



TUGAS PERTEMUAN: 9

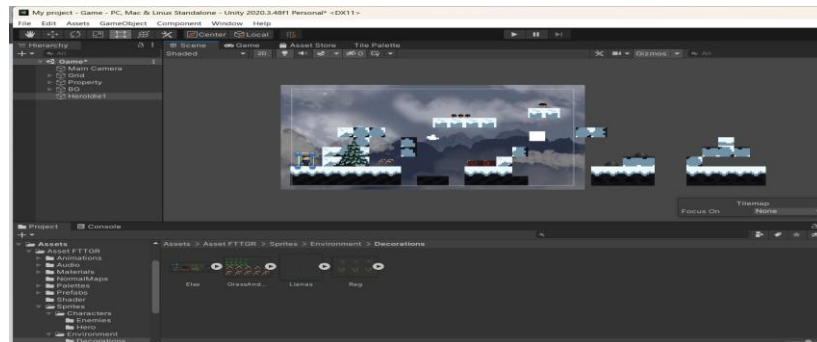
GAME ANIMATION

NIM	:	2118062
Nama	:	Nayla Dwi Salsabila
Kelas	:	A
Asisten Lab	:	Bagas Anardi Surya W (2118004)
Baju Adat	:	Baju Adat Batak Simalungun (Provinsi-Indonesia Barat)
Referensi	:	https://id.pngtree.com/freepng/traditional-clothes-of-batak-north-sumatera-indonesia_8129003.html

9.1 Tugas 9 : Membuat Animasi Game

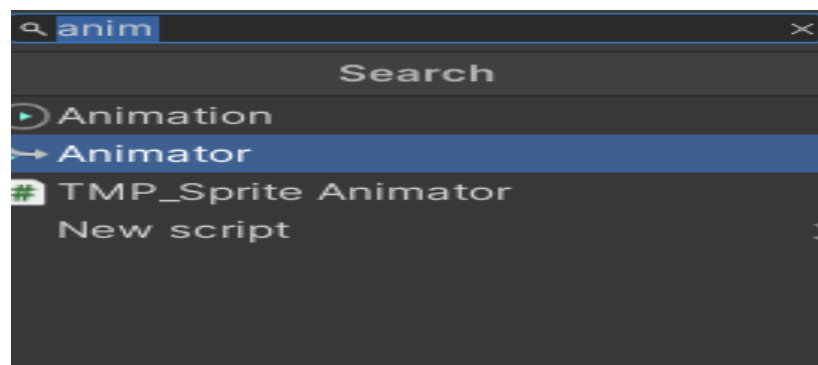
A. Membuat Animasi Game

1. Buka project unity yang sudah dibuat sebelumnya.



Gambar 1.1 Tampilan Unity

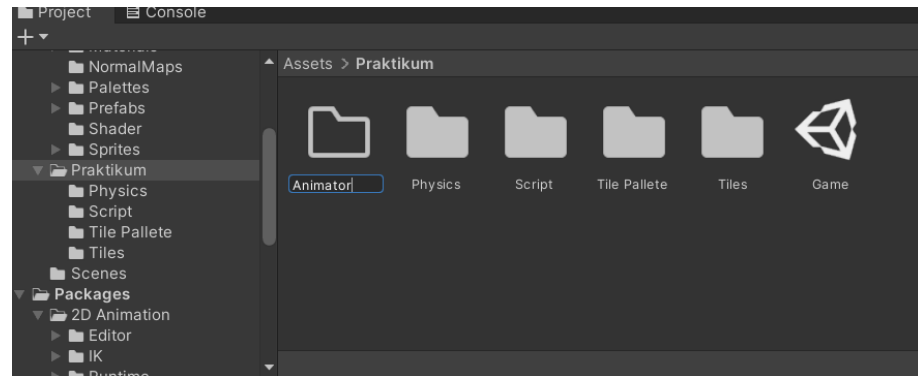
2. Klik Karakter, lalu tambahkan component Animator pada Inspector



Gambar 1.2 Tampilan Add Component

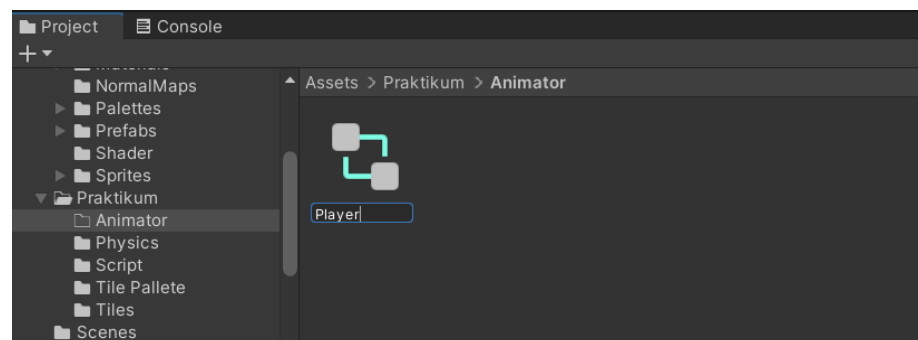


3. Buat folder baru dan beri nama Animator.



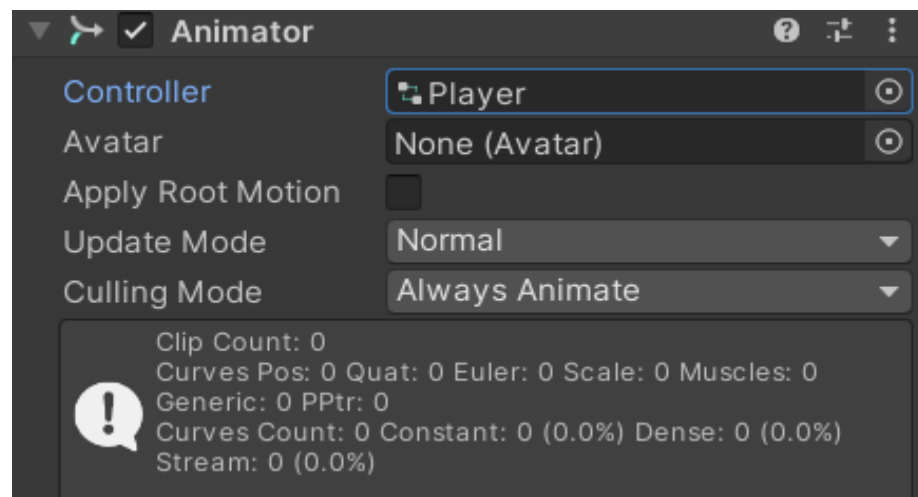
Gambar 1.3 Tampilan Buat Folder

4. Buat file Animator Controller pada folder Animator, ubah nama menjadi Player.



Gambar 1.4 Tampilan Animator Controller

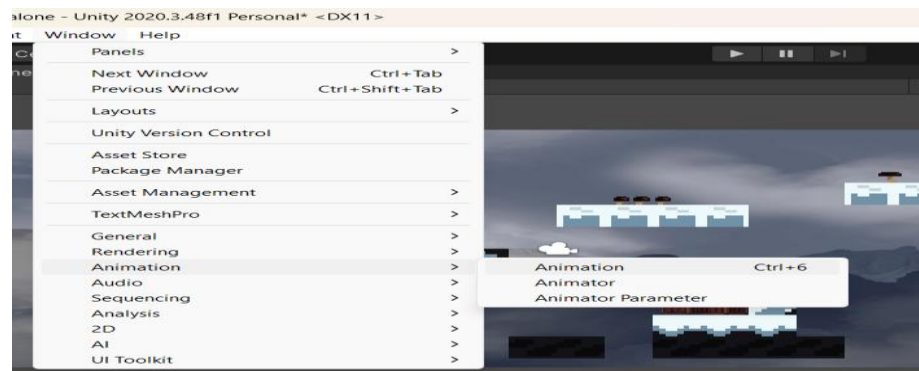
5. Klik Karakter, lalu pada Inspector cari komponen Animator, setelah itu pada setting Controller ubah menjadi Player.



Gambar 1.5 Tampilan Ubah Controller

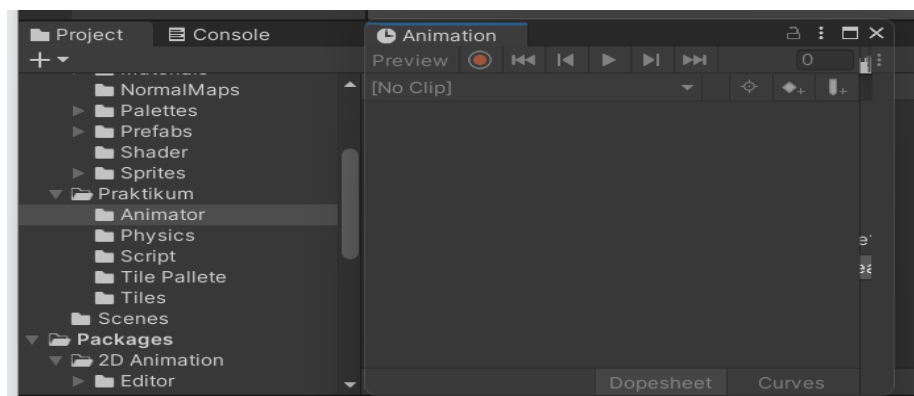


6. Tambahkan menu panel Animation pada menu Window lalu Animation lalu Animation.



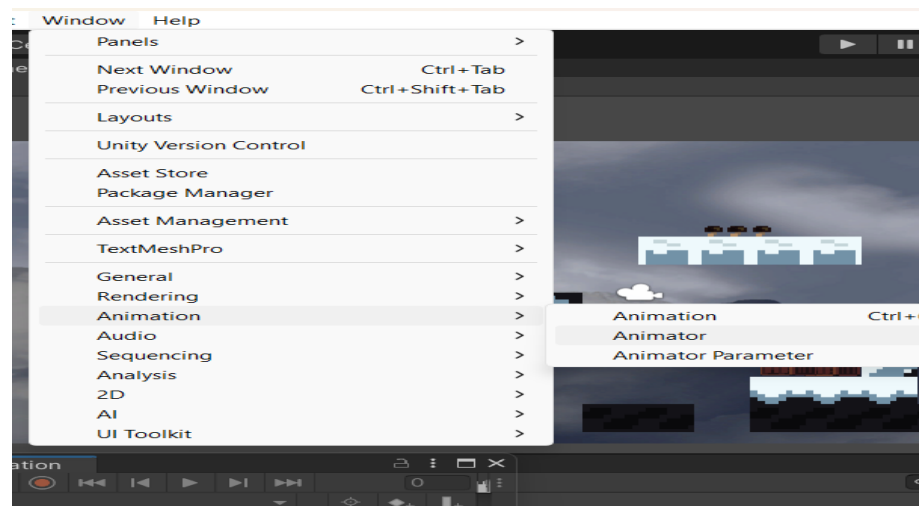
Gambar 1.6 Tampilan Window Animation

7. Akan muncul panel baru, taruh dibawah sebelah Project



Gambar 1.7 Tampilan Panel Animation

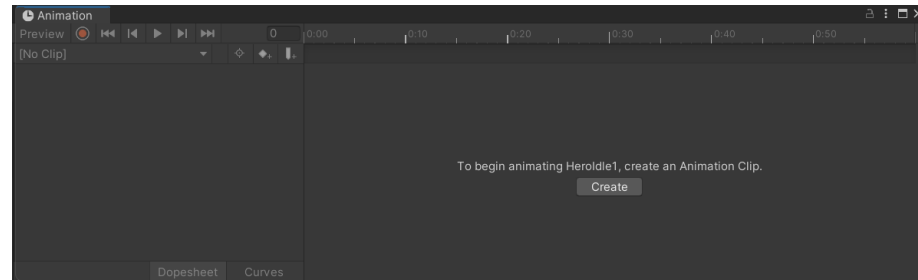
8. Tambahkan juga panel Animator.



Gambar 1.8 Tampilan Panel Animator

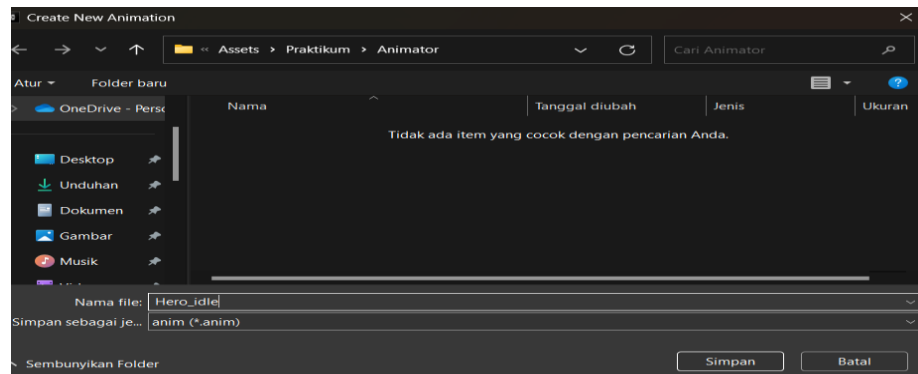


9. Klik Karakter, lalu ke panel Animation, klik Create



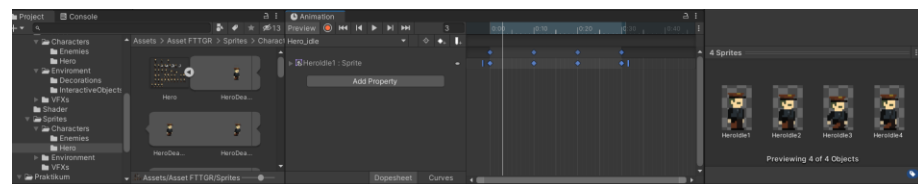
Gambar 1.9 Tampilan Membuat Animasi

10. Simpan pada folder Animator, dan beri nama Hero_idle



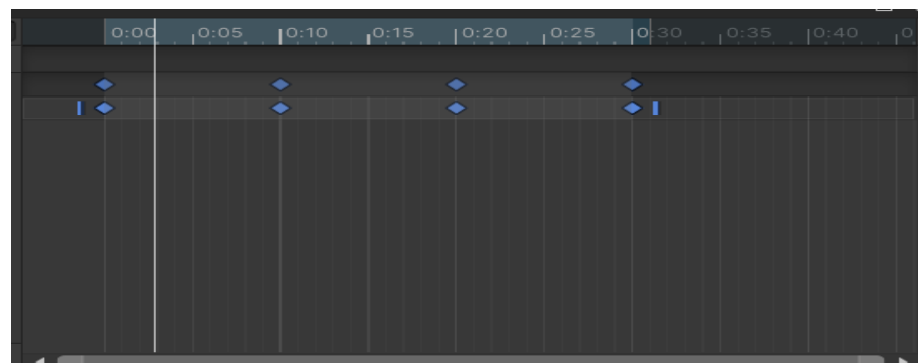
Gambar 1.10 Tampilan Simpan File

11. Cari asset hero, lalu drag n drop ke panel Animation



Gambar 1.11 Tampilan Menambahkan Asset

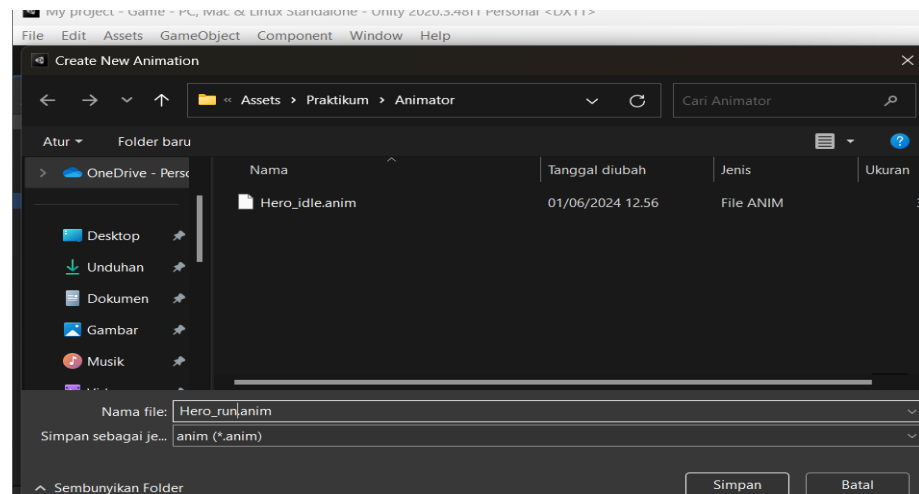
12. Kemudian, geser kotak kecil yang terdapat disebelah keyframe, geser hinggal fram 01:00, agar animasi tidak terlalu cepat



Gambar 1.12 Tampilan Geser Keyframe

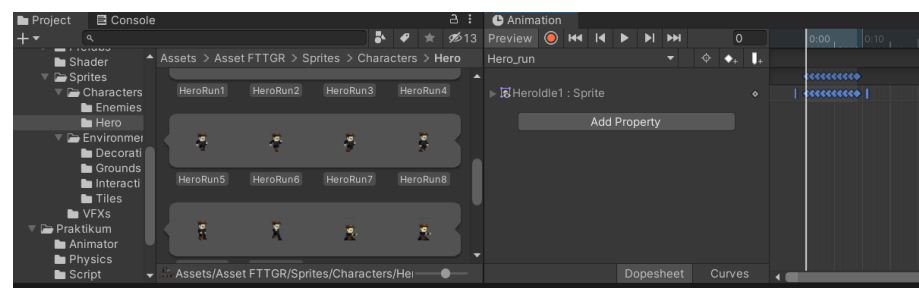


13. Buat animasi baru lagi, beri nama Hero_run, simpan pada folder Animator juga.



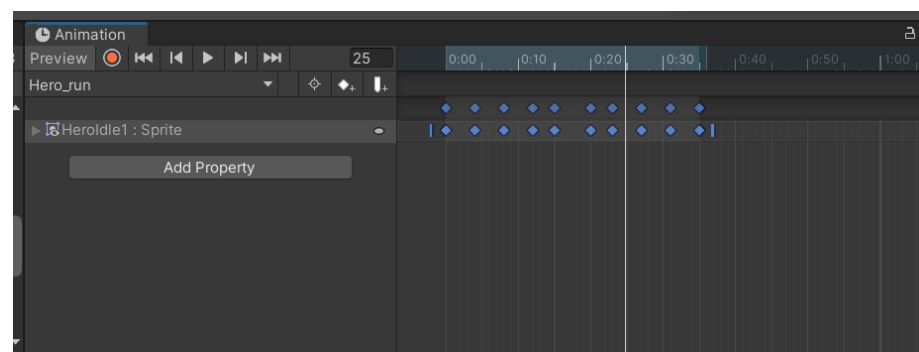
Gambar 1.13 Tampilan Buat Animasi

14. Setelah itu, cari asset hero_run, lalu drag n drop kedalam panel Animation



Gambar 1.14 Tampilan Asset Animasi

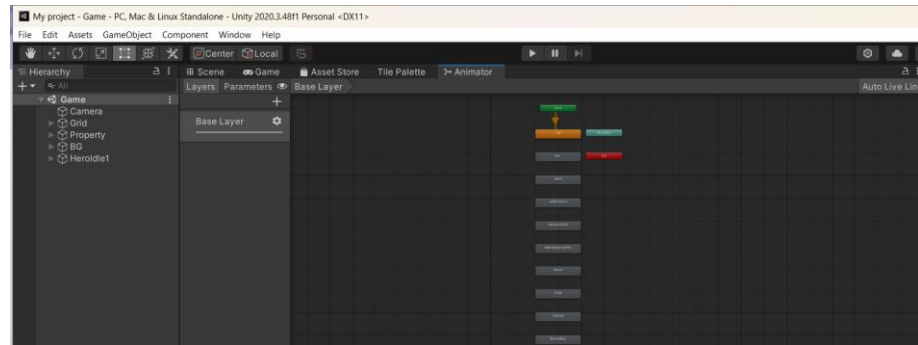
15. Setelah itu geser kotak kecil yang terdapat disebelah keyframe, geser hinggal fram 01:00, agar animasi tidak terlalu cepat



Gambar 1.15 Tampilan Geser Keyframe

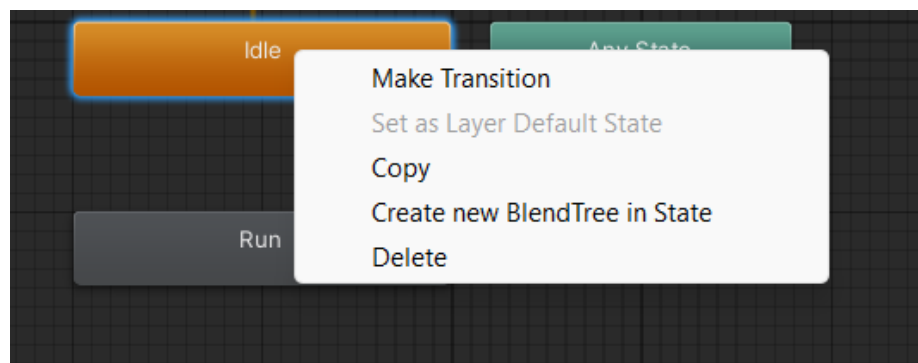


16. Setelah itu masuk ke panel Animator, dan akan tampil seperti dibawah



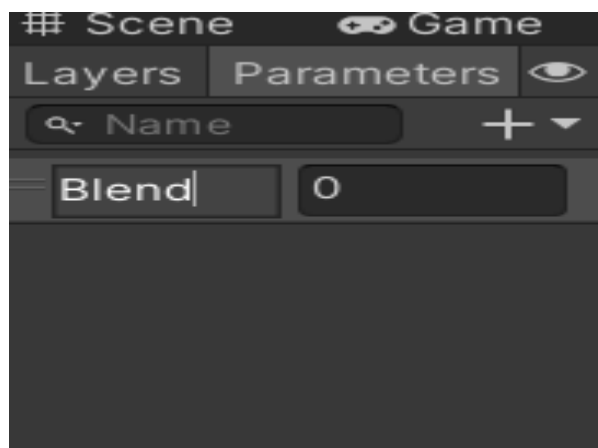
Gambar 1.16 Tampilan Panel Animator

17. Buat transisi antara hero_idle dan hero_run, dengan cara klik kanan pada hero_idle lalu pilih Make Transition, lalu arahkan ke hero_run.



Gambar 1.17 Tampilan Buat Transisi

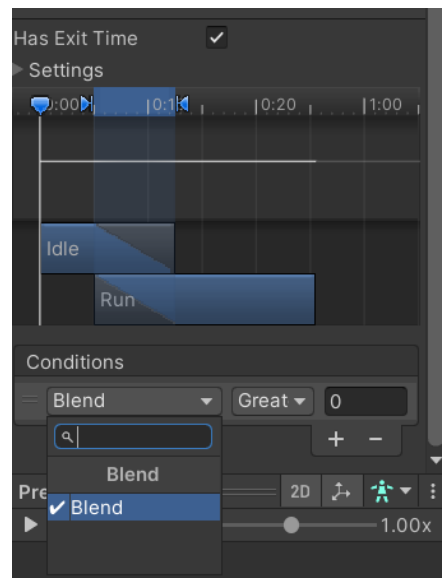
18. Tambahkan parameter baru pada Tab Parameters, buat parameter Float dan beri nama Blend



Gambar 1.18 Tampilan Buat Parameter

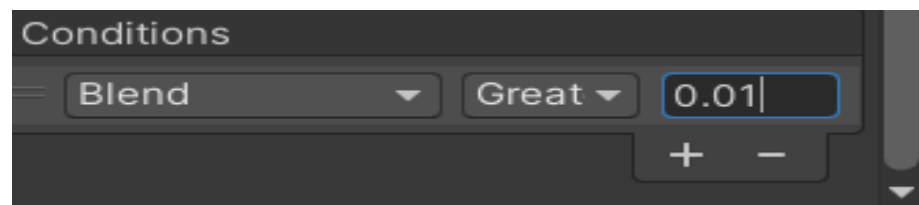


19. Klik panah putih diantara hero_idle dan hero_run, lalu pada bagian Conditions tambahkan parameter Blend.



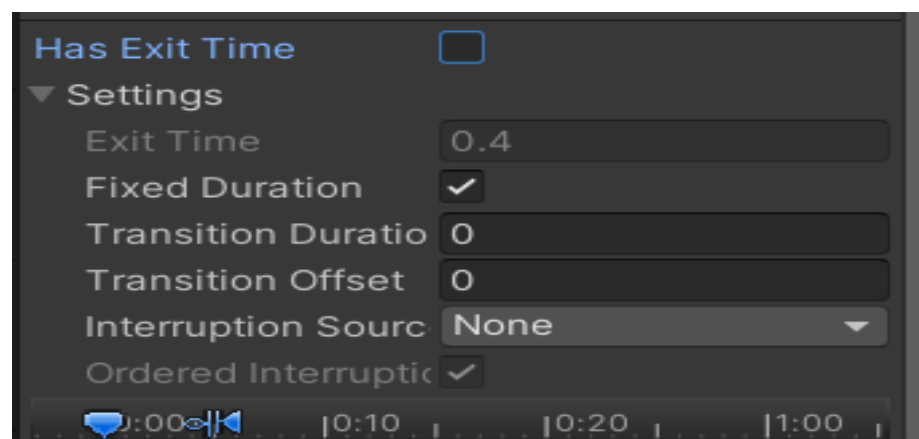
Gambar 1.19 Tampilan Tambah Parameter

20. Atur nilai conditions Blend tadi menjadi 0.01



Gambar 1.20 Tampilan Atur Nilai

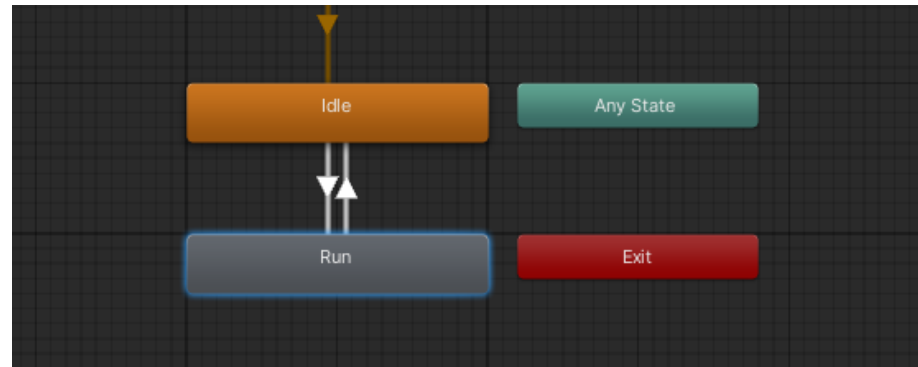
21. Pada bagian Settings, hilangkan centang Has Exit Time, lalu ganti nilai Transition Duration menjadi 0



Gambar 1.21 Tampilan Settings Animasi

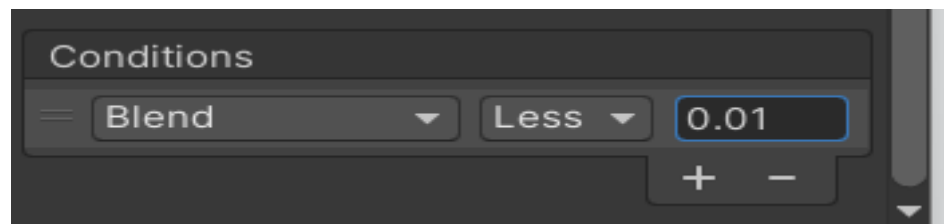


22. Buat transisi juga dari hero_run ke hero_idle



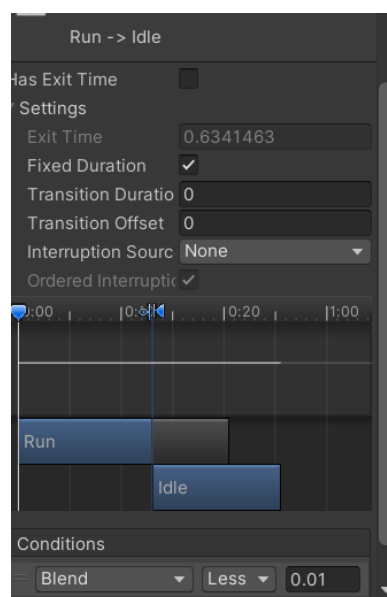
Gambar 1.22 Tampilan Buat Transisi

23. Klik panah yang mengarah dari hero_run ke hero_idle, lalu pada bagian Conditions, tambahkan parameter Blend, ubah operator menjadi Less, dan ubah nilai menjadi 0.01



Gambar 1.23 Tampilan Tambah Conditions

24. Pada bagian Settings, hilangkan centang Has Exit Time, lalu ganti nilai Transition Duration menjadi 0



Gambar 1.24 Tampilan Settings Animasi



25. Buka script Player, lalu tambahkan kode dibawah ini

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

    Rigidbody2D rb;
```

Gambar 1.25 Tampilan Menambahkan Kode

26. Tambahkan juga script komponen animator pada void Awake seperti dibawah

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

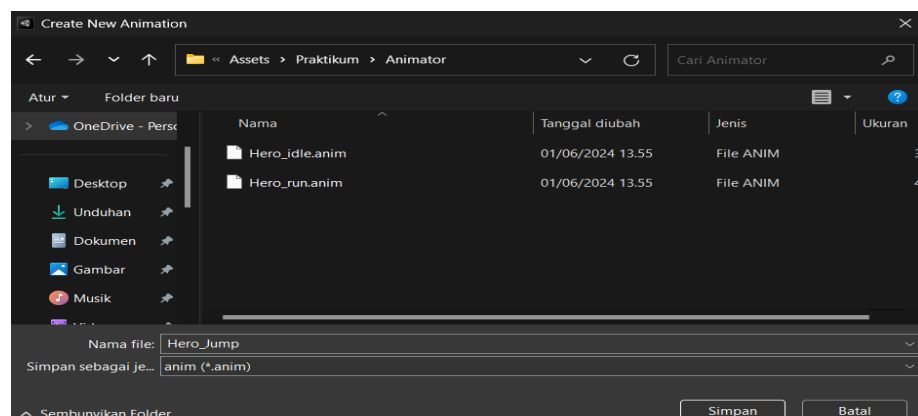
Gambar 1.26 Tampilan Menambahkan Script

27. Tambahkan script juga pada void FixedUpdate seperti dibawah ini

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

Gambar 1.27 Tampilan Menambahkan Script

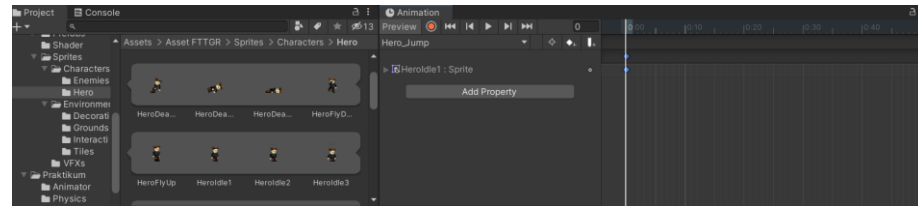
28. Kemudian buat animasi baru lalu beri nama hero_jump, simpan pada folder Animator



Gambar 1.28 Tampilan Buat Animasi

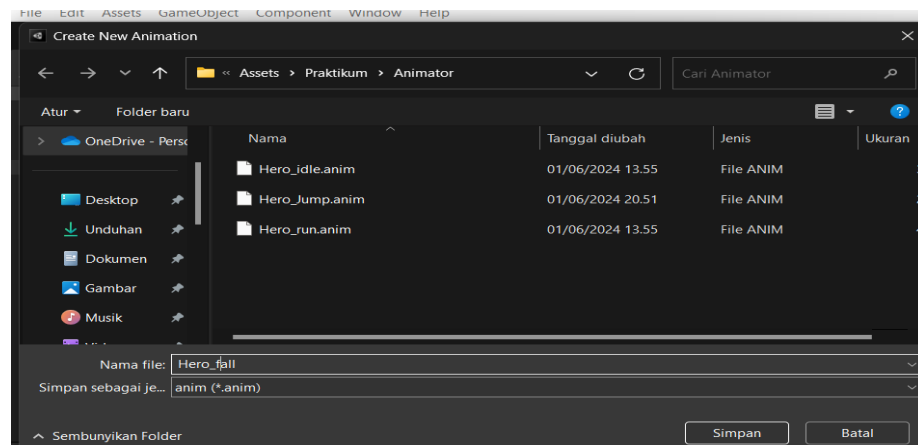


29. Cari asset hero jump, lalu drag n drop kedalam panel Animation



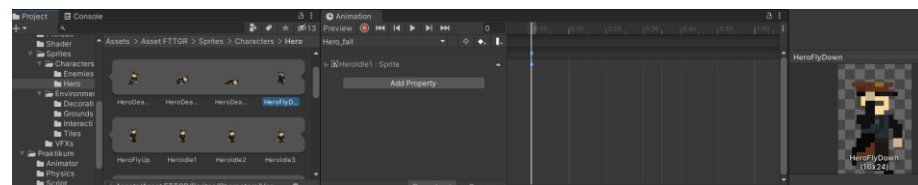
Gambar 1.29 Tampilan Menambahkan Asset

30. Buat animasi baru lagi, beri nama hero_fall, simpan pada folder Animator



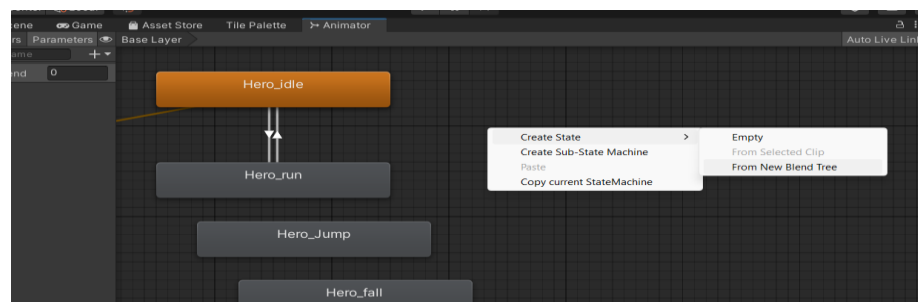
Gambar 1.30 Tampilan Buat Animasi

31. Cari asset hero fall, lalu drag n drop kedalam panel Animation



Gambar 1.31 Tampilan Menambahkan Asset

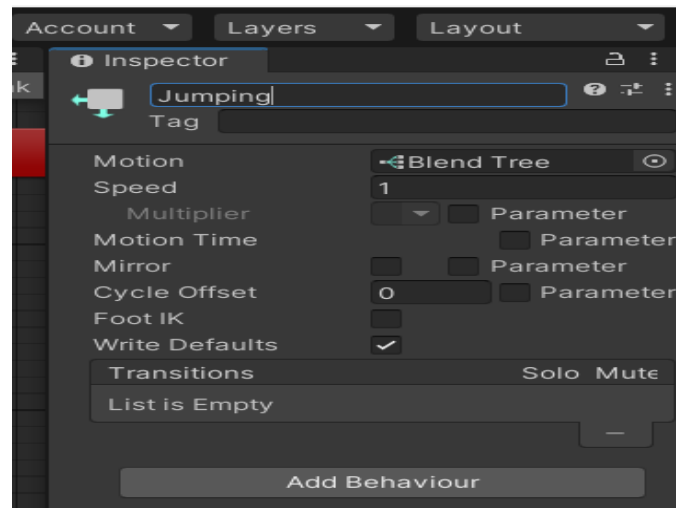
32. Kemudian pergi ke panel Animator, klik kanan pada area kosong, lalu pilih Create State lalu, From New Blend Tree



Gambar 1.32 Tampilan Buat State

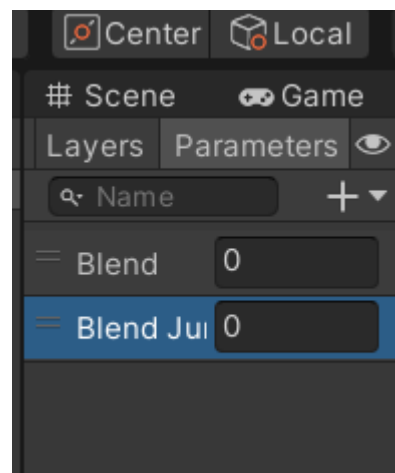


33. Cari asset karakter jump, lalu drag n drop kedalam panel Animation



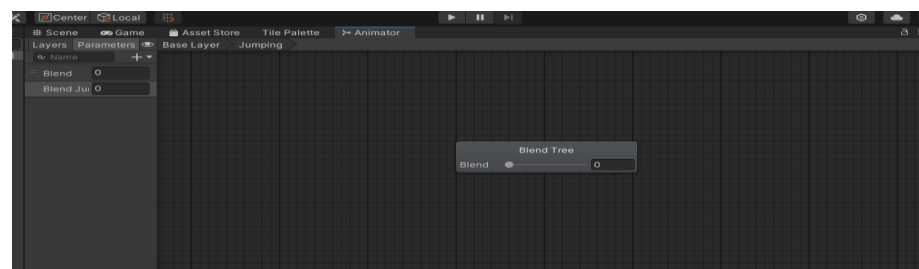
Gambar 1.33 Tampilan Rename State

34. Pada bagian parameter, tambahkan parameter dengan tipe data Float dan beri nama Blend Jump



Gambar 1.34 Tampilan Buat Parameter

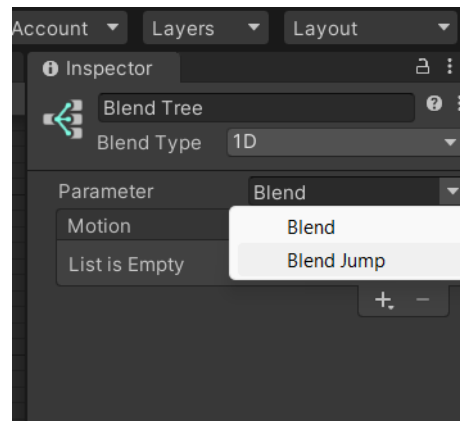
35. Lalu klik dua kali pada Blend Tree Jumping, lalu akan masuk ke panel seperti dibawah ini



Gambar 1.35 Tampilan Blend Tree

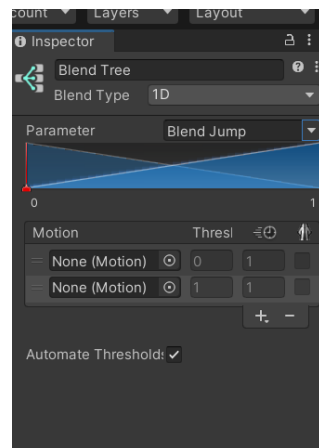


36. Lalu pada Inspector Blend Tree, ubah parameter menjadi Blend Jump



Gambar 1.36 Tampilan Ubah Parameter

37. Lalu tekan icon tambah dan pilih Add Motion Field, tambahkan dua Motion Field



Gambar 1.37 Tampilan Add Motion

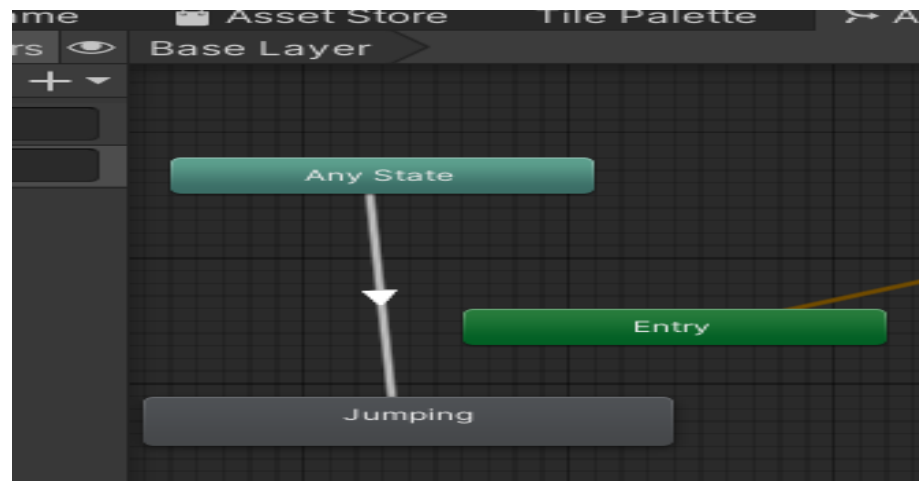
38. Cari motion seperti dibawah ini, sesuaikan urutannya, lalu hilangkan centang pada Automate Thresholds, dan atur nilai Thresh seperti dibawah



Gambar 1.38 Tampilan Atur Threshold

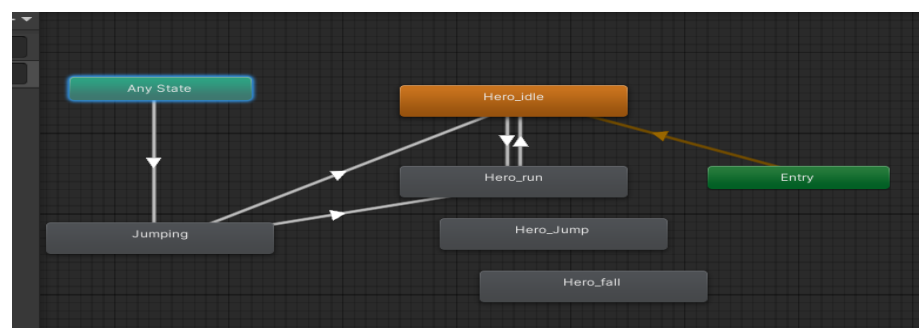


39. Kembali ke Base Layer, klik kanan pada Any State, lalu pilih Make Transition dan arahkan ke Jumping



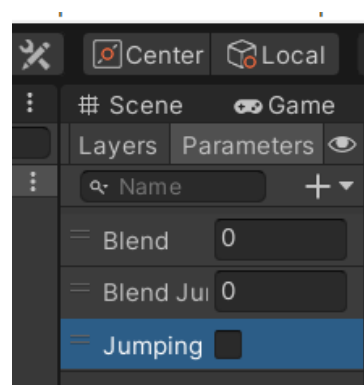
Gambar 1.39 Tampilan Make Transition

40. Klik kanan pada Jumping, lalu Make Transition dan arahkan ke hero_idle dan hero_run



Gambar 1.40 Tampilan Make Transition

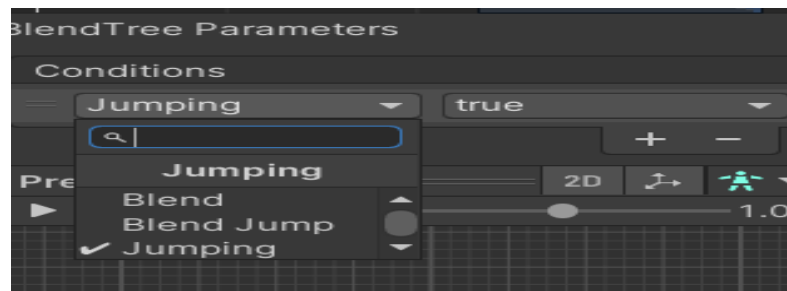
41. Tambahkan parameter transisi dengan tipe data Bool, lalu beri nama Jumping



Gambar 1.41 Tampilan Buat Parameter

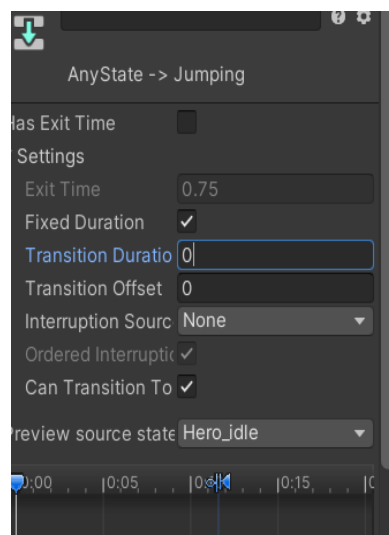


42. Klik panah yang mengarah dari Any State ke Jumping, lalu pada inspector tambahkan Conditions Jumping dan ubah nilai menjadi True



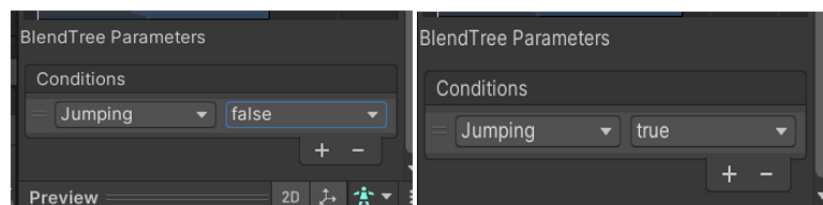
Gambar 1.42 Tampilan Add Condition

43. Lalu hilangkan juga centang Has Exit Time dan pada bagian Settings, ubah nilai Transition Duration ke 0



Gambar 1.43 Tampilan Setting Animasi

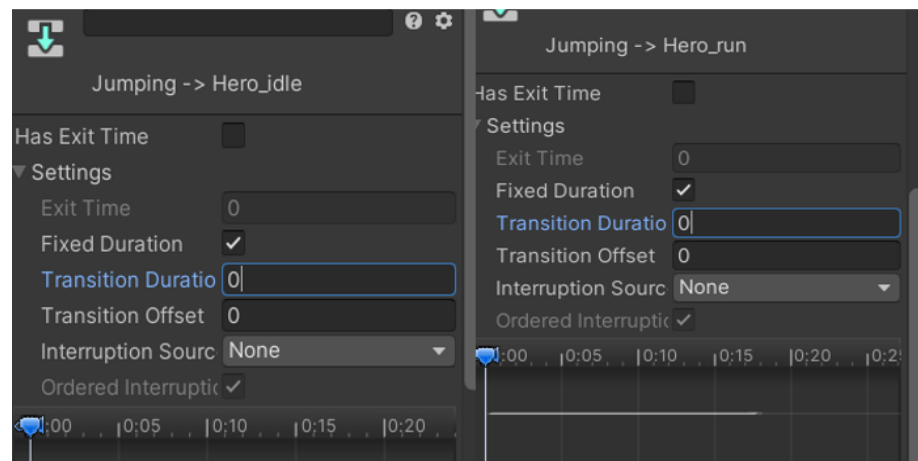
44. Lalu Make Transition juga dari Jumping ke karakter_idle dan karakter_walk, lalu pada Inspector tambahkan Conditions Jumping, untuk panah yang mengarah ke karakter_idle ubah Conditions Jumping menjadi false, lalu untuk panah yang mengarah ke karakter_walk ubah Conditions Jumping menjadi true



Gambar 1.44 Tampilan Transisi Jump



45. Lalu pada masing-masing panah, klik Settings, lalu ubah nilai Transition Duration ke 0 dan hilangkan centang pada Has Exit Time



Gambar 1.45 Tampilan Setting Animasi

46. Lalu buka kembali Script Player, tambahkan source code berikut pada fungsi void Update

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

Gambar 1.46 Tampilan Script Update

47. Lalu pada fungsi void FixedUpdate, tambahkan source code berikut

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

Gambar 1.47 Tampilan Script FixedUpdate

48. Lalu pada fungsi void GroundCheck, tambahkan source code berikut

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

Gambar 1.48 Tampilan Script GroundCheck



49. Setelah itu jalankan, coba untuk berjalan dan melompat, setelah itu render game yang sudah dibuat



Gambar 1.49 Tampilan Play Game

B. Link Github Pengumpulan

Link :



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);
    }
}
```

Analisa :

Pada void HandleJumpInput, fungsi animator.SetBool memerlukan dua parameter yaitu nama parameter boolean dalam Animator dan nilai boolean yang akan diatur, didalam source code diatas, nilai boolean tidak ada.

Lalu pada void HandleMovementInput, ada beberapa kesalahan, yaitu pada percabangan if yang pertama, move != 1 salah seharusnya move == 0, karena lebih tepat untuk menentukan bahwa karakter benar-benar tidak bergerak (idle), karena move bernilai 0 ketika tidak ada input gerakan



horizontal. Lalu handle animasi juga kurang dan juga ada logika yang salah, dan perbaikannya seperti dibawah

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

Lalu pada percabangan if yang kedua, terdapat logika scaling yang salah, scaling di kode sebelumnya di hardcode. Seharusnya bukan if(move != 0), melainkan if (move < 0) dan juga block kode yang dijalankan didalam if juga salah. berikut adalah perbaikan yang benar

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