

# TUGAS PERTEMUAN: 9

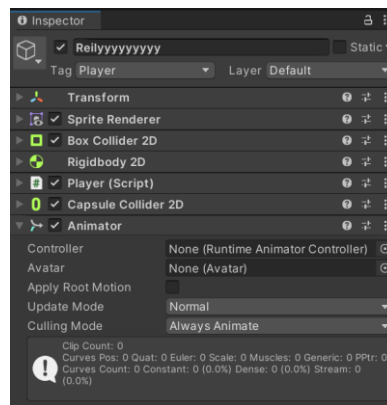
## GAME ANIMATION

NIM	:	2118132
Nama	:	Madelbertha Fridolin Puka
Kelas	:	D
Asisten Lab	:	Abidzar Al Giffari (2218013)

### 9.1 Tugas 9: Membuat Game Animation

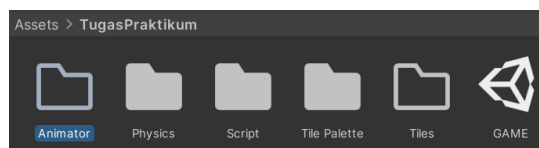
#### A. Membuat Game Animation

1. Pada karakter klik *Inspector* kemudian pilih *Add component Animator*



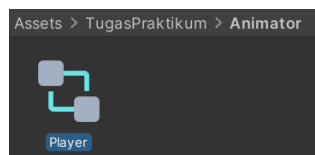
Gambar 9.1 Tampilan *Add Component Animator*

2. Pada folder Tugas Praktikum buat folder baru dengan nama Animator



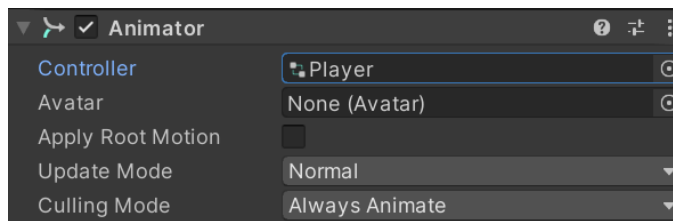
Gambar 9.2 Tampilan Folder Animator

3. Buat *file animator controller* pada folder Animator, ubah namanya menjadi "Player"



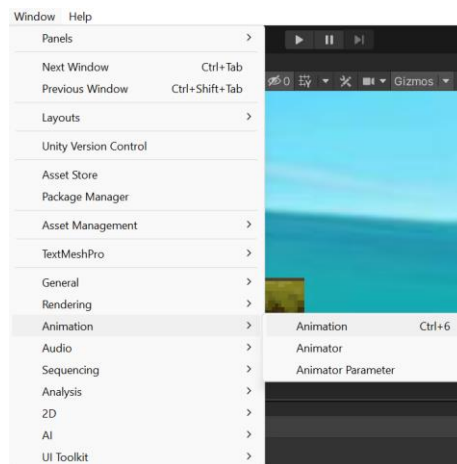
Gambar 9.3 Tampilan *File Player*

4. Klik *Player* pada *Hierarchy* kemudian cari *component Animator*, pada *setting controller* ubah menjadi *Player*.



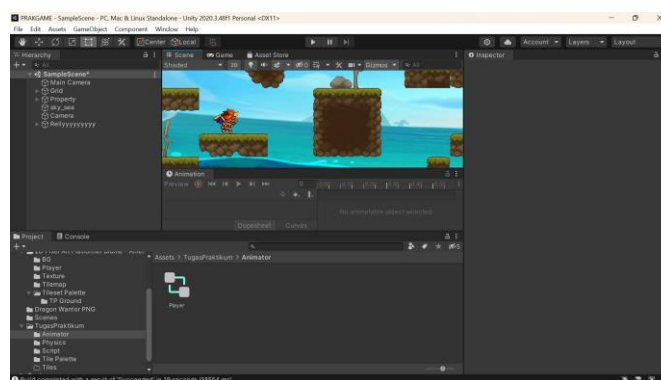
Gambar 9.4 Tampilan Ubah Nama

5. Tambahkan menu panel Animation di menu *Window*, pilih Animation > Animation atau tekan *Shortcut CTRL + 6*



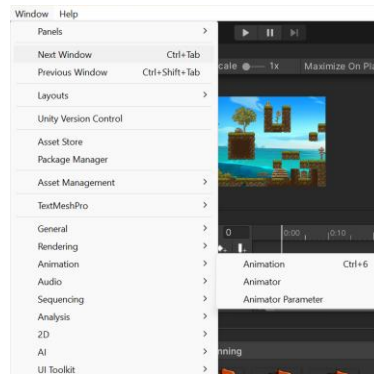
Gambar 9.5 Tampilan Window

6. Akan muncul menu *panel* baru, geser *panel* tersebut dibawah sendiri seperti gambar dibawah ini, *Split Panel Project* dan Animation



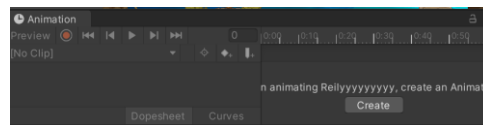
Gambar 9.6 Tampilan Split Panel Project

7. Tambahkan menu panel Animator



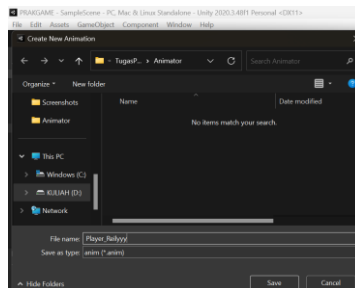
Gambar 9.7 Tampilan Panel Animator

8. Geser Panel tersebut sesuai dengan gambar. Untuk membuat animasi klik Reillyyyy pada *Hierarchy*, kemudian ke menu panel Animation, pilih *Create*



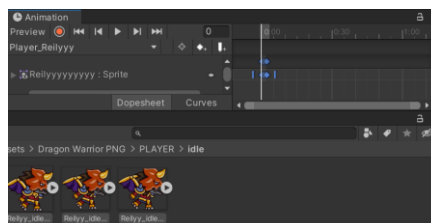
Gambar 9.8 Tampilan Create Animation

9. Simpan pada folder Animator dan beri nama “Player\_Reillyyyy”



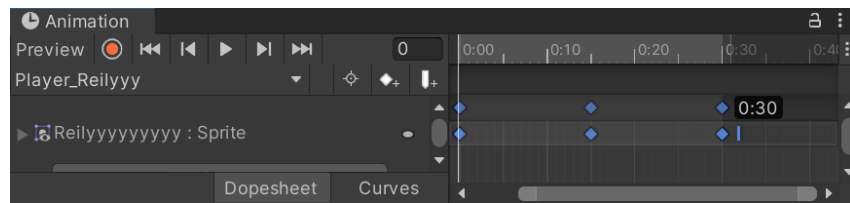
Gambar 9.9 Tampilan Player\_Reillyyyy

10. Pada menu *Project* buka folder player lalu pilih Reilly, kemudian *drag* ke *tab* Animation



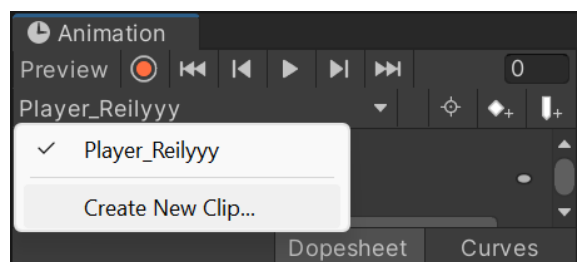
Gambar 9.10 Tampilan Drag Reillyy

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



Gambar 9.11 Tampilan *Timeline Frame*

12. Buat animasi baru, Klik pada “Player\_Reillyyy” kemudian pilih *Create New Clip*, dan beri nama “Player\_run”, Simpan pada Folder Animator



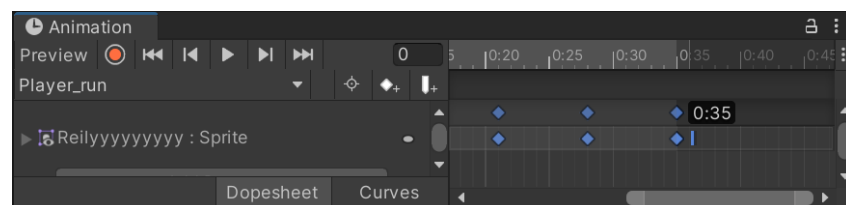
Gambar 9.12 Tampilan Player\_run

13. Buka menu *Project* kemudian cari *folder* Player > run, Pilih *player-run*, *drag and drop* pada menu Animation



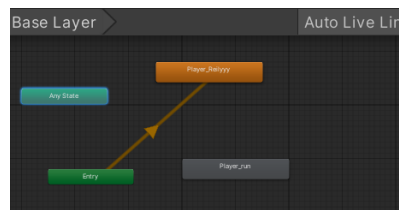
Gambar 9.13 Tampilan *Drag* Player\_run

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



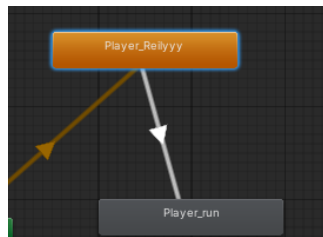
Gambar 9.14 Tampilan *Timeline Frame*

15. Pilih ke menu Animator yang telah dibuka sebelumnya dan akan tampil seperti berikut



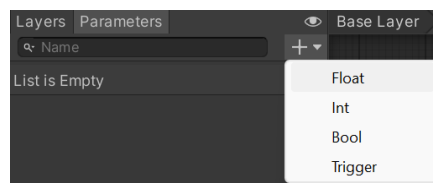
Gambar 9.16 Tampilan Menu Animator

16. Kemudian buat transisi antara player\_Reilly dan player\_run dengan cara klik kanan pada player\_Reilly dan pilih *Make Transition* dan tarik ke player\_run



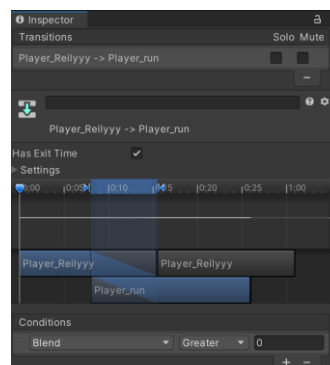
Gambar 9.17 Tampilan *Transisi*

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



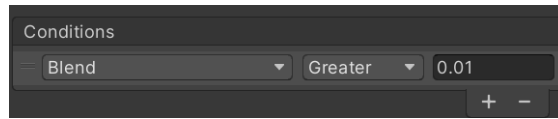
Gambar 9.18 Tampilan *Parameters*

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



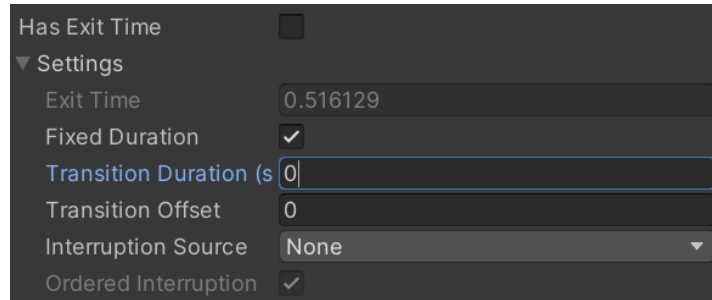
Gambar 9.19 Tampilan Pengaturan *Conditions*

19. Atur nilai *conditions blend* tersebut menjadi 0.01



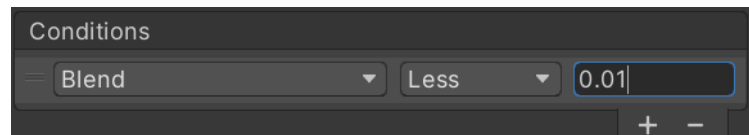
Gambar 9.20 Tampilan Pengaturan *Conditions*

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



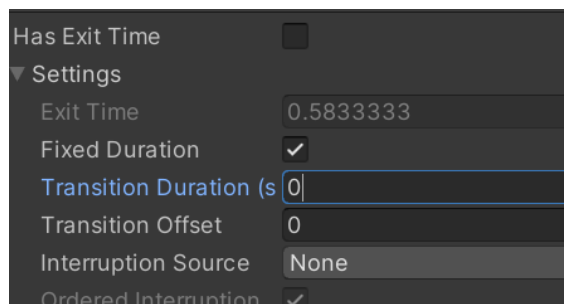
Gambar 9.21 Tampilan *Settings*

21. Buat transisi juga dari *player\_run* ke *player\_Reilly* dengan cara klik kanan pada *player\_run* dan pilih *Make Transition*. Tambahkan parameter transisi dengan tipe data *Float*. Klik ikon tambah dan *rename* menjadi “Blend”. Setelah itu, ubah operator dari *Greater* menjadi *Less* dan atur nilainya menjadi 0.01.



Gambar 9.22 Tampilan *Conditions*

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



Gambar 9.23 Tampilan *Settings*

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

```
Public Animator animator;
```

24. Tambahkan *Script* Komponen Animator

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

25. Dan pada fungsi *FixedUpdate* tambahkan *source code* berikut

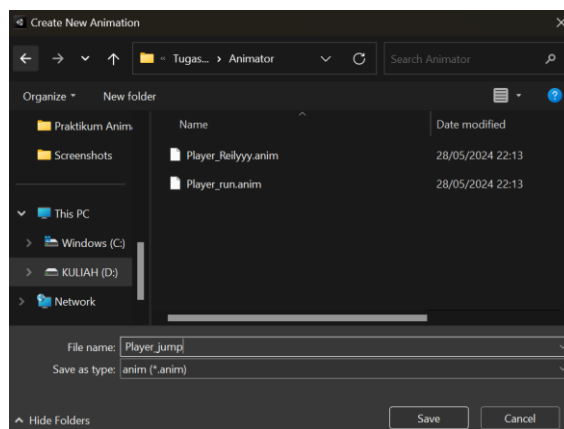
```
animator.SetFloat("Blend Jump", rb.velocity.x);
```

26. Jika dijalankan maka *player* dapat memiliki animasi ketika berhenti atau pun ketika berjalan



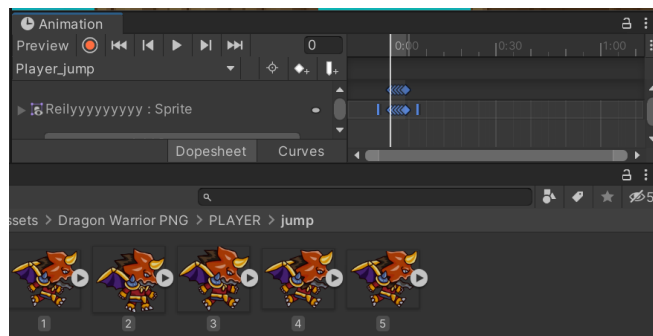
Gambar 9.24 Tampilan Player Ketika Berjalan

27. Kemudian buat animasi baru tekan tulisan “Player\_run” kemudian pilih *Create New Clip*, dan beri nama “Player\_jump”



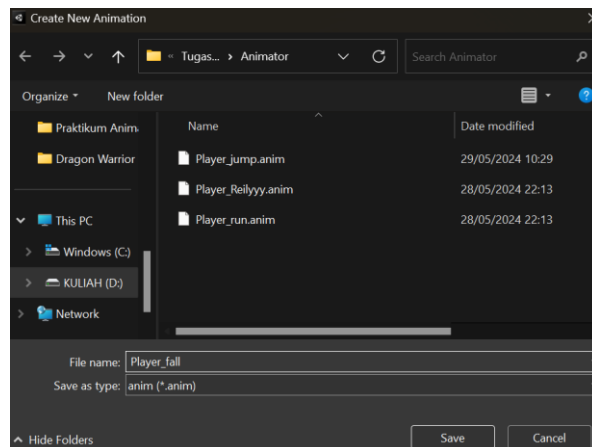
Gambar 9.25 Tampilan Player\_jump

28. Pada folder *player* buka *jump* lalu pilih gambar jump, kemudian *drag* ke *tab* Animation.



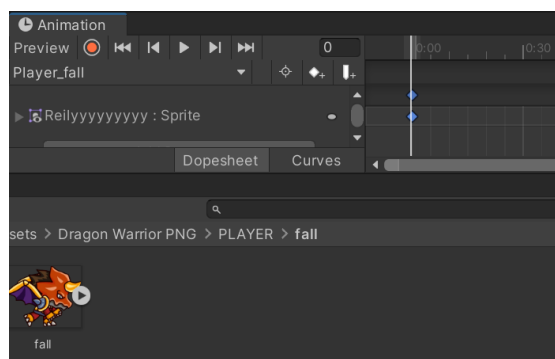
Gambar 9.26 Tampilan *Player Jump*

29. Buat animasi baru dengan cara tekan tulisan “Player\_jump” kemudian pilih *Create New Clip*, dan beri nama “Player\_fall”



Gambar 9.27 Tampilan *Player\_fall*

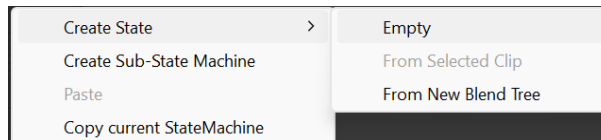
30. Pada *tab* *Project* buka folder karakter lalu pilih fall dan pilih gambar player-fall, kemudian *drag* ke *tab* Animation.



Gambar 9.28 Tampilan *Drag Player*

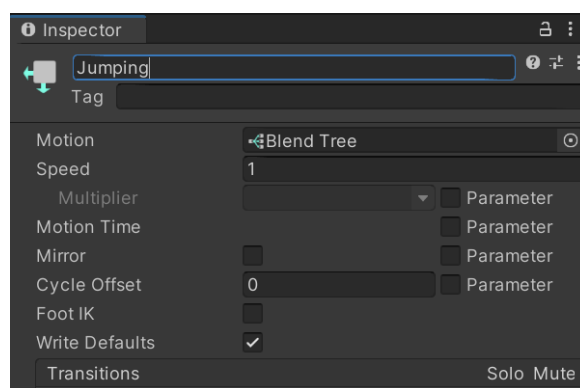


31. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong, pilih *Create State>From New Blend Tree*.



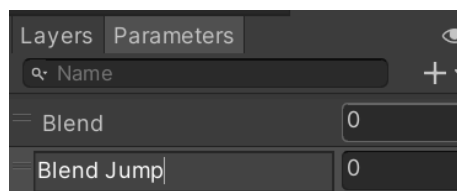
Gambar 9.29 Tampilan *Create State*

32. Pada Animator klik *Blend Tree*, di menu *Inspector*, ubah namanya menjadi *Jumping*



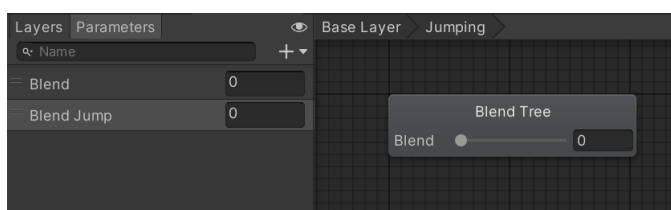
Gambar 9.30 Tampilan Pengubahan *Jumping*

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



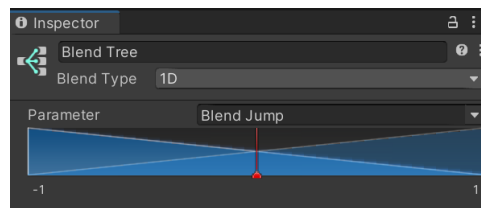
Gambar 9.31 Tampilan *Blend Jump*

34. Pada menu Animator, Klik dua kali pada *Blend Tree* “Jumping”, Tekan pada *Blend Tree*



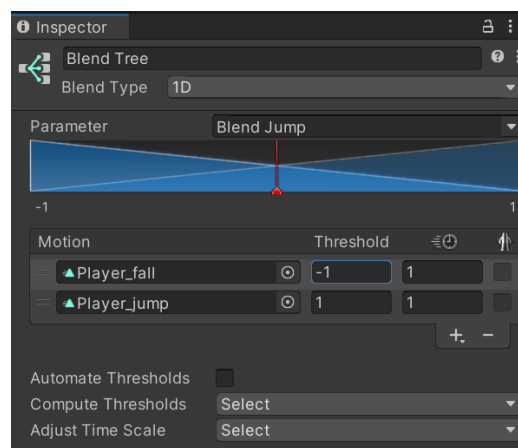
Gambar 9.32 Tampilan *Blend Tree*

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



Gambar 9.33 Tampilan *Inspector*

36. Buka menu *Inspector*, tekan icon + dan pilih *Add Motion Field*.  
Tambahkan dua *Motion Field*



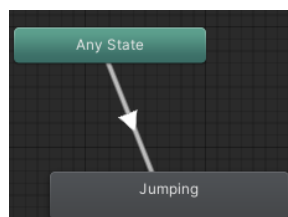
Gambar 9.34 Tampilan *Motion Field*

37. Klik bagian icon *None (Motion)*, maka akan muncul *Windows Motion*,  
Tambahkan Sesuai dengan urutan.



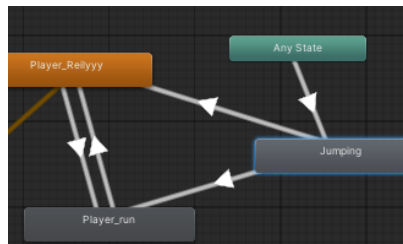
Gambar 9.35 Tampilan *Motion*

38. Hilangkan centang “*Automate Thresholds*” dan atur nilai *Threshold* seperti berikut. Kembali ke *Base Layer*, klik kanan *Any State*, pilih *Make Transition* dan arahkan panahnya ke *Jumping*



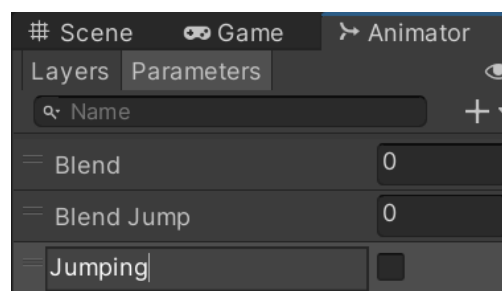
Gambar 9.36 Tampilan *Transition*

39. Klik kanan *Jumping*, pilih *Make Transition* dan arahkan panahnya ke *Player\_Reillyyy* dan *Player\_run*



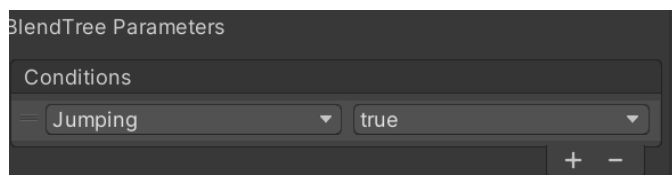
Gambar 9.37 Tampilan *Transition*

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



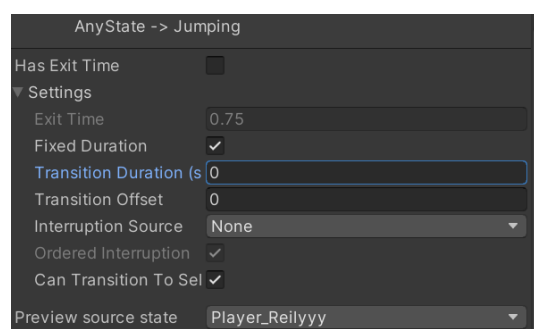
Gambar 9.38 Tampilan *Parameters*

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



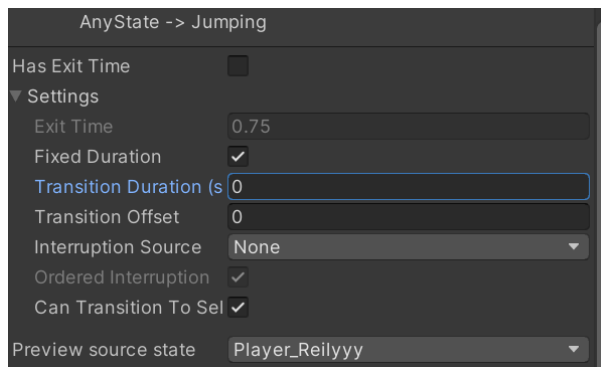
Gambar 9.39 Tampilan *Conditions*

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



Gambar 9.40 Tampilan *Settings*

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



Gambar 9.41 Tampilan *Settings*

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

```
Animator.SetBool("Jumping", true);
```

45. Pada Fungsi *FixedUpdate* tambahkan seperti berikut

```
Animator.SetFloat("Blend Jump", rb.velocity.y);
```

46. Tambahkan baris kode seperti dibawah ini dalam *method GroundCheck*

```
Animator.SetBool("Jumping", !isGrounded);
```

47. Jika di *play* maka karakter sudah bisa bergerak dengan animasi



Gambar 9.42 Tampilan Hasil Akhir

## KUIS

### Soucer Code

```
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 codingan diatas ini ada beberapa kesalahan sintak yang menyebabkan eror pada program. Pada fungsi *HendleJumpInput()*, baris *animator.SetBool("isJumping",);* tidak memiliki nilai *boolean* yang ditetapkan seharusnya codingannya itu *animator.SetBool("isJumping",true);* atau *animator.SetBool("isJumping",false);*.

Kemudian pada fungsi *HandleMovementInput()* baris ke-5 yaitu *transform.Translate(Vector3.left \_move\_ Time.deltaTime);* memiliki kesalahan penulisan. Seharusnya *transform.Translate(Vector3.left \* move \**

*Time.deltaTime*);. Selanjutnya Baris *animator.SetBool("isWalking", false)*; berada di dalam *else* yang seharusnya dieksekusi ketika karakter sedang berjalan. Seharusnya *animator.SetBool("isWalking", true)*;. Kemudian Baris *transform.localScale = new Vector3(-4, 1, 1)*; menggunakan nilai skala yang cukup besar dan berbeda jauh dengan ukuran y, z (-4) untuk sumbu X, yang dapat menyebabkan masalah pada tampilan karakter. Kemudian Baris *transform.localScale = new Vector3(1, 2, 1)*; menggunakan nilai skala yang cukup besar (2) untuk sumbu Y, yang juga dapat menyebabkan masalah pada tampilan karakter.