



TUGAS PERTEMUAN: 9

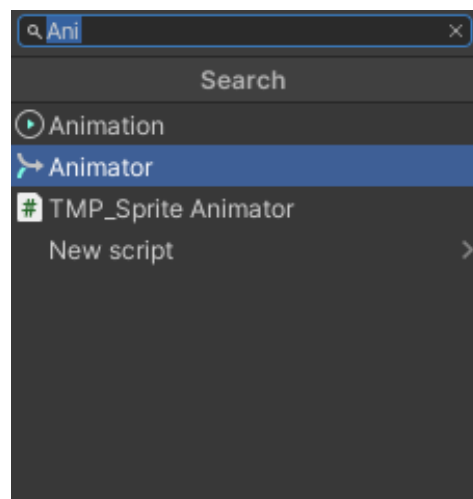
GAME ANIMATION

NIM	:	2118085
Nama	:	Muhammad Firman Romadhoni
Kelas	:	A
Asisten Lab	:	M. Zainul Musyafa' (2118050)
Nama Asset	:	Sunny Land Forrest

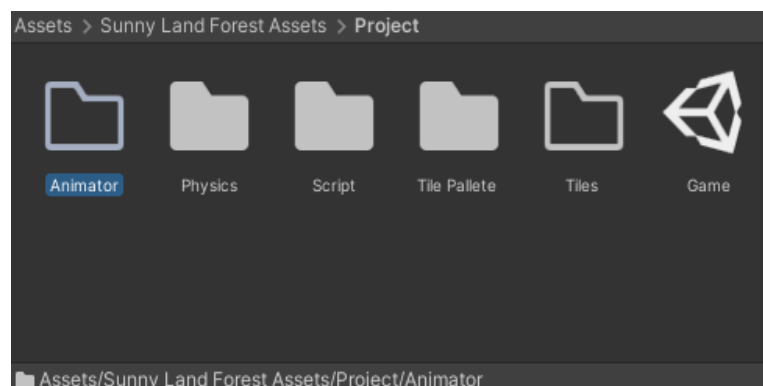
1.1 Tugas 1 : Langkah – Langkah Membuat Game Animation

A. Langkah – langkah membuat Character Animation

1. Pada karakter klik inspector kemudian pilih Add Component Animator



2. Pada **folder Praktikum** Buat Folder baru “**Animator**”.

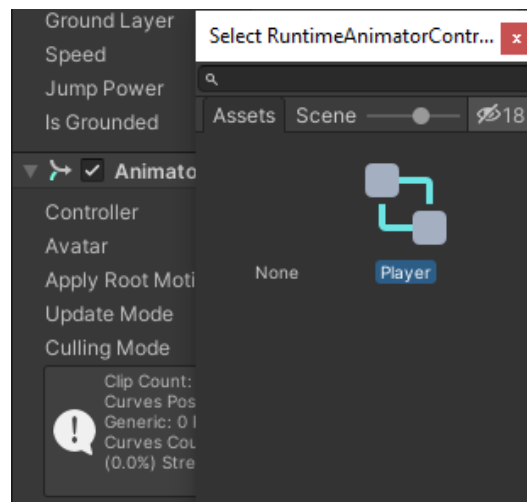




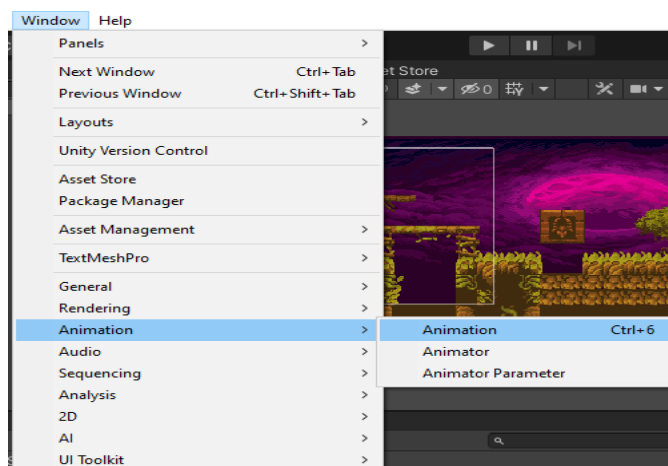
3. Buat File **Animator Controller** pada folder **Animator** , ubah namanya menjadi **Player**.



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

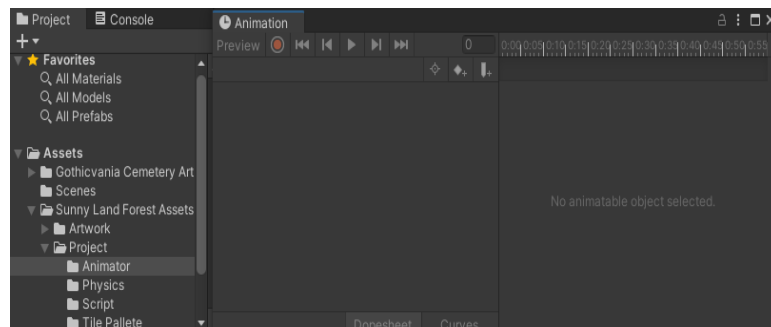


5. Tambahkan menu panel **Animation** di menu **Window**, pilih **Animation** > **Animation** atau tekan **Shortcut CTRL + 6**.

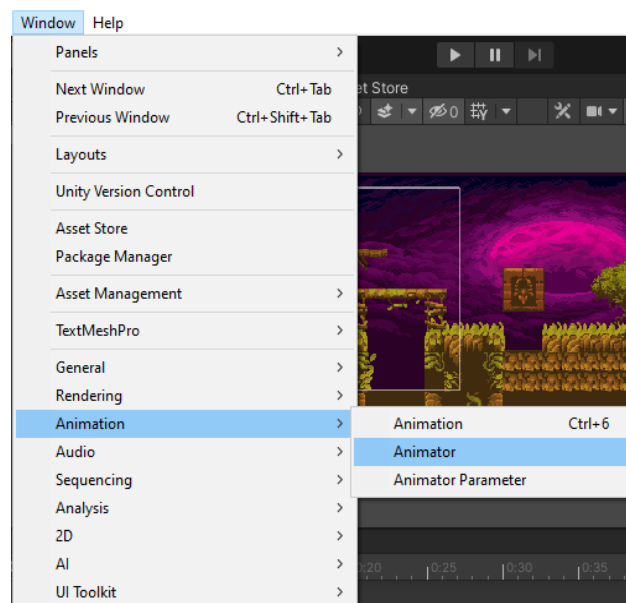




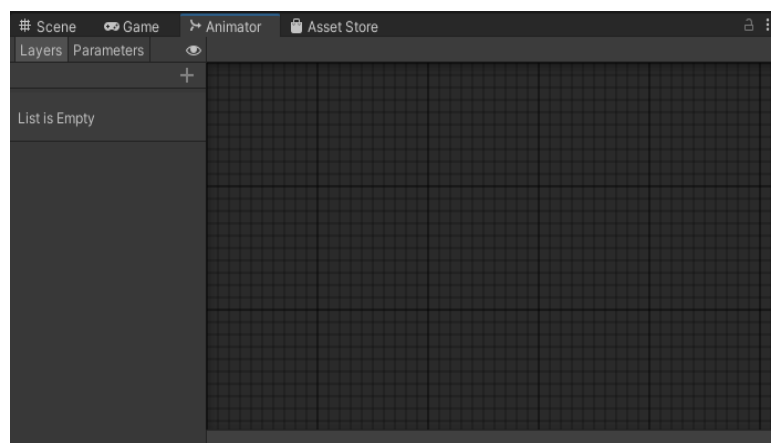
6. Akan muncul menu **panel** baru, geser **panel** tersebut dibawah sendiri seperti gambar dibawah ini, **Split Panel Project dan Animation.**



7. Tambahkan menu panel **Animator.**

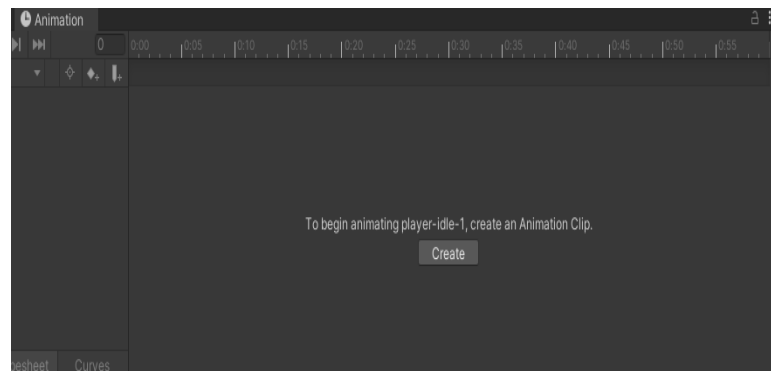


8. Geser Panel tersebut sesuai dengan gambar.

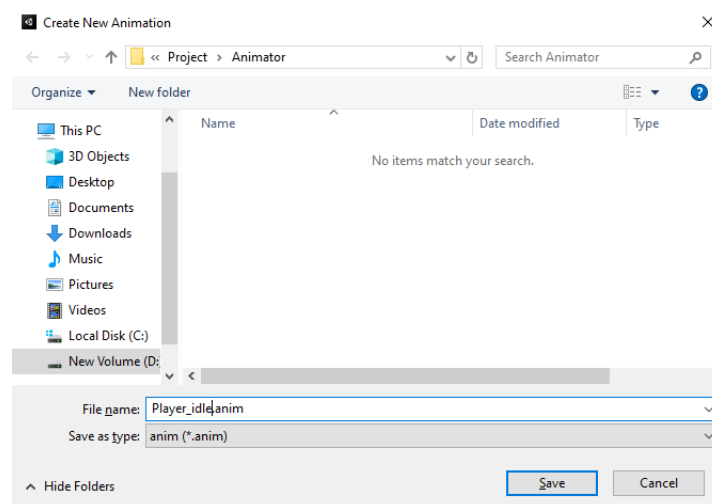




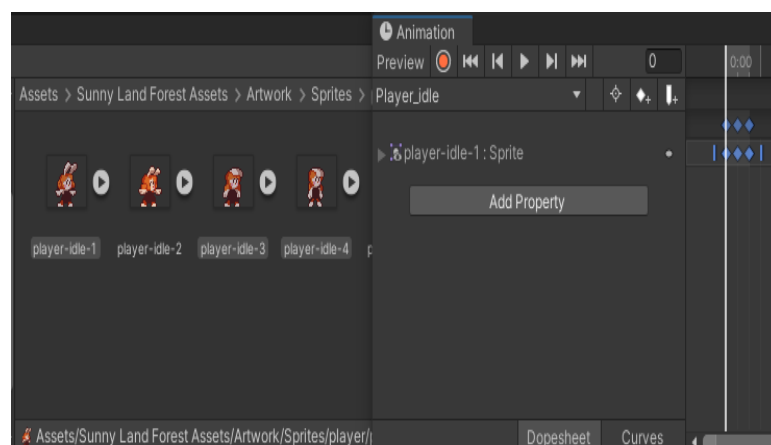
9. Untuk membuat animasi klik **player-idle1** pada **Hierarchy**, kemudian ke menu panel **Animation**, pilih **Create**.



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

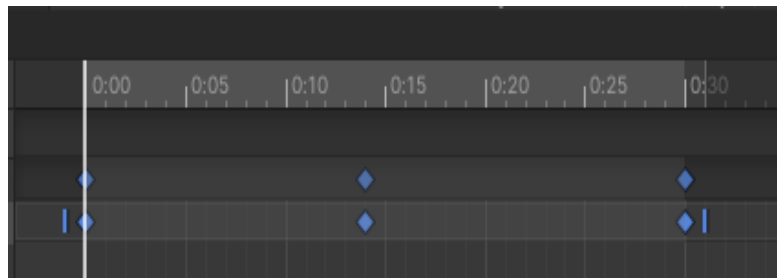


11. Pada menu Project buka folder player lalu pilih **Idle** dan pilih gambar player-idle-1, player-idle-3 dan player-idle-4, kemudian drag ke tab Animation.

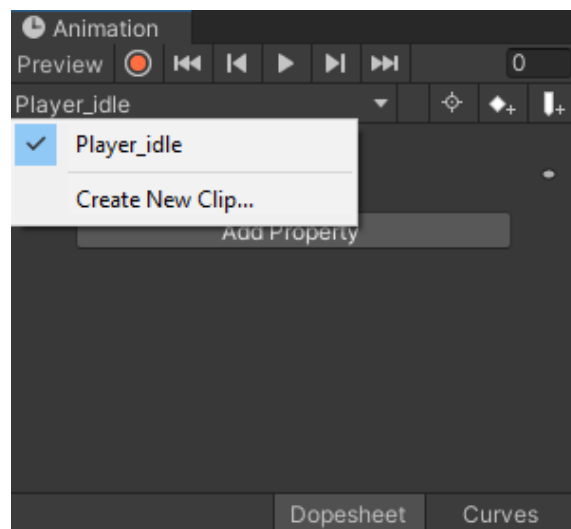




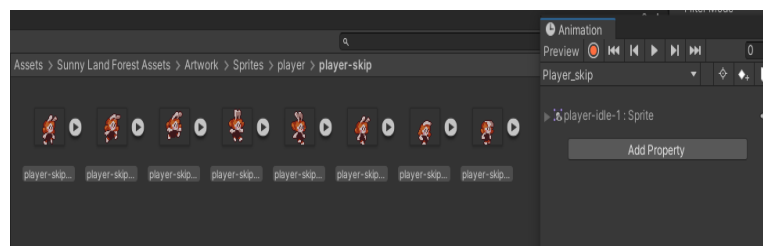
12. Tekan CTRL + A pada menu **panel Animation** geser kotak kecil pada timeline sampai frame 0:30 agar animasinya tidak terlalu cepat.



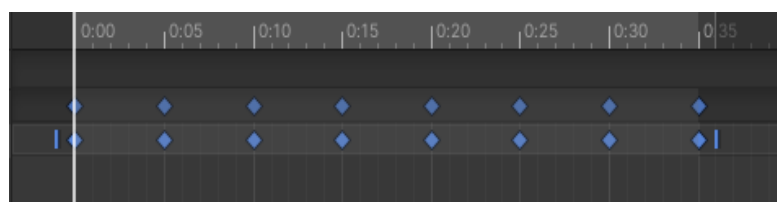
13. Buat animasi baru, Klik pada “Player_idle” kemudian pilih **Create New Clip**, dan beri nama “Player_skip”, Simpan pada Folder **Animator**.



14. Buka menu Project kemudian cari folder **Player** > **skip**, Pilih player-skip-1 sampai player-skip-8, drag and drop pada menu **Animation**.

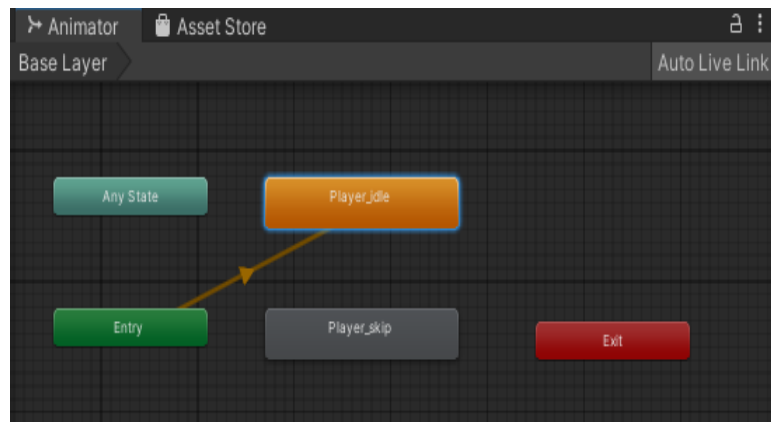


15. pada panel **timeline** tekan Ctrl+A di keyboard, klik bagian kotak kecil disamping keyframe terakhir dan geser sampai waktu 0:35.

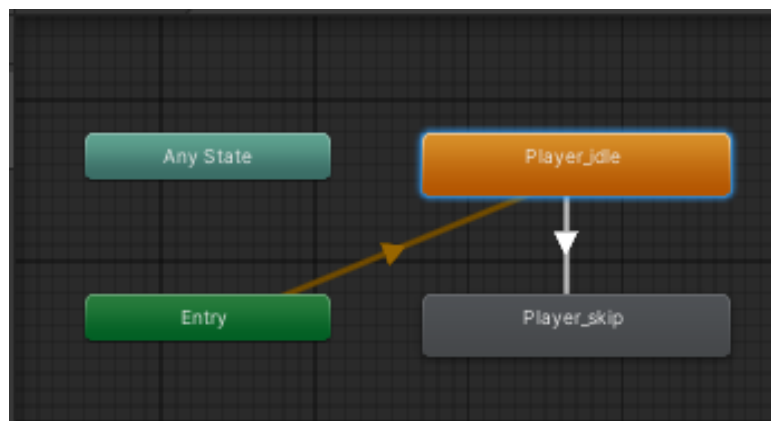




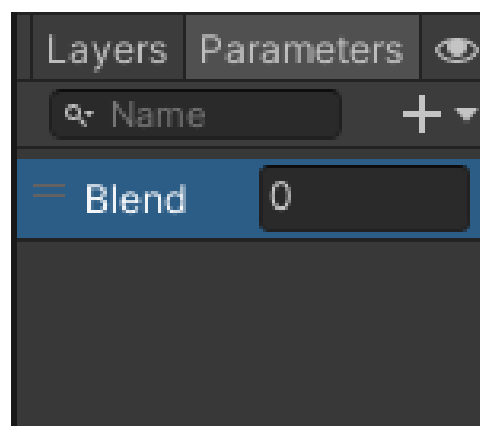
16. pilih ke menu Animator yang telah dibuka sebelumnya dan akan tampil seperti berikut.



17. Kemudian buat transisi antara player_idle dan player_run dengan cara klik kanan pada player_idle dan pilih **Make Transition** dan tarik ke player_skip.

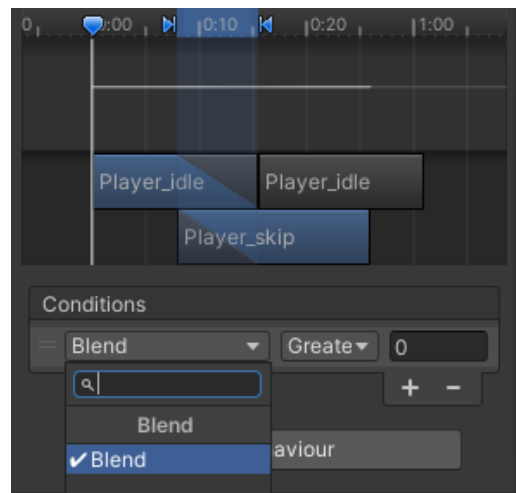


18. Masuk ke tab parameter, tambahkan tipe data bdengan cara tekan icon tambah dan ubah namanya menjadi “Blend”.

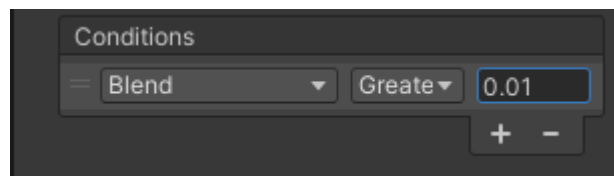




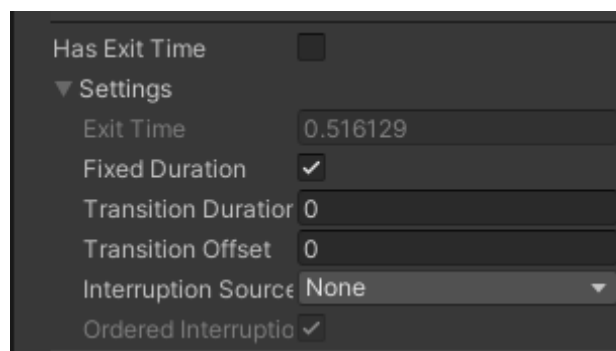
19. Klik panah putih tersebut, pada bagian conditions klik icon tambah kemudian atur menjadi “Blend”.



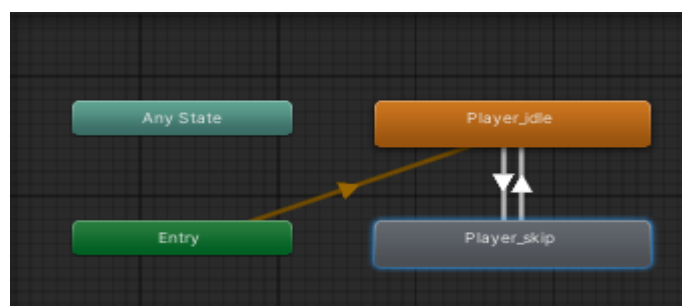
20. Atur nilai conditions blend tersebut menjadi 0.01.



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

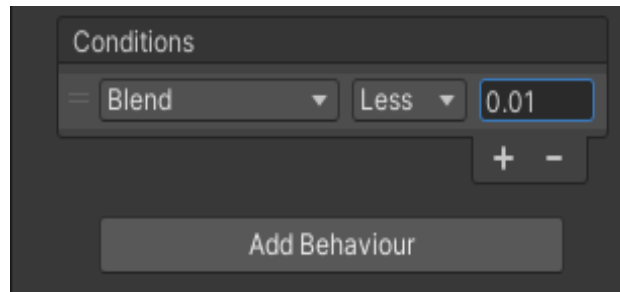


22. Buat transisi juga dari player_skip ke player_idle dengan cara klik kanan pada **player_skip** dan pilih **Make Transition**.

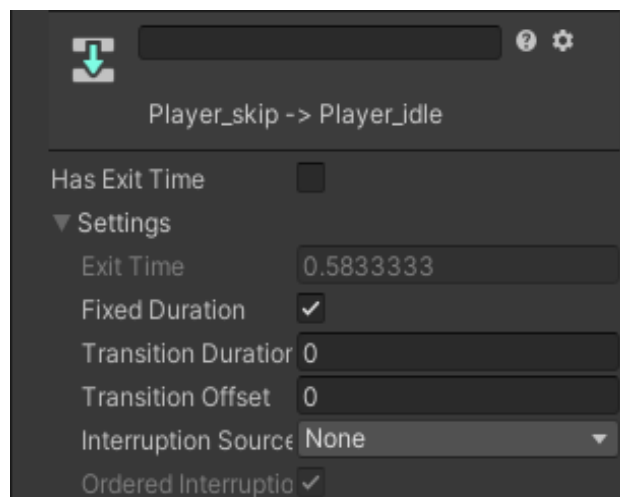




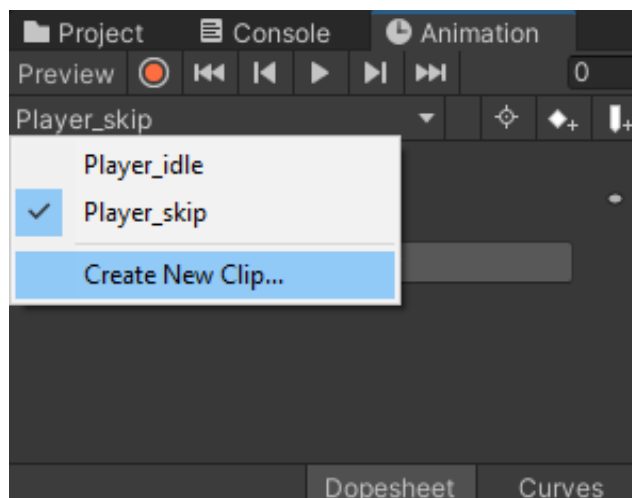
23. Tambahkan parameter transisi dengan tipe data Float. Klik ikon tambah dan rename menjadi “Blend”. Setelah itu, ubah operator dari Greater menjadi Less dan atur nilainya menjadi 0.01.



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

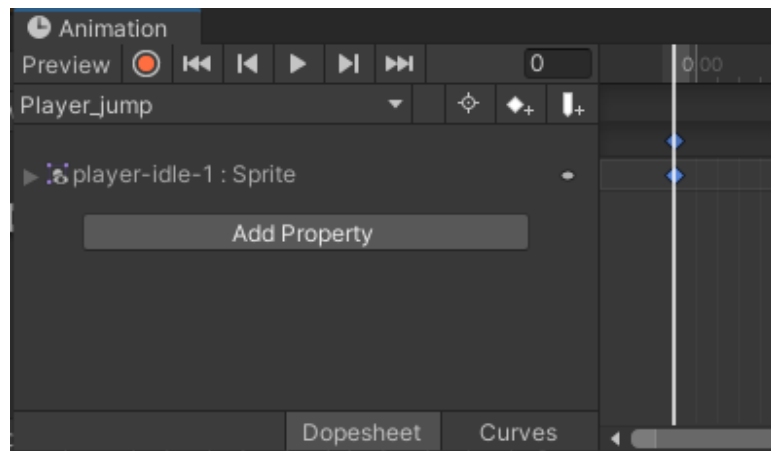


25. Kemudian buat animasi baru tekan tulisan “Player_skip” kemudian pilih **Create New Clip**, dan beri nama “Player_jump”.

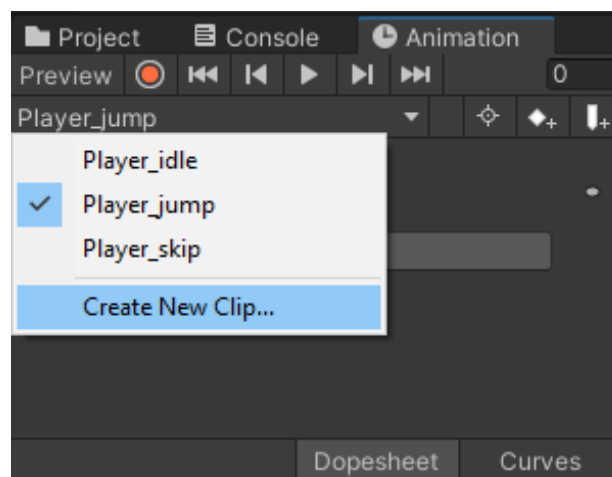




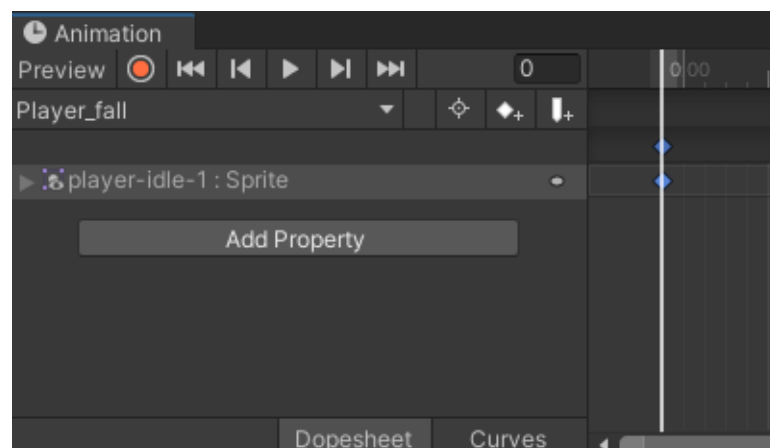
26. Pada folder player buka jump lalu pilih gambar player-jump-1, kemudian drag ke tab Animation.



27. Buat animasi baru dengan cara tekan tulisan "Player_jump" kemudian pilih **Create New Clip**, dan beri nama "Player_fall".

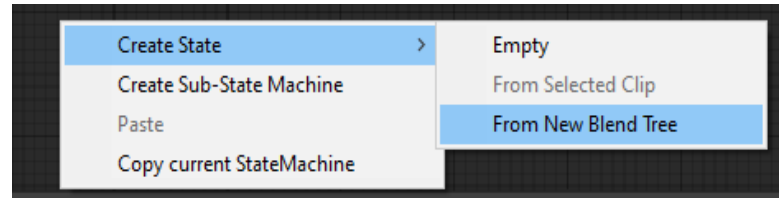


28. Pada tab Project buka folder karakter lalu pilih Idle dan pilih gambar player-fall, kemudian drag ke tab Animation.

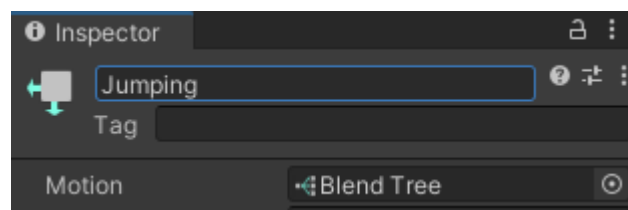




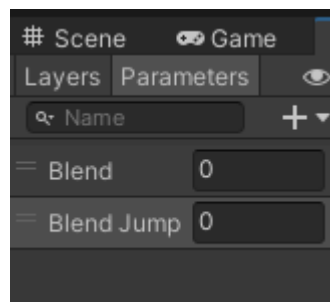
29. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong , pilih Create State>From New Blend Tree.



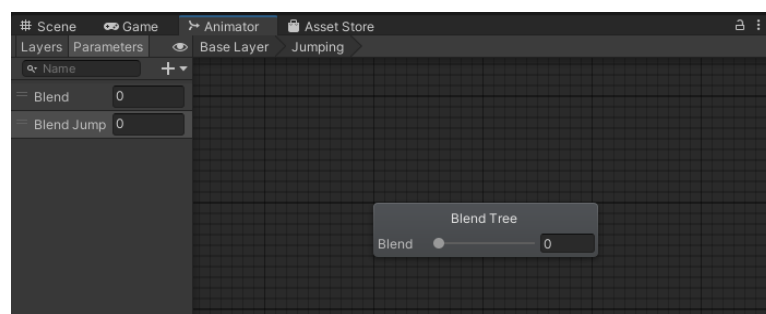
30. Pada Animator klik Blend Tree, di menu Inspector, ubah namanya menjadi Jumping.



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

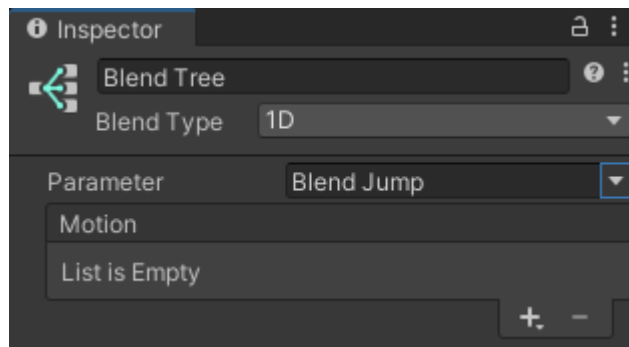


32. Pada menu Animator, Klik dua kali pada Blend Tree “Jumping”, Tekan pada Blend Tree.

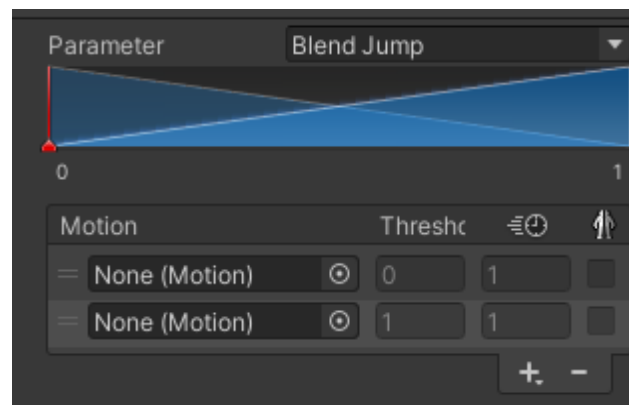




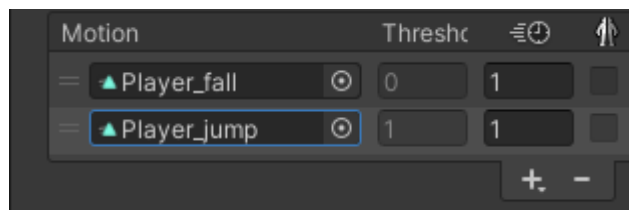
33. Klik 2X Blend Tree “Jumping”, pada inspector ubah parameter menjadi “Blend Jump”.



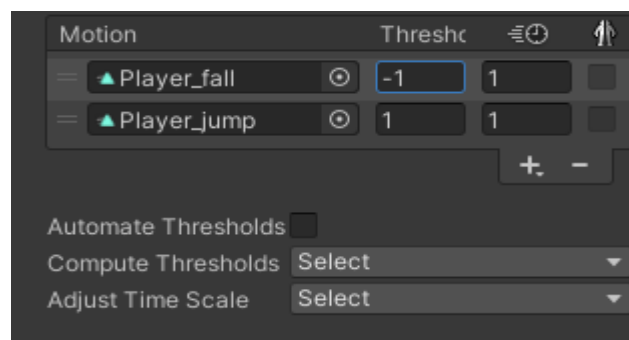
34. Buka menu Inspector, tekan icon + dan pilih **Add Motion Field**.
Tambahkan dua Motion Field.



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

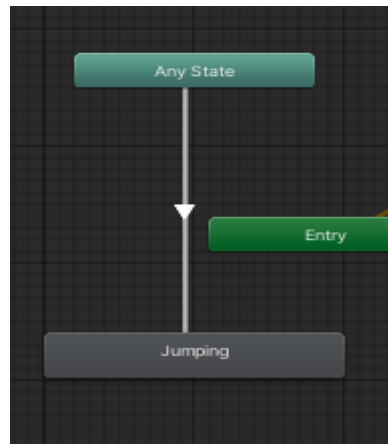


36. Hilangkan centang “Automate Thresholds” dan atur nilai Threshold seperti berikut.

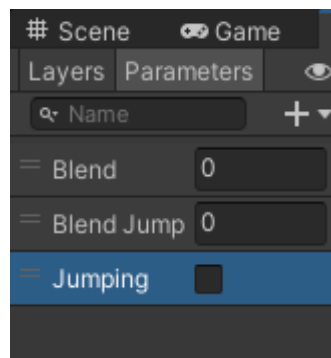




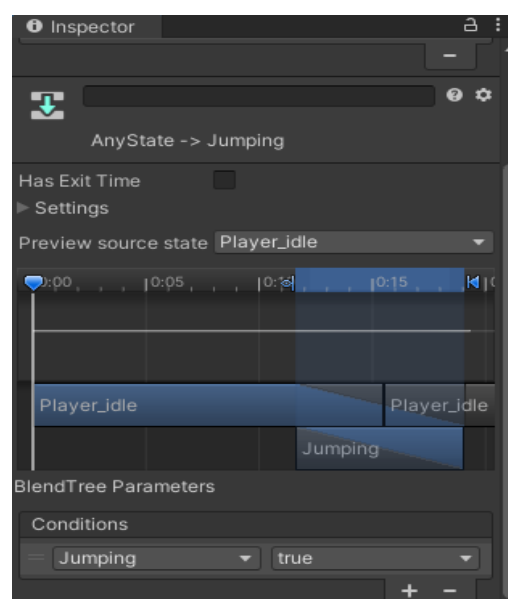
37. Kembali ke Base Layer, klik kanan Any State, pilih Make Transition dan arahkan panahnya ke Jumping.



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

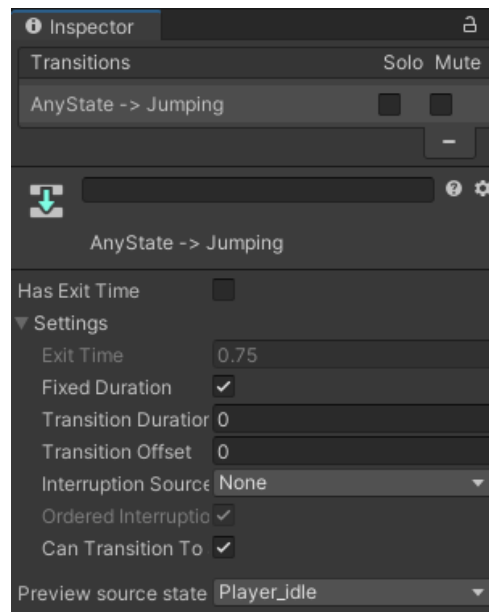


39. Klik panah yang mengarah ke Jumping, pada inspector tambahkan condition, pilih condition Jumping dan ubah nilainya menjadi true.

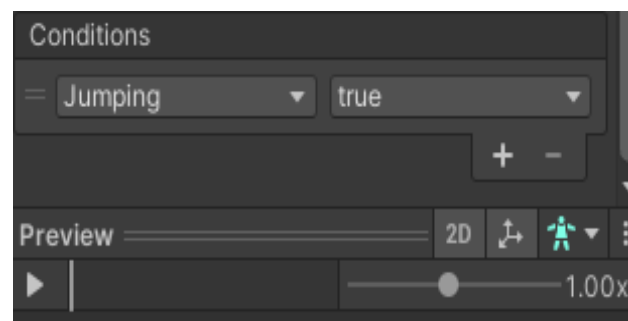
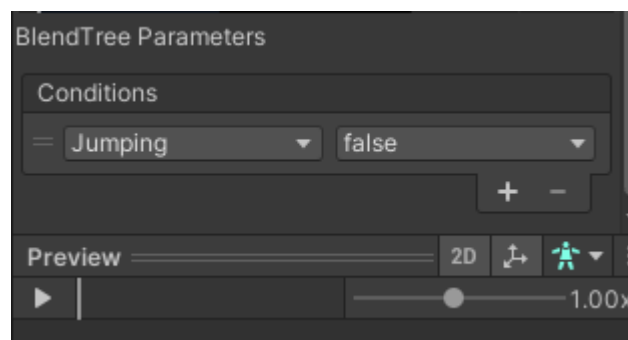




40. Klik Settings dan ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time.

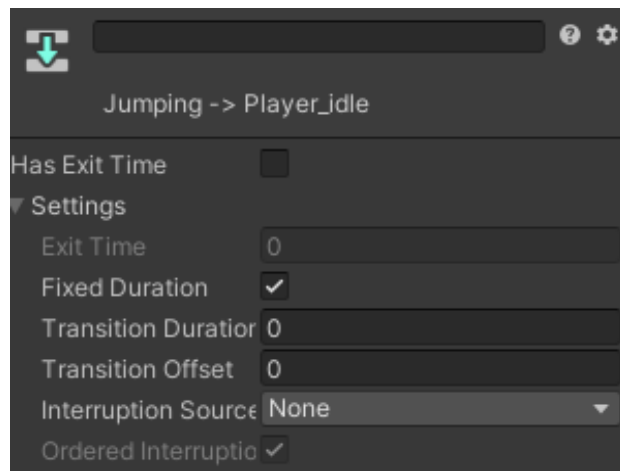


41. Klik panah yang mengarah ke **Player_idle** dan **Player_skip**, pada **inspector** tambahkan **condition**, pilih **condition Jumping**, pada arah panah ke **player_idle** ubah menjadi **false**, pada arah panah ke **player_skip** ubah menjadi **true**.





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



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



B. Kuis Pertemuan 9

```
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);
    }
}
```

Penjelasan :

Kesalahan pada SetBool dalam HandleJumpInput dan HandleMovementInput, nilai boolean untuk SetBool harus ditentukan apakah nilainya true atau false. Kesalahan logika dalam HandleMovementInput, Ketika move tidak sama dengan 1, menetapkan animator ke "isIdle", dan ketika sama dengan 1, menetapkan "isWalking" ke false. Logika ini bisa diubah sesuai kebutuhan animasi. Ada ketidaksesuaian dalam pemeriksaan nilai `move` untuk pembalikan skala.

Perbaikan Code :

```
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);
        }
    }
}
```

Penjelasan :

HandleJumpInput(): Menambahkan nilai boolean (true atau false) ke metode SetBool dari animator, Mengatur isJumping menjadi false jika tidak ada input Space.

HandleMovementInput(): Mengatur isIdle menjadi true jika move adalah 0, dan isWalking menjadi false, Mengatur isIdle menjadi false dan isWalking menjadi true jika move tidak 0, Mengubah posisi objek dengan transform.Translate, Mengatur skala objek untuk menghadap arah yang benar sesuai dengan nilai move.