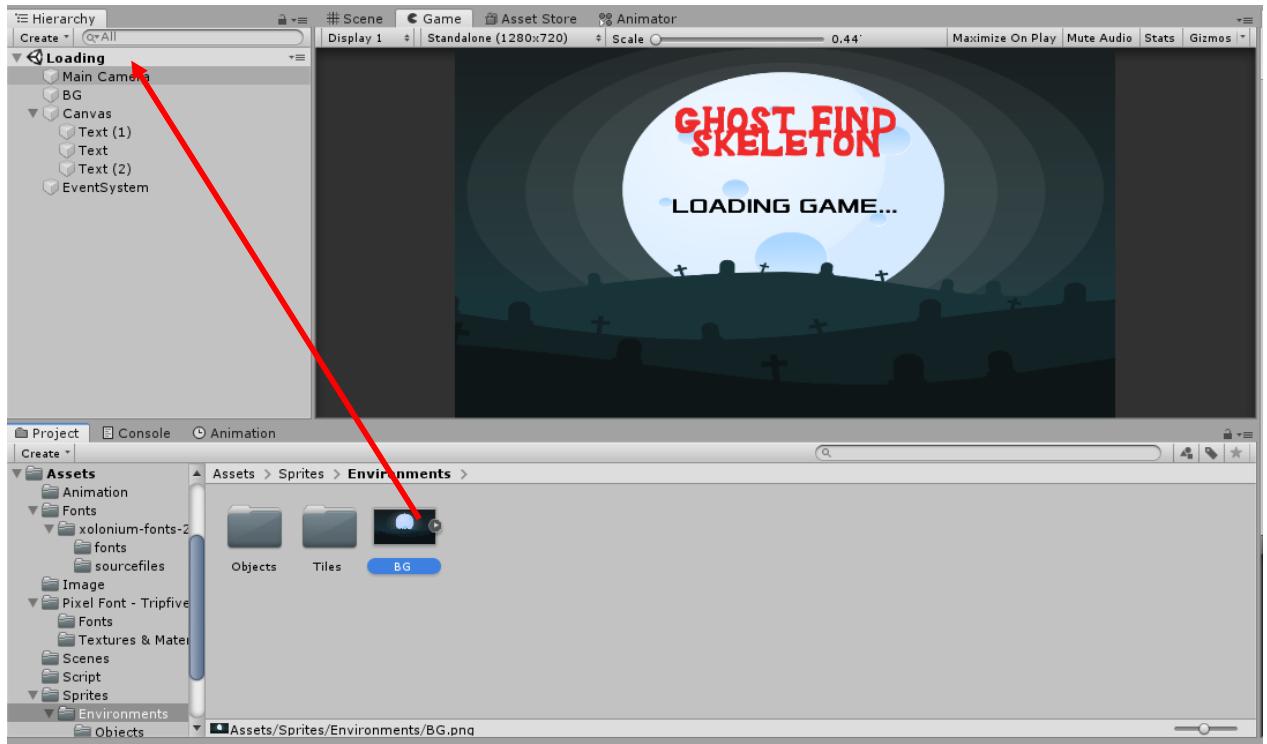


## Langkah 7 : Membuat Loading Game untuk Mulai

1. Membuat **Scene** baru dengan nama **Loading**. New Scene > tulis nama Scene Loading.



2. DRAG dan DROP pada Assets > Sprites > Environment > pilih BG ke Hierarchy.



3. Menambahkan 3 UI Text, yaitu :

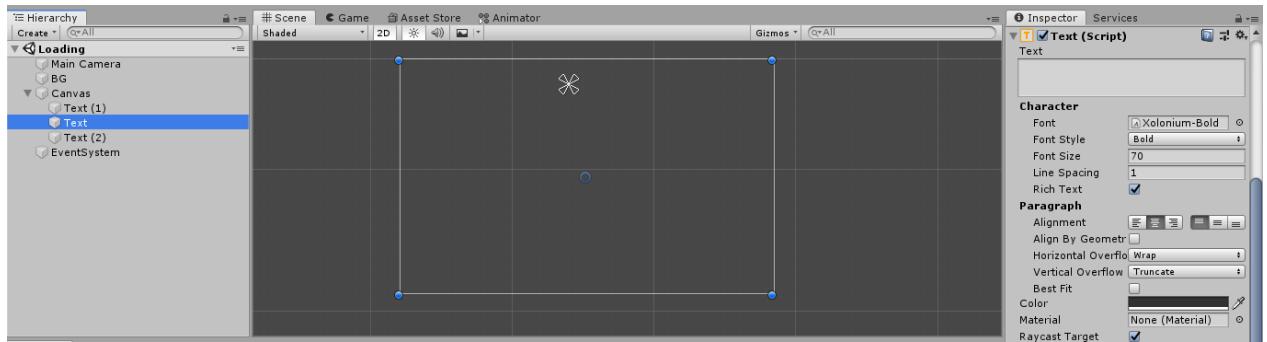
**UI Text pertama** untuk nama Game.



**UI Text kedua** untuk tulisan “Loading Game....”



## UI Text ketiga untuk menampilkan countdown dengan perhitungan waktu mundur.



## 4. Membuat Script dengan nama CountDownTimer. Lalu DRAG and DROP ke MainCamera

The screenshot shows the Microsoft Visual Studio interface with the 'CountDownTimer.cs' script open in the code editor. The script contains the following C# code:

```

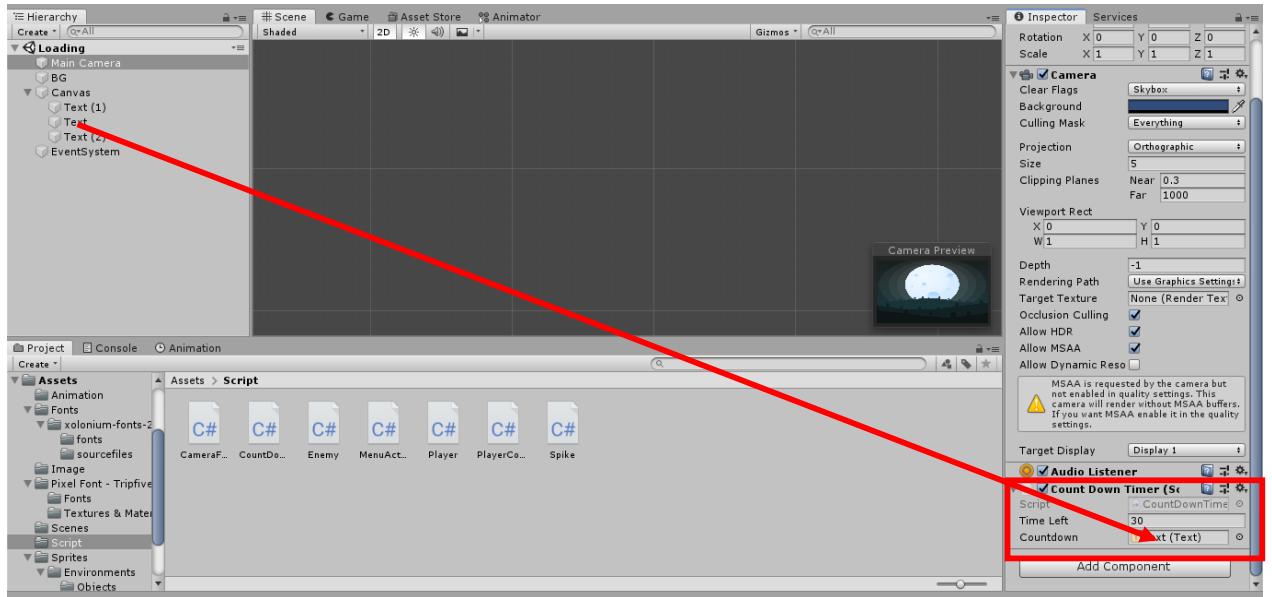
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System.Threading;

public class CountDownTimer : MonoBehaviour
{
    public int timeLeft = 30; //Seconds Overall
    public Text countdown; //UI Text Object
    void Start()
    {
        StartCoroutine("LoseTime");
        Time.timeScale = 1; //Just making sure that the timeScale is right
    }
    void Update()
    {
        countdown.text = ("**" + timeLeft); //Showing the Score on the Canvas
    }
    //Simple Coroutine
    IEnumerator LoseTime()
    {
        while (timeLeft > 0)
        {
            yield return new WaitForSeconds(1);
            timeLeft--;
        }
        Application.LoadLevel(1);
    }
}

```

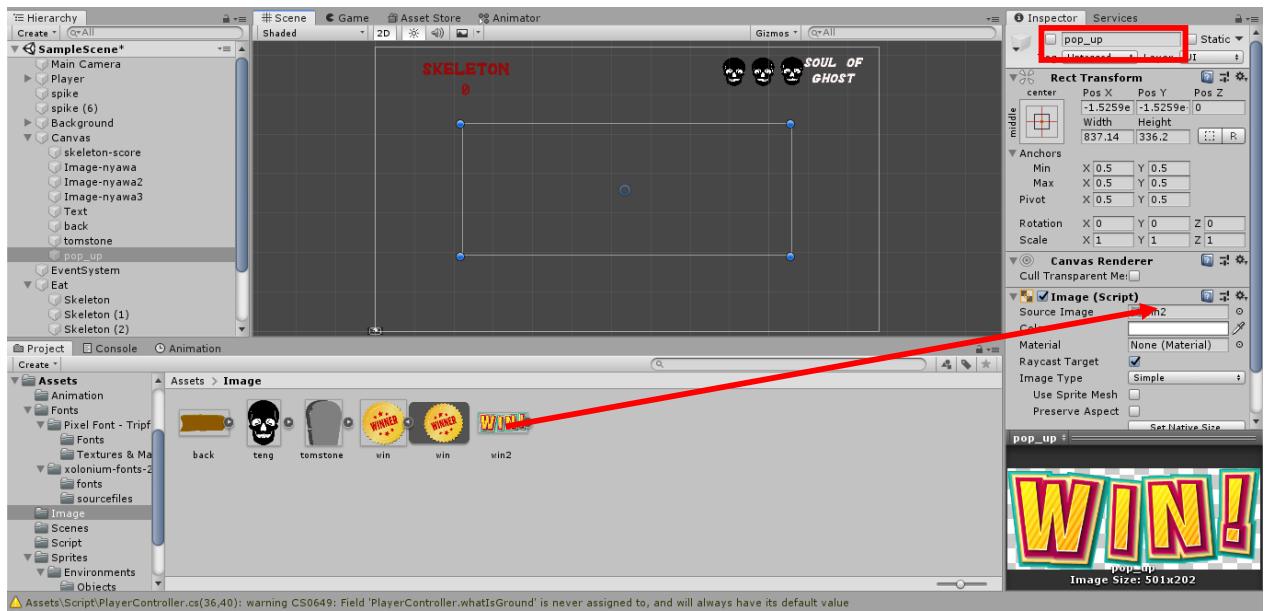
Below the code editor is the Unity Editor's Hierarchy panel, which shows the 'Main Camera' selected. A red arrow points from the 'Main Camera' node in the Hierarchy panel down towards the 'CountDownTimer.cs' script in the Project panel. The Project panel shows the script is located at 'Assets/Script/CountDownTimer.cs'.

5. Pada **Inspector MainCamera** setting **Time Left** yang digunakan untuk mengatur permulaan waktu hitung mundur dengan **30 detik**, lalu **DRAG** dan **DROP** UI Text ketiga untuk menampilkan waktu.

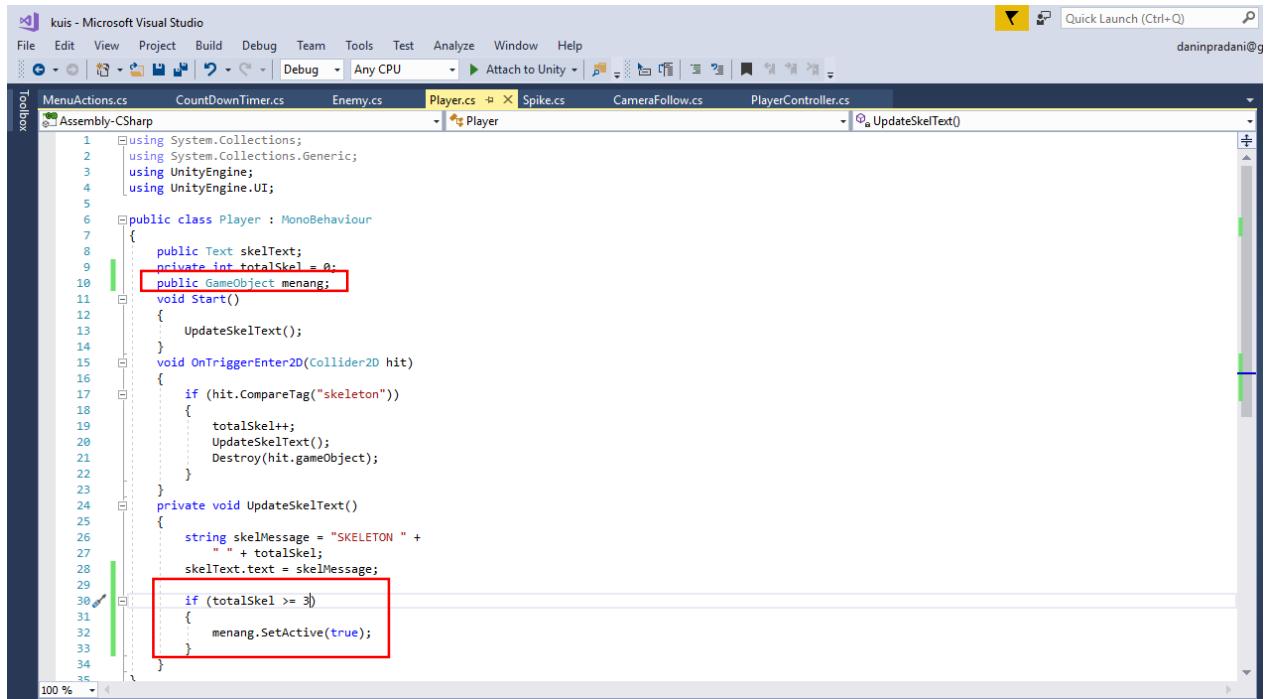


### Langkah 8 : Menambahkan Pop Up Jika Score/Nilai Berjumlah 4

1. Tambahkan UI Image dengan nama **pop\_up**. Jika sudah jangan ceklis check box dengan nama **pop\_up**. Lalu **DRAG** dan **DROP** object **win2**.



## 2. Tambahkan action Pop Up pada Script Player.



```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Player : MonoBehaviour
{
    public Text skelText;
    private int totalSkel = 0;
    public GameObject menang;

    void Start()
    {
        UpdateSkelText();
    }

    void OnTriggerEnter2D(Collider2D hit)
    {
        if (hit.CompareTag("skeleton"))
        {
            totalSkel++;
            UpdateSkelText();
            Destroy(hit.gameObject);
        }
    }

    private void UpdateSkelText()
    {
        string skelMessage = "SKELETON " +
            " " + totalSkel;
        skelText.text = skelMessage;

        if (totalSkel >= 3)
        {
            menang.SetActive(true);
        }
    }
}

```

## 3. Pada GameObject Player DRAG and DROP object pop\_up.

