



## TUGAS PERTEMUAN: 9

### GAME ANIMATION

NIM	:	2118016
Nama	:	Dimas Rizky Pratama
Kelas	:	A
Asisten Lab	:	NATASYA OCTAVIA(2118034)

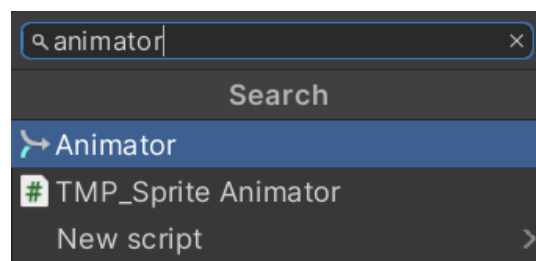
#### 1 Tugas 1 : Membuat Character Animation

1. Buka kembali proyek sebelumnya yang sudah dapat digerakkan dengan *keyboards*.



Gambar 9.1 Membuka Proyek Unity

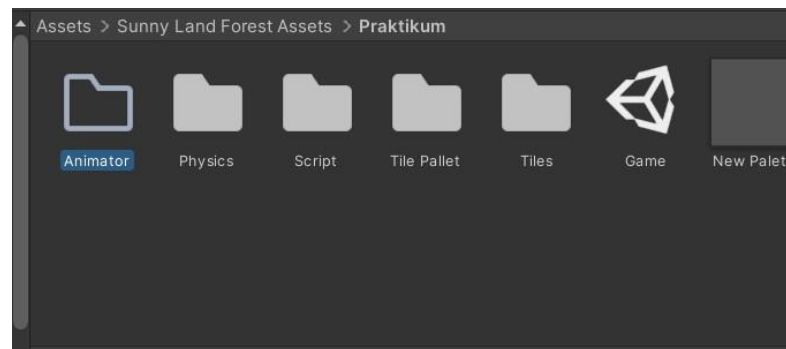
2. Kemudian klik karakternya lalu pada *inspector* tambahkan *component Animator*.



Gambar 9.2 Menambahkan Component Animator



3. Tambahkan *folder* baru pada *folder* “Praktikum” berikan nama *folder* dengan nama “Animator”.



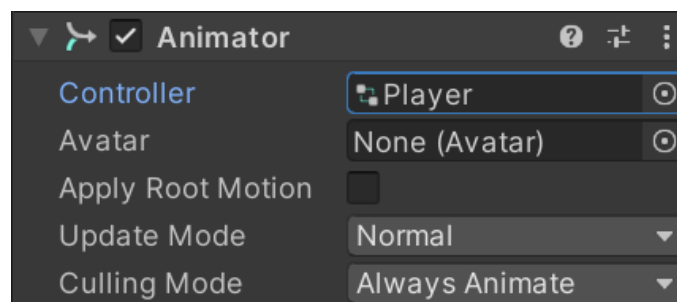
Gambar 9.3 Menambahkan Folder Baru

4. Tambahkan *file* “Animator Controller” pada *folder* “Animator” dengan cara klik kanan pada *folder* lalu *Create > Animator Controller*, ubah nama menjadi “Player”.



Gambar 9.4 Menambahkan File Animator Controller

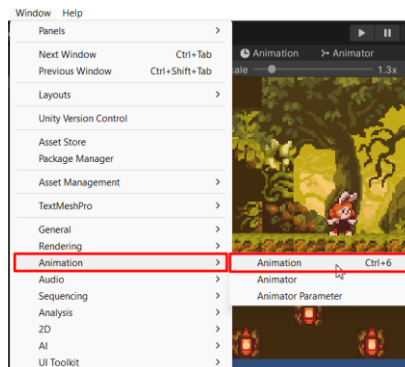
5. Lalu klik pada karakter bagian *Component Animator*, pada Controller ubah menjadi *Player*.



Gambar 9.5 Setting Controller

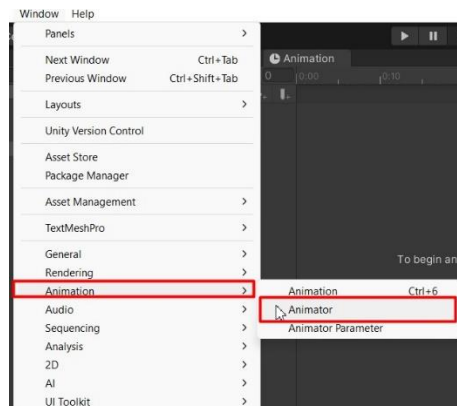


6. Tambahkan *menu panel Animation* dengan cara klik *menu Window > Animation > Animation*.



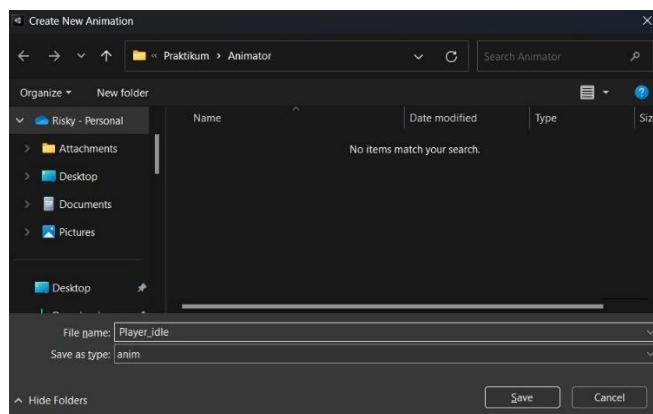
Gambar 9.6 Menambahkan Window Animation

7. Tambahkan lagi *Window* baru dengan cara klik *menu Window > Animation > Animator*.



Gambar 9.7 Menambahkan Window Animator

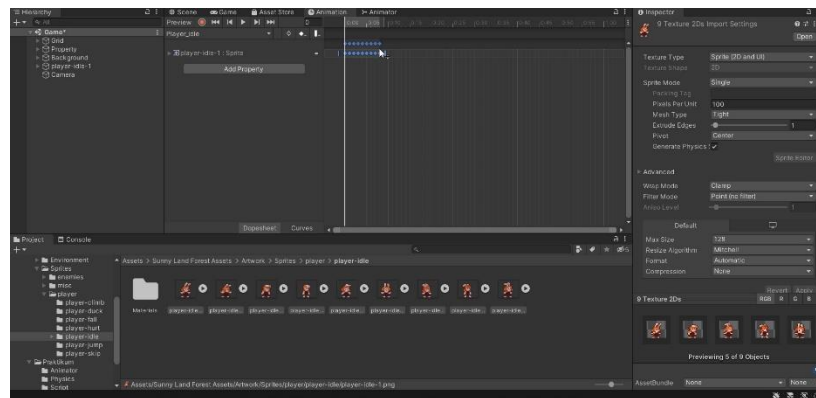
8. Klik *Create* pada *Animation* lalu simpan pada folder *animator* dan beri nama “*Player\_idle*”.



Gambar 9.8 Menyimpan File

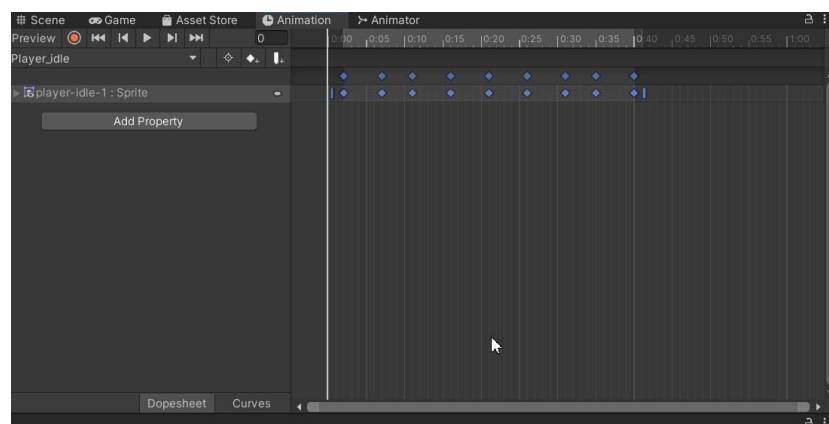


9. Kemudian masukkan *asset player idle*, masukkan semua gambar *player idle* ke *tab Animation*.



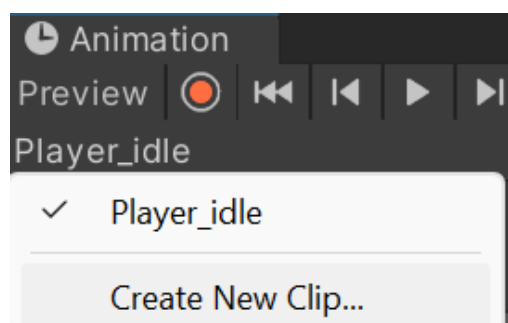
Gambar 9.9 Meletakkan Asset Iddle

10. Lalu tekan **Ctrl + A** pada *tab Animation* kemudian geser kotak ke *frame 40*.



Gambar 9.10 Menambahkan Frame

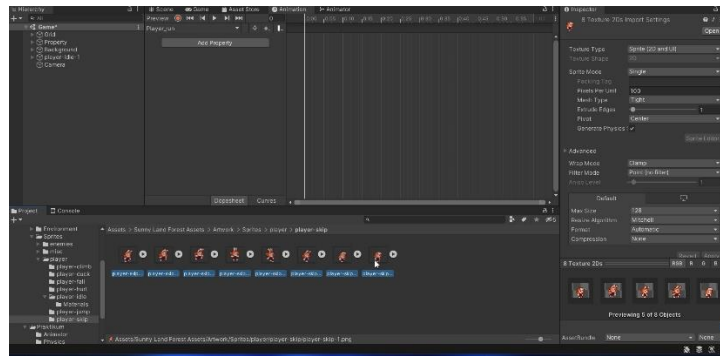
11. Tambahkan animasi bari dengan klik pada "Player\_idle" lalu pilih *Create New Clip*, beri nama "Player\_run" dan simpan pada *folder Animator*.



Gambar 9.11 Menambahkan Animasi

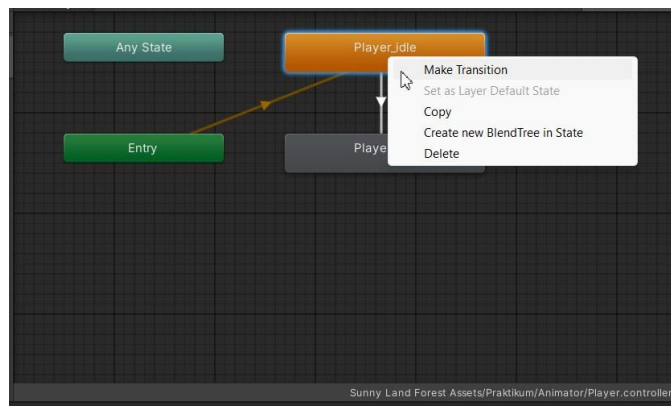


12. Selanjutnya tambahkan *asset run*, masukkan semua gambar animasi *run* dan masukkan ke *tab Animation*, lalu geser frame sampai frame 35.



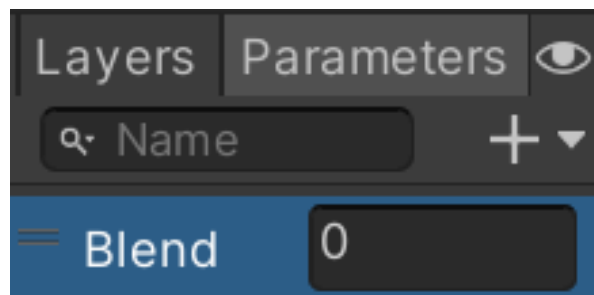
Gambar 9.12 Menambahkan Animasi Run

13. Lalu pada *menu Animator*, buat transisi antara *Player\_idle* dan *Player\_run* dengan klik kanan pada *Player\_idle* pilih *Make Transition* lalu tarik ke *Player\_run*.



Gambar 9.13 Membuat Transisi

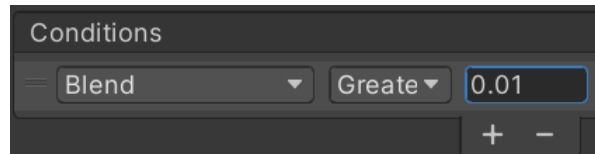
14. Lalu pada *tab parameter* tambahkan tipe data *Float* dengan menekan *icon* tambah dan rubah namanya menjadi “Blend”.



Gambar 9.14 Menambahkan Parameter

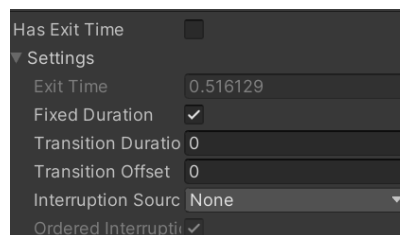


15. Klik panah putih antara Player\_idle dan Player\_run, Kemudian pada bagian *conditions* klik *icon* tambah atur menjadi “Blend” dan beri nilai *conditions* menjadi 0.01.



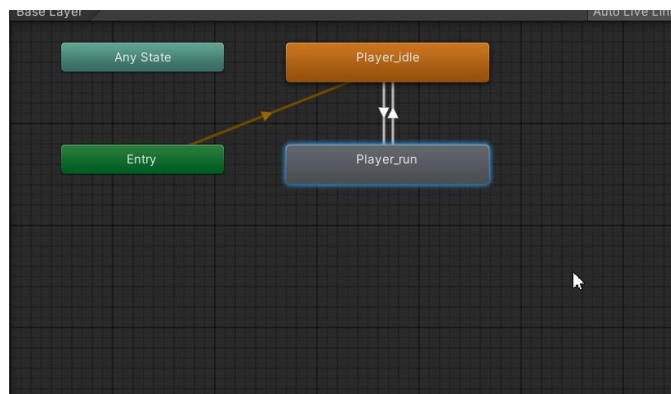
Gambar 9.15 Menambahkan Conditions

16. Pada bagian *settings* hilangkan centang pada *Has Exit Time* dan beri nilai *Transition Duration* menjadi 0.



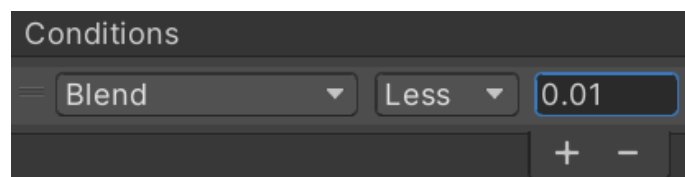
Gambar 9.16 Mengatur Settings

17. Lalu buat transisi kembali dari Player\_run ke Player\_idle dengan cara yang sama.



Gambar 9.17 Membuat Transisi

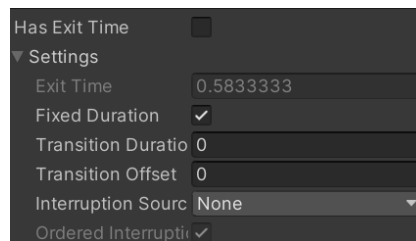
18. Tambahkan lagi *parameter* dengan tipe *data float*, ubah nama menjadi “Blend” lalu rubah operator menjadi *Less* dan berikan nilai 0.01.



Gambar 9.18 Menambahkan Parameter Baru



19. Pada bagian *settings* hilangkan juga centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



Gambar 9.19 Mengatur Setting

20. Lalu rubah *code* pada *class Player* dengan *code* dibawah ini.

```
Public Animator animator;
```

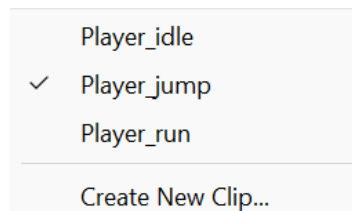
21. Tambahkan juga *code* dibawah ini pada *void Awake()*.

```
Animator = GetComponent<Animator>();
```

22. Tambahkan *code* dibawah ini pada *void FixedUpdate()*.

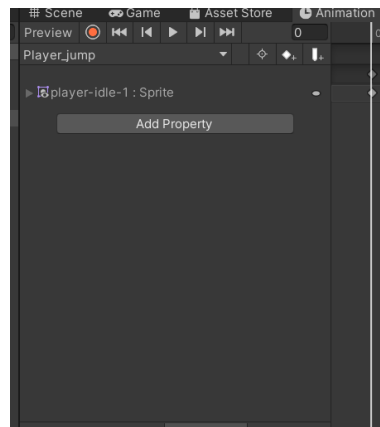
```
animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
```

23. Kemudian tambahkan animasi baru dengan nama "Player\_jump" dan letakkan pada folder *animator*.



Gambar 9.20 Menambahkan Animasi

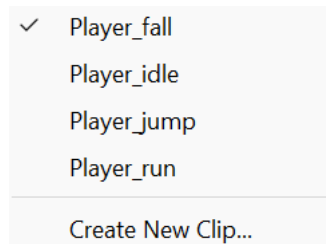
24. Lalu tambahkan *asset* melompat pada *folder* lalu letakkan pada *tab Animation*.



Gambar 9.21 Meletakkan Asset Melompat

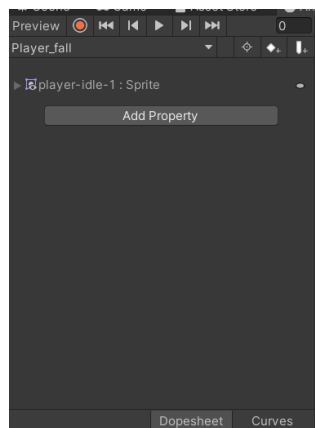


25. Kemudian buat animasi baru lagi dengan nama “Player\_fall” dan simpan pada *folder Animator*.



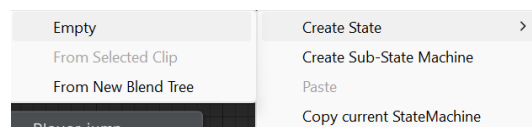
Gambar 9.22 Menambahkan Animasi Fall

26. Lalu tambahkan *asset player fall* pada *tab Animation*.



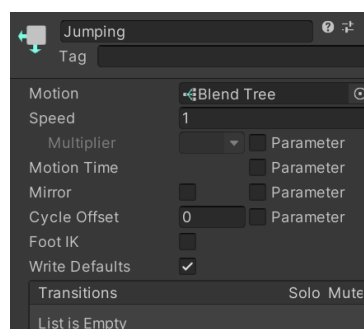
Gambar 9.23 Menambahkan Asset Fall

27. Lalu klik kanan pada *menu Animator* lalu pilih *Create State > From New Blend Tree*.



Gambar 9.24 Membuat Animasi Jump

28. Lalu klik *Blend Tree*, rubah namanya pada *inspector* menjadi *Jumping*.

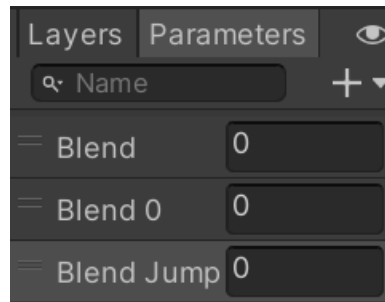


Gambar 9.25 Merubah Nama Blend Tree



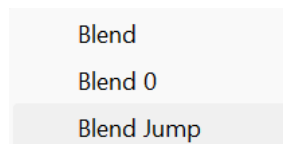


29. Tambahkan *parameter* baru dengan tipe data *Float* lalu rubah namanya menjadi “*Blend Jump*”.



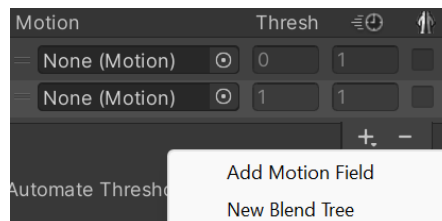
Gambar 9.26 Meletakkan Script CameraFollow

30. Klik 2 kali pada *Blend Tree* “*Jumping*” lalu pada *inspector* ubah parameternya menjadi “*Blend Jump*”.



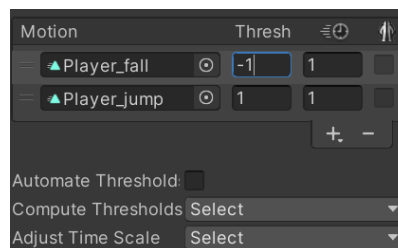
Gambar 9.27 Merubah Parameter

31. Tambahkan dua *Motion Field* dengan menekan *icon* tambah dan pilih *Add Motion Field*.



Gambar 9.28 Menambahkan Motion Field

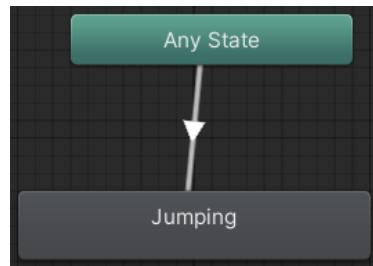
32. Pada bagian *None* tambahkan “*Player\_fall*” dan “*Player\_jump*”, Kemudian hilangkan centang pada “*Automate Thresholds*” dan berikan nilai seperti dibawah.



Gambar 9.29 Menambahkan Nilai Threshold

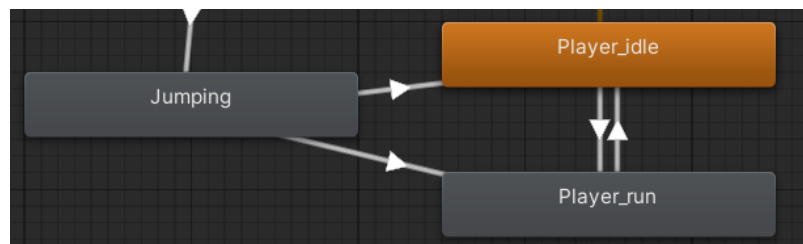


33. Pada *Base Layer* klik kanan pada *Any State* pilih *Make Transition* dan arahkan pada *Jumping*.



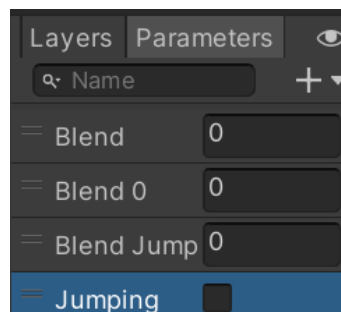
Gambar 9.30 Menghubungkan Dengan Jumping

34. Lalu hubungkan *Jumping* dengan *Player\_idle* dan *Player\_run*.



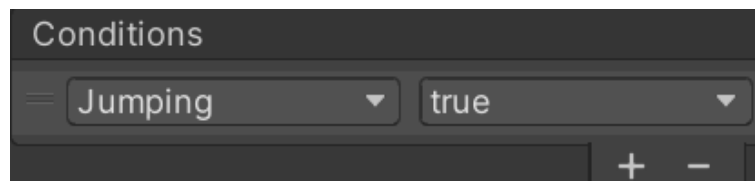
Gambar 9.31 Menghubungkan Jumping

35. Tambahkan *parameter* dengan tipe data *Bool* dan rubah namanya menjadi *Jumping*.



Gambar 9.32 Menambahkan Parameter Jumping

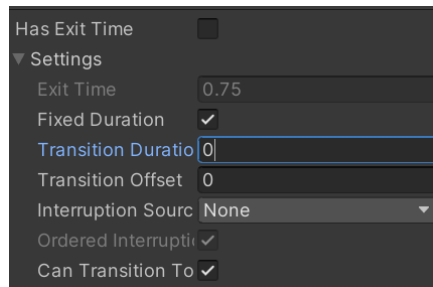
36. Klik panah yang mengarah ke *Jumping*, pada *inspector* tambahkan *condition Jumping* dan rubah nilainya menjadi *True*.



Gambar 9.33 Menambahkan Conditions Jumping

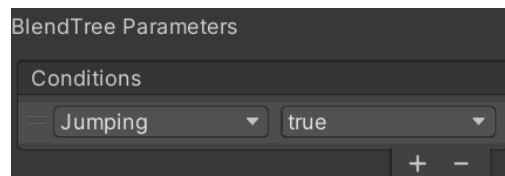


37. Lalu pada *Settings* hapus centang pada *Has Exit Time* dan rubah nilai *Transition Duration* menjadi 0.



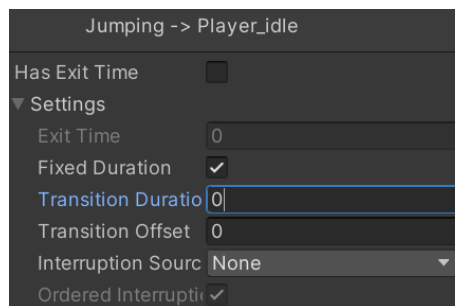
Gambar 9.34 Mengatur Settings

38. Lalu tambahkan *condition Jumping* pada panah Player\_idle dan Player\_run, kemudian untuk Player\_idle menjadi *false* dan Player\_run menjadi *true*.



Gambar 9.35 Menambahkan Conditions Jumping

39. Lalu pada *Setting* Player\_idle dan Player\_run berikan nilai 0 untuk *Transition Duration* dan hapus centang pada *Has Exit Time*.



Gambar 9.36 Mengatur Setting

40. Lalu pada *script Player* tambahkan *code* berikut pada fungsi *Update*.

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

41. Tambahkan *code* berikut pada fungsi *FixedUpdate*

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

42. Kemudian tambahkan *code* berikut pada *method* *GroundCheck()*.

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



43. Jika di play maka karakter sudah bisa bergerak dengan animasi



Gambar Hasil runing

## 2 Kuis

### A. Kesalahan

```
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 != 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);
    }
}
```



```
}
```

## B. Tambahan

```
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);
    }
    else
    {
        animator.SetBool("isJumping", false);
    }
}

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

    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);
    }

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

## Analisa

Kesalahan pada *source code* diatas terdapat pada fungsi *HandleJumpInput* yang memiliki kondisi *if else*, pada kondisi *if* tidak adanya parameter kedua yaitu bernilai *false* atau *true* yang harusnya jika *space* ditekan maka menghasilkan nilai *true*, maka ("*isJumping*", *true*) dan pada kondisi *else if* jika *space* masih ditekan maka kondisinya ("*isJumping*", *true*) kemudian



seharusnya ada kondisi lagi yaitu jika *space* tidak ditekan dengan menambahkan kondisi *else* (*"isJumping", false*). Lalu pada fungsi *HandleMovementInput* terdapat kesalahan pada kondisi yang ada, pada kondisi *if* yang awalnya (*!=1*) harusnya (*==0*) karena kondisi tersebut untuk memberi perintah animasi karakter tidak digerakkan. Maka pada kondisi tersebut memiliki perintah (*"isIdle", true*) kemudian (*"isWalking", false*) dan untuk *transform.Translate()* harusnya diletakkan pada kondisi jika karakter digerakkan yaitu kondisi *else* karena kondisi (ini merupakan perintah jika karakter digerakkan).