MEMBUAT SISTEM GAME

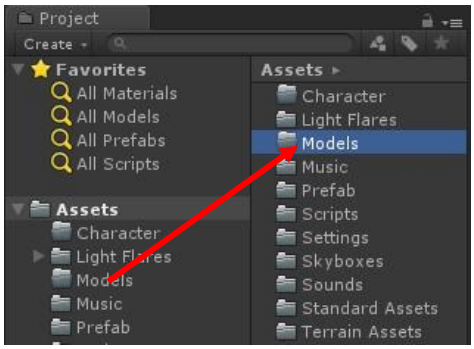
(Model dan Animasi)

Pada praktik sebelumnya, kita telah melakukan simulasi pembuatan player dan objek musuh dengan objek bawaan Unity. Pada praktik kali ini kita akan mengganti objek dengan model sendiri sehingga memiliki tampilan yang lebih menarik. Bahan model dan animasi dapat diperoleh dari asset yang telah dibagikan pada pertemuan sebelumnya.

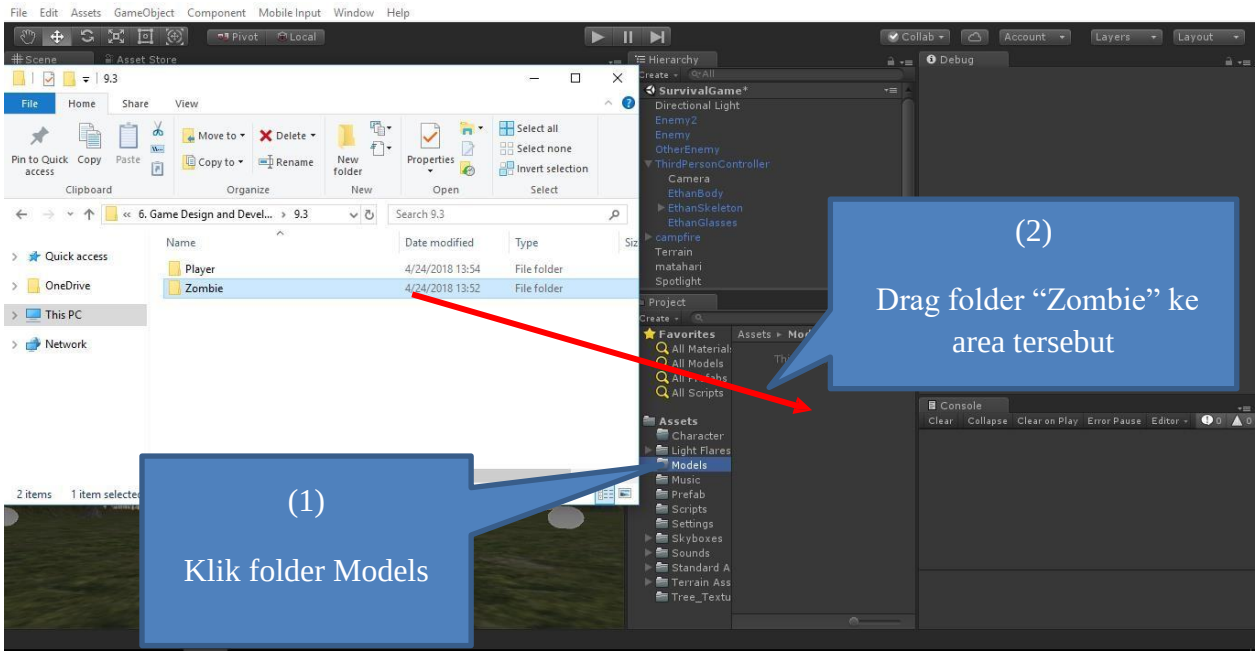
A. Membuat 3D Model Zombie

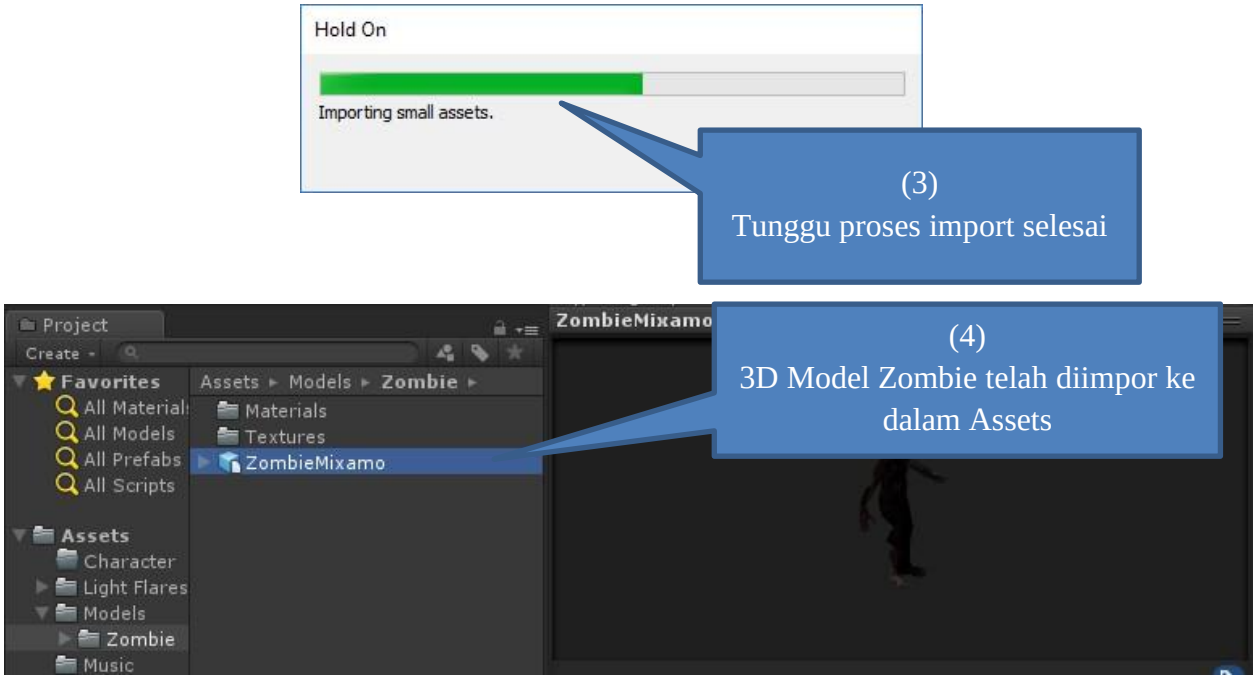
Zombie merupakan salah satu model musuh. Untuk membuat model zombie ikuti langkah- langkah berikut:

1. Buat folder baru di dalam asset dengan nama "**Models**".

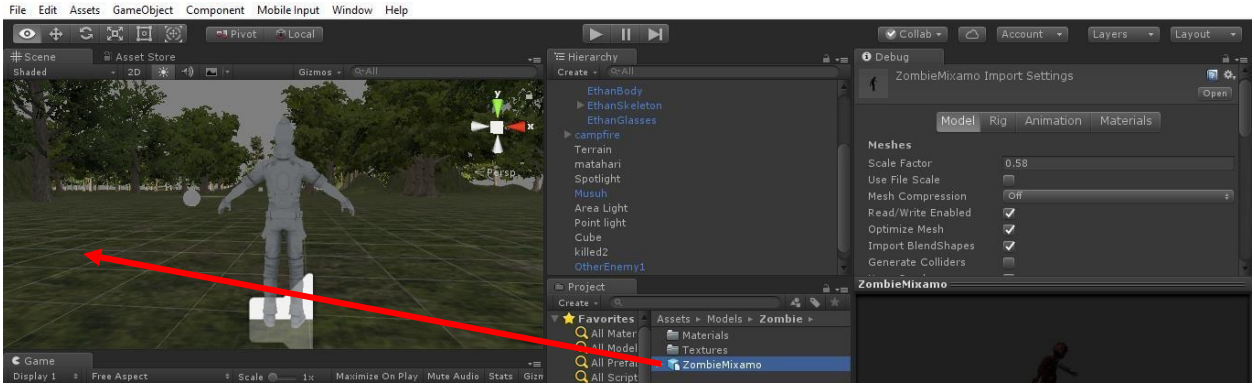


2. Import 3D model dengan cara drag folder "**Zombie**" ke dalam folder "**Models**" yang telah dibuat sebelumnya.

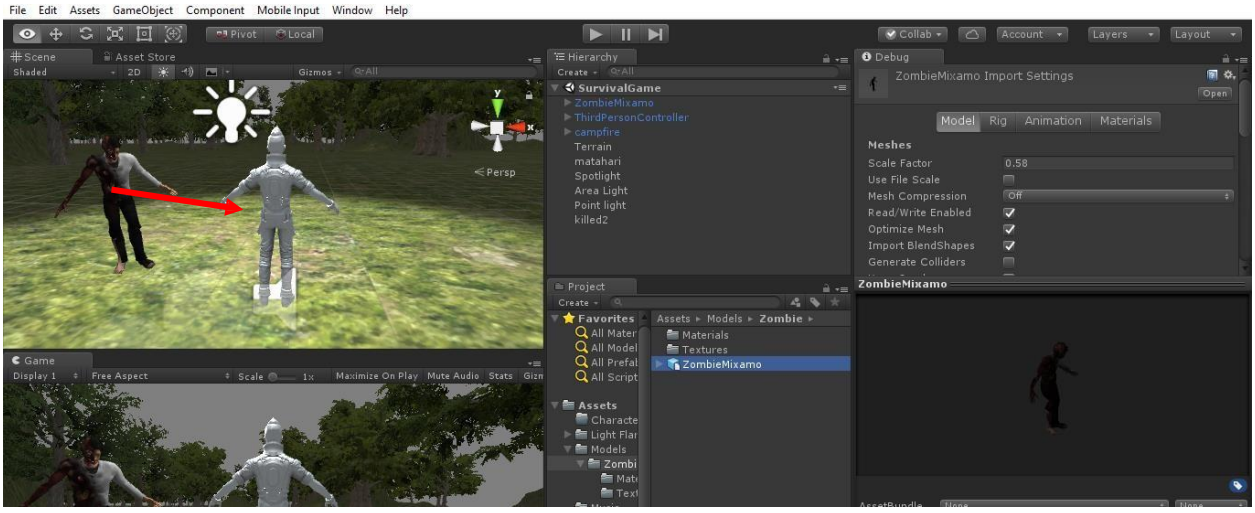




3. Drag “ZombieMixamo” ke dalam “Scene View”.

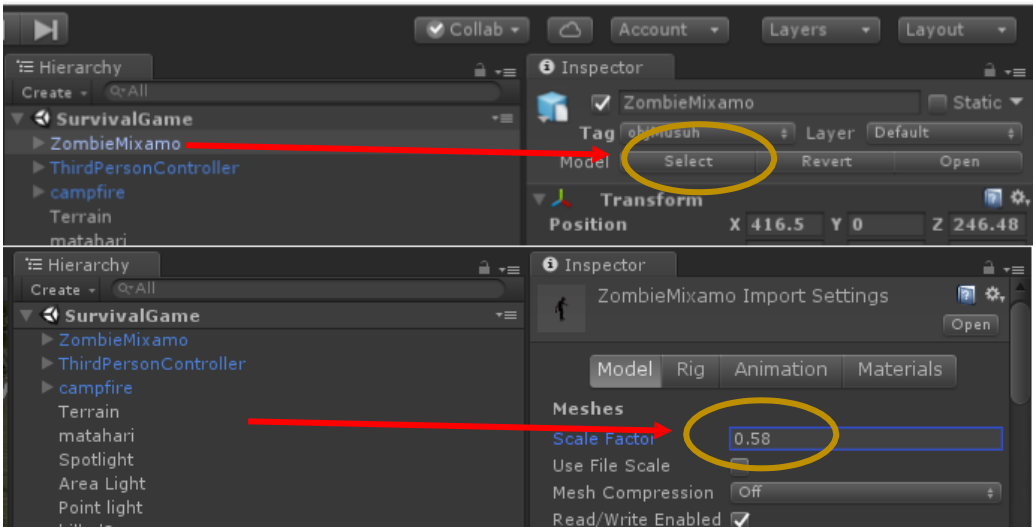


4. Lakukan rotasi pada zombie sehingga menghadap ke karakter utama.

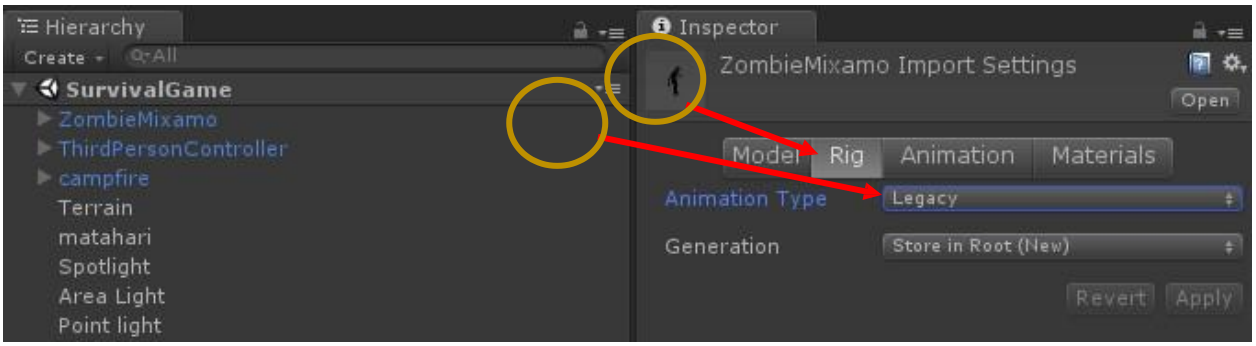


5. Hapus SEMUA OBJEK MUSUH selain “ZombieMixamo” dari peta game atau hirarki.

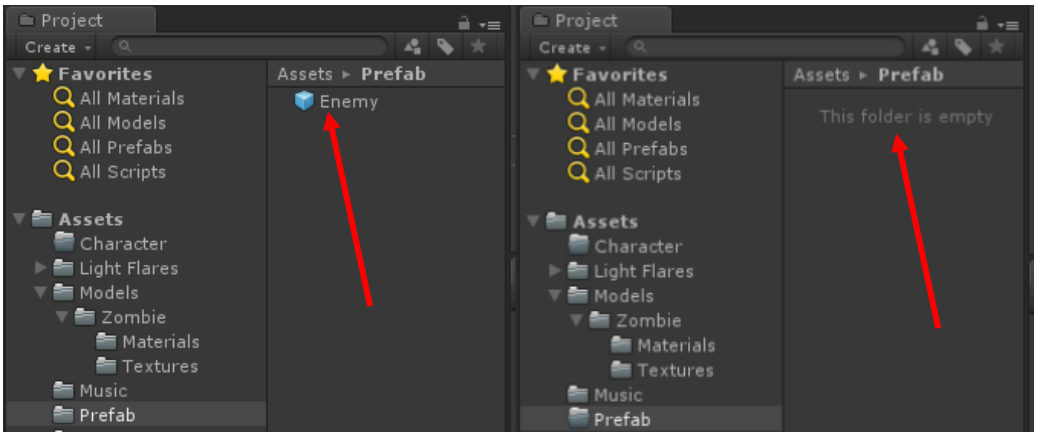
6. Jika objek “Zombie” tidak terlihat atau terlalu kecil (skala defaulnya 0.01), maka ubah skalanya (Scale Factor). Ubah Scale Factor dari zombie menjadi 0.58, caranya: klik pada ZombieMixamo pilih “Select” ubah nilai Scale Factor = 0.58



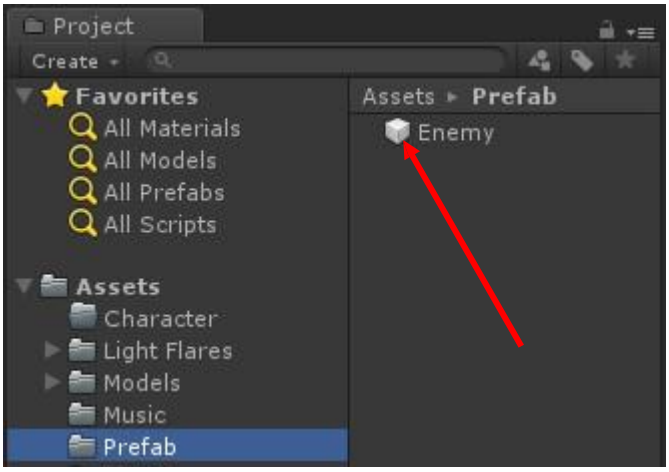
7. Ubah tipe Animasi menjadi “Legacy” dengan cara pilih tab “Rig” pada Animation Type pilih “Legacy”.



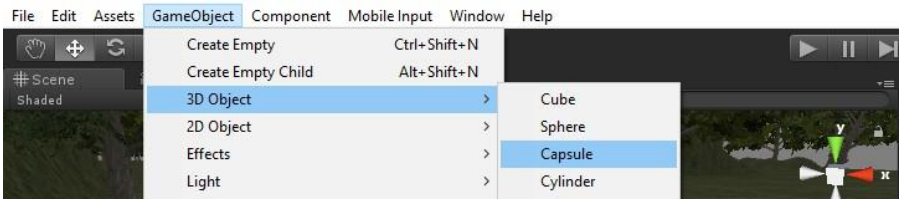
8. Hapus prefab” Enemy” pada kotak hirarki.



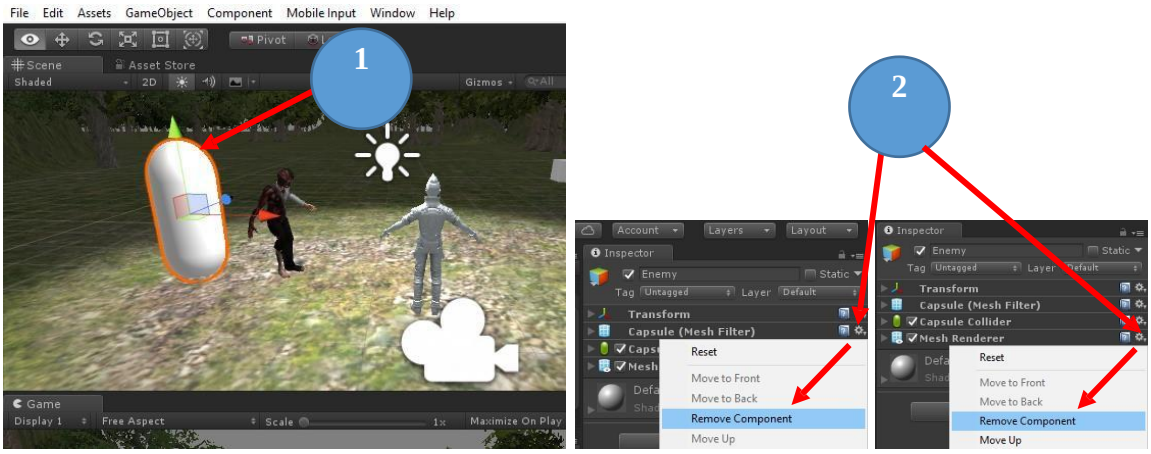
9. Buat Prefab baru di dalam folder “Prefab” dengan nama “Enemy”.



10. Terlihat bahwa prefab tersebut masih berwarna putih, hal ini menandakan bahwa tidak ada komponen atau objek apapun di dalam prefab Enemy.
11. Selanjutnya buat Game Object baru. Klik menu GameObject 3D Object Capsule rename game objek Capsule dengan "Enemy".



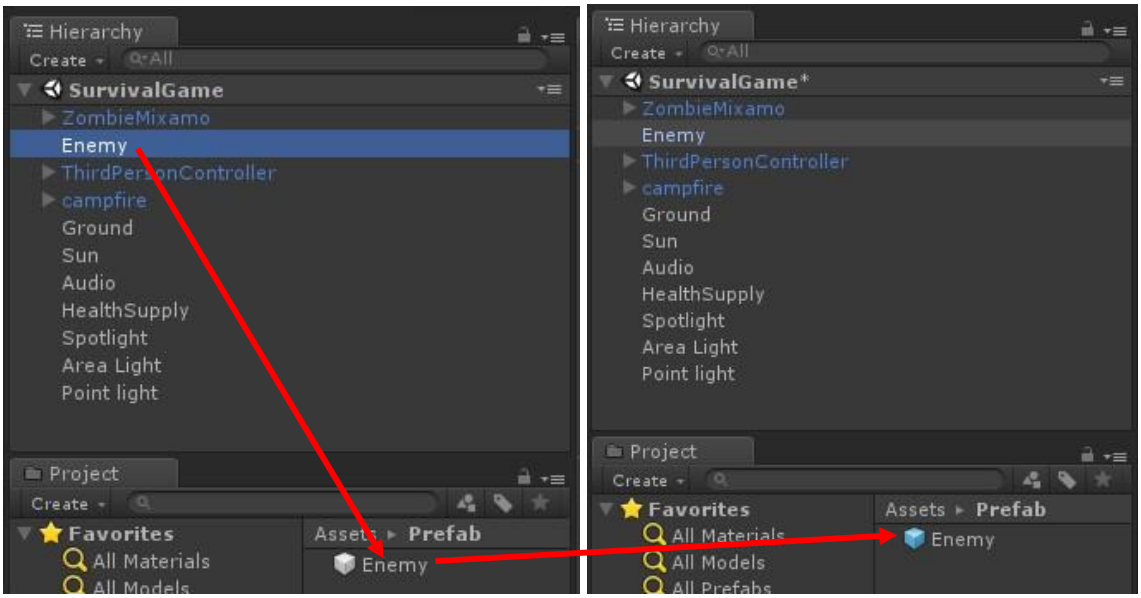
12. Secara default, di dalam game objek "Capsule" terdapat beberapa komponen, yaitu: Capsule (Mesh Filter), Capsule Collider, dan Mesh Renderer. Selanjutnya atur ukuran capsule agar sesuai dengan ukuran Enemy (sedikit lebih tinggi dari Enemy)[1] . Kemudian hapus Capsule (Mesh Filter) dan Mesh Renderer[2].



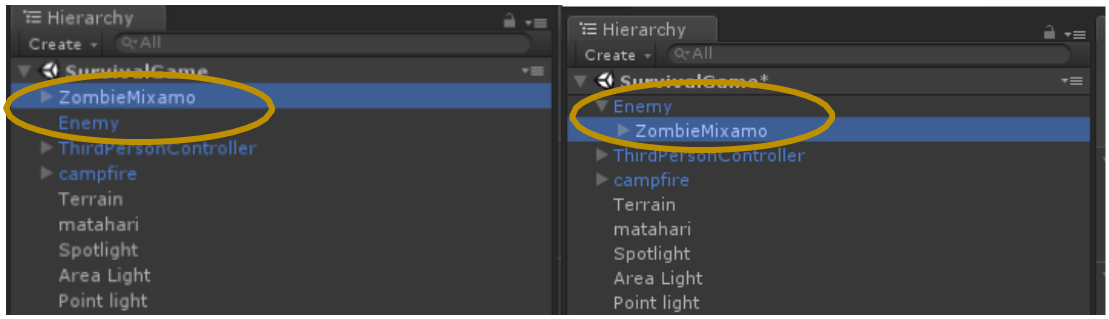
13. Jika sudah, maka objek Enemy sekarang terlihat tidak memiliki bentuk. Klik pada komponen “Capsule Collider” dari objek Enemy untuk melihat posisi objek “Enemy”.

14. Atur posisi posisi Zombie agar berada di tengah-tengah objek objek Enemy.

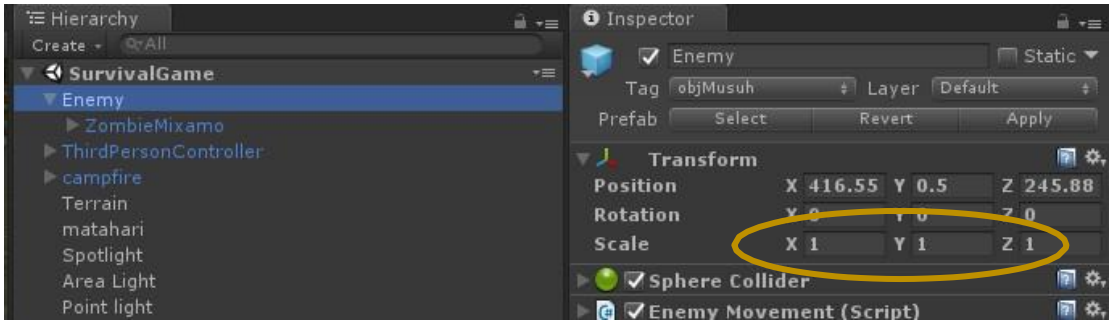
15. Sekarang kembali Drag “objek Enemy” ke dalam “Prefab Enemy”. Selanjutnya, prefab Enemy akan terlihat menjadi berwarna biru. Hal ini menandakan bahwa sudah terdapat game objek di dalam prefab Enemy.

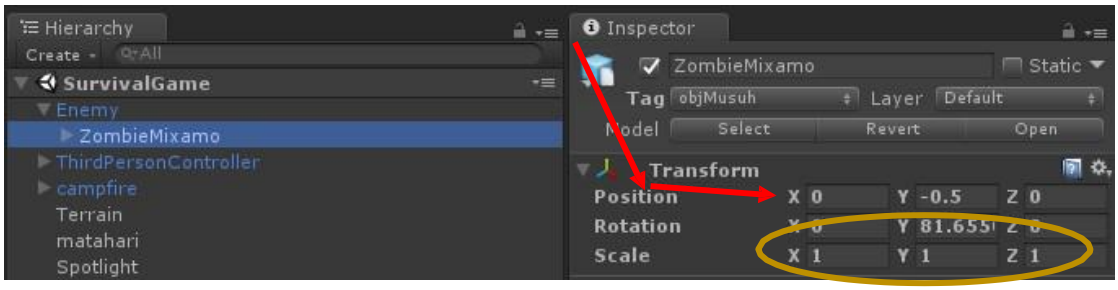


16. Selanjutnya, drag model “Zombie” ke objek “Enemy”, sehingga ZombieMixamo menjadi Child dari game objek Enemy.

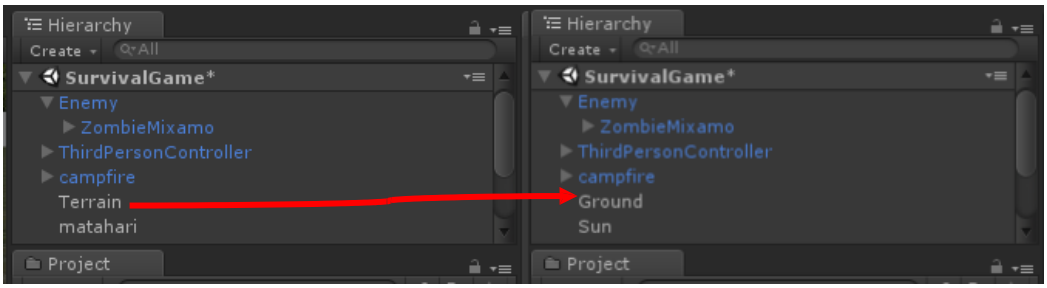


17. Atur skala “Zombie” dan Enemy menjadi 1,1,1. Kemudian Atur posisi sumbu X,Y,Z sehingga zombie terlihat pada posisi menyentuh tanah.

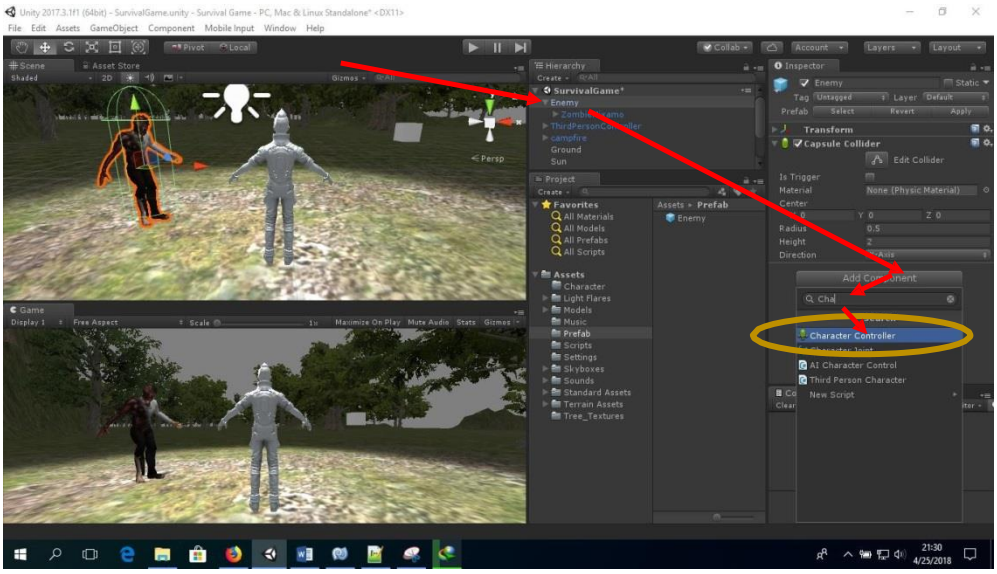




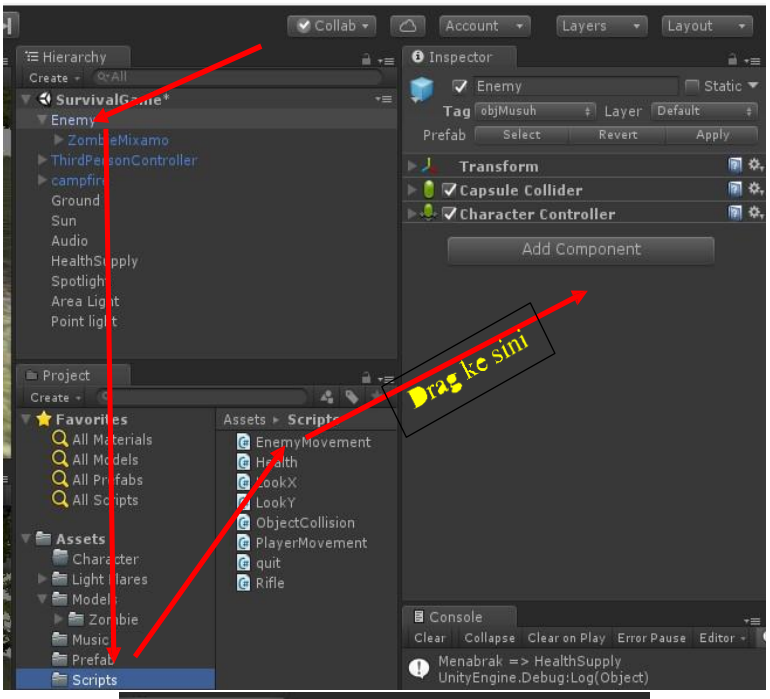
18. Selanjutnya rename objek “Terrain” menjadi “Ground”

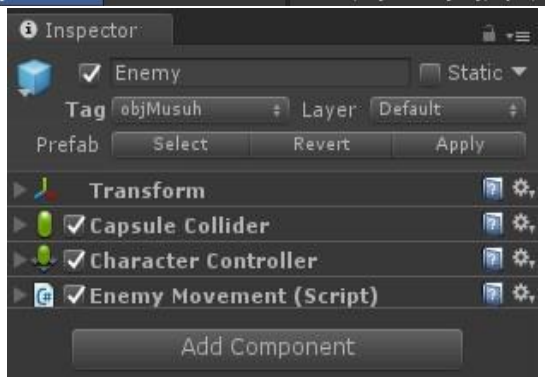


19. Tambahkan komponen “Character Controller” pada Enemy. Caranya: klik Enemy klik tombol “Add Component” pada kotak pencarian ketik “Cha” pilih “Character Controller”.



20. Pastikan posisi zombie berada tepat di tengah-tengah character controller dan objek Enemy.
21. Tambahkan skrip “EnemyMovement” ke dalam objek Enemy. Caranya: klik objek Enemy klik folder Script kemudian drag skrip “EnemyMovement” ke dalam objek Enemy.





22. Jalankan Game, maka anda akan melihat beberapa bug, yaitu animasi berhenti (padahal seharusnya tidak) dan mungkin objek zombie sedikit melayang. Hal ini sebenarnya sederhana, tinggal menambahkan gravitasi pada skrip "EnemyMovement". Lihat kode berikut:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnemyMovement : MonoBehaviour {
6     CharacterController _controller;
7     Transform _player;
8
9     [SerializeField]
10    float _moveSpeed = 0.8f;
11
12    [SerializeField]
13    float _gravity = 2.0f;
14
15    float _yVelocity = 0.0f;
16
17    void Start () {
18        GameObject playerObject = GameObject.FindGameObjectWithTag("Player");
19        _player = playerObject.transform;
20        _controller = GetComponent<CharacterController>();
21    }
22
23    void Update () {
24        Vector3 direction = _player.position - transform.position;
25        direction.Normalize();
26        Vector3 velocity = direction *  moveSpeed;
27        if (_controller.isGrounded) {
28        } else {
29            _yVelocity -= _gravity;
30        }
31        velocity.y = _yVelocity;
32        _controller.Move(velocity * Time.deltaTime);
33    }
34 }
```

B. Membuat Zombie Menghadap Karakter (Player)

Saat anda simulasikan, objek zombie akan selalu menghadap ke arah default mereka. Kita dapat membuat objek zombie selalu menghadap ke karakter dengan mengubah skrip berikut:

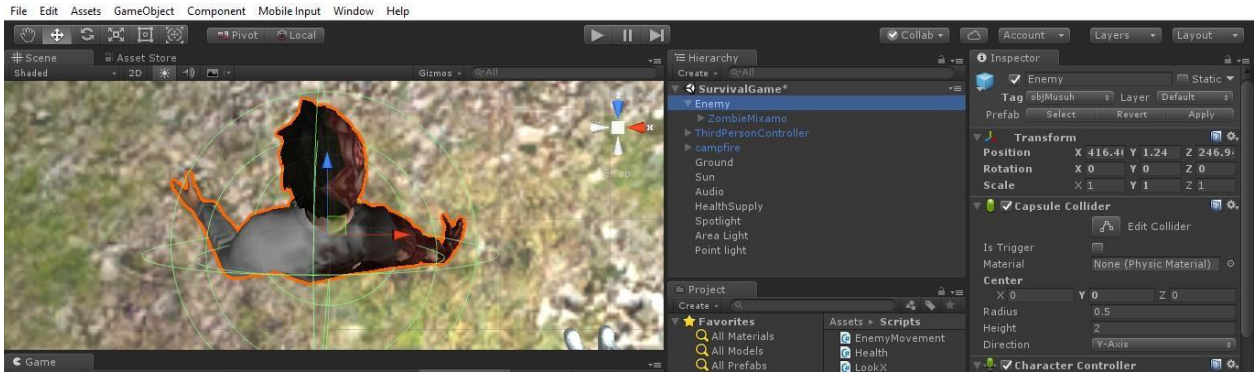
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnemyMovement : MonoBehaviour {
6     CharacterController _controller;
7     Transform _player;
8
9     [SerializeField]
10    float _moveSpeed = 0.8f;
11
12    [SerializeField]
13    float _gravity = 2.0f;
14
15    float _yVelocity = 0.0f;
```

```
17      void Start () {
18          GameObject playerObject = GameObject.FindGameObjectWithTag("Player");
19          _player = playerObject.transform;
20          _controller = GetComponent<CharacterController>();
21      }
22
23      void Update () {
24          Vector3 direction = _player.position - transform.position;
25          direction.Normalize();
26          Vector3 velocity = direction * _moveSpeed;
27          if (_controller.isGrounded) {
28          } else {
29              _yVelocity -= _gravity;
30          }
31          velocity.y =  yVelocity;
32          transform.rotation = Quaternion.LookRotation(direction);
33          _controller.Move(velocity * Time.deltaTime);
34      }
35  }
```

Saat kita simulasikan maka otomatis zmbie akan menghadap ke karakter. Namun, ketika karakter melompat maka zombie akan jatuh ke tanah karena posisi karakter berada di atas zombie. Untuk mengatasi permasalahan ini tambahkan kode berikut:

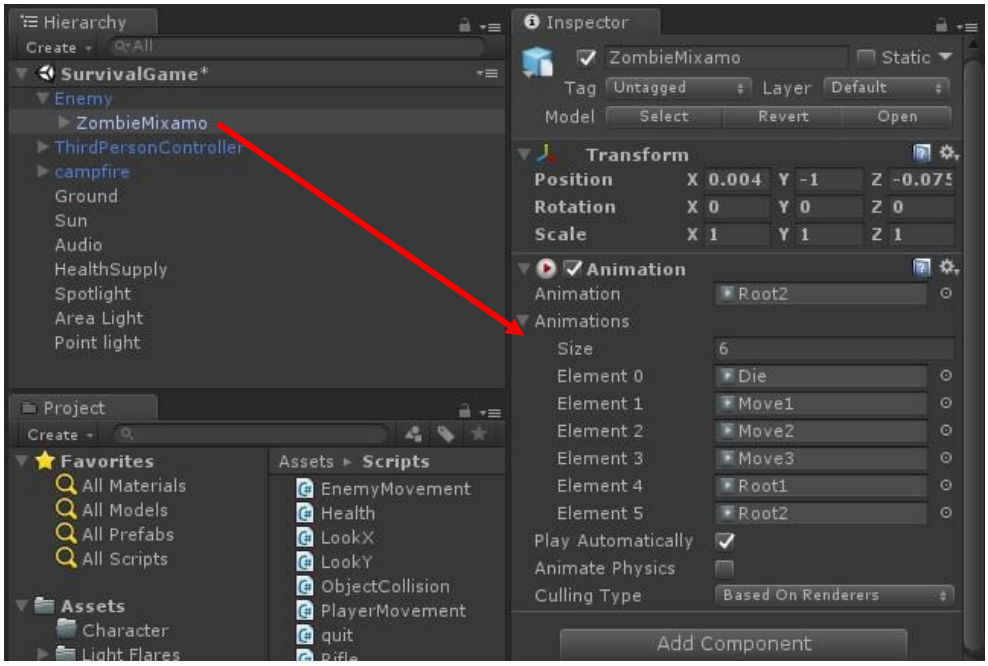
```
void Update () {
    Vector3 direction = _player.position - transform.position;
    direction.Normalize();
    Vector3 velocity = direction * _moveSpeed;
    if (_controller.isGrounded) {
    } else {
        _yVelocity -= _gravity;
    }
    velocity.y = _yVelocity;
    direction.y = 0;
    transform.rotation = Quaternion.LookRotation(direction);
    _controller.Move(velocity * Time.deltaTime);
}
```

Catatan:
Pastika zombie telah berada tepat di tengah objek Enemy. Dan rotasi dari zombie adalah 0,0,0.



C. Membuat Animasi Zombie

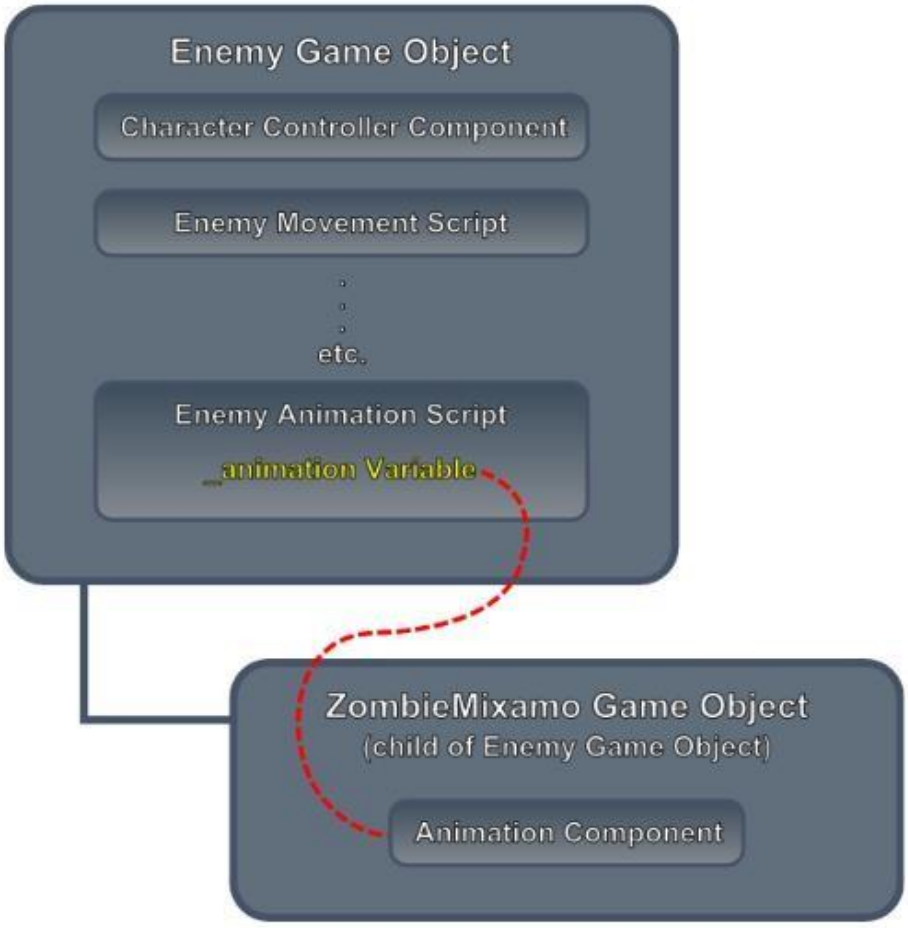
Sekarang kita dapat melanjutkan fokus dalam pembuatan game. Anda akan melihat zombie berdiri diam dan meluncur dari lantai ketika bergerak ke arah Karakter (player). Pada kasus ini kita akan memperbaiki kekurangan tersebut. Kita akan membuat zombie dapat memainkan animasi berjalannya. Pilih objek game "ZombieMixamo" Lalu lihatlah Inspector. Maka akan terlihat komponen animasi di sana. Salah satu propertinya adalah daftar animasi. Daftar animasi tersebut adalah animasi yang dimiliki oleh model 3d zombie.



Pada gambar di atas dapat anda lihat, terdapat 6 animasi yaitu “Die”, “Move1”, ”Move2”, ”Move3”, ”Root1”, ”Root2” . masing-masing dari jenis animasi tersebut memiliki animasi yang berbeda-beda yang dapat digunakan. Selanjutnya kita lakukan sedikit sentuhan terhadap animasi- animasi tersebut dengan menggunakan skrip agar kita dapat melakukan kontrol terhadap animasi- animasi tersebut sesuai keinginan kita. Kita mulai dengan membuat skrip baru dengan nama “EnemyAnimation” (pastikan letak skrip adalah di dalam folder Scripts).

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnemyAnimation : MonoBehaviour {
6     Animation _animation;
7
8     void Start () {
9         _animation = GetComponentInChildren<Animation> ();
10        _animation ["Move1"].wrapMode = WrapMode.Loop;
11        _animation.Play ("Move1");
12    }
13 }
```

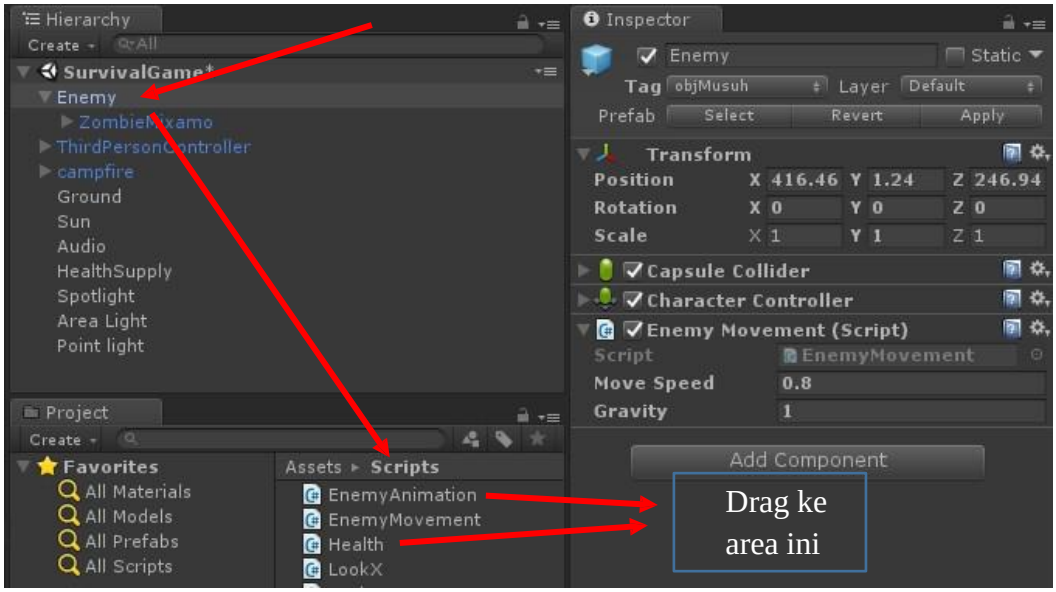
Pada baris ke 6, merupakan kode yang digunakan untuk menghandle komponen animasi. Pada baris ke 9 merupakan inisialisasi dari handling komponen animasi. Komponen Animasi berada di dalam "ZombieMixamo", dan kita akan melakukan handling pada komponen-komponen tersebut. Namun, skrip EnemyAnimation ini akan dimasukkan ke dalam objek game "Enemy". Tapi seperti yang kita ketahui, komponen Animasi yang dibutuhkan untuk berada di sisi lain, ada di dalam objek game ZombieMixamo. Mengingat, ZombieMixamo adalah child dari objek game Enemy. Dalam situasi ini, kita dapat menggunakan GetComponentInChildren untuk mendapatkan komponen Animasi dengan benar. Diagram berikut menunjukkan apa yang terjadi ketika kita menggunakan GetComponentInChildren:



Menggunakan GetComponentInChildren, variabel "_animation" selanjutnya akan "dihubungkan" ke komponen Animasi dalam child objek game, yaitu ZombieMixamo. GetComponentInChildren<Animation>() akan mengembalikan komponen Animasi pertama yang dapat ditemukan di antara child objek game. Meskipun anehnya, GetComponentInChildren juga melakukan pencarian terhadap parent objek game.

Baris ke 11 adalah membuat kode untuk memainkan animasi berjalan (walk) yang tepat

untuk zombie kita. Kita cukup memberikan nama animasi yang ingin kami mainkan. Di sini, kita menggunakan animasi bernama "Move1". Animasi walk yang kita gunakan dimaksudkan untuk dimainkan berulang kali. Baris 10 melakukan beberapa konfigurasi untuk membuatnya agar animasi yang kita gunakan akan berulang dengan menggunakan perintah "WrapMode.Loop". Tambahkan skrip “EnemyAnimation” dan “Health” ke Inspector objek game Enemy.



Selanjutnya jalankan game. Seharusnya pada tahap ini Zombie harus dapat berjalan dengan benar.

Setelah itu tambahkan variasi pada animasi “walk”. Caranya ubah kode skrip “EnemyAnimation” menjadi seperti berikut:

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

public class EnemyAnimation : MonoBehaviour {
    Animation _animation;

    void Start () {
        _animation = GetComponentInChildren<Animation> ();
        _animation ["Move1"].wrapMode = WrapMode.Loop;
        _animation.Play ("Move1");
        _animation ["Move1"].normalizedTime = Random.value;
    }
}
```

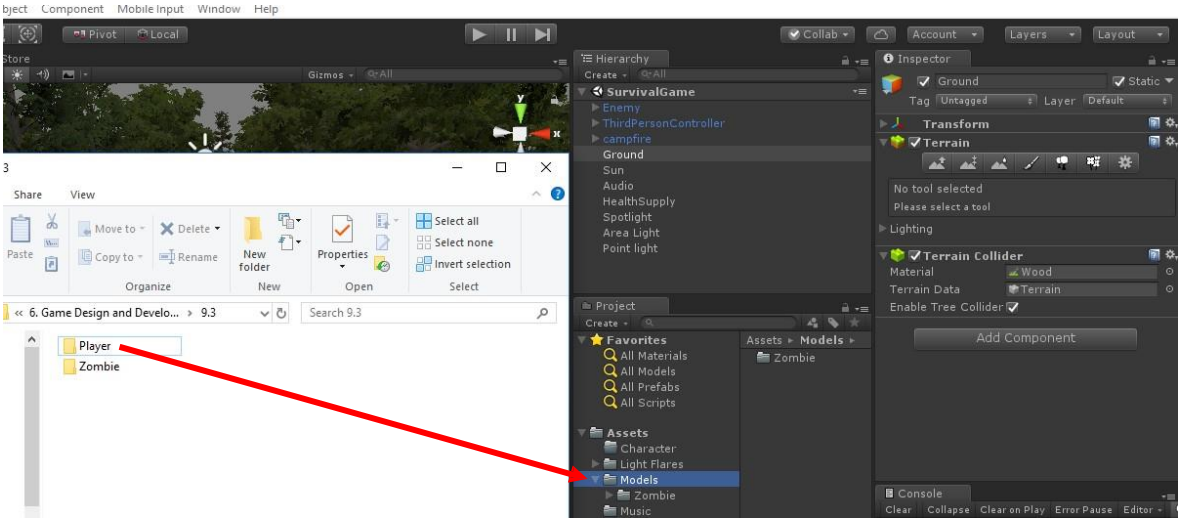
Kode tambahan di atas melakukan random starting point pada animasi kita. Random.value memberikan nilai yang berbeda ketika digunakan. Rentangnya adalah 0.0 sampai 1.0. Nilai random tersebut dimasukkan ke dalam normalizedTime. normalizedTime inilah yang mengontrol pada titik mana animasi aktif. Misalkan, nilai di atur 0.5, maka animasi akan dijalankan pada posisi tengah. 0.0 maksudnya adalah awal animasi dimainkan, dan 1.0 adalah akhir dari animasi dimainkan. Selanjutnya lakukan pengacakan terhadap animasi yang akan dimainkan. Caranya ubah skrip "EnemyAnimation" di atas menjadi seperti berikut:

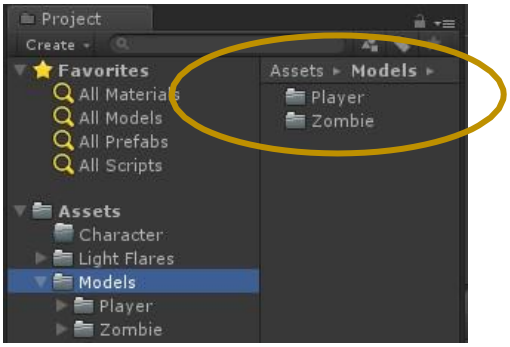
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class EnemyAnimation : MonoBehaviour {
    Animation _animation;

    void Start () {
        _animation = GetComponentInChildren<Animation> ();

        string animationToPlay = "";
        switch (Random.Range (0, 3)) {
            case 0:
                animationToPlay = "Move1";
                break;
            case 1:
                animationToPlay = "Move2";
                break;
            case 2:
                animationToPlay = "Move3";
                break;
            default:
                animationToPlay = "Move1";
                break;
        }

        _animation [animationToPlay].wrapMode = WrapMode.Loop;
        _animation.Play (animationToPlay);
        _animation [animationToPlay].normalizedTime = Random.value;
    }
}
```

D. Membuat 3D Model Player

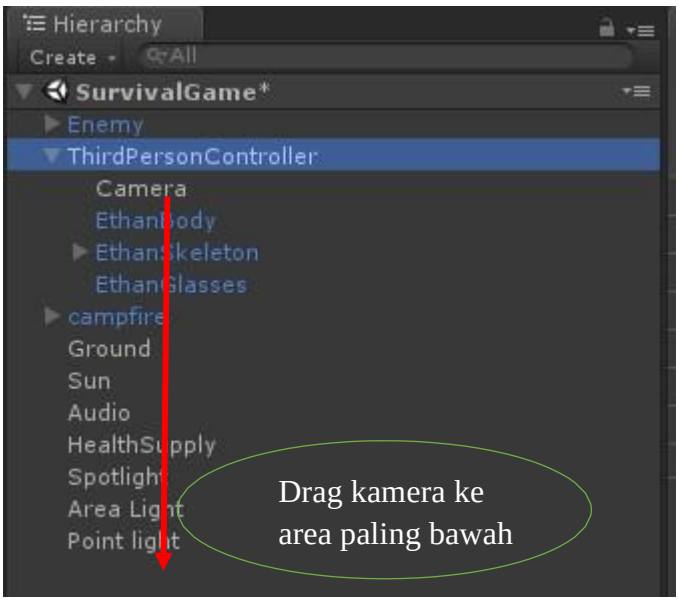
Setelah kita selesai membuat model dan animasi untuk musuh, sekarang kita lanjutkan dengan membuat model dan animasi untuk karakter utama (Player). Untuk lelakukannya ikuti langkah-langkah berikut:

1. Drag folder “Player” ke dalam folder “Models” yang ada di Assets.

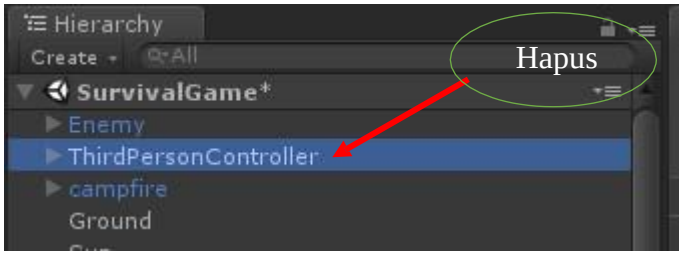




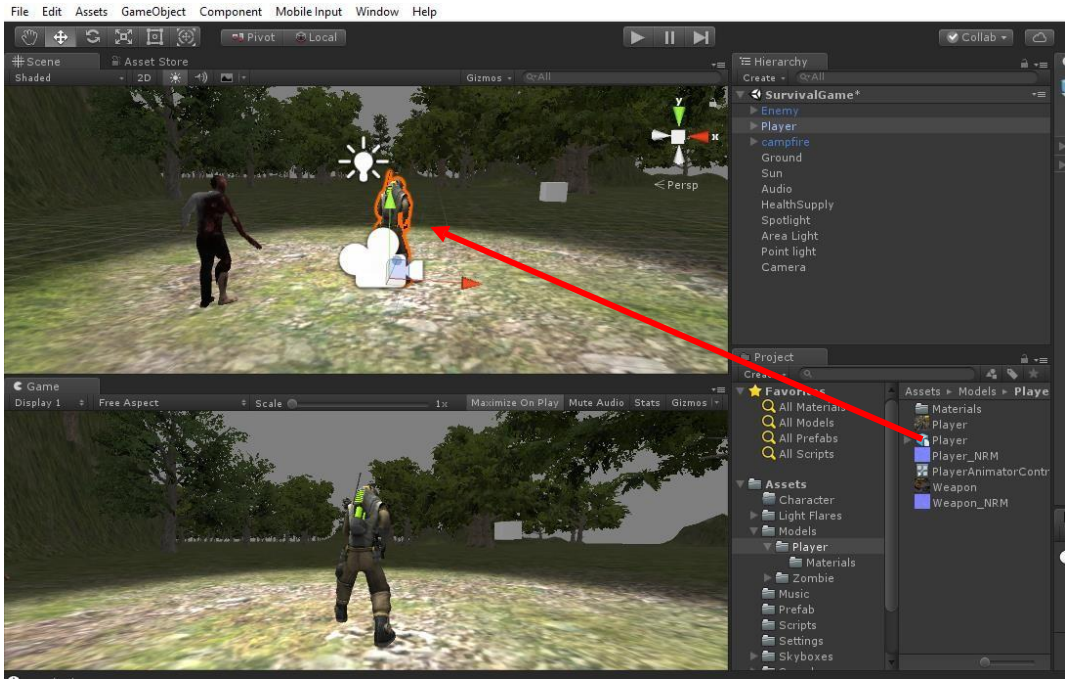
2. Karena kamera utama berada di dalam objek game “ThridPersonController”, kita keluarkan kamera terlebih dahulu.



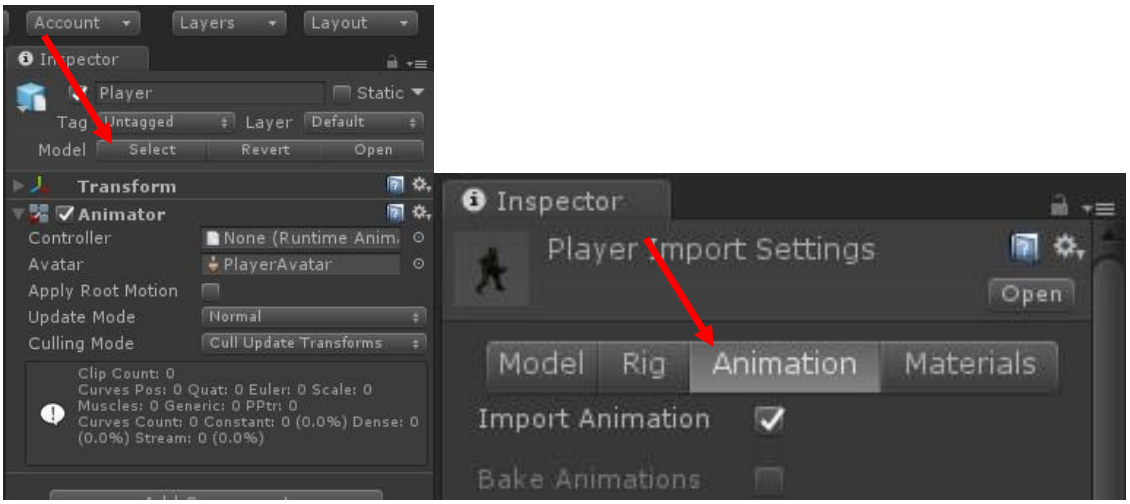
3. Hapus game objek “ThridPersonController”.



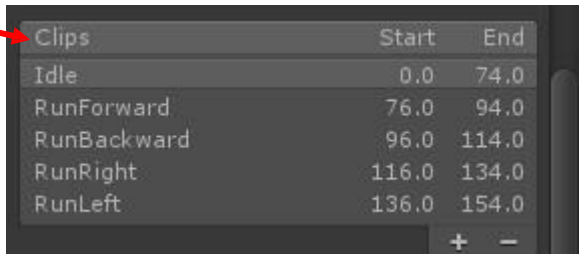
4. Drag prefab “Player” ke dalam Scene View.



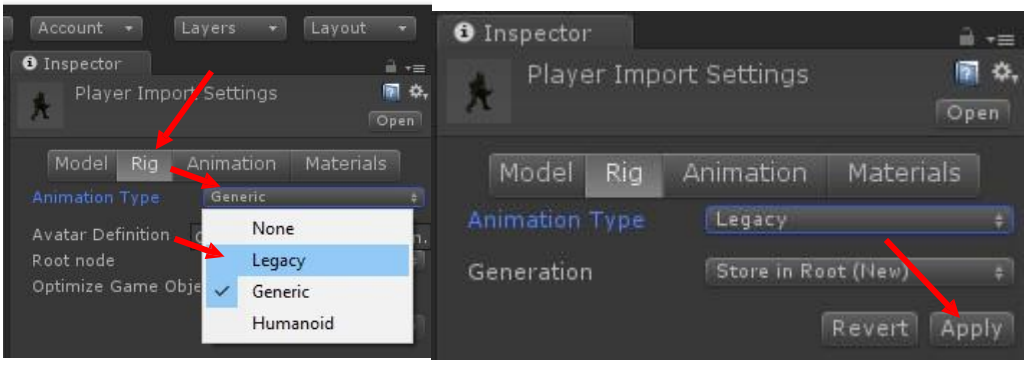
5. Pada bagian Model pilih “Select” pilih “Animation” pastikan properti animasi terlihat seperti gambar [2].



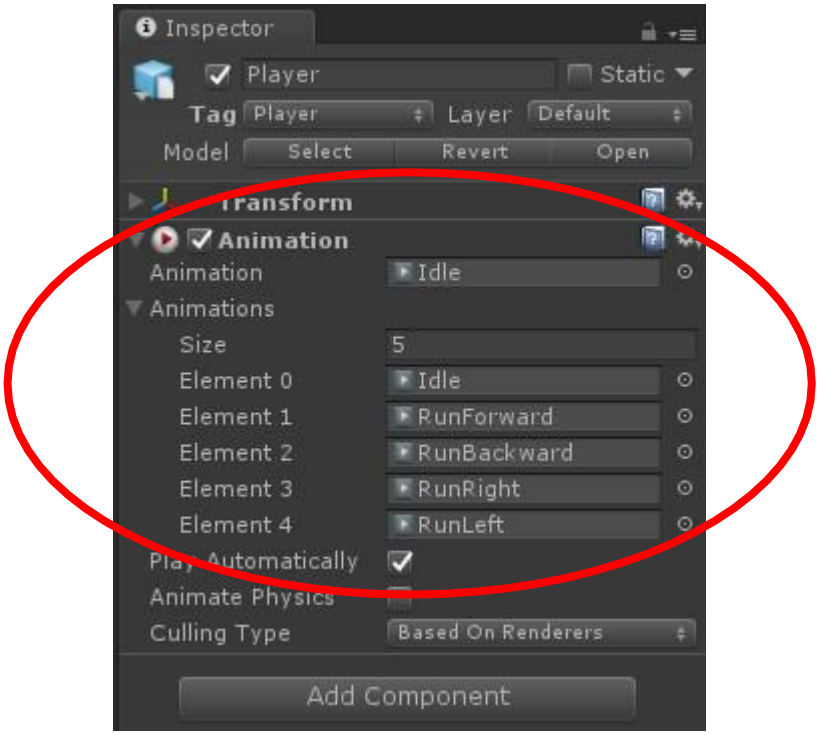
Gambar [2]



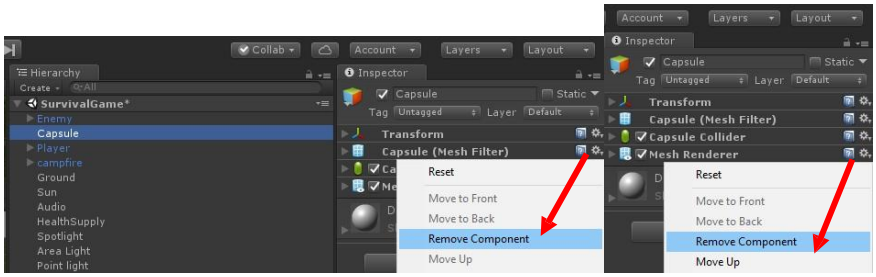
6. Selanjutnya kembali ke Tab “Rig” pada Animation Type pilih Legacy klik Apply.



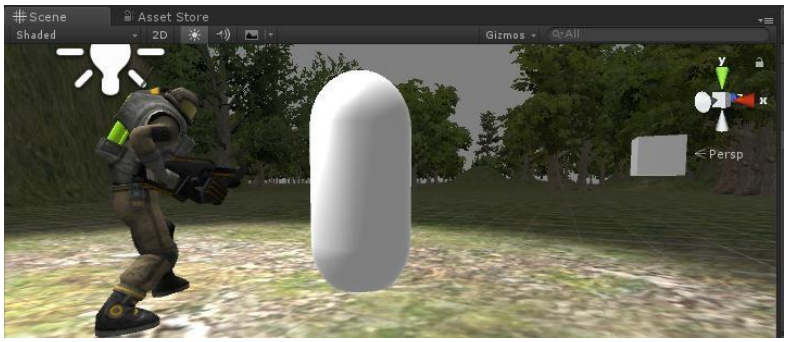
7. Selanjunya klik objek game “Player”, maka anda akan melihat properti pada “Player” terlihat menjadi seperti berikut:

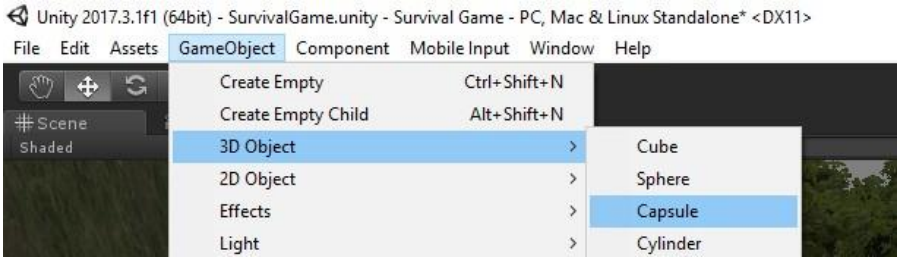


8. Sama dengan Enemy, kita buat satu game objek baru (Capsule). Caranya klik menu GameObject 3D Object Capsule.

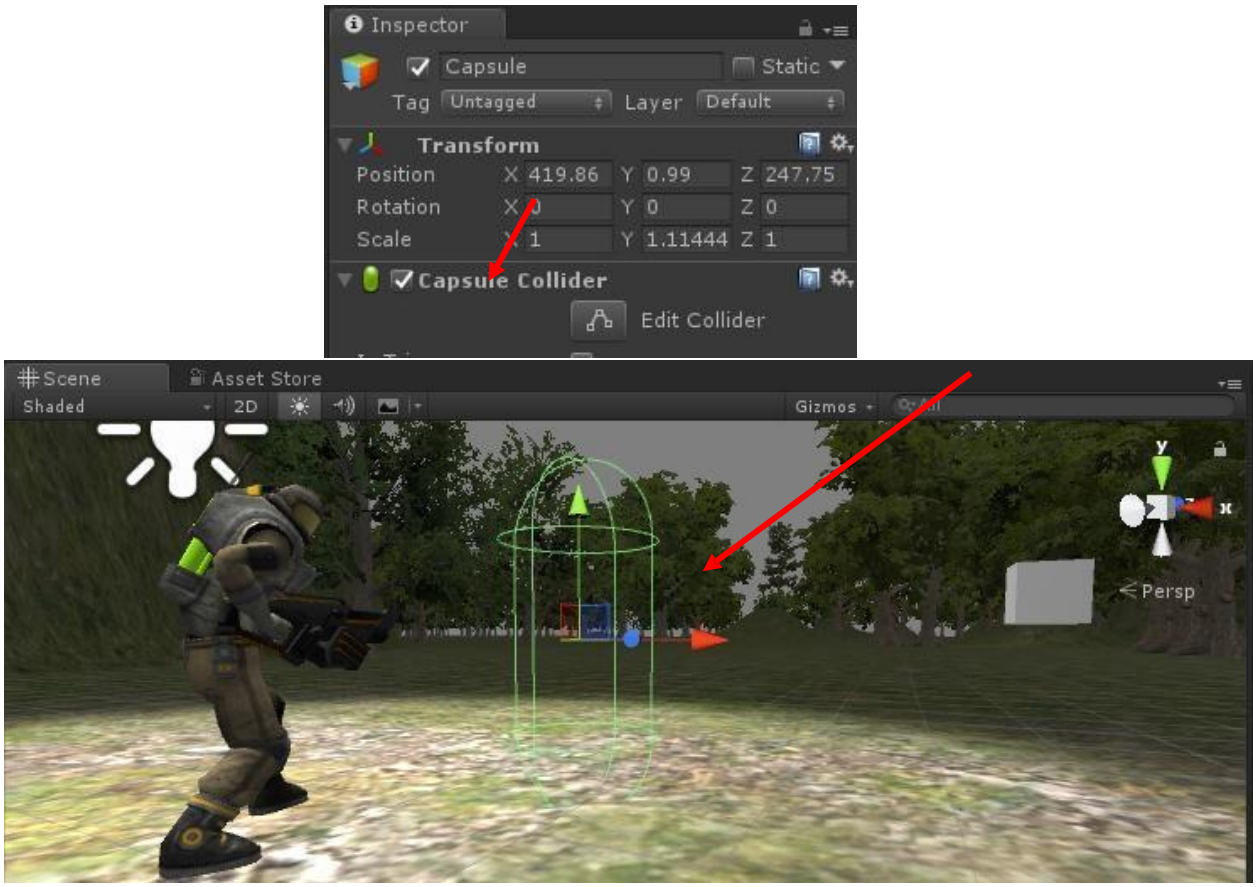


9. Hapus Mesh Filter dan Mesh Renderer.

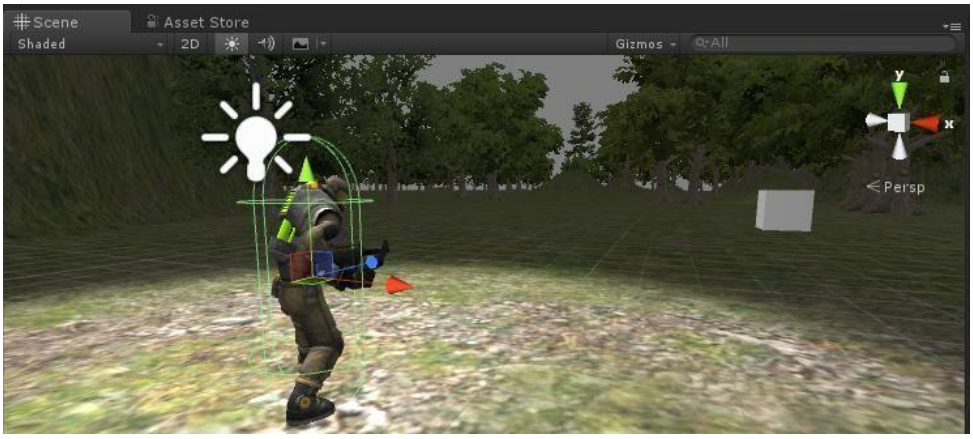




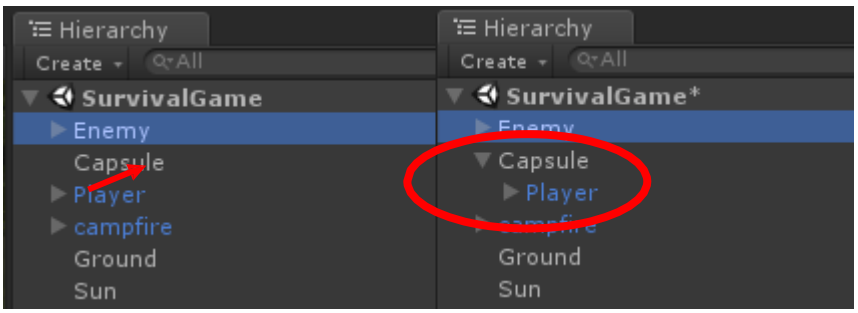
10. Klik pada Capsule Collider untuk melihat objek “Capsule” yang akan dijadikan parent dari Player.



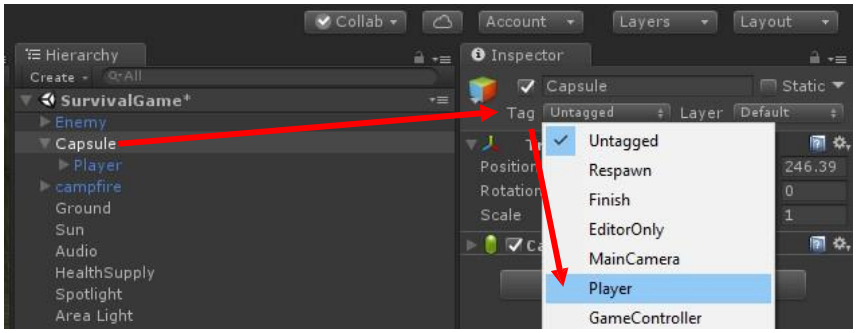
11. Selanjutnya pindah posisi “Capsule” di lokasi Player berada sehingga posisi Player menjadi berada tepat di tengah-tengah Capsule.



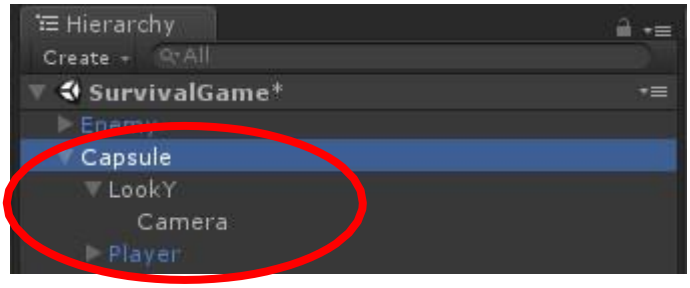
12. Drag Player ke dalam Capsule. Sehingga sekarang Player menjadi child dari Capsule.



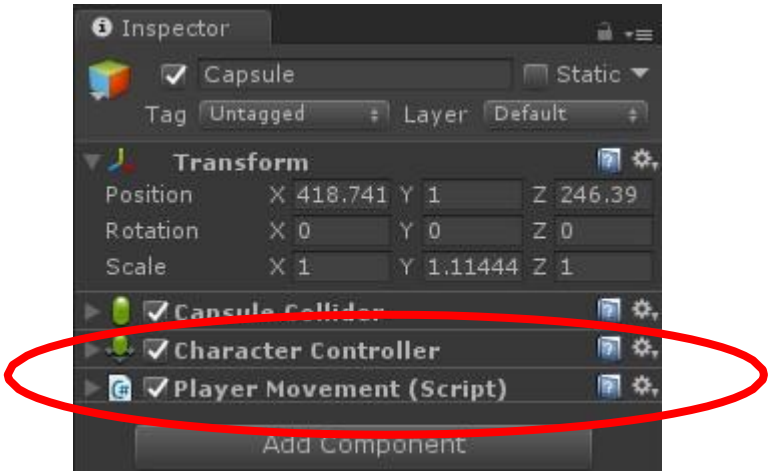
13. Ubah Tag Capsule menjadi “Player”.



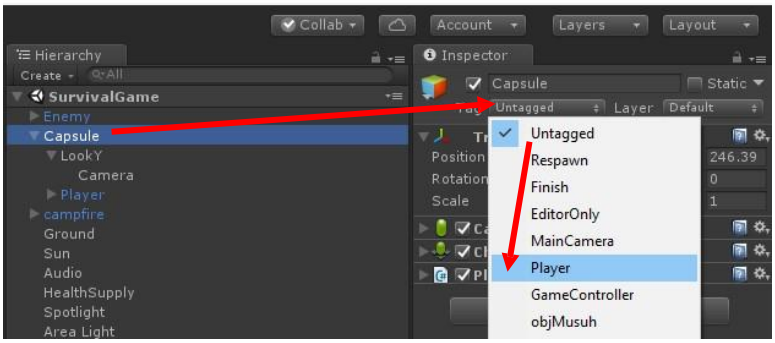
14. Buat “Empty Object” di dalam “Capsule”. Caranya klik kanan pada Capsule pilih Create Empty pindah posisi GameObject ke atas Player rename GameObject dengan nama “LookY” drag Camera ke dalam “LookY”. Sehingga struktur akan terlihat menjadi seperti berikut.



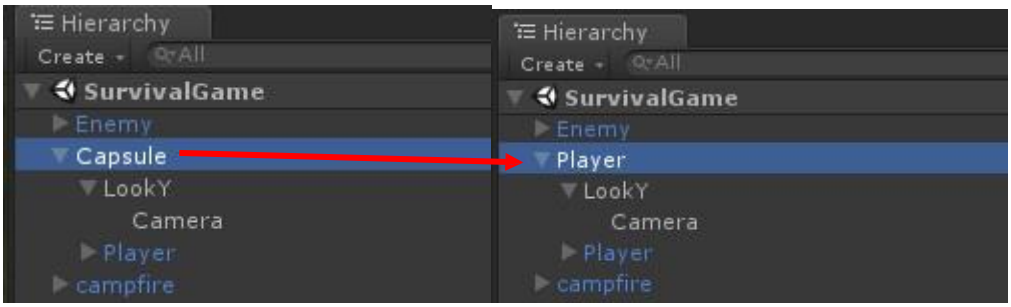
15. Selanjutnya tambahkan komponen “Character Controller” dan skrip “PlayerMovement” ke dalam Capsule yang menjadi parent dari Player.



16. Ubah Tag Capsule yang menjadi parent dari Player dari “untagged” menjadi “Player”.



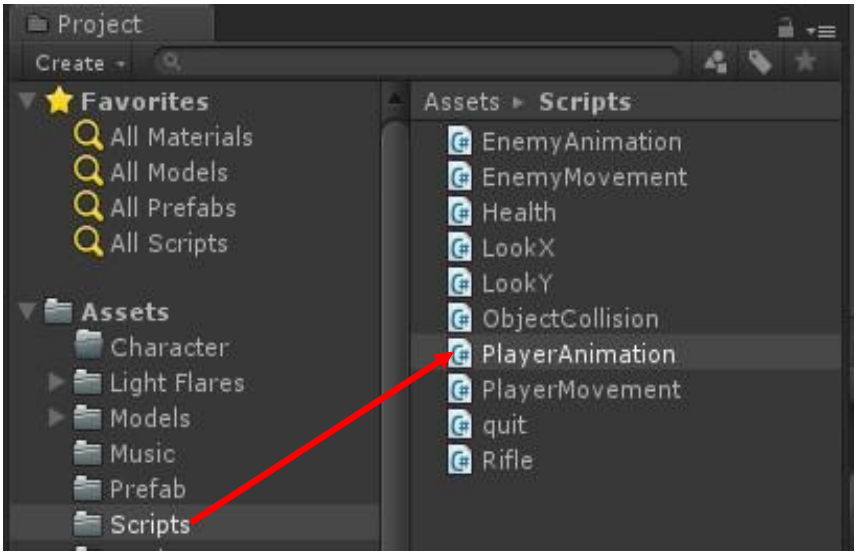
17. Rename objek “Capsule” dengan nama “Player”



E. Membuat Animasi Player

Untuk membuat karakter menjadi lebih hidup, kita dapat menambahkan gerakan animasi. Untuk melakukannya, ikuti langkah-langkah berikut:

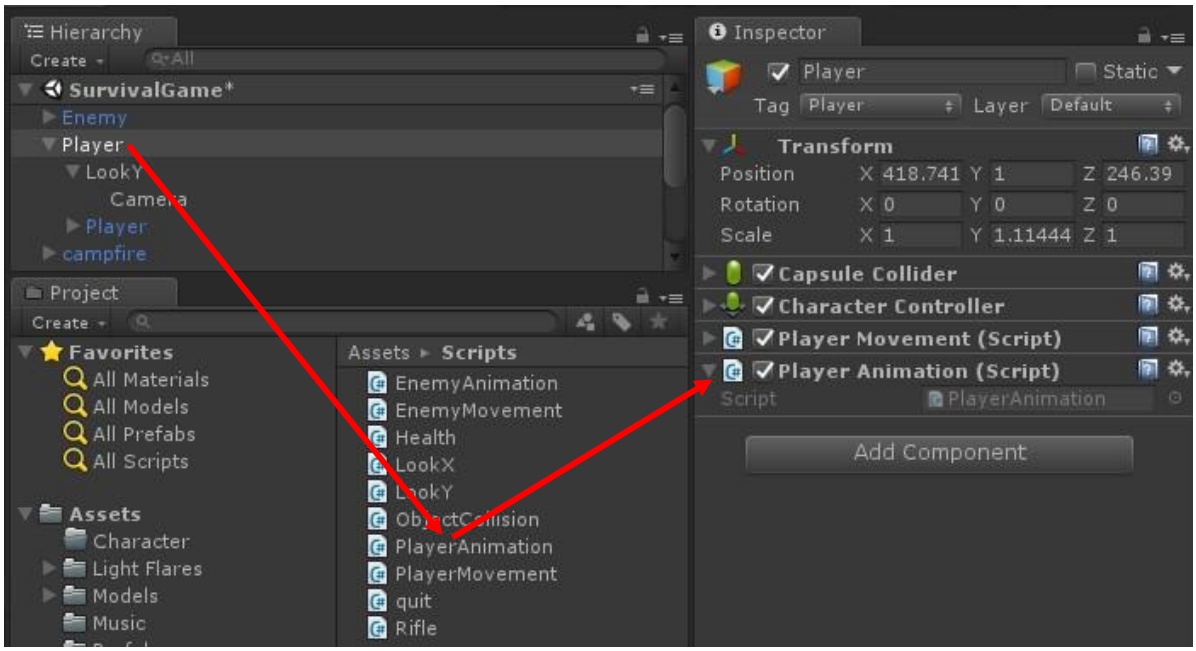
1. Buat skrip baru dengan nama “PlayerAnimation”.



2. Kemudian ubah skrip “PlayerAnimation” menjadi seperti berikut:

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

public class PlayerAnimation : MonoBehaviour {
    Animation _animation;

    void Start () {
        _animation = GetComponentInChildren<Animation> ();
        _animation.SyncLayer (0);
    }

   void Update () {
       float v = Input.GetAxisRaw ("Vertical");
       float h = Input.GetAxisRaw ("Horizontal");
       bool runningForward = v > 0;
       bool runningBackward = v < 0;
       bool notRunningForwardOrBackward = v == 0;
       bool runningForwardOrStanding = v >= 0;
       bool runningLeft = h < 0;
       bool runningRight = h > 0;
       bool notRunningLeftOrRight = h == 0;
       bool notRunningAtAll = notRunningForwardOrBackward && notRunningLeftOrRight;

       if (runningForward) {
           _animation.Blend ("RunForward");
           _animation.Blend ("Idle", 0);
           _animation.Blend ("RunBackward", 0);
       } else if (runningBackward) {
           _animation.Blend ("RunBackward");
           _animation.Blend ("Idle", 0);
           _animation.Blend ("RunForward", 0);
       } else if (notRunningForwardOrBackward) {
           _animation.Blend ("RunForward", 0);
           _animation.Blend ("RunBackward", 0);
       }

       if ((runningForwardOrStanding && runningRight) || (runningBackward && runningLeft)) {
           _animation.Blend ("RunRight");
           _animation.Blend ("Idle", 0);
           _animation.Blend ("RunLeft", 0);
       } else if ((runningForwardOrStanding && runningLeft) || (runningBackward && runningRight)) {
           _animation.Blend ("RunLeft");
           _animation.Blend ("Idle", 0);
           _animation.Blend ("RunRight", 0);
       } else if (notRunningLeftOrRight) {
           _animation.Blend ("RunLeft", 0);
           _animation.Blend ("RunRight", 0);
       }

       if (notRunningAtAll) {
           _animation.Blend ("Idle");
       }
   }
}
```

3. Selanjutnya tambahkan skrip “PlayerAnimation” ke dalam objek game “Player” (player parent).



4. Jalankan Game.