

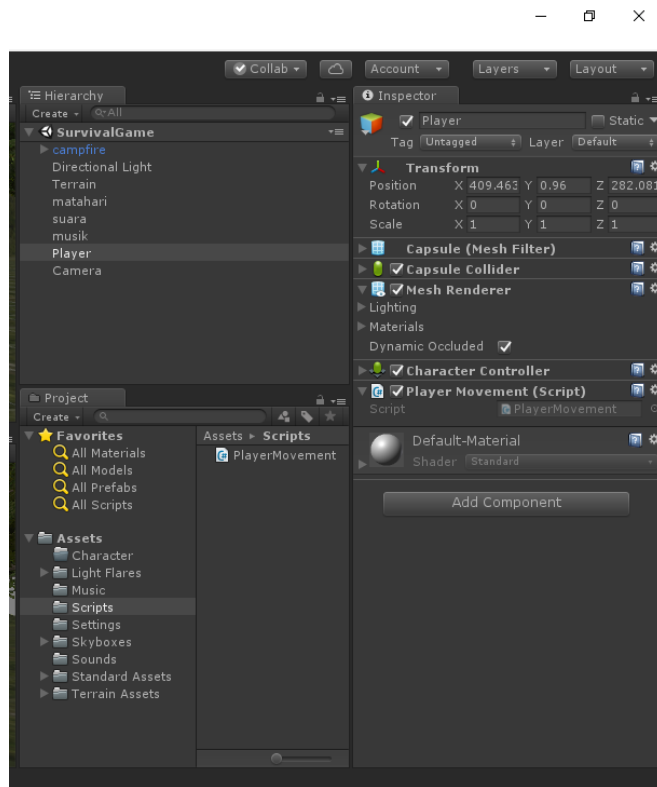
## SKRIP DAN KARAKTER

Pada bab ini akan dijelaskan skrip yang digunakan Unity serta langkah pembuatan karakter. Seperti pada pembuatan game pada umumnya yang tidak lepas dari pembuatan skrip. Pada Unity, kita juga membutuhkan skrip untuk mengatur sistem dalam game, termasuk proses pembuatan karakter. Karakter yang akan dibuat pada bab ini merupakan karakter utama dan karakter musuh.

### 6.1. Membuat karakter

Karakter merupakan pemeran utama dalam game. Langkah pertama untuk membuat karakter adalah membuat game objek dari karakter. Untuk membuatnya ikuti langkah berikut:

1. Klik menu **GameObject** → **3D Object** → **Capsule**.
2. Rename objek dengan nama “**Player**”.
3. Klik pada objek **Player** → klik menu **Component** → **Physics** → **Character Controller**.
4. Atur posisi kamera agar berada di belakang karakter dan dapat melihat karakter dengan baik, selanjutnya **Drag kamera** ke dalam karakter.
5. Buat folder baru di dalam Assets dengan nama “**Scripts**”, kemudian buat **C# Script** dan rename dengan nama “**PlayerMovement**”.
6. Klik **Player** kemudian **Drag** script **PlayerMovement** ke dalam karakter (**Player**).
7. Jika sudah selesai, maka struktur project dan hirarki akan terlihat seperti gambar berikut:



### 6.1.1. Membuat isi skrip

Setelah skrip ditambahkan ke dalam objek game, berikutnya kita buat skrip untuk menjalankan karakter dengan langkah sebagai berikut:

1. Buka script **PlayerMovement** (tunggu hingga **Visual Studio** atau **MonoDevelop** terbuka).
2. Edit kode program hingga menjadi seperti berikut:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerMovement : MonoBehaviour {
6      CharacterController _controller;
7
8      // Use this for initialization
9      void Start () {
10         _controller = GetComponent<CharacterController>();
11     }
12
13     // Update is called once per frame
14     void Update () {
15         Vector3 direction = new Vector3(Input.GetAxis("Horizontal"),0,Input.GetAxis("Vertical"));
16         _controller.Move(direction);
17     }
18 }
```

### 6.1.2. Modifikasi isi skrip

Kita dapat melakukan modifikasi pada skrip yang telah kita buat. Pada step ini, kita akan membuat pengaturan frame dan mengatur kecepatan gerak serta membuat karakter dapat melompat. Pengaturan frame bertujuan untuk mengatur kecepatan game berjalan sesuai dengan kemampuan komputer. Jika komputer anda memiliki spesifikasi yang minimum, maka game akan menyesuaikan kemampuan komputer sehingga game dapat tetap berjalan dengan normal. Untuk mengaturnya, kita dapat menggunakan perintah “**Time.deltaTime**” Langkahnya adalah sebagai berikut:

1. Tambahkan kode dengan memodifikasi skrip sehingga menjadi kode seperti di bawah ini:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerMovement : MonoBehaviour {
6      CharacterController _controller;
7
8      [SerializeField]
9      float _moveSpeed = 5.0f;
10
11     // Use this for initialization
12     void Start () {
13         _controller = GetComponent<CharacterController>();
14     }
15
16     // Update is called once per frame
17     void Update () {
18         Vector3 direction = new Vector3(Input.GetAxis("Horizontal"),0,Input.GetAxis("Vertical"));
19         Vector3 velocity = direction * _moveSpeed;
20         _controller.Move(velocity * Time.deltaTime);
21     }
22 }
```

2. Selanjutnya kita buat karakter dapat melakukan lompatan ketika kita tekan **space** pada keyboard. Untuk melakukannya, modifikasi kode program sehingga menjadi seperti berikut ini:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerMovement : MonoBehaviour {
6      CharacterController _controller;
7
8      [SerializeField]
9      float _moveSpeed = 5.0f;
10
11     [SerializeField]
12     float _jumpSpeed = 20.0f;
13
14     [SerializeField]
15     float _gravity = 1.0f;
16
17     float _yVelocity = 0.0f;
18
19     // Use this for initialization
20     void Start () {
21         _controller = GetComponent<CharacterController>();
22     }
23
24     // Update is called once per frame
25     void Update () {
26         Vector3 direction = new Vector3(Input.GetAxis("Horizontal"),0,Input.GetAxis("Vertical"));
27         Vector3 velocity = direction * _moveSpeed;
28         if (_controller.isGrounded) {
29             if (Input.GetButtonDown("Jump")) {
30                 _yVelocity = _jumpSpeed;
31             }
32         } else {
33             _yVelocity -= _gravity;
34         }
35         velocity.y = _yVelocity;
36         _controller.Move(velocity * Time.deltaTime);
37     }
38 }
```

Keterangan:

- A. **\_jumpSpeed**. Digunakan untuk mengatur tinggi rendahnya lompatan.
- B. **\_gravity**. Digunakan untuk membuat karakter kembali ke tanah.
- C. **\_yVelocity**. Digunakan untuk mengatur perubahan lompatan dan jatuhnya karakter.
- D. Pada baris ke-28, **isGrounded** bekerja saat karakter berada di permukaan tanah.
- E. Pada baris ke-29, **GetInputDown** digunakan untuk mengecek saat tombol ditekan. Jika yang ditekan adalah **space**, maka **\_yVelocity** secara otomatis akan aktif dan membuat karakter loncat.

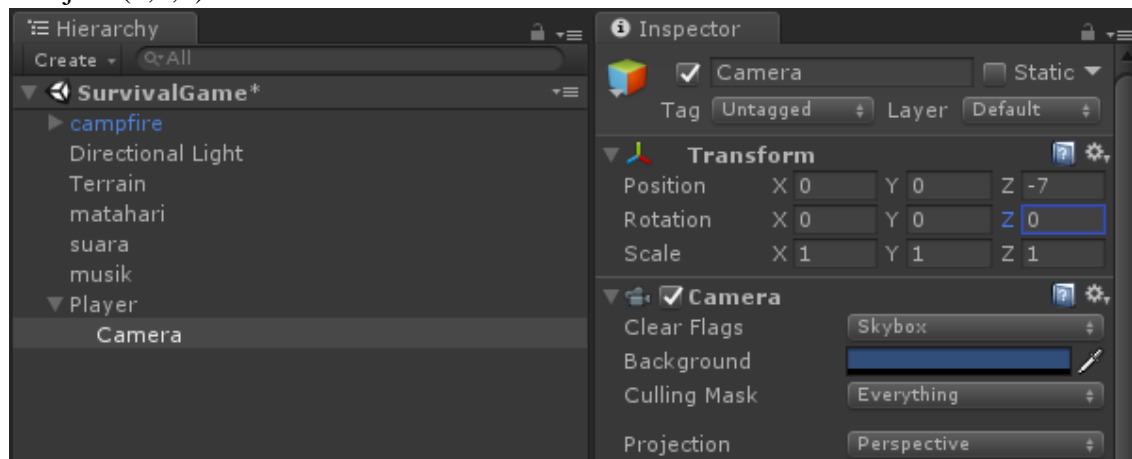
## 6.2. Modifikasi sistem kamera

Pada langkah sebelumnya, kita telah membuat sistem kamera. Selanjutnya kita lakukan modifikasi sistem kamera tersebut agar dapat melihat lingkungan yang lebih luas. Hasil dari modifikasi ini akan membuat kamera dapat melihat secara horizontal dan vertical. Berikut langkah untuk memodifikasi kamera.

1. Buat skrip baru (letakkan di dalam folder **Scripts**).
2. Rename skrip menjadi **“LookX”** kemudian edit kode sehingga menjadi seperti berikut:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class LookX : MonoBehaviour {
6      [SerializeField]
7      float _mouseX = 0.0f;
8      // Use this for initialization
9      void Start () {
10
11      }
12
13      // Update is called once per frame
14      void Update () {
15          _mouseX = Input.GetAxis("Mouse X");
16      }
17  }
```

3. Klik objek kamera, kemudian **Drag** skrip **LookX** ke dalam inspector objek kamera yang ada di kotak hirarki. Kemudian buat posisi kamera menjadi (0,0,-7) dan rotasi kamera menjadi (0,0,0).



4. Jalankan game, kemudian klik mouse pada sembarang area, maka anda akan melihat nilai Mouse X akan berubah.

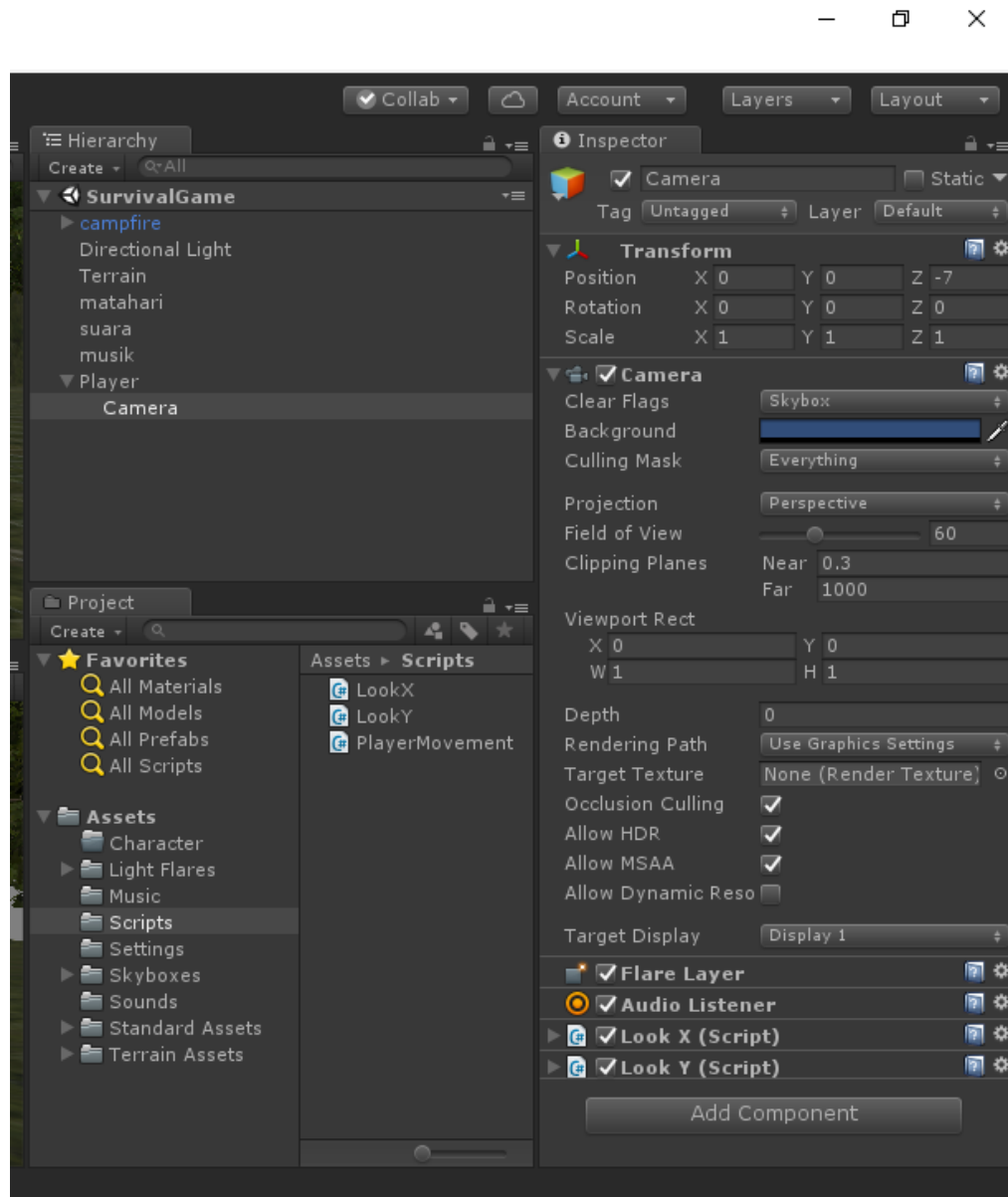
5. Ubah kode program pada **LookX** menjadi seperti berikut:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class LookX : MonoBehaviour {
6      [SerializeField]
7      float _mouseX = 0.0f;
8      // Use this for initialization
9      void Start () {
10
11      }
12
13      // Update is called once per frame
14      void Update () {
15          _mouseX = Input.GetAxis("Mouse X");
16          Vector3 rot = transform.localEulerAngles;
17          rot.y += _mouseX;
18          transform.localEulerAngles = rot;
19      }
20 }
```

6. Klik **Camera** pada hirarki, kemudian **Drag** skrip **LookX** ke dalam inspector **Camera**.
7. Buat skrip baru dengan nama "**LookY**" kemudian edit kode program menjadi seperti berikut:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class LookY : MonoBehaviour {
6      [SerializeField]
7      float _sensitivity = 5.0f;
8
9      float _mouseY = 0.0f;
10     // Use this for initialization
11     void Start()
12     {
13
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19         _mouseY = Input.GetAxis("Mouse Y");
20         Vector3 rot = transform.localEulerAngles;
21         rot.x += _mouseY * _sensitivity;
22         transform.localEulerAngles = rot;
23     }
24 }
```

8. Klik kamera pada hirarki, kemudian **drag LookY** ke dalam inspector **Camera**. Sehingga struktur proyek akan terlihat seperti gambar berikut:



9. Jika sudah, jalankan game.

### 6.3. Skrip dasar musuh

Setelah selesai membuat kaakte utama, selanjunya kita akan membuat musuh dan menambahkan skrip pada musuh. Perbedaan skrip yang ada di karakter utama dan skrip pada musuh adalah skrip pada musuh pergerakan objek musuh secara otomatis.

### 6.3.1. Membuat Objek Musuh

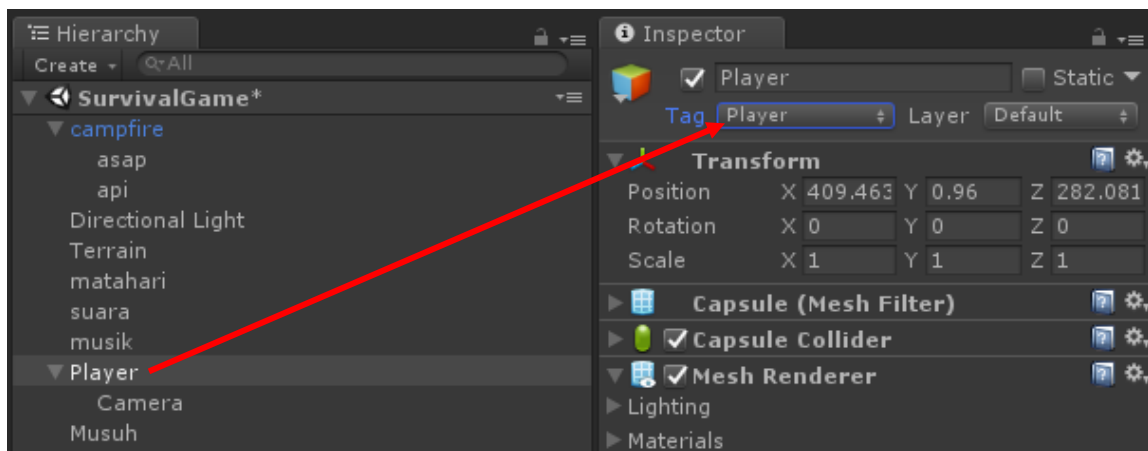
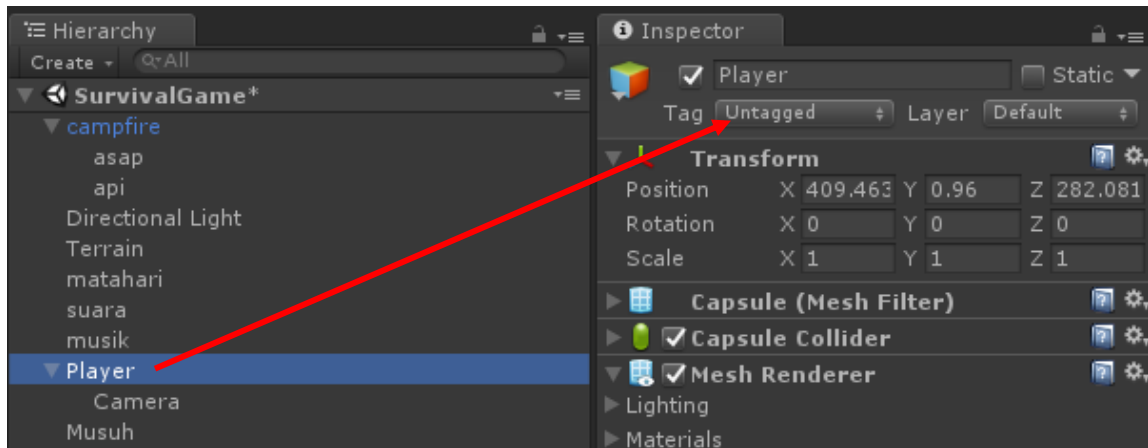
Langkah pembuatan objek musuh adalah sebagai berikut:

1. Klik menu **GameObject** → **3D Object** → **Sphere**
2. Rename dengan nama “**Musuh**”
3. Tambahkan komponen pengendali karakter dengan cara klik pada objek **Musuh** → kemudian klik menu **Component** → **Physics** → **Character Controller**.

### 6.3.2. Menambahkan Skrip pada Objek Musuh

Sebelum menambahkan skrip pada objek musuh, pertama-tama kita harus memberikan tanda **Tag** pada objek atau karakter utama (**Player**). Cara memberikan **Tag** adalah sebagai berikut:

1. Klik pada objek **Player**.
2. Pada kotak Inspector ubah **Tag** dari **Untagged** menjadi **Player**.

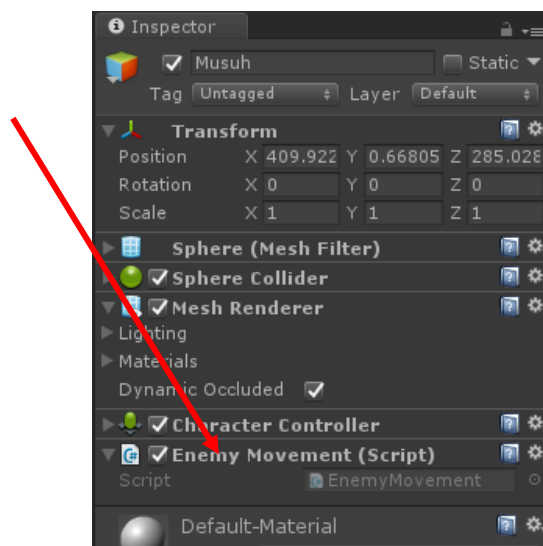


Setelah membuat objek musuh dan memberikan **Tag** Player, berikutnya tambahkan skrip pada musuh dengan cara sebagai berikut:

1. Buka folder **Scripts** pada Assets, kemudian buat skrip (**C# Script**) baru dengan nama **“EnemyMovement”**.
2. Edit skrip menjadi seperti 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      // Use this for initialization
10     void Start () {
11         GameObject playerObject = GameObject.FindGameObjectWithTag("Player");
12         _player = playerObject.transform;
13         _controller = GetComponent<CharacterController>();
14     }
15
16     // Update is called once per frame
17     void Update () {
18         Vector3 direction = _player.position - transform.position;
19         _controller.Move(direction * Time.deltaTime);
20     }
21 }
```

3. Kode di atas memberikan perintah kepada musuh untuk menjadi karakter yang memiliki Tag **Player** (karakter utama). Selanjutnya, musuh akan selalu mengikuti pergerakan karakter. Silahkan jalankan game, kemudian move karakter, maka anda akan melihat musuh akan otomatis mencari karakter.
4. Klik objek musuh **“Musuh”** kemudian **drag** skrip **EnemyMovement** ke inspector **Musuh**.



5. Jalankan Game.



### 6.3.3. Mengatur Kecepatan Gerrak Musuh

Jika anda jalankan game, maka pergerakan musuh akan sangat cepat. Hal ini akan mengakibatkan karakter mengalami kesulitan dalam mengalahkan musuh (kecepatan musuh yang seperti ini biasanya digunakan pada game level yang tinggi). Untuk itu perlu dilakukan modifikasi kecepatan musuh pada skip. Untuk mengatur kecepatan musuh, ikuti langkah berikut:

1. Buka skrip **EnemyMovement**, kemudian modifikasi menjadi seperti kode di bawah ini:

```
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 = 5.0f;
11
12     // Use this for initialization
13     void Start () {
14         GameObject playerObject = GameObject.FindGameObjectWithTag("Player");
15         _player = playerObject.transform;
16         _controller = GetComponent<CharacterController>();
17     }
18
19     // Update is called once per frame
20     void Update () {
21         Vector3 direction = _player.position - transform.position;
22
23         direction.Normalize();
24         Vector3 velocity = direction * _moveSpeed;
25         _controller.Move(velocity * Time.deltaTime);
26     }
27 }
```

2. Dengan kode di atas, maka anda dapat menyesuaikan kecepatan musuh (**misal: 0.6**) pada kotak inspector, lihat gambar berikut:

