



## TUGAS PERTEMUAN: 9

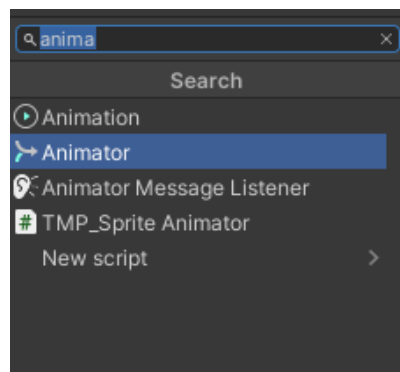
### GAME ANIMATION

NIM	:	2118027
Nama	:	RAFID ARTUR PRASETYA
Kelas	:	C
Asisten Lab	:	APRILLIA DWI DYAH S. (2118143)

#### 1.1 Tugas 1 : Membuat Character Animation

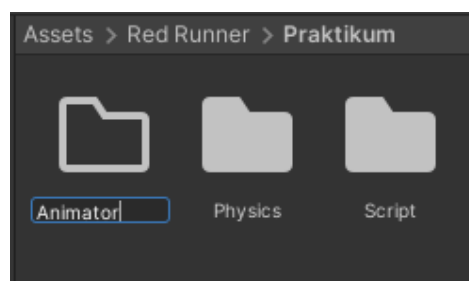
##### A. Character Animation

1. Klik character pada hirarki, pada Inspector tambahkan komponen “Animator”.



Gambar 9.1 Add Component Animator

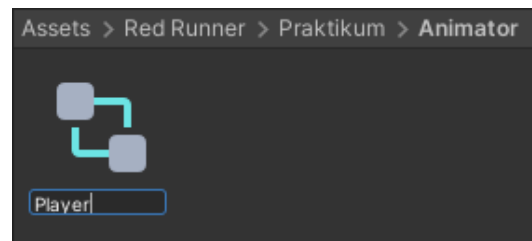
2. Buat folder baru dalam folder Praktikum, kemudian beri nama Animator.



Gambar 9.2 Membuat Folder Animator

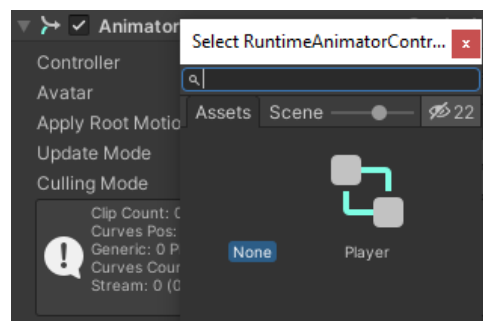


3. Masuk ke dalam folder Animator yang sudah dibuat, kemudian buat Animator Controller dan beri nama Player.



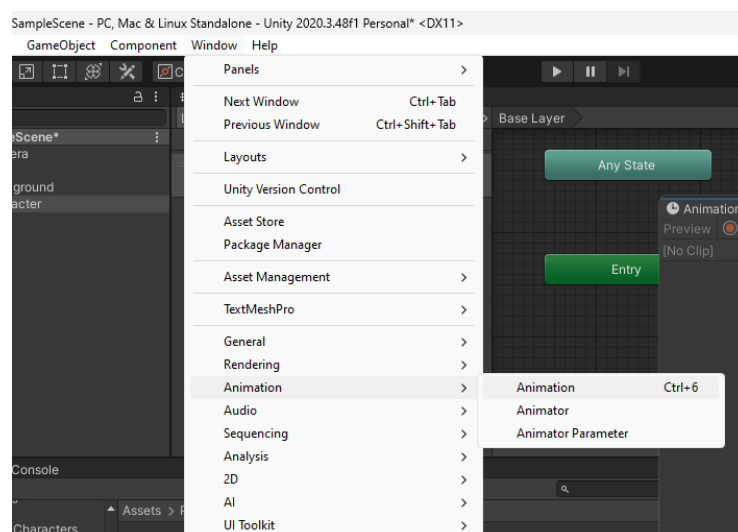
Gambar 9.3 Membuat Animator Controller

4. Kembali ke Inspector character, pada komponen Animator yang sudah ditambahkan sebelumnya, pada Controller pilih Player (klik 2 kali untuk apply).



Gambar 9.4 Select Player Controller

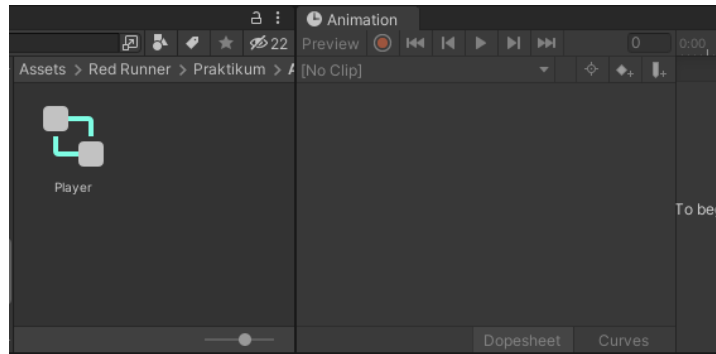
5. Tambahkan menu panel Animation seperti pada gambar di bawah ini atau dengan shortcut Ctrl+6.



Gambar 9.5 Menambah Menu Animation

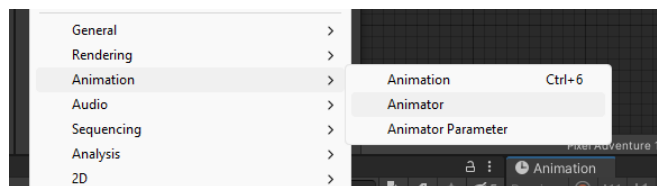


6. Letakkan di bawah tepat di samping panel Project.



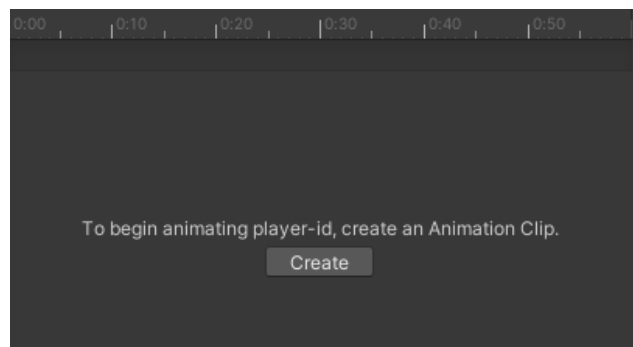
Gambar 9.6 Mengatur Letak Panel Animation

7. Tambahkan juga menu panel Animator. Secara otomatis dia akan berada di samping menu panel Game.



Gambar 9.7 Menambah Panel Animator

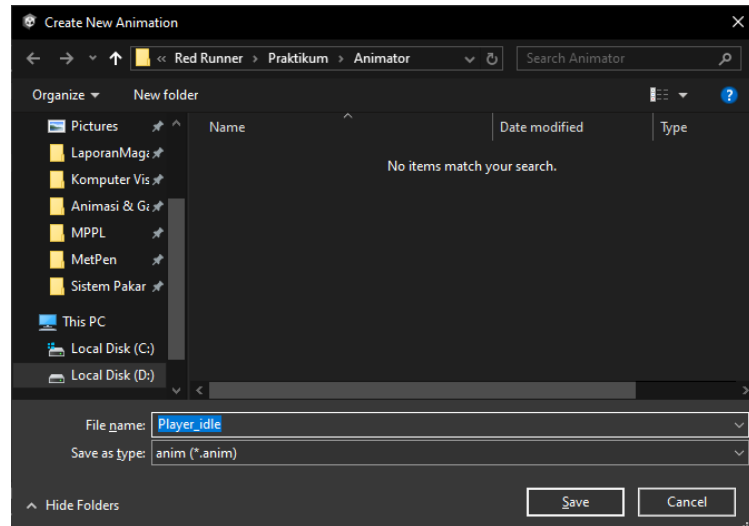
8. Untuk membuat animasi klik character pada Hirarki, kemudian pada panel Animation pilih Create.



Gambar 9.8 Create Animation

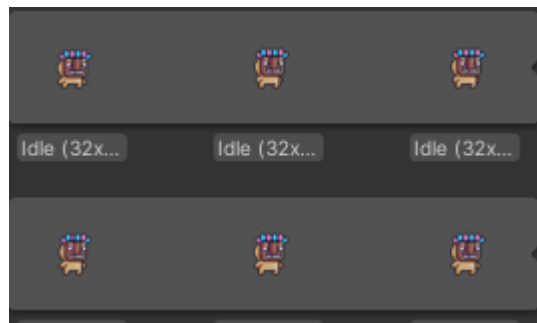


9. Simpan pada folder Animator.



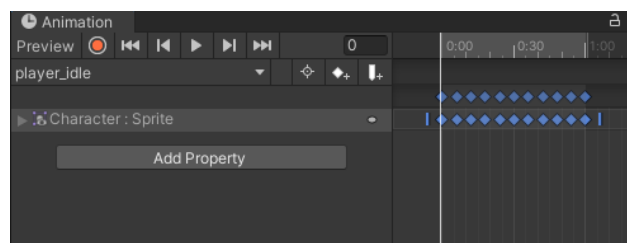
Gambar 9.9 Menyimpan Animasi

10. Pada folder Character Mage, cari folder Idle dan pilih semua karakter.



Gambar 9.10 Memilih Semua Karakter Idle

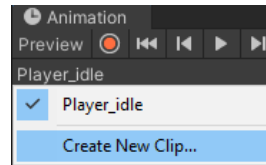
11. Kemudian drag and drop ke tab Animation. Atur durasinya menjadi 3 menit agar gerakannya tidak berdekatan.



Gambar 9.11 Drag and Drop ke Tab Animation

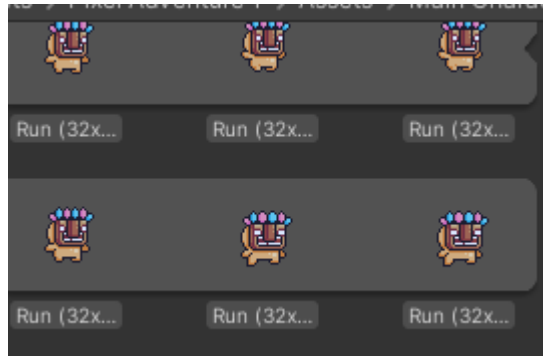


12. Create New Clip dan beri nama Player\_run.



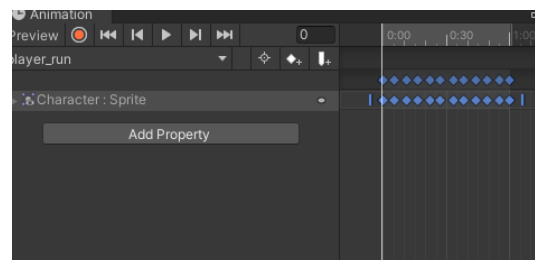
Gambar 9.12 Membuat Clip Player\_run

13. Pilih semua karakter pada folder Run.



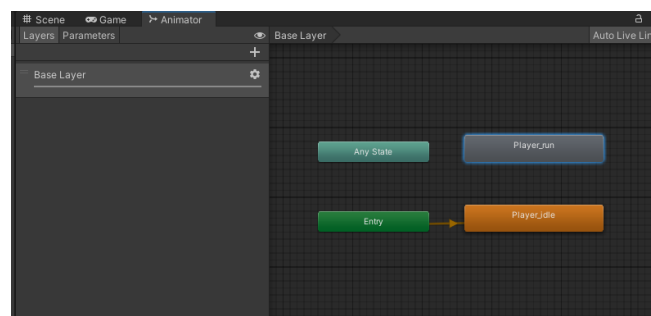
Gambar 9.13 Memilih Semua Karakter Run

14. Kemudian drag and drop ke tab Animation. Atur durasinya menjadi 45 detik agar gerakannya tidak berdekatan.



Gambar 9.14 Drag and Drop ke Tab Animation

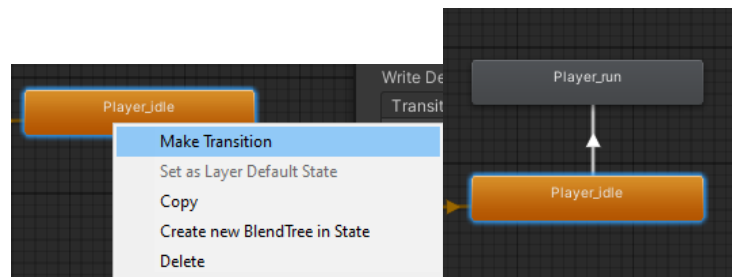
15. Pergi ke menu panel Animator, maka akan tampil seperti berikut.



Gambar 9.15 Tampilan Menu Panel Animator

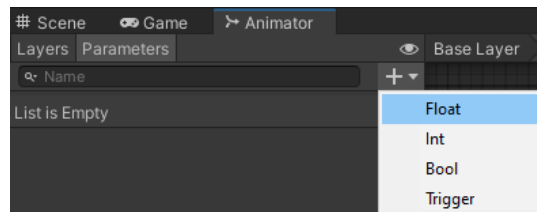


16. Klik kanan pada Player\_idle, kemudian Make Transition dan arahkan panahnya ke Player\_run.



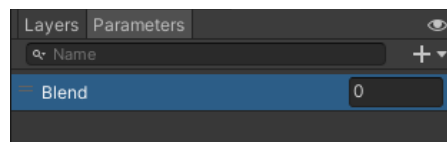
Gambar 9.16 Membuat Transition Player\_idle ke Player\_run

17. Masuk ke tab parameter, buat tipe data baru dengan cara klik pada ikon +, kemudian pilih yang Float.



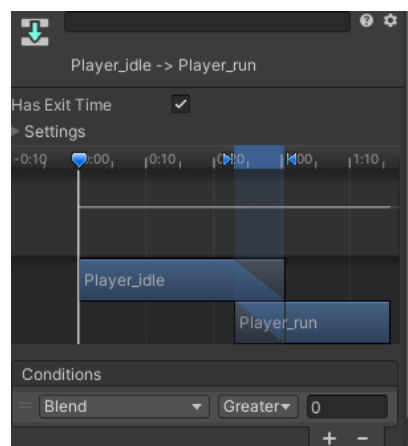
Gambar 9.17 Membuat Tipe Data Baru

18. Beri nama Blend.



Gambar 9.18 Memberi Nama pada Tipe Data

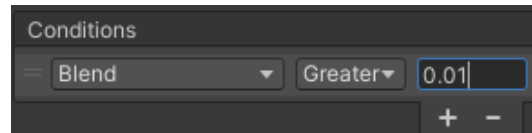
19. Klik panah putih yang mengarah dari Player\_idle ke Player\_run, pada Inspector tambahkan Conditions dan atur menjadi “Blend”.



Gambar 9.19 Menambah Conditions

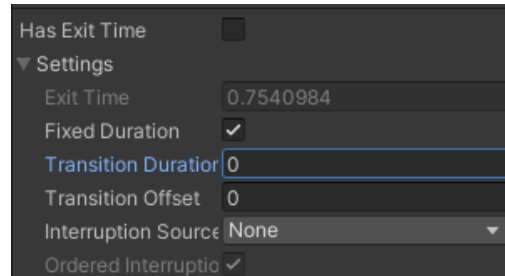


20. Ubah nilai conditions menjadi 0,01.



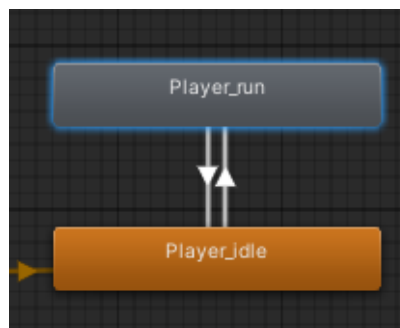
Gambar 9.20 Mengubah Nilai Conditions

21. Pada Setting ubah Transition Duration menjadi 0 dan hilangkan centang pada Has Exit Time.



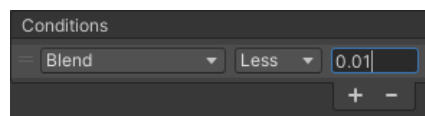
Gambar 9.21 Mengubah Setting

22. Klik kanan pada Player\_run dan buat transisi ke Player\_idle.



Gambar 9.22 Membuat Transition ke Player\_idle

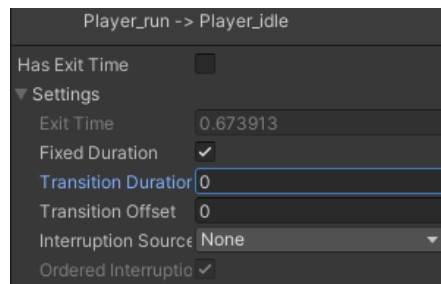
23. Klik pada panah yang baru saja dibuat. Tambahkan juga Conditions Blend dan ubah dari Greater menjadi Less, serta atur nilainya menjadi 0,01.



Gambar 9.23 Mengatur Conditions



24. Atur juga pada Setting yaitu beri nilai 0 pada Transition Duration dan hilangkan centang pada Has Exit Time.



Gambar 9.24 Mengatur Setting

25. Buka script Player, kemudian tambahkan source code di bawah ini. Letakkan dalam kelas Player dan paling atas.

```
public Animator animator;
```

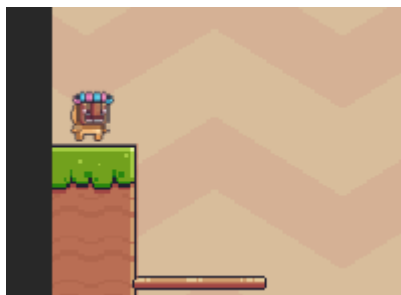
26. Tambahkan script komponen Animator pada fungsi void Awake().

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

27. Dan pada fungsi FixedUpdate tambahkan source code berikut.

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

28. Apabila dijalankan maka Player mempunyai animasi ketika berhenti dan berjalan.

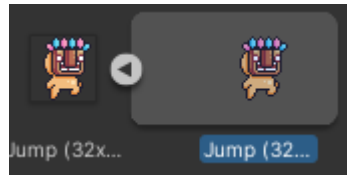


Gambar 9.25 Hasil Play



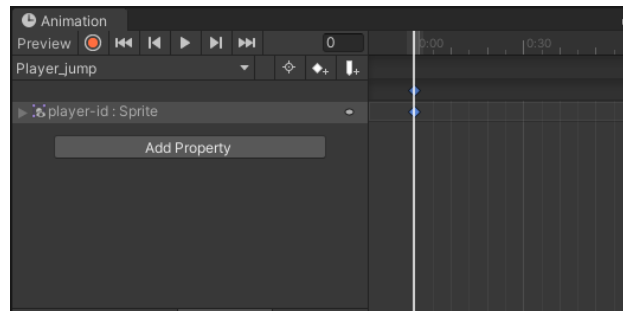


29. Buat clip baru dan beri nama Player\_jump, kemudian pilih karakter jump1.



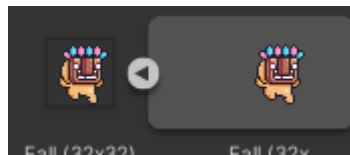
Gambar 9.26 Memilih Karakter Jump1

30. Drag and drop ke tab Animation.



Gambar 9.27 Drag and Drop ke Tab Animation

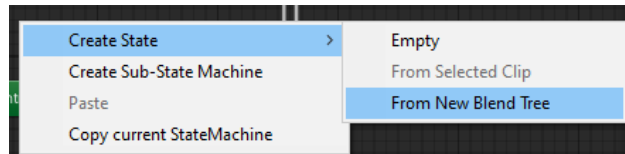
31. Buat juga clip untuk Player\_fall, kemudian pilih karakter Fall dan drag and drop ke Tab Animation.



Gambar 9.28 Membuat Clip Player\_fall

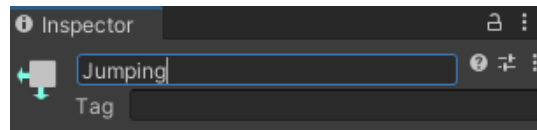


32. Untuk menambah animasi ketika melompat, kembali ke menu panel Animator, klik kanan pada bagian kosong manapun dan Create State lalu pilih From New Blend Tree.



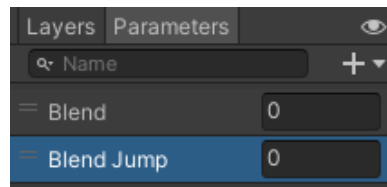
Gambar 9.29 Create State From New Blend Tree

33. Setelah terbuat, pergi ke Inspector dan ubah namanya menjadi Jumping.



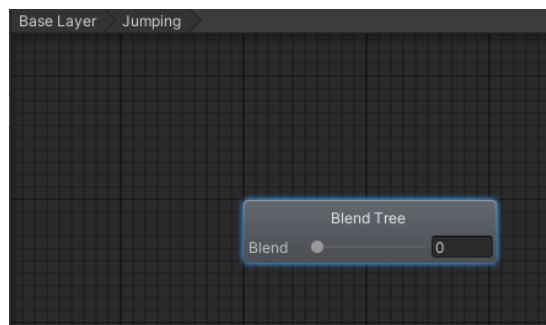
Gambar 9.30 Merubah Nama Menjadi Jumping

34. Buat tipe data baru Float dan beri nama Blend Jump.



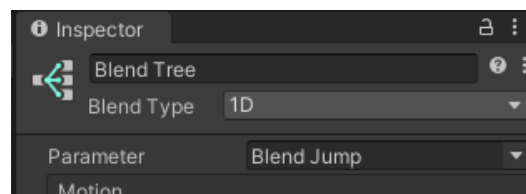
Gambar 9.31 Membuat Tipe data Blend Jump

35. Klik 2 kali pada Blend Tree “Jumping”, tekan pada Blend Tree.



Gambar 9.32 Double Click pada Blend Tree Jumping

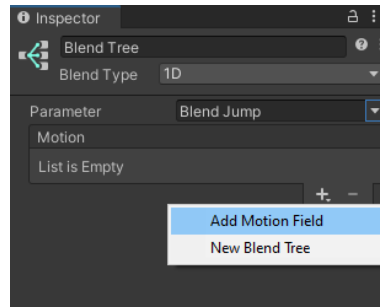
36. Ubah parameternya menjadi “Blend Jump”.



Gambar 9.33 Mengubah Parameter Blend Jump

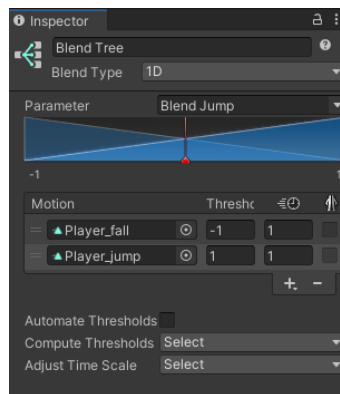


37. Tekan ikon + dan pilih Add Motion Field. Tambahkan dua Motion Field.



Gambar 9.34 Add Motion Field

38. Pada bagian icon None (Motion), akan muncul Windows Motion. Tambahkan sesuai dengan urutan. Hilangkan centang “Automate Thresholds” dan ubah nilainya seperti pada gambar di bawah.



Gambar 9.35 Setting Motion Field

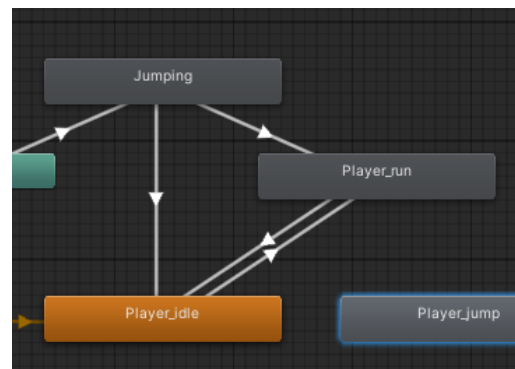
39. Kembali ke Base Layer, klik kanan pada Any State, pilih Make Transition dan arahkan panahnya ke Jumping.



Gambar 9.36 Membuat Transisi dari Any State ke Jumping

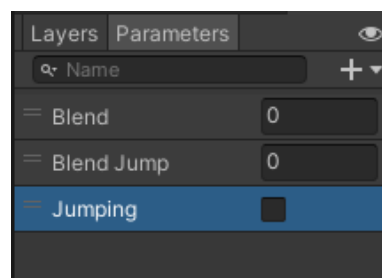


40. Klik kanan pada Jumping, Make Transition dan arahkan panahnya ke Player\_idle dan Player\_run.



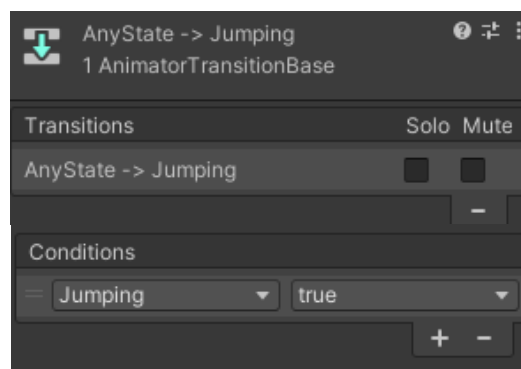
Gambar 9.37 Membuat Transisi dari Jumping

41. Buat tipe data baru Boolean dan beri nama Jumping.



Gambar 9.38 Membuat Tipe Data Jumping

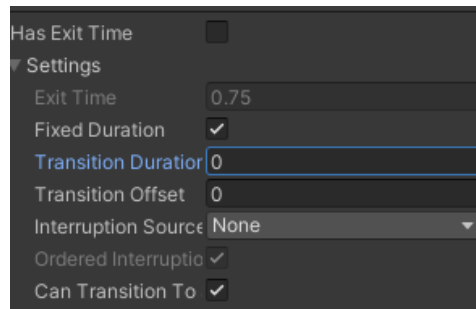
42. Klik panah yang mengarah ke Jumping dan buat Conditions dan ubah ke Jumping dengan nilai true.



Gambar 9.39 Menambahkan Condition Jumping

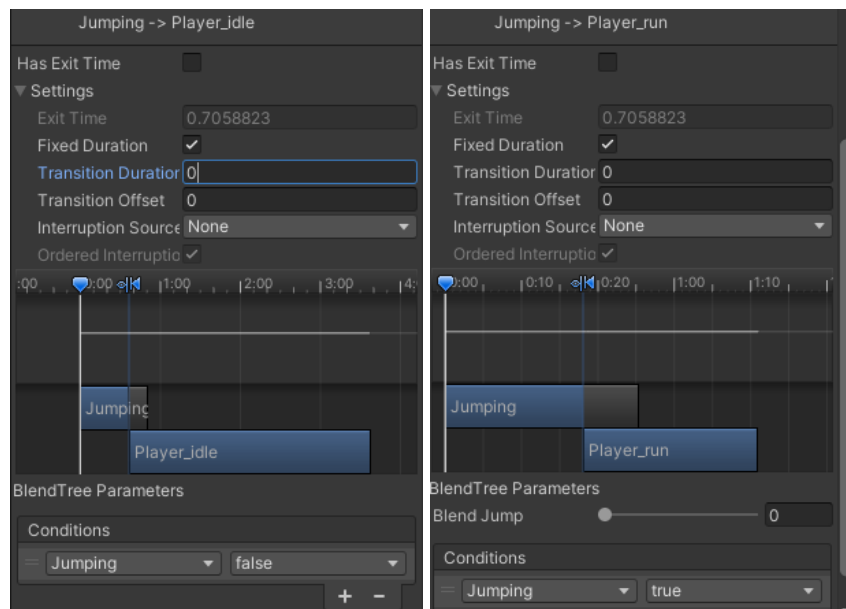


43. Atur settingnya dengan mengubah nilai Transition Duration menjadi 0 dan hilangkan centang pada Has Exit Time.



Gambar 9.40 Mengubah Setting

44. Klik panah yang mengarah dari Jumping ke Player\_idle. Tambahkan Conditions Jumping dengan nilai false. Sementara yang mengarah ke Player\_run juga ditambahkan Conditions Jumping tetapi nilainya diubah menjadi true.



Gambar 9.41 Setting Inspector

45. Tambahkan source code yang berlabel kuning pada fungsi update.

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



46. Pada fungsi FixedUpdate tambahkan source code yang dilabel kuning.

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

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

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

47. Tambahkan baris kode yang berlabel kuning 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);
}
```

48. Jika di-play maka karakter akan bisa bergerak dengan animasi.



Gambar 9.42 Hasil Play

Kuis :

```
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", 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 :

Pada source code diatas isJumping, harus ada tambahan 1 paramater Boolean, dan Ketika if dia bernilai true Ketika else dia bernilai false. Karena Ketika tombol space di tekan player akan lompat, jika tidak player akan diam atau tidak lompat. Kemudian move nya bernilai 0 berarti idle akan true, walking akan false, begitu juga sebaliknya. Menambahkan kondisi pada kedua pergerakan untuk menghindari bug animasi. Kemudian Ketika move lebih dari 0 berarti dia berjalan ke kanan Ketika move lebih kecil dari 0 maka dia akan berjalan kekiri, maka dari itu nilai vector positive Ketika lebih dari 0, dan negative Ketika kurang dari 0.