



TUGAS PERTEMUAN: 9

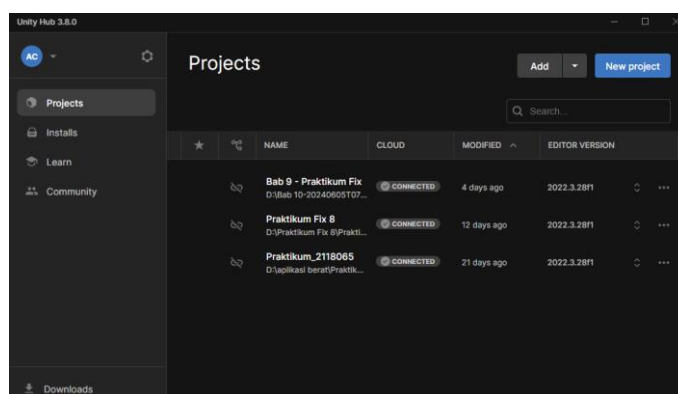
ANIMATION

NIM	:	2118122
Nama	:	Egia Suranta Perangin-Angin
Kelas	:	D
Asisten Lab	:	Berchmans Bayu Bin Jaya (2218034)
Referensi	:	https://assetstore.unity.com/packages/2d/characters/pixel-adventure-1-155360

9.1 Tugas 1 : ANIMATION

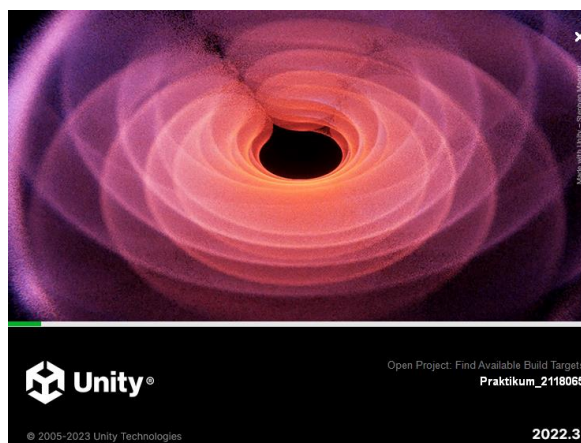
A. Animation

1. Masuk ke dalam aplikasi unity



Gambar 9.1 Membuka Unity

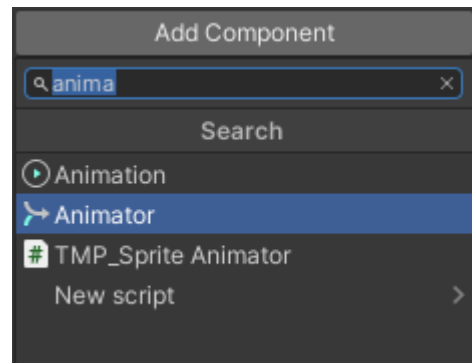
2. Pilih folder yang ingin dibuka atau yang sudah diunduh property



Gambar 9.2 me buka file unity

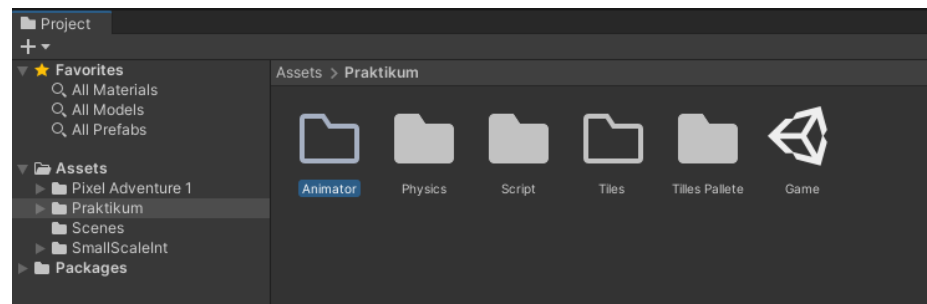


3. Pada hirarki player tambahkan komponen animator



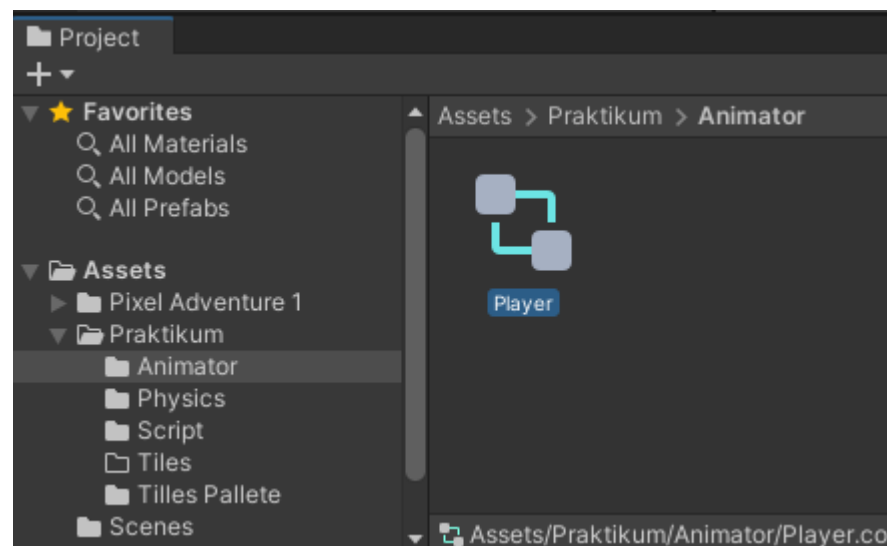
Gambar 9.3 Menambahkan komponen animator

4. Buat folder baru Bernama Animator dalam folder praktikum



Gambar 9.4 menambahkan folder animator

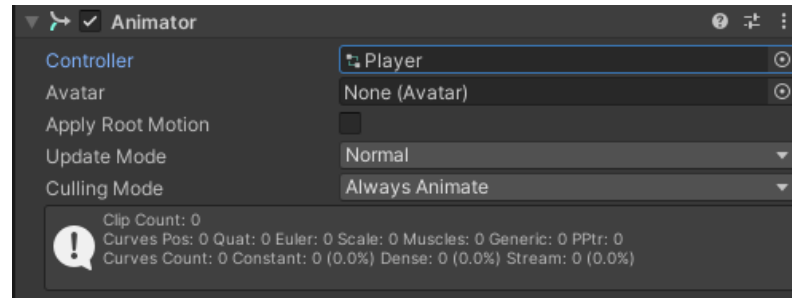
5. Membuat animator kontroler pada folder animator dgn nama player



Gambar 9.5 menambahkan kontrollerr

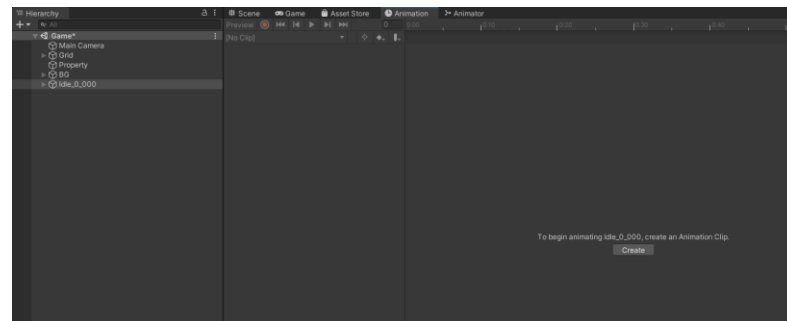


6. Pada komponen hirarki player, ubah target controller ke player



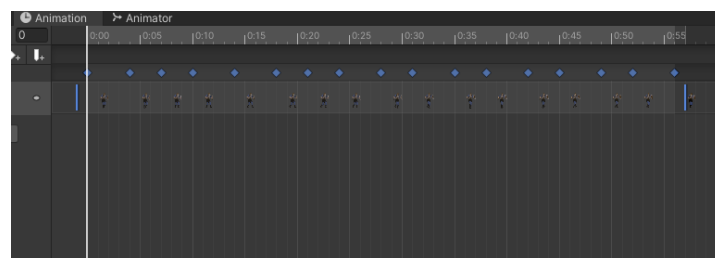
Gambar 9.6 mengubah controller ke player

7. Untuk membuat animasi klik karakter pada Hierarchy, kemudian ke menu panel Animation, pilih Create



Gambar 9.7 menambahkan animation dan animator

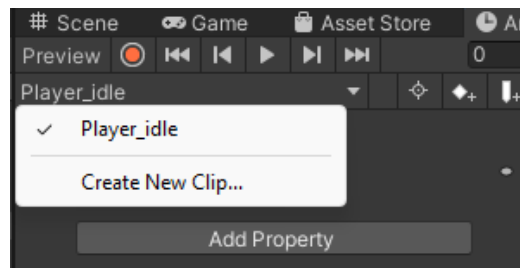
8. Pada menu Project buka folder player lalu pilih Idle dan pilih gambar player-idle-1, player-idle-3 dan player-idle-4, dan seterusnya kemudian drag ke tab Animation. Tekan CTRL + A pada menu **panel Animation** geser kotak kecil pada timeline sampai frame 0:30 agar animasinya tidak terlalu cepat



Gambar 9.8 penerapan movement

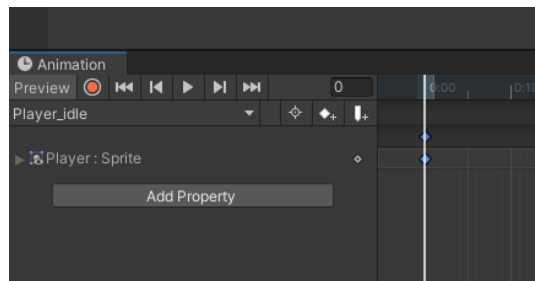


9. Buat animasi baru, Klik pada “Player_idle” kemudian pilih Create New Clip, dan beri nama “Player_run”, Simpan pada Folder Animator



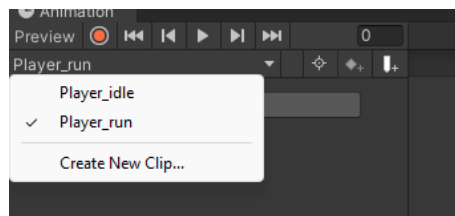
Gambar 9.9 Buat animasi

10. Pada grid property asset ambil animasi karakter idle lalu drag and drop pada animation



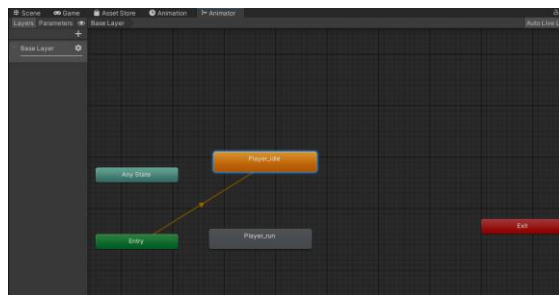
Gambar 9.10 Menambahkan idle animation

11. Buka menu Project kemudian cari folder Player > run, Pilih player-run-1 sampai player-run-6, drag and drop pada menu Animation



Gambar 9.11 Membuka player run

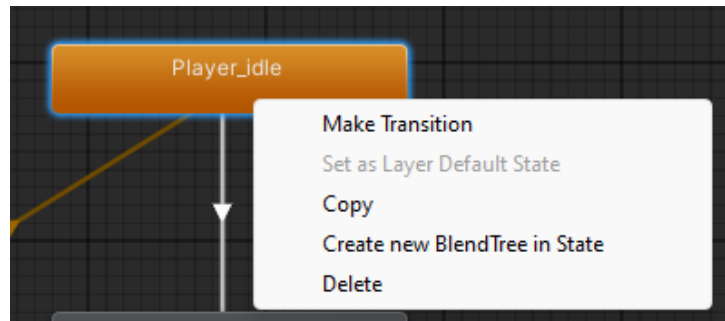
12. pilih ke menu Animator yang telah dibuka sebelumnya dan akan tampil seperti berikut



Gambar 9.12 Menu animator

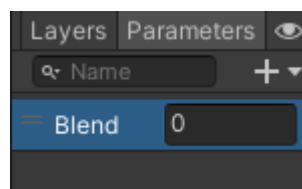


13. Kemudian buat transisi antara player_idle dan player_run dengan cara klik kanan pada player_idle dan pilih Make Transition dan tarik ke player_run



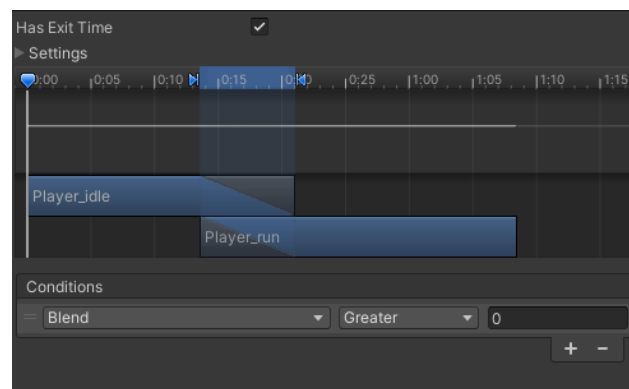
Gambar 9.13 Menambahkan make transition

14. Masuk ke tab parameter, tambahkan tipe data bdengan cara tekan icon tambah dan ubah namanya menjadi “Blend”



Gambar 9.14 Membuat blend tipe data float

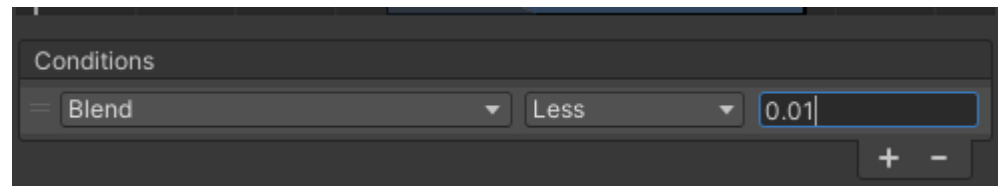
15. Klik panah putih tersebut, pada bagian conditions klik icon tambah kemudian atur menjadi “Blend”



Gambar 9.15 Menambahkan conditions

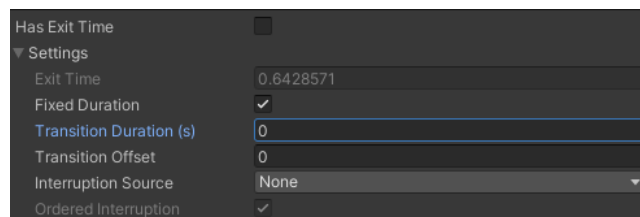


16. Atur nilai conditions blend tersebut menjadi 0.01



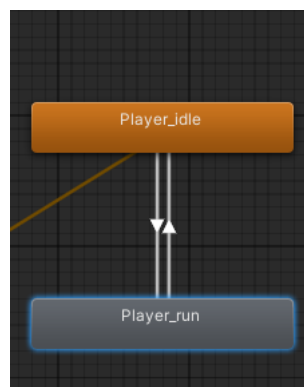
Gambar 9.16 Mengatur nilai condition

17. Pada Pada bagian Settings, hilangkan centang pada Has Exit Time dan atur nilai Transition Duration menjadi 0



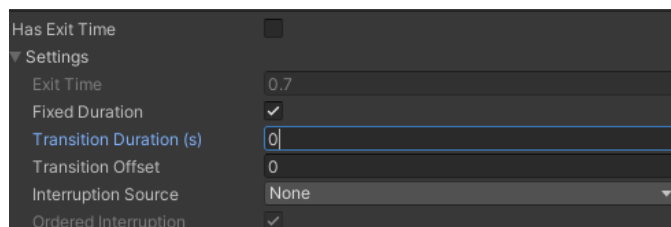
Gambar 9.17 Ubah bagian setting transition

18. Buat transisi juga dari player_run ke player_idle dengan cara klik kanan pada player_run dan pilih Make Transition.



Gambar 9.18 Membuat transisi player_run ke player_idle

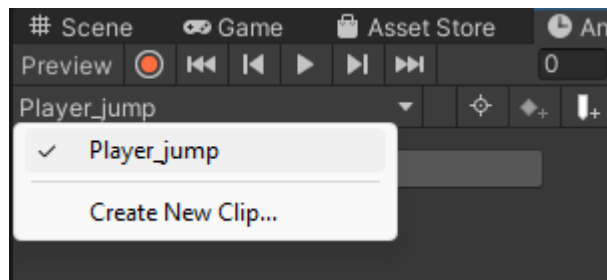
19. Pada bagian Settings, hilangkan centang pada Has Exit Time dan atur nilai Transition Duration menjadi 0



Gambar 9.19 Tampilan settingan

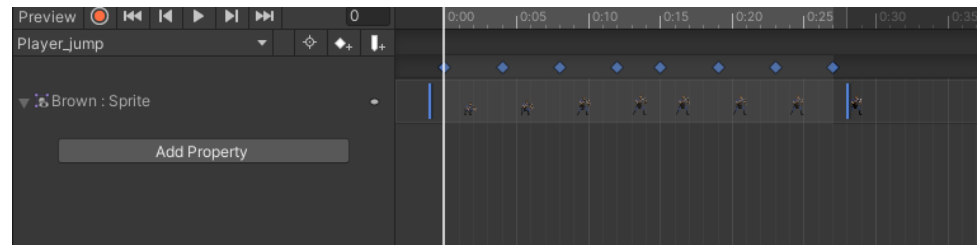


20. Kemudian buat animasi baru tekan tulisan “Player_run” kemudian pilih Create New Clip, dan beri nama “Player_jump”



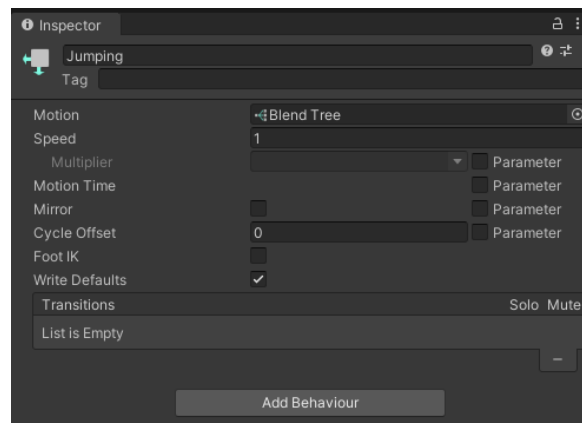
Gambar 9.20 Membuat player_jump

21. Pada folder player buka jump lalu pilih gambar player-jump-1, kemudian drag ke tab Animation.



Gambar 9.21 Drag gambar player jump

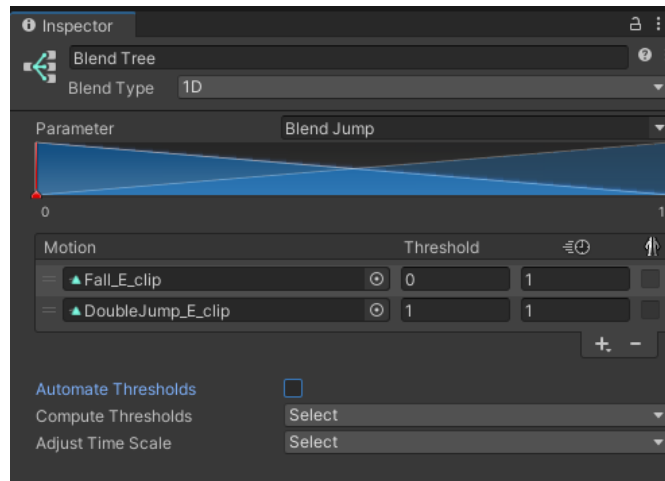
22. Pada Animator klik Blend Tree, di menu Inspector, ubah namanya menjadi Jumpig



Gambar 9.22 Mengubah nama menjadi jumping

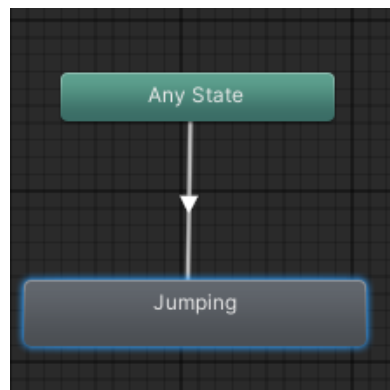


23. Klik 2X Blend Tree “Jumping”, pada inspector ubah parameter menjadi “Blend Jump”



Gambar 9.23 Mengubah parameter menjadi blend jump

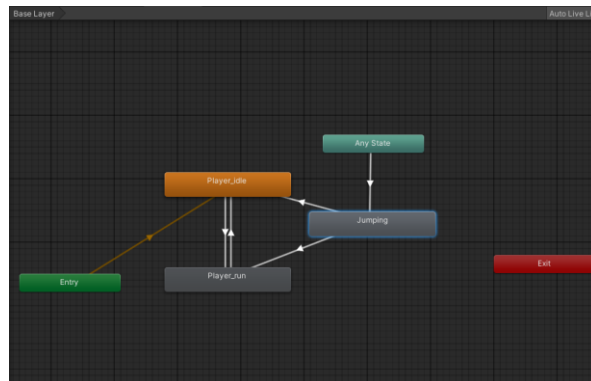
24. Kembali ke Base Layer, klik kanan Any State, pilih Make Transition dan arahkan panahnya ke Jumping



Gambar 9.24 Make Transition any state ke jumping

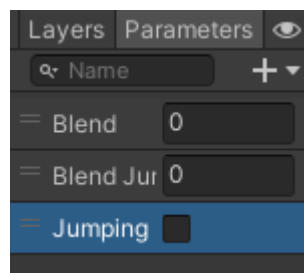


25. Klik kanan Jumping, pilih Make Transition dan arahkan panahnya ke Player_idle dan Player_run



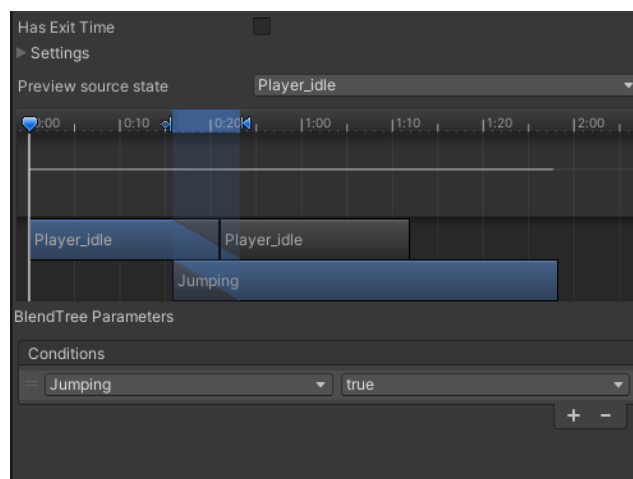
Gambar 9.25 Make Transition jumping ke player_idle dan player_run

26. Tambahkan parameter transisi dengan tipe data Bool tekan icon + dan ubah namanya menjadi “Jumping”



Gambar 9.26 Menambahkan parameter jumping

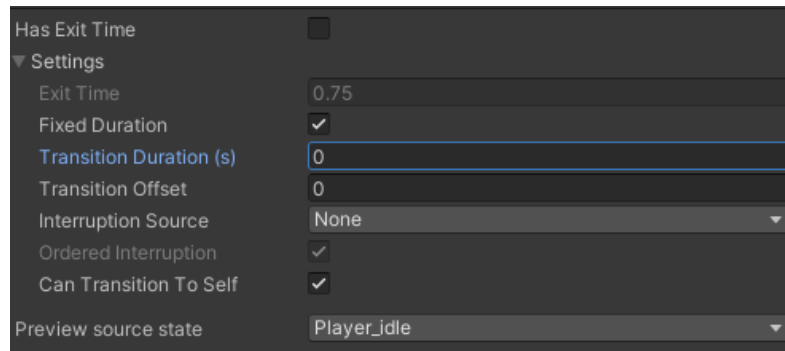
27. Klik panah yang mengarah ke Jumping, pada inspector tambahkan condition, pilih condition Jumping dan ubah nilainya menjadi true



Gambar 9.27 Mengatur condition jumping



28. Klik Settings dan ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time



Gambar 9.28 Setting jumping to idle

B. Kuis

```
void HandleJumpInput()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        animator.SetBool("isJumping", );
        rb.AddForce(Vector2.up * jumpForce,
ForceMode2D.Impulse);
    }
    else if (Input.GetKey(KeyCode.Space))
    {
        animator.SetBool("isJumping",);
    }
}

void HandleMovementInput()
{
    float move = Input.GetAxis("Horizontal");

    if (move != 1)
    {
        animator.SetBool("isIdle", true);
        transform.Translate(Vector3.left * move *
Time.deltaTime);
    }
    else
    {
        animator.SetBool("isWalking", false);
    }

    if (move != 0)
    {
        transform.localScale = new Vector3(-4, 1, 1);
    }
    else if (move > 0)
    {
        transform.localScale = new Vector3(1, 2, 1);
    }
}
```



Penjelasan :

1. Dalam fungsi HandleJumpInput:

- `animator.SetBool("isJumping",);` harus memiliki nilai boolean kedua, seperti true atau false.
- `else if (Input.GetKey(KeyCode.Space))` seharusnya `else if (Input.GetKeyUp(KeyCode.Space))` untuk mendeteksi ketika tombol space dilepas.

2. Dalam fungsi HandleMovementInput:

- `if (move != 1)` seharusnya `if (move != 0)` untuk mengecek apakah ada input gerakan.
- `animator.SetBool("isWalking", false);` seharusnya `animator.SetBool("isWalking", true);` ketika karakter bergerak.
- `transform.Translate(Vector3.left * move * Time.deltaTime);` menggunakan `Vector3.left` yang tidak perlu dikalikan dengan `move` karena sudah merepresentasikan arah ke kiri. Sebaiknya gunakan `Vector3.right` dan kalikan dengan `move` untuk mendapatkan arah yang benar.
- Blok kode yang mengatur `transform.localScale`:
- `if (move != 0)` dan `else if (move > 0)` saling bertentangan karena `move > 0` adalah subset dari `move != 0`. Ini harus diatur ulang untuk memastikan skala karakter berubah sesuai dengan arah gerakan.