



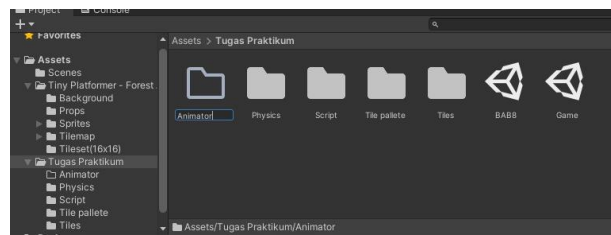
TUGAS PERTEMUAN: 9

GAME ANIMATON

NIM	:	2118059
Nama	:	Arika Risma Nabella
Kelas	:	B
Asisten Lab	:	Bagas Anardi Surya W (2118004)

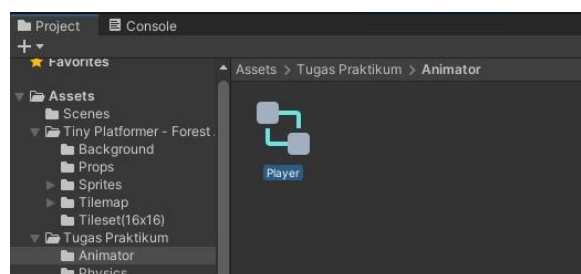
9.1 Tugas 1 : Penerapan Game Animation

1. Pada karakter, klik inspector kemudian pilih Add Component Animator. Di dalam folder Praktikum, buat folder baru bernama "Animator".



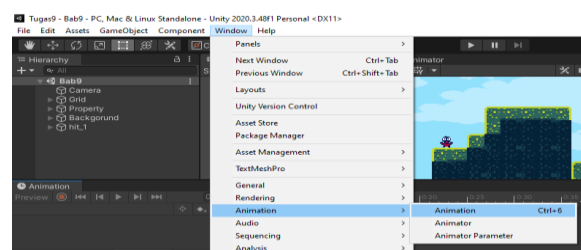
Gambar 9.1 Buat folder baru

2. Buat file Animator Controller di folder Animator, dan ubah namanya menjadi Player.



Gambar 9.2 Animator Controller

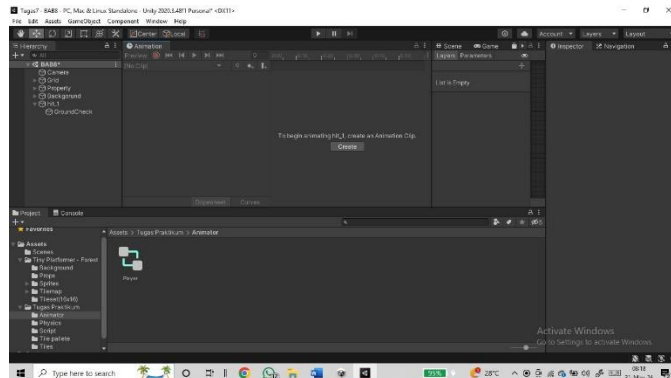
3. Tambahkan panel menu Animation di menu Window, pilih Animation > Animation atau tekan shortcut CTRL + 6.



Gambar 9.3 Panel menu Animation

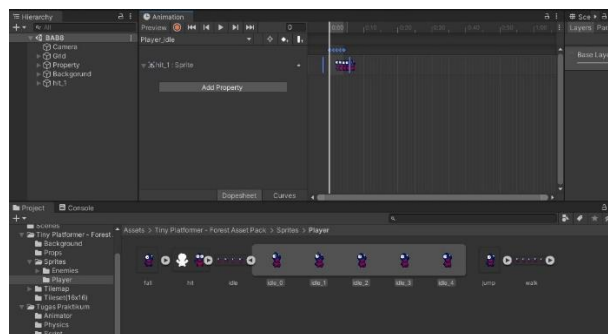


4. Akan muncul panel menu baru, geser panel tersebut ke bagian bawah seperti gambar di bawah ini, bagi panel Project dan Animation. Tambahkan panel menu Animator. Geser panel tersebut sesuai dengan gambar.



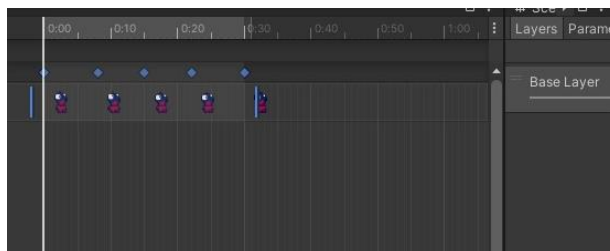
Gambar 9.4 Panel menu baru

5. Untuk membuat animasi, klik Player-idle1 pada Hierarchy, kemudian ke panel menu Animation, pilih Create Simpan di folder Animator dan beri nama "Player_idle". Pada menu Project, buka folder Player, lalu pilih Idle dan pilih gambar Player-idle-0, Player-idle-1, Player-idle-2, Player-idle-3 dan Player-idle-4, kemudian drag ke tab Animation.



Gambar 9.5 Drag Player-idle

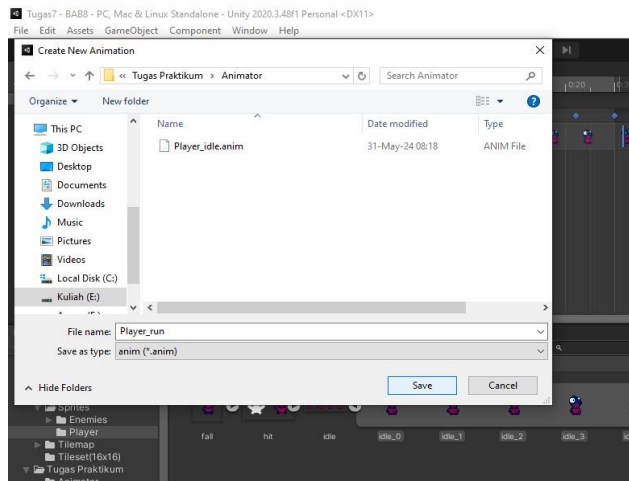
6. Tekan CTRL + A pada panel menu Animation, geser kotak kecil pada timeline sampai frame 0:30 agar animasinya tidak terlalu cepat.



Gambar 9.6 Timeline

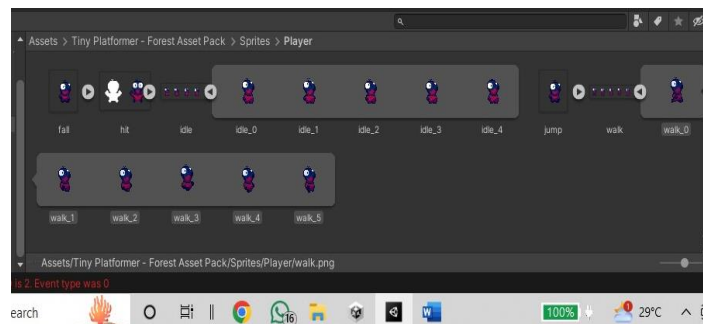


7. Buat animasi baru, klik pada "Player_idle" kemudian pilih Create New Clip, dan beri nama "Player_run". Simpan di folder Animator.



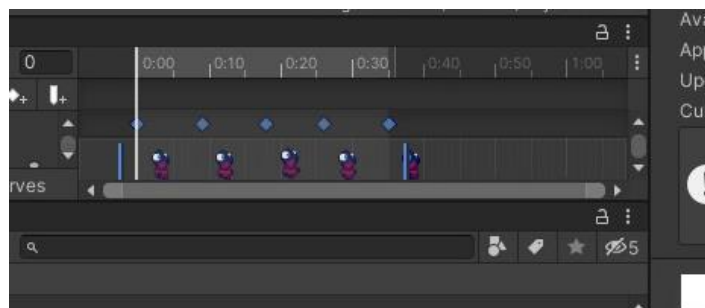
Gambar 9.7 Player_run

8. Buka menu Project, kemudian cari folder Player, pilih Walk-0 sampai Walk-5, drag and drop ke menu Animation.



Gambar 9.8 Drag Player Walk

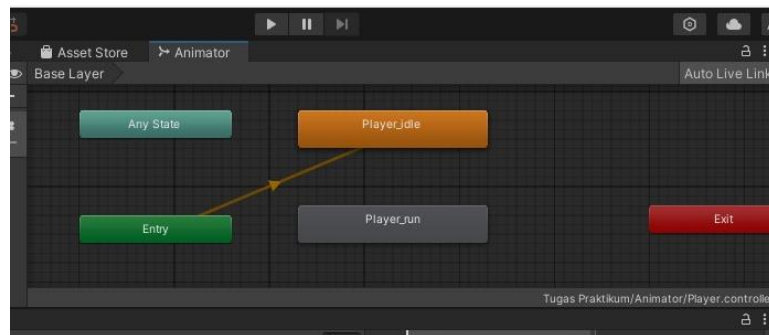
9. Pada panel timeline, tekan Ctrl + A di keyboard, klik kotak kecil di samping keyframe terakhir dan geser sampai waktu 0:35.



Gambar 9.9 TimeLine

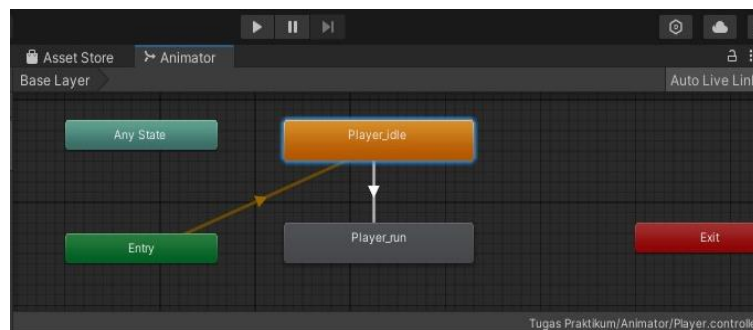


10. Pilih menu Animator yang telah dibuka sebelumnya dan akan tampil seperti berikut.



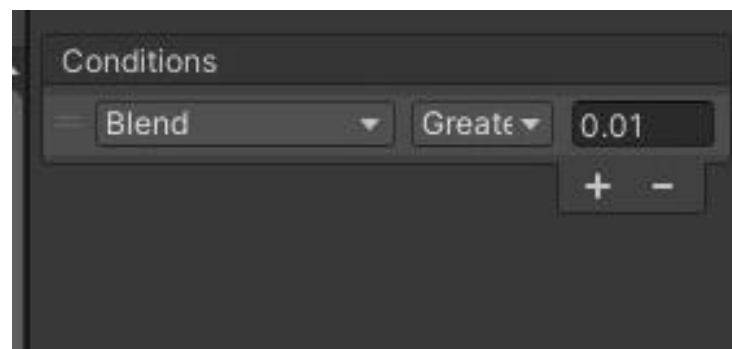
Gambar 9.10 Base Layer

11. Kemudian buat transisi antara Player_idle dan Player_run dengan cara klik kanan pada Player_idle dan pilih Make Transition, lalu tarik ke Player_run.



Gambar 9.11 Transisi antara Player_idle dan Player_run

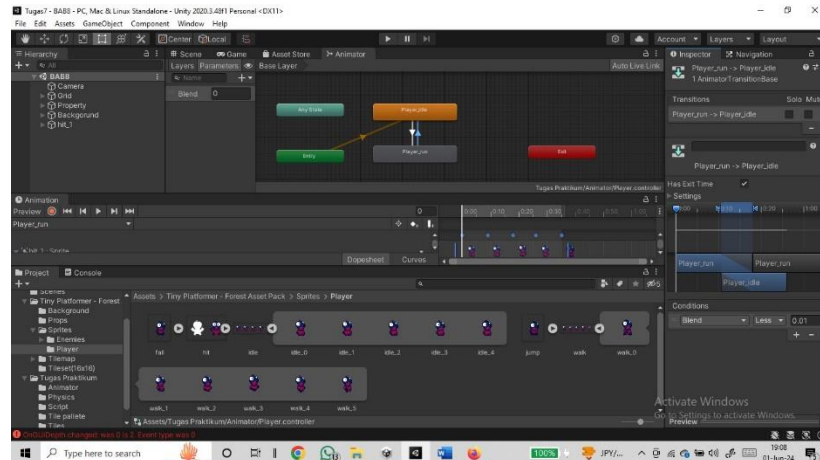
12. Masuk ke tab Parameter, tambahkan tipe data dengan cara tekan ikon tambah dan ubah namanya menjadi "Blend". Klik panah putih tersebut, pada bagian Conditions klik ikon tambah kemudian atur menjadi "Blend". Atur nilai Conditions Blend tersebut menjadi 0.01.



Gambar 9.12 Conditions Blend

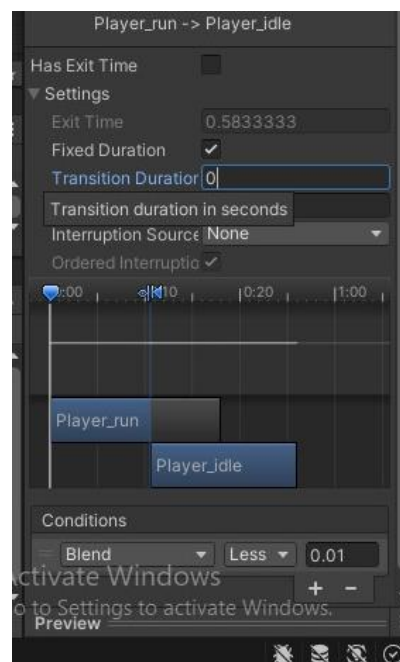


13. Pada bagian Settings, hilangkan centang pada Has Exit Time dan atur nilai Transition Duration menjadi 0. Buat transisi juga dari Player_run ke Player_idle dengan cara klik kanan pada Player_run dan pilih Make Transition.



Gambar 9.13 Make Transition.

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



Gambar 9.14 Transition Duration



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

```
public class Player : MonoBehaviour  
{  
    public Animator animator;  
}
```

Gambar 9.15 Script Player

16. Tambahkan Script Komponen Animator.

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

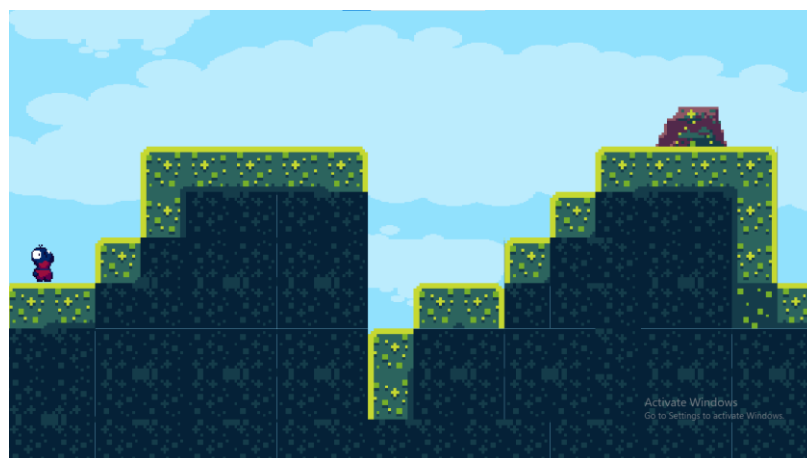
Gambar 9.16 Script Komponen Animator

17. Pada fungsi FixedUpdate tambahkan source code berikut.

```
void FixedUpdate()  
{  
    GroundCheck();  
    Move(horizontalValue, jump);  
    animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));  
}
```

Gambar 9.17 Script FixedUpdate

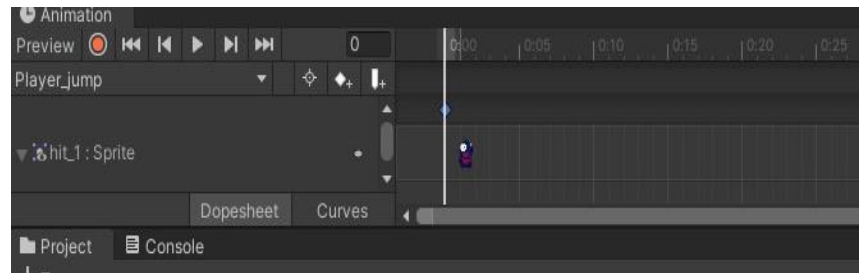
18. Jika dijalankan, maka Player dapat memiliki animasi ketika berhenti ataupun berjalan.



Gambar 9.18 Hasil Animasi Player

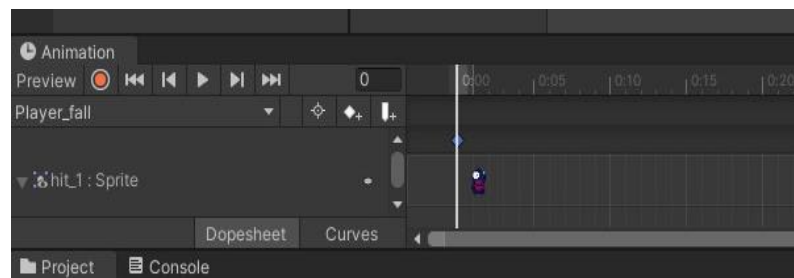


19. Buat animasi baru, tekan tulisan "Player_run" kemudian pilih Create New Clip, dan beri nama "Player_jump". Di folder Player, buka Jump lalu pilih gambar Player-jump-1, kemudian drag ke tab Animation.



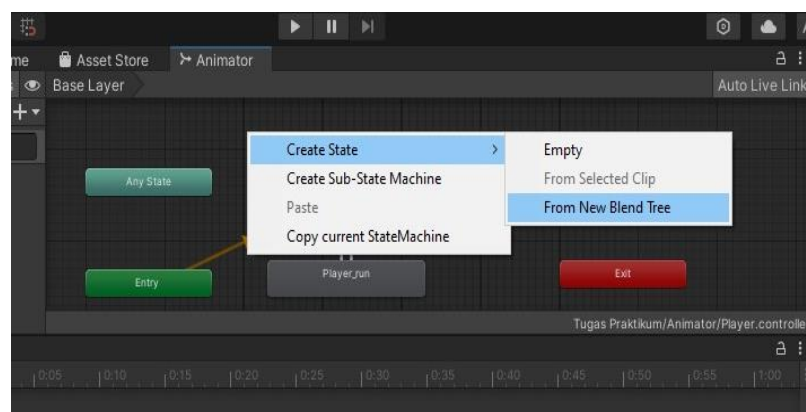
Gambar 9.19 Create Player_jump

20. Buat animasi baru dengan cara tekan tulisan "Player_jump" kemudian pilih Create New Clip, dan beri nama "Player_fall".m Pada tab Project, buka folder karakter lalu pilih Idle dan pilih gambar Player-fall, kemudian drag ke tab Animation.



Gambar 9.20 Create Player_fall

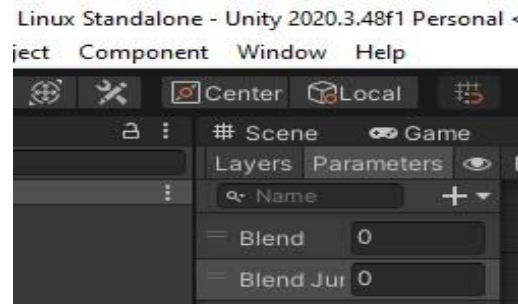
21. Untuk menambahkan animasi ketika melompat, klik kanan pada menu Animator, di area kosong, pilih Create State > From New Blend Tree.



Gambar 9.21 Create State

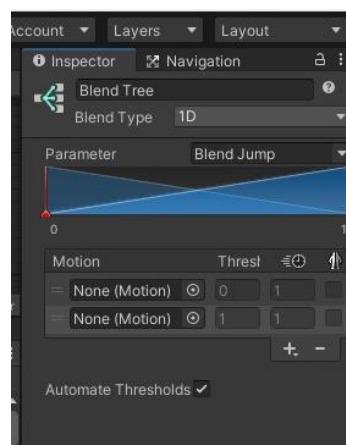


22. Pada Animator, klik Blend Tree, di menu Inspector, ubah namanya menjadi Jumping. Pada menu Parameters, tambahkan parameter tipe data Float, tekan ikon + dan ubah namanya menjadi "Blend Jump".



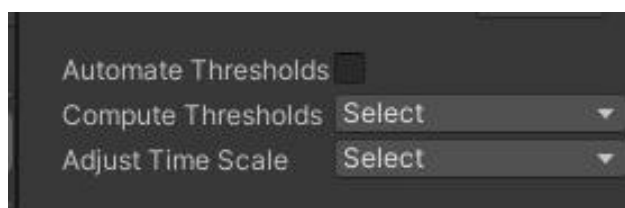
Gambar 9.22 Blend Jump

23. Pada menu Animator, klik dua kali pada Blend Tree "Jumping", tekan pada Blend Tree. Klik dua kali Blend Tree "Jumping", pada Inspector ubah parameter menjadi "Blend Jump". Buka menu Inspector, tekan ikon + dan pilih Add Motion Field. Tambahkan dua Motion Field.



Gambar 9.23 Motion Field

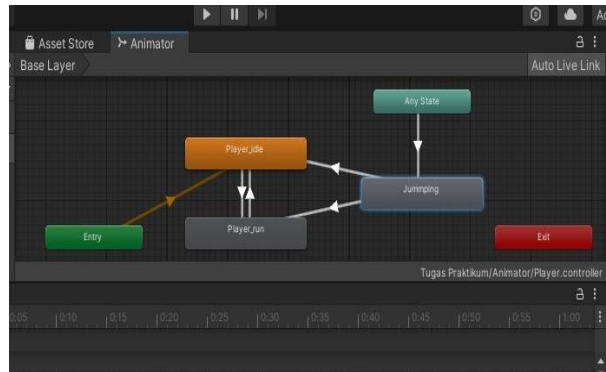
24. Klik bagian ikon None (Motion), maka akan muncul Windows Motion, tambahkan sesuai dengan urutan. Hilangkan centang "Automate Thresholds" dan atur nilai Threshold seperti berikut.



Gambar 9.24 Atur nilai Threshold

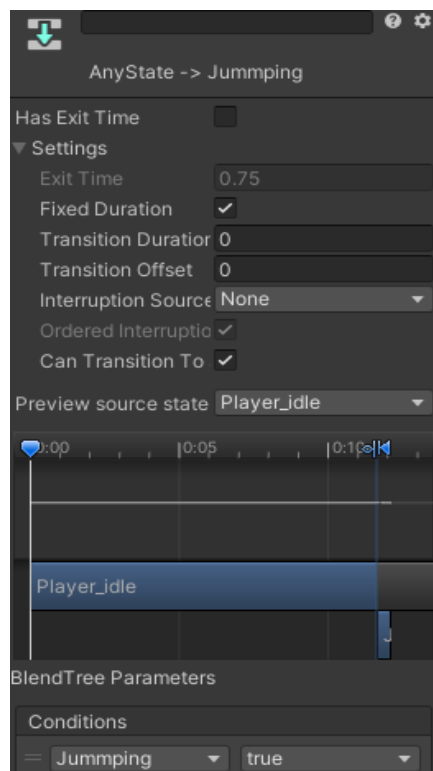


25. Kembali ke Base Layer, klik kanan Any State, pilih Make Transition dan arahkan panahnya ke Jumping. Klik kanan Jumping, pilih Make Transition dan arahkan panahnya ke Player_idle dan Player_run.



Gambar 9.25 Make Transition

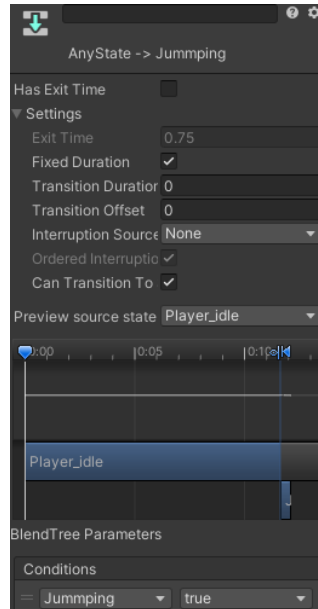
26. Tambahkan parameter transisi dengan tipe data Bool, tekan ikon + dan ubah namanya menjadi "Jumping". Klik panah yang mengarah ke Jumping, pada Inspector tambahkan condition, pilih condition Jumping dan ubah nilainya menjadi true. Klik Settings dan ubah nilai Transition Duration menjadi 0 dan hilangkan centang Has Exit Time.



Gambar 9.26 Transition Duration



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



Gambar 9.27 Has Exit Time

28. Buka script Player, dan tambahkan source code berikut pada fungsi Update.

```
public class Player : MonoBehaviour
{
    public Animator animator;
```

Gambar 9.28 Script fungsi Update

29. Pada fungsi FixedUpdate tambahkan seperti berikut.

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

Gambar 9.29 Script fungsi FixedUpdate

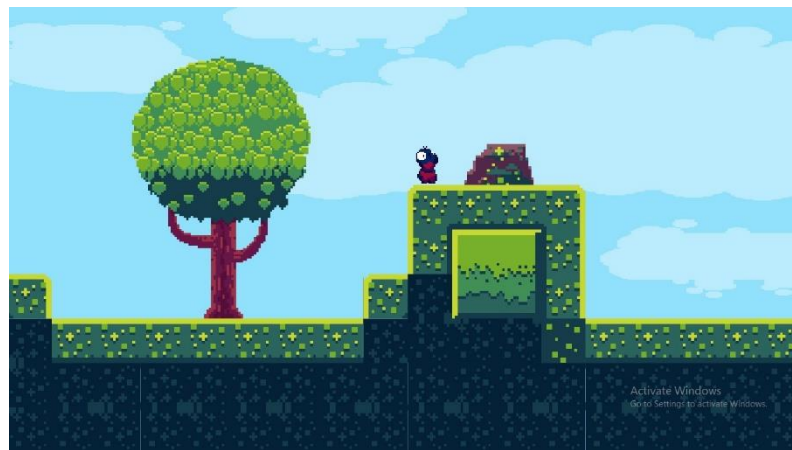


30. Tambahkan baris kode seperti di bawah ini dalam method GroundCheck.

```
void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump);
    animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
}
```

Gambar 9.30 Script GroundCheck

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



Gambar 9.31 Hasil Render

Link Github :

https://github.com/arikarisma15/2118059_PRAK_ANIMASIGAME.git



KUIS

Melengkapi Script

```
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:

1. HandleJumpInput:

- Menambahkan true pada animator.SetBool("isJumping", true) ketika tombol spasi ditekan.
- Menambahkan false pada animator.SetBool("isJumping", false) ketika tombol spasi tidak ditekan.

2. HandleMovementInput:

- Mengatur animasi idle dan walking berdasarkan nilai move.
- Menggerakkan karakter menggunakan transform.Translate.
- Menyesuaikan skala lokal karakter untuk menghadap ke arah yang benar berdasarkan nilai move.