



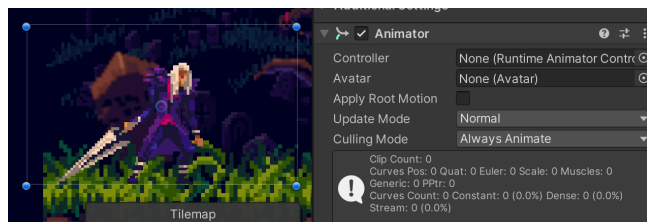
TUGAS PERTEMUAN: 9

Game Animation

NIM	:	Miftakhussurur Al Maliki
Nama	:	2118103
Kelas	:	C
Asisten Lab	:	Rifal Rifqi Rhomadon (2218106)
Baju Adat	:	Baju Safari (Bali-Indonesia Timur)
Referensi	:	https://www.inilah.com/8-jenis-pakaian-adat-bali-yang-memiliki-makna-dan-filosofi

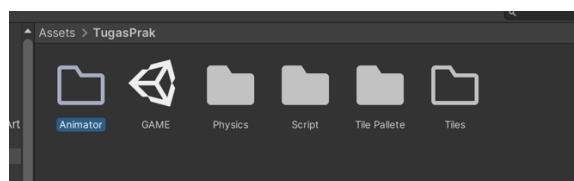
9.1 Tugas 9 : Membuat Game Animation

1. Buka file projek Unity sebelumnya pada bab 8 pada karakter klik inspector kemudian pilih Add Component Animator



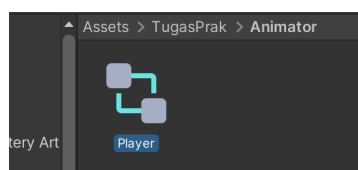
Gambar 1.1 Menambahkan component animator

2. Pada folder TugasPrak Buat Folder baru “Animator”



Gambar 1.2 Membuat folder Animator

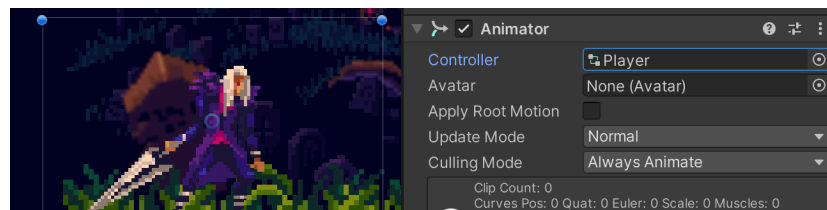
3. Buat File Animator Controller pada folder Animator , ubah namanya menjadi Player



Gambar 1.3 Membuat file animator controller

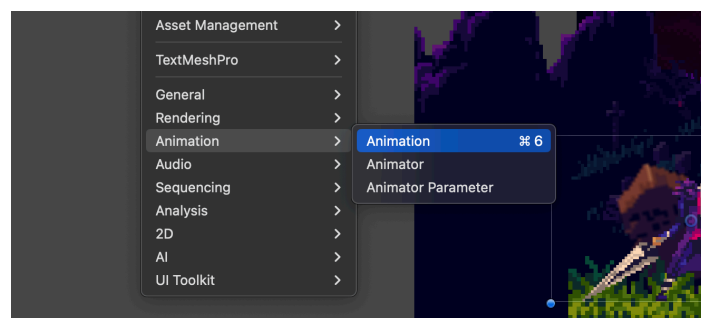


4. Klik player pada Hierarchy, kemudian cari Component Animator, pada setting Controller ubah menjadi Player



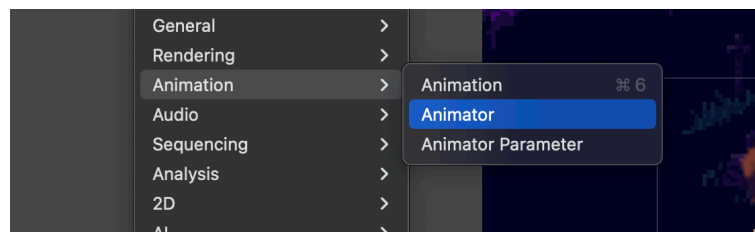
Gambar 1.4 Mengatur controller player

5. Tambahkan menu panel Animation di menu Window, pilih Animation > Animation



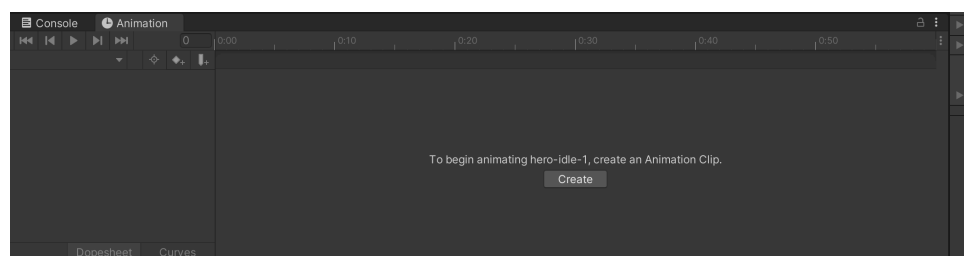
Gambar 1.5 Menampilkan panel animation

6. Tambahkan menu panel Animator



Gambar 1.6 Menampilkan panel animator

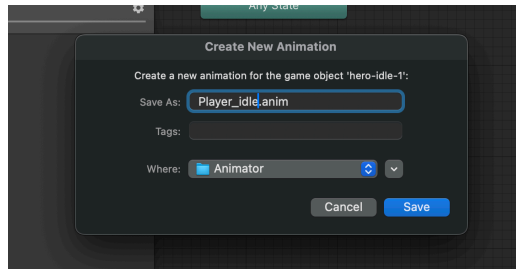
7. Untuk membuat animasi klik hero-idle-1 pada Hierarchy, kemudian ke menu panel Animation, pilih Create



Gambar 1.7 Menambah animation clip player

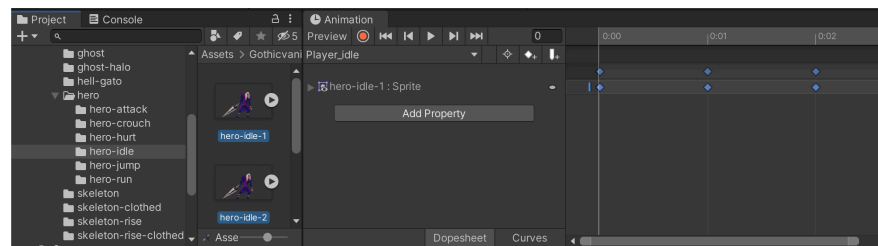


8. Simpan pada folder Animator dan beri nama “Player_idle”



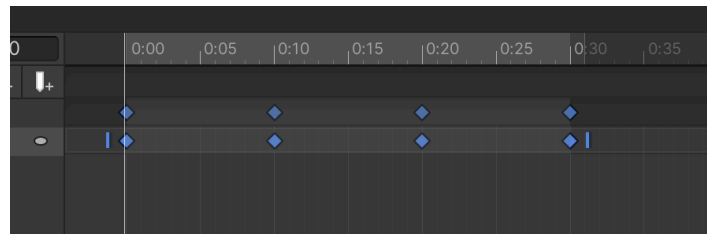
Gambar 1.8 Membuat clip player idle

9. Pada menu Project buka folder hero lalu pilih hero-idle dan pilih gambar hero-idle-1 sampai hero-idle-4, kemudian drag ke tab Animation.



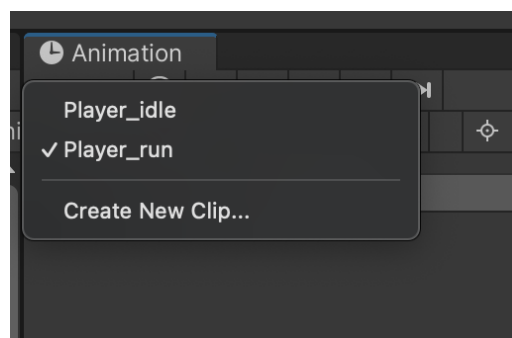
Gambar 1.9 Memasukkan assets hero idle ke timeline animation

10. Tekan Cmd+A pada menu panel Animation geser kotak kecil pada timeline sampai frame 0:30 agar animasinya tidak terlalu cepat



Gambar 1.10 Mengubah frame clip player idle

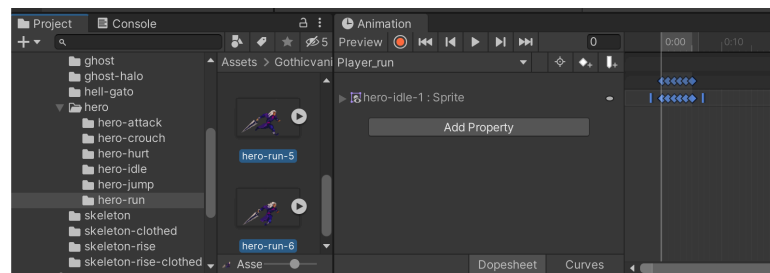
11. Buat animasi baru, Klik pada “Player_idle” kemudian pilih Create New Clip, dan beri nama “Player_run”, Simpan pada Folder Animator



Gambar 1.11 Menambah clip player run

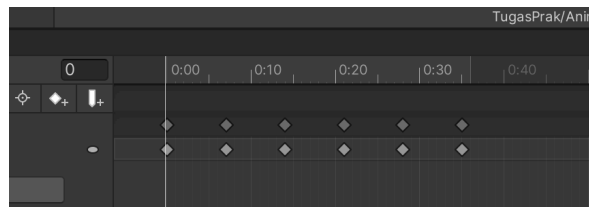


12. Buka menu Project kemudian buka folder hero lalu pilih hero-run, pilih hero-run-1 sampai hero-run-6, drag and drop pada menu Animation



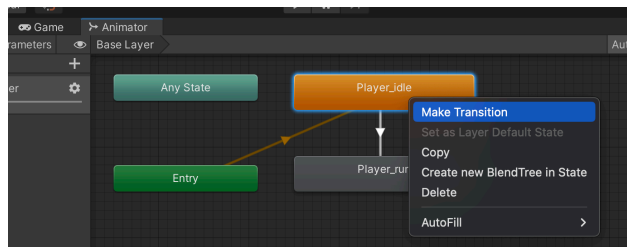
Gambar 1.12 Memasukkan assets hero run ke timeline animation

13. pada panel timeline tekan Ctrl+A di keyboard, klik bagian kotak kecil disamping keyframe terakhir dan geser sampai waktu 0:35



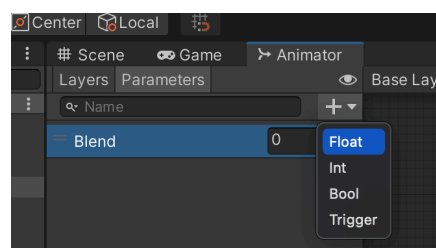
Gambar 1.13 Mengubah frame clip player run

14. Pilih ke menu Animator yang telah dibuka sebelumnya, 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 1.14 Menambahkan transisi dari player idle ke player run

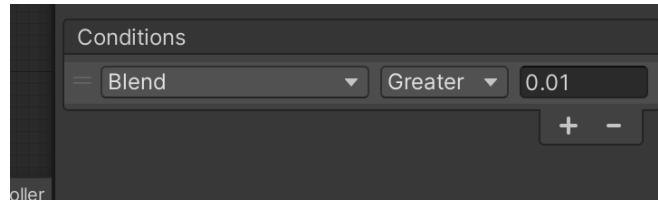
15. Masuk ke tab parameter, tambahkan tipe data dengan cara tekan icon tambah dan ubah namanya menjadi “Blend”



Gambar 1.15 Menambah parameters animator

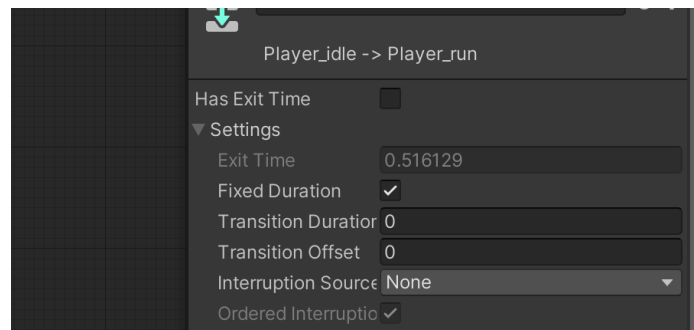


16. Klik panah putih tersebut, pada bagian conditions klik icon tambah kemudian atur menjadi “Blend”. Atur nilai conditions blend tersebut menjadi 0.01



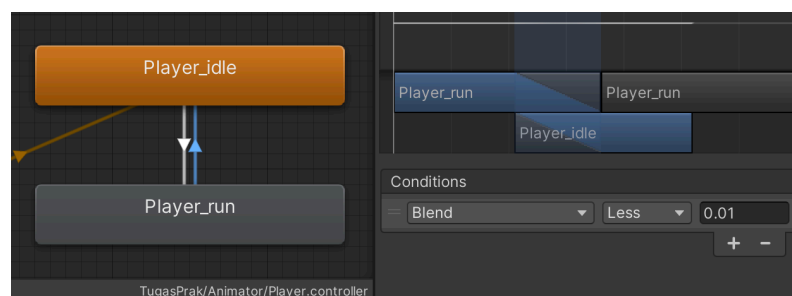
Gambar 1.16 Mengatur kondisi transisi player idle ke player run

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



Gambar 1.17 Mengatur transisi player idle ke player run

18. Buat transisi juga dari player_run ke player_idle dengan cara klik kanan pada player_run dan pilih Make Transition. pada bagian conditions klik icon tambah kemudian atur menjadi “Blend”. Atur nilai conditions blend tersebut menjadi 0.01 dan ubah dari Greater menjadi Less. Dan ulangi langkah 17.



Gambar 1.18 Menambahkan transisi dari player run ke player idle

19. Agar animasi dapat sesuai ketika berjalan, buka script Player dan tambahkan source code berikut pada class Player.



```
public class Player : MonoBehaviour
{
    public Animator animator;

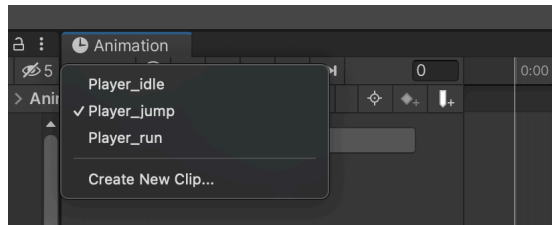
    Rigidbody2D rb;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();
    }

    void FixedUpdate()
    {
        GroundCheck();
        Move(horizontalValue, jump);
        animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
    }
}
```

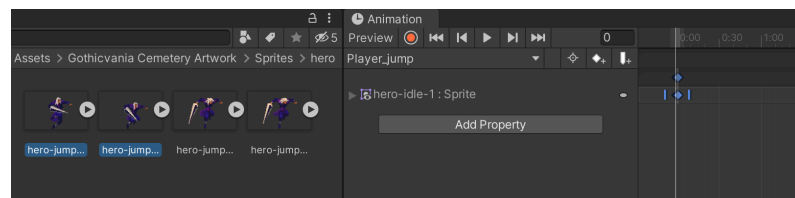
Gambar 1.19 Mengubah script player

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



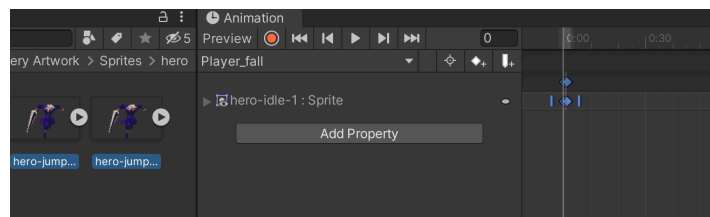
Gambar 1.20 Menambahkan clip player jump

21. Pada folder hero buka hero-jump lalu pilih gambar hero-jump-1 dan hero-jump-2, kemudian drag ke tab Animation.



Gambar 1.21 Memasukkan assets hero jump ke timeline animation

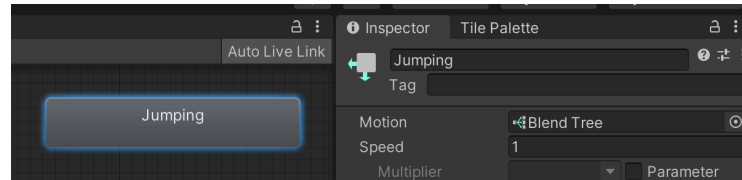
22. Buat animasi baru dengan cara tekan tulisan “Player_jump” kemudian pilih Create New Clip, dan beri nama “Player_fall”, buka folder hero buka hero-jump lalu pilih gambar hero-jump-3 dan hero-jump-4, kemudian drag ke tab Animation



Gambar 1.22 Mengubah frame clip player jump

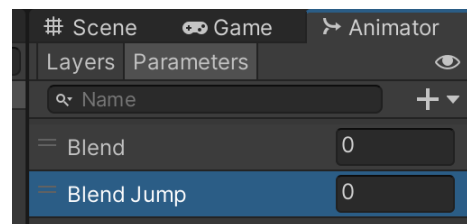


23. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong , pilih Create State>From New Blend Tree, ubah namanya menjadi Jumping



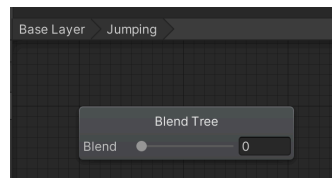
Gambar 1.23 Menambah state jumping

24. Pada menu Parameters tambahkan parameter tipe data Float tekan icon + dan ubah namanya menjadi “Blend Jump”.



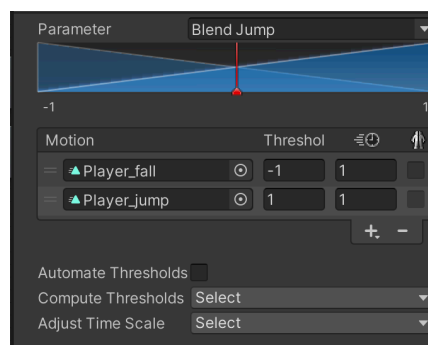
Gambar 1.24 Menambah parameter blend jump

25. Pada menu Animator, Klik dua kali pada Blend Tree “Jumping”, Tekan pada Blend Tree, pada inspector ubah parameter menjadi “Blend Jump”.



Gambar 1.25 Mengubah parameter jumping dengan blend jump

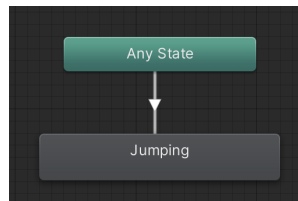
26. Buka menu Inspector, tekan icon + dan pilih Add Motion Field. Tambahkan dua Motion Field, hilangkan centang “Automate Thresholds” dan atur nilai Threshold seperti berikut



Gambar 1.26 Menambahkan motion field

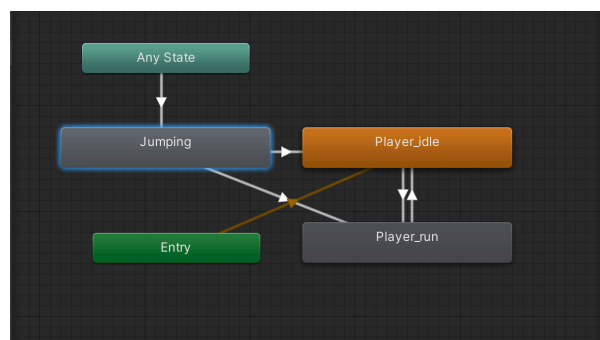


27. Kembali ke Base Layer, klik kanan Any State, pilih Make Transition dan arahkan panahnya ke Jumping



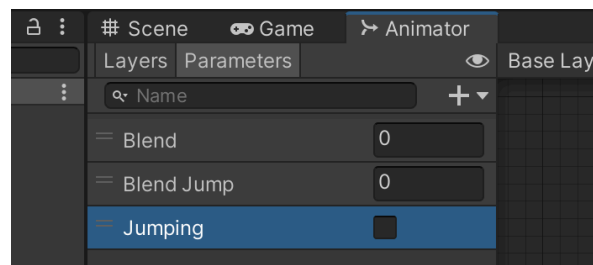
Gambar 1.27 Menambah transisi dari any state ke jumping

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



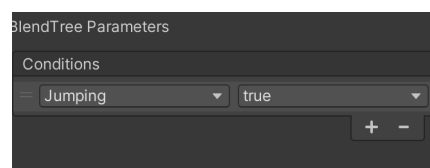
Gambar 1.28 Menambah transisi dari jumping ke player idle & run

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



Gambar 1.29 Menambah parameter jumping

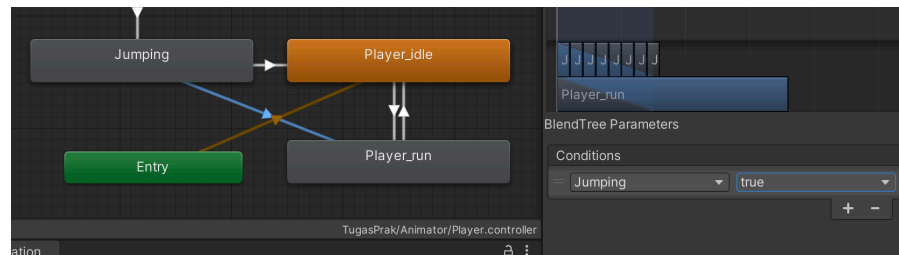
30. Klik panah yang mengarah ke Jumping, pada inspector tambahkan condition, pilih condition Jumping dan ubah nilainya menjadi true, ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time



Gambar 1.30 Merubah kondisi transisi jumping



31. Klik panah yang mengarah ke Player_idle dan Player_run, pada inspector tambahkan condition, pilih condition Jumping, pada arah panah ke player_idle ubah menjadi false, pada arah panah ke player_run ubah menjadi true, dan ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time



Gambar 1.31 Merubah kondisi transisi player idle & run

32. Buka script Player, dan tambahkan source code berikut pada fungsi update

```
void Update ()
{
    horizontalValue = Input.GetAxisRaw("Horizontal");
    if (Input.GetButtonDown("Jump")){
        animator.SetBool("Jumping", true);
        jump = true;
    }
    else if (Input.GetButtonUp("Jump"))
        jump = false;
}
```

Gambar 1.32 Menambahkan script jumping pada fungsi update()

33. Pada Fungsi FixedUpdate tambahkan seperti berikut

```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);
    animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
    animator.SetFloat("Blend Jump", rb.velocity.y);
}
```

Gambar 1.33 Menambahkan script jumping pada fungsi FixedUpdate()

34. Tambahkan baris kode seperti dibawah ini dalam method GroundCheck

```
void GroundCheck()
{
    isGrounded = false;
    Collider2D[] colliders = Physics2D.OverlapCircleAll(ground
    if (colliders.Length > 0){
        isGrounded = true;
    }
    animator.SetBool("Jumping", !isGrounded);
}
```

Gambar 1.34 Menambahkan script jumping pada fungsi GroundCheck()



35. Berikut adalah hasil rendernya



Gambar 1.35 Hasil render

9.2 Kuis

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

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

    if (move == 0)
    {
        animator.SetBool("isIdle", true);
        transform.Translate(Vector3.left * move *
Time.deltaTime);
    }else{
        animator.SetBool("isIdle", false);
        animator.SetBool("isWalking", true);
        transform.Translate(Vector3.right * move *
Time.deltaTime);
    }

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



Analisa :

Kode tersebut mengalami error karena parameter yang kosong dalam metode animator.SetBool pada kedua kondisi dalam metode HandleJumpInput. Metode SetBool memerlukan dua parameter: nama boolean dan nilai boolean (true atau false). Ketika parameter kedua dibiarkan kosong, compiler tidak dapat memahami maksud kode tersebut dan menghasilkan error. Selain itu, dalam metode HandleMovementInput, kondisi if (move != 1) dan else if (move > 0) salah penempatan sehingga menyebabkan kondisi idle dan skala transformasi yang tidak sesuai dengan arah gerakan karakter. Kesalahan-kesalahan ini mengakibatkan logika animasi dan gerakan tidak berfungsi seperti yang diharapkan.