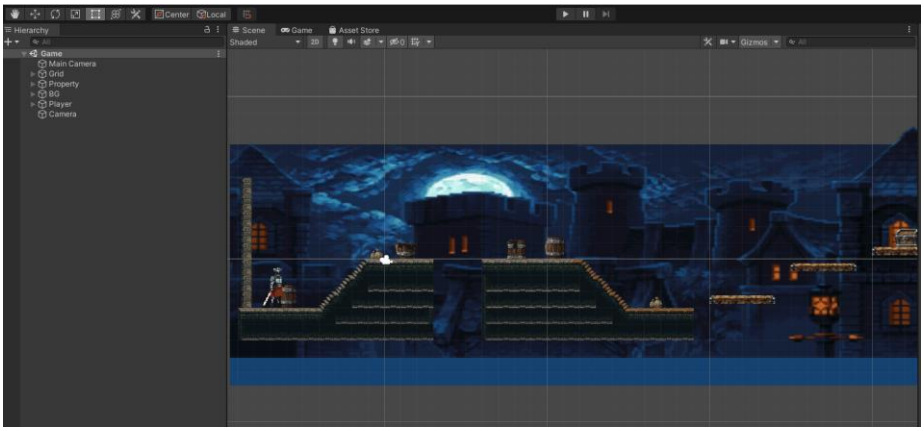
TUGAS PERTEMUAN: 9 GAME ANIMATION

NIM	:	2118080
Nama	:	Syahdan Izzur Rosuli
Kelas	:	B
Asisten Lab	:	M. Zainul Musyafa' (2118050)
Baju Adat	:	Cak dan Ning (Jawa Timur)
Referensi	:	https://cdnwpseller.gramedia.net/wp-content/uploads/2021/09/01210916/Baju-Cak-dan-Ning.jpg

9.1 Tugas 9 : Membuat Character Animation

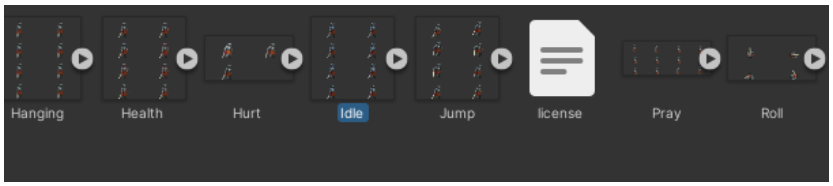
A. Membuat Pergerakan Player

1. Buka kembali file projek Unity sebelumnya pada tugas bab 8 untuk digunakan kembali.



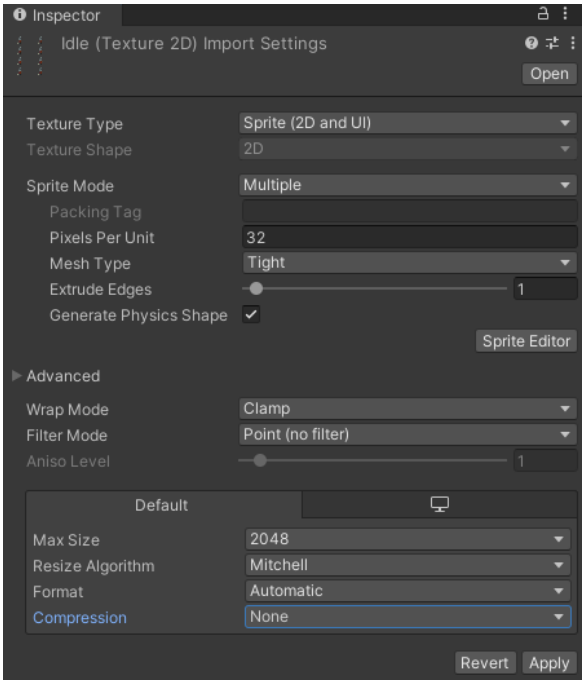
Gambar 9.1 Buka Project

2. Selanjutnya Pilih idle pada karakter.



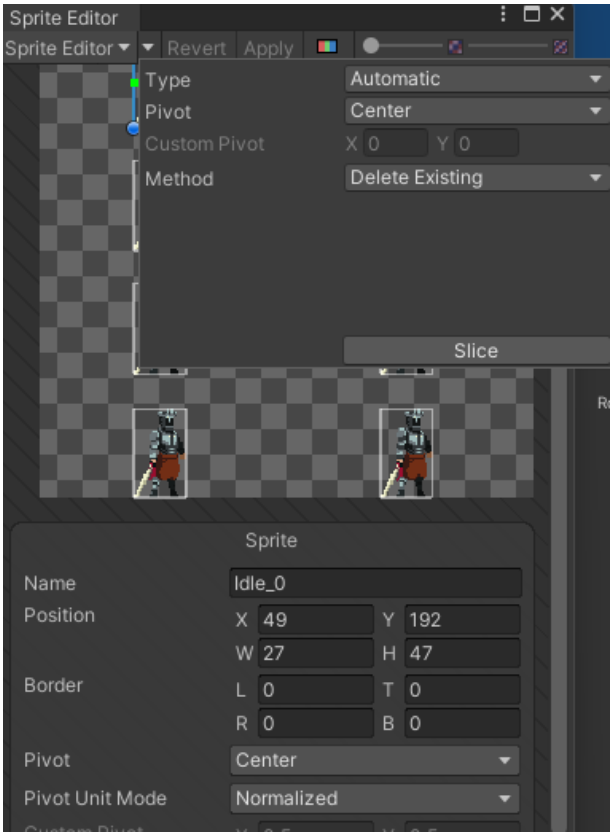
Gambar 9.2 Pilih Idle

3. Ubah Inspector seperti pada gambar



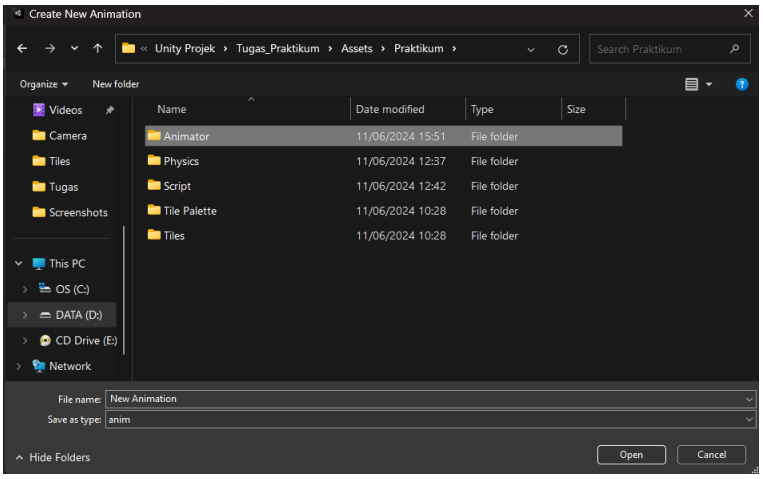
Gambar 9.3 Setting Idle character

4. Masuk ke sprite editor kemudian ubah bagian slice seperti dibawah kemudian klik apply.



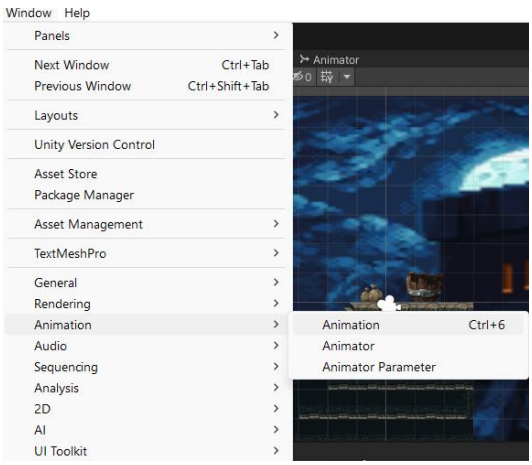
Gambar 9.4 Settingan sprite editor

5. Kemudian drag and drop animasi ke projek maka akan keluar tampilan untuk menyimpan folder animasinya.



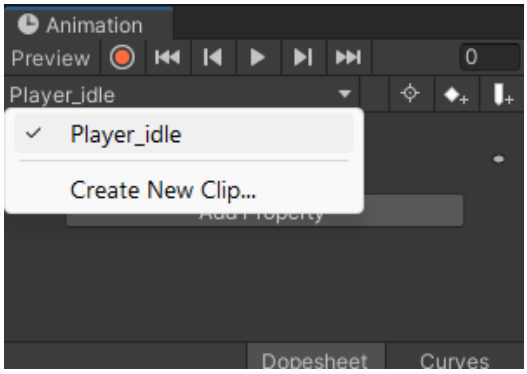
Gambar 9.5 Simpan animator

6. Pilih tab window kemudian pilih animation > animation untuk membuka tab animasi.



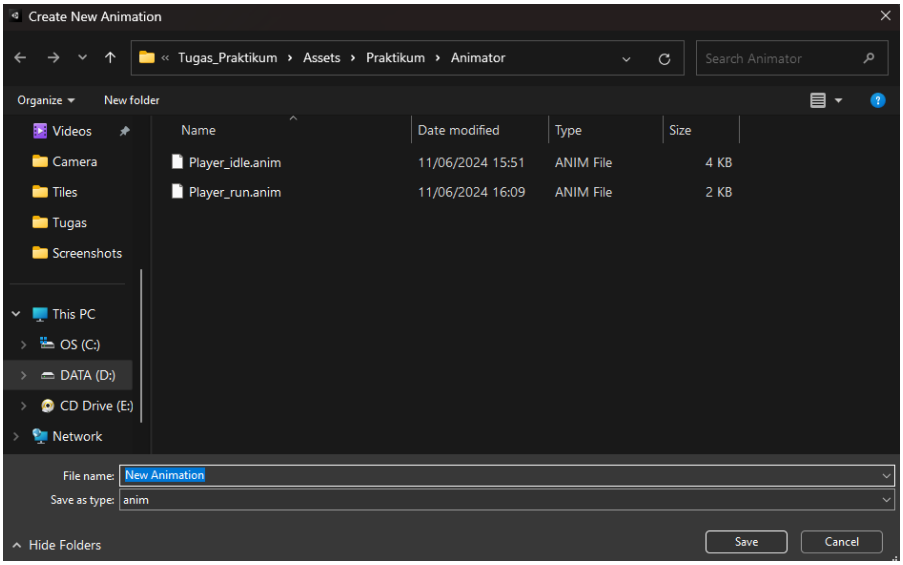
Gambar 9.6 Tab animation

7. Kemudian pada jendela animation pilih Player_idle kemudian create new clip.



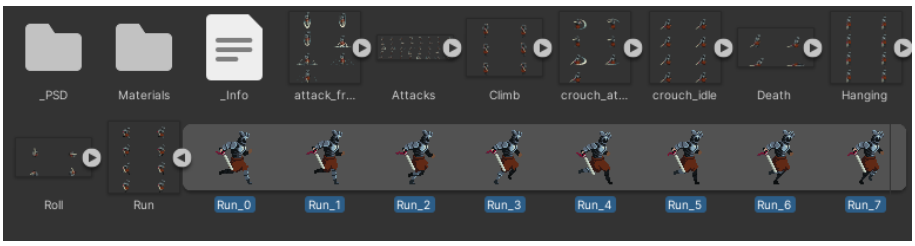
Gambar 9.7 Create new clip

8. Tambahkan player_run kemudian klik save.



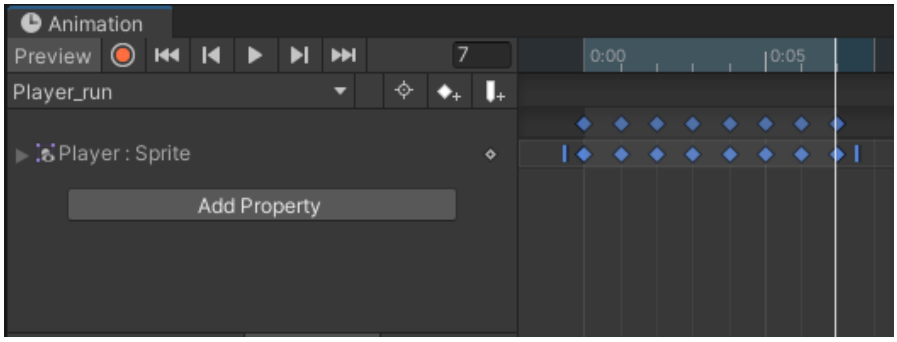
Gambar 9.8 Player_run

9. Lalu pilih run_0 hingga run_7 pada bagian asset.



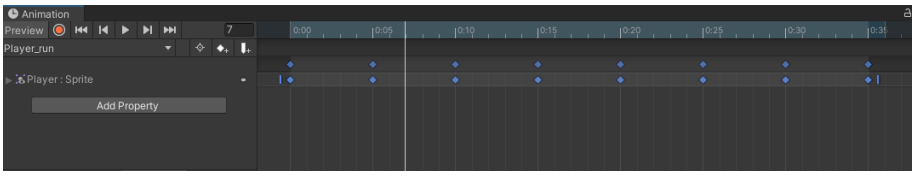
Gambar 9.9 Pilih animasi run

10. Selanjutnya masukkan ke timeline pada animation, pastikan berada pada Player_run.



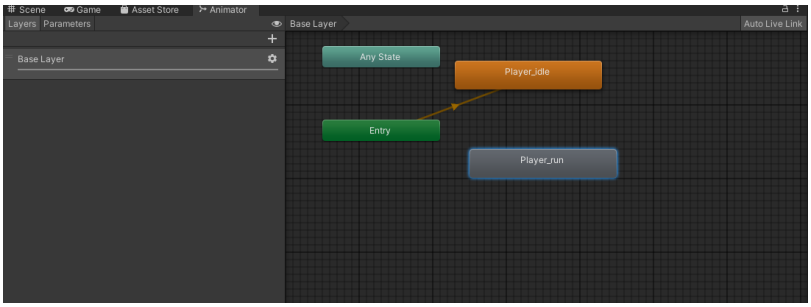
Gambar 9.10 Import ke timeline

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



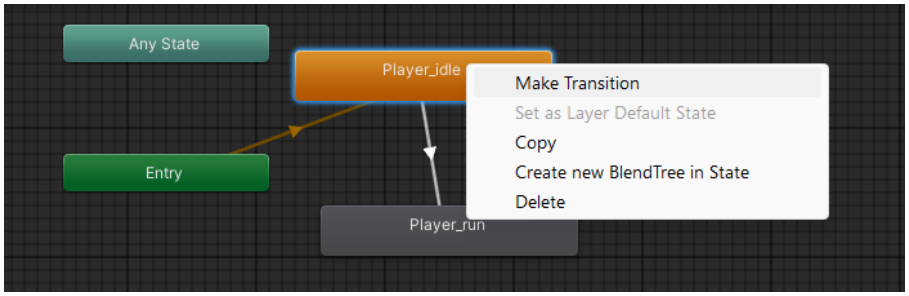
Gambar 9.11 Penyesuaian keyframe

12. pilih ke menu Animator maka akan muncul tampilan seperti berikut.



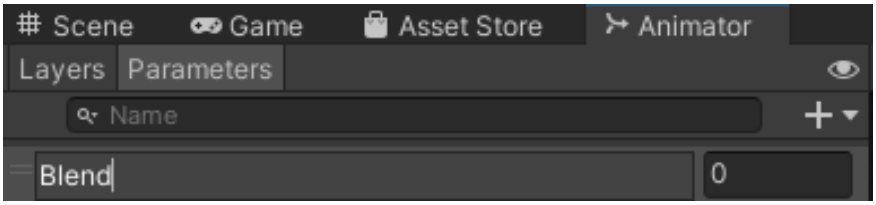
Gambar 9.12 Tampilan Animator

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.



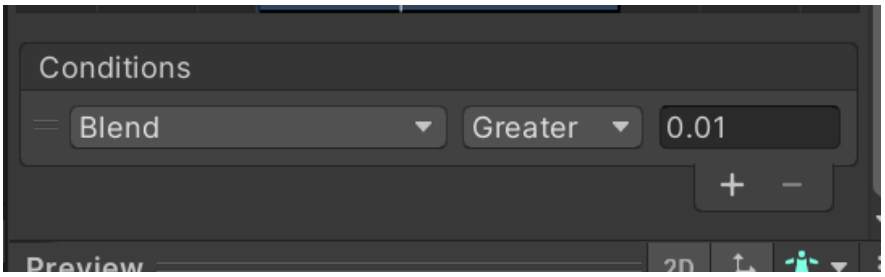
Gambar 9.13 Buat transisi dari idle menuju run

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



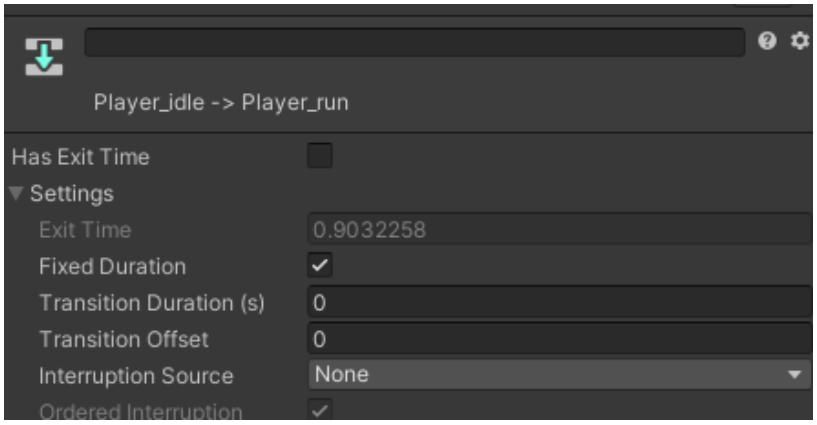
Gambar 9.14 Tambah tipe data

15. Klik panah putih tersebut, pada bagian conditions klik icon tambah kemudian atur menjadi “Blend” dan atur value menjadi 0.01.



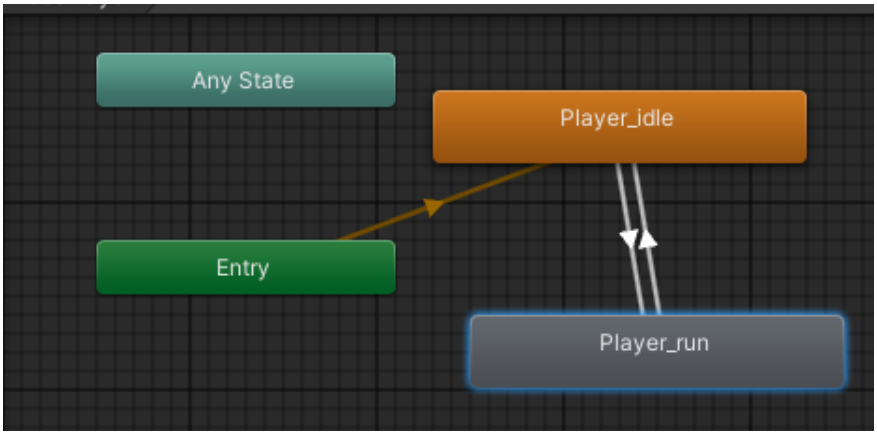
Gambar 9.15 Setting blend

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



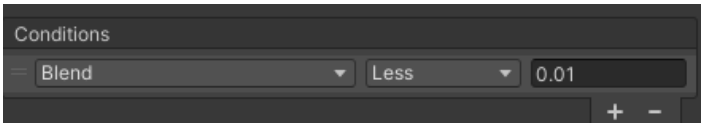
Gambar 9.16 Setting Transisi

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



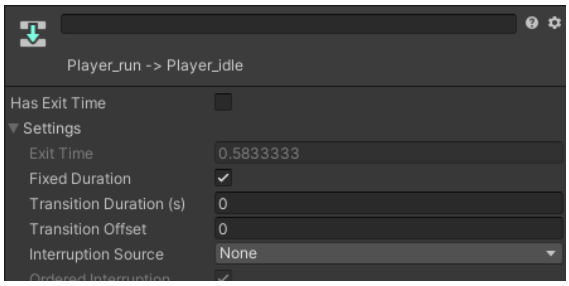
Gambar 9.17 Tambah Transisi Baru

18. 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 9.18 Buat parameter baru

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



Gambar 9.19 Ubah Settings

20. 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;
```

Gambar 9.20 Tambah Script

21. Tambahkan Script Komponen Animator.

```
private void Awake()
{
    rb = GetComponent<Rigidbody2D>();
    animator = GetComponent<Animator>();
    originalScale = transform.localScale; // Simpan skala asli
}
```

Gambar 9.21 Tambah komponen animator

22. Dan pada fungsi FixedUpdate tambahkan source code berikut.

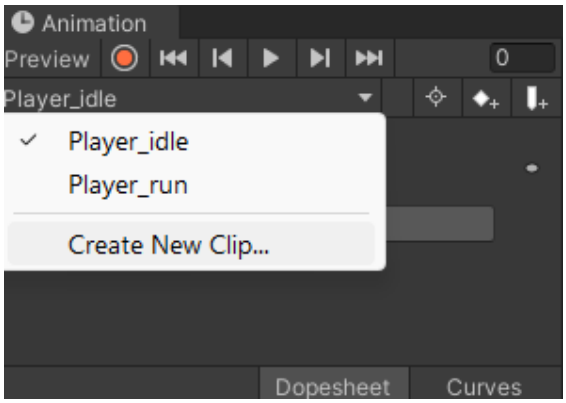
```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);

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

Gambar 9.22 Tambah fungsi

23. Selanjutnya tekan play, maka player telah memiliki animasi idle dan berlari.

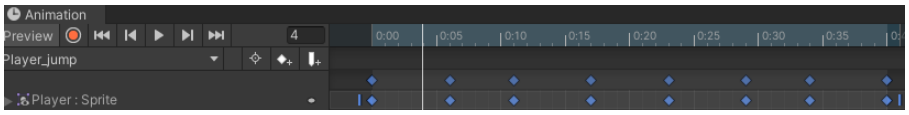
Gambar 9.23 Pengaplikasian Animation

24. Kemudian buat animasi baru kemudian pilih Create New Clip, dan beri nama “Player_jump”.



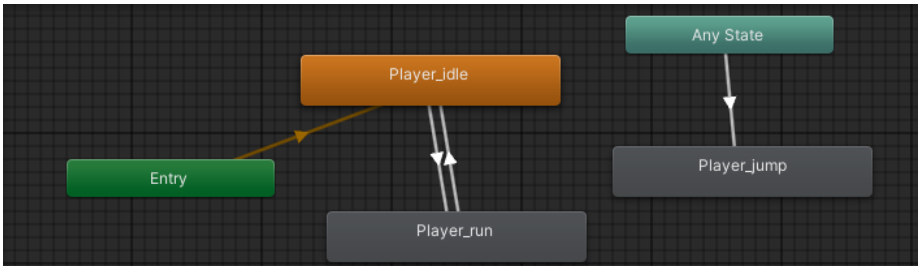
Gambar 9.24 Buat player jump

25. Lalu masukkan animasi jump pada folder asset dan sesuaikan keyframe.



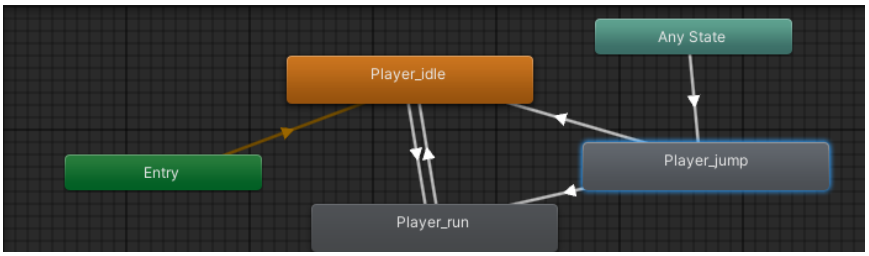
Gambar 9.25 Sesuaikan keyframe

26. Pada Animator, klik kanan Any State, pilih Make Transition dan arahkan panahnya ke Player_jump.



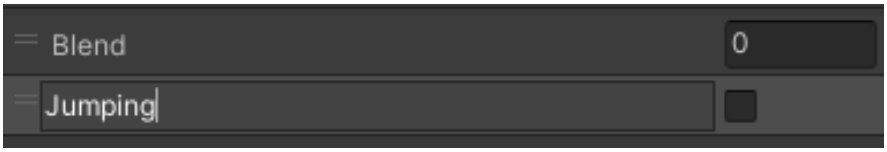
Gambar 9.26 Hubungkan ke jump

27. Klik kanan player_jump, pilih Make Transition dan arahkan panahnya ke Player_idle dan Player_run.



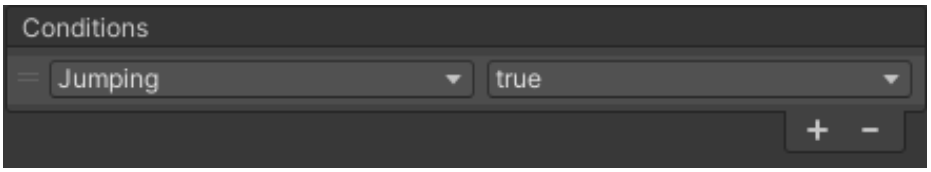
Gambar 9.27 Transisi dari player jump

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



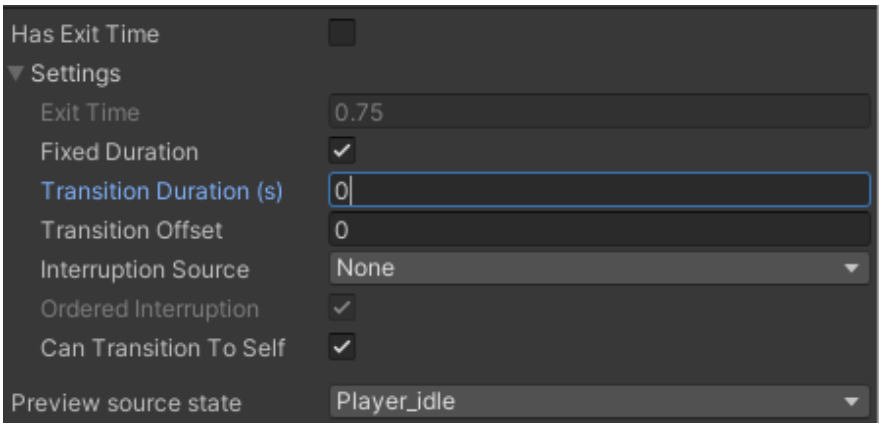
Gambar 9.28 Parameter jumping

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



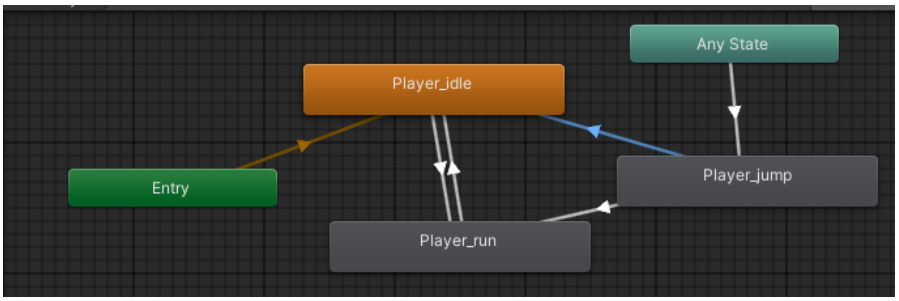
Gambar 9.29 Condition Jumping

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



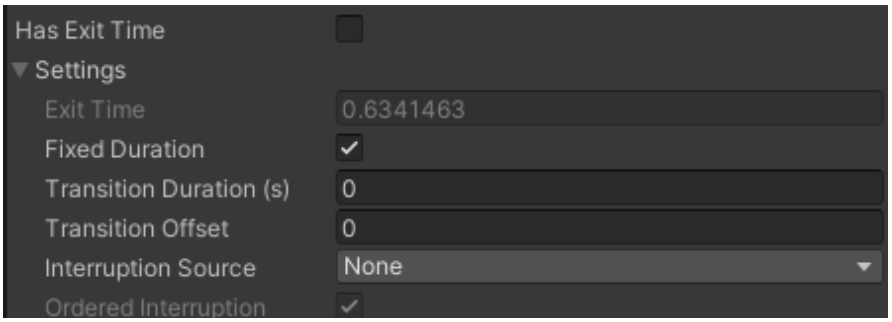
Gambar 9.30 Ubah Settings

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.



Gambar 9.31 Tambah Condition Jump

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



Gambar 9.32 Ubah Settings

33. 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;
    }
```

Gambar 9.33 Ubah fungsi Update

34. Tambahkan baris kode seperti dibawah ini dalam method GroundCheck

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

Gambar 9.34 Ubah Groundcheck

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

Gambar 9.35 Hasil Akhir