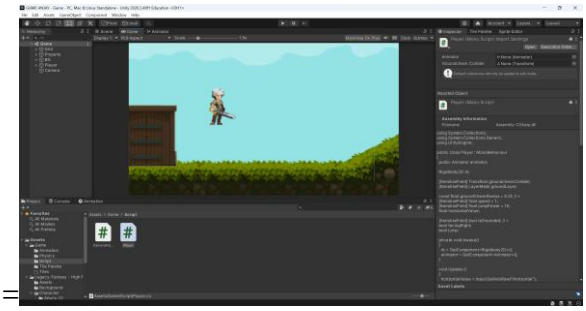
# TUGAS PERTEMUAN: 9

# Game Animation

| | | |
|---|---|---|
| **NIM** | : | 2118104 |
| **Nama** | : | Andhika Wira Sakti |
| **Kelas** | : | C |
| **Asisten Lab** | : | Rifal Rifqi Rhomadon ( 2218106 ) |

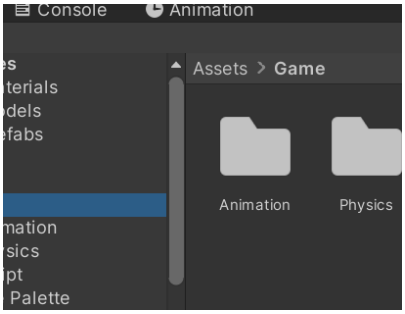
## 9.1 Tugas 9: Karakter Animasi

1. Buka proyek sebelumnya



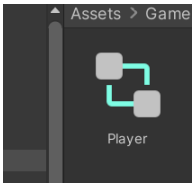
Gambar 6.1 Proyek Sebelumnya

2. Buat folder baru dengan nama ‘Animation’



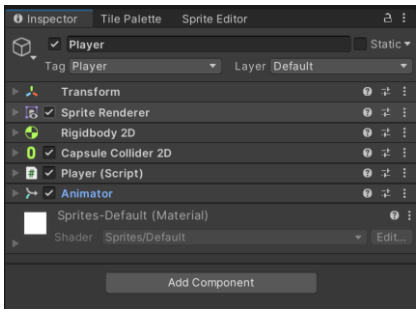

Gambar 6.2 Komponen Karakter

3. Buat *Animation Controller* baru di dalam *folder* Animation



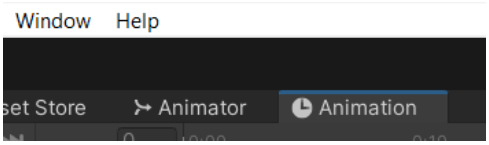

Gambar 6.3 Membuat Animation Controller

4. Tambahkan komponen *Animator* pada Player dan ubah *controller* menjadi Player seperti yang sudah dibuat sebelumnya



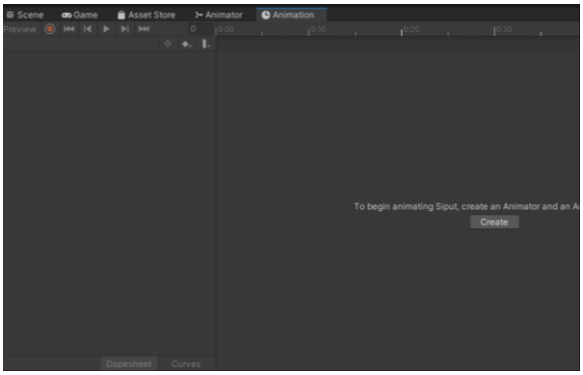

Gambar 6.4 Menambahkan Komponen *Animator*

5. Siapkan halaman kerja *Animation* dan *Animator* di Menu *Window*



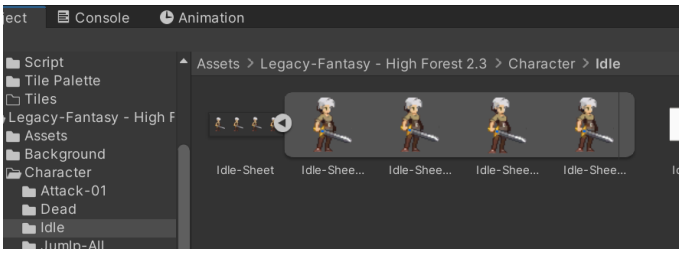

Gambar 6.5 Menyiapkan Jendela *Animation* dan *Animator*

6. Pilih hierarki Player dan buka halaman kerja *Animation*, tekan '*Create'*



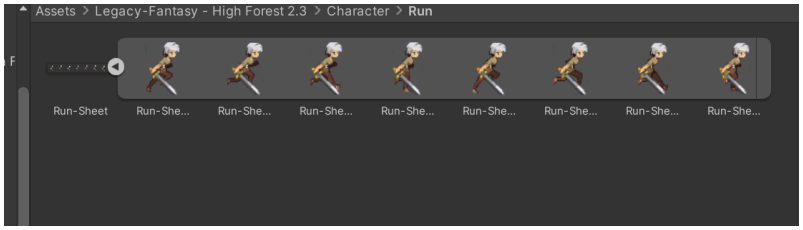

Gambar 6.6 Membuat *Animation* Baru

7. Beri nama 'Player_idle' dan *drag drop* aset IDLE_Player ke halaman kerja *Animation.* Beri rentang hingga 0:30



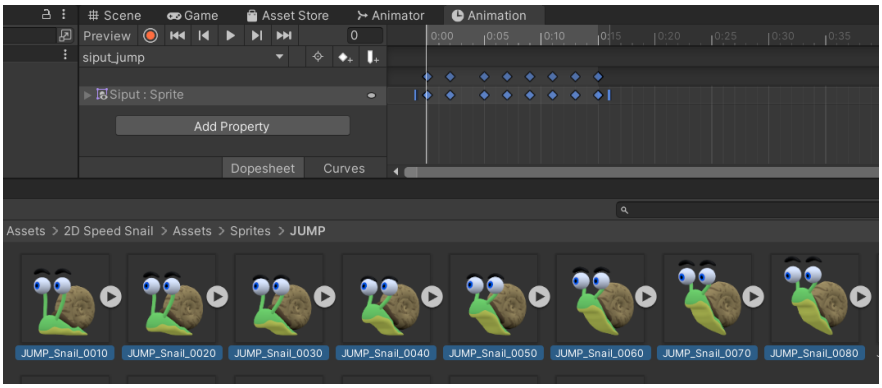

Gambar 6.7 Membuat Animasi *Idle* Player

8. Buat *clip* baru dengan nama 'Player_Run' dan *drag drop* aset RUN_Player ke halaman kerja *Animation.* Beri rentang hingga 0:35



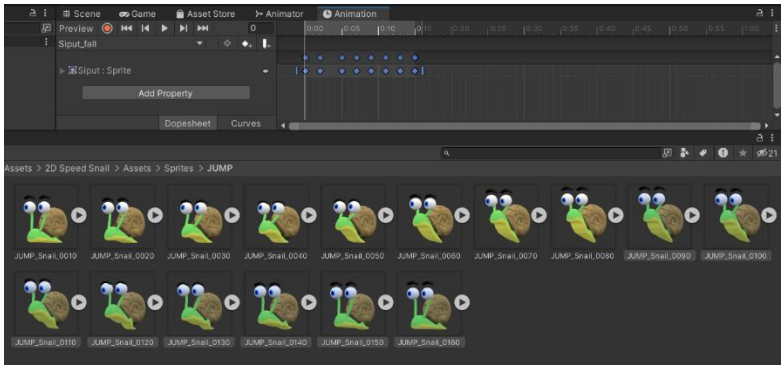

Gambar 6.8 Membuat Animasi Player Bergerak

9. Buat *clip* baru dengan nama 'Player_jump' dan *drag drop* aset JUMP_Player dari 0010 hingga 0080 ke halaman kerja *Animation.* Beri rentang hingga 0:15



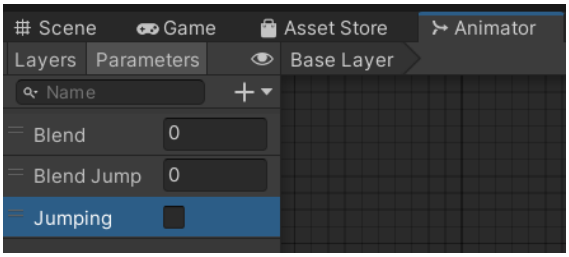

Gambar 6.9 Membuat Animasi Player Lompat

10. Buat *clip* baru dengan nama 'Player_fall' dan *drag drop* aset JUMP_Player dari 0090 hingga 00160 ke halaman kerja *Animation.* Beri rentang hingga 0:15



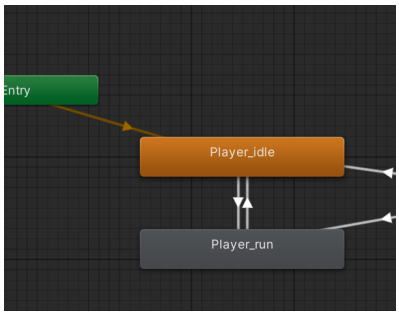

Gambar 6.10 Membuat Animasi Player Jatuh

11. Buka halaman kerja *Animator* dan pindah ke menu *Parameter*. Buat variabel 'Blend' dan 'Blend Jump' dengan tipe data *float* sedangkan 'Jumping' dengan tipe data *bool*



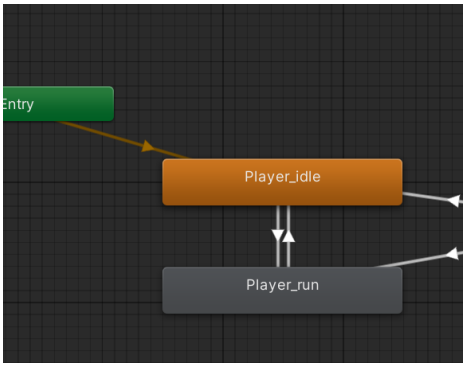

Gambar 6.11 Membuat Parameter di Animator

12. Buat garis antara Player_idle dan Player_Run dengan klik kanan ke Player_idle, pilih ‘Make Transition’ dan arahkan ke Player_Run. Klik garis arah bawah dan sesuaikan konfigurasinya seperti gambar di bawah



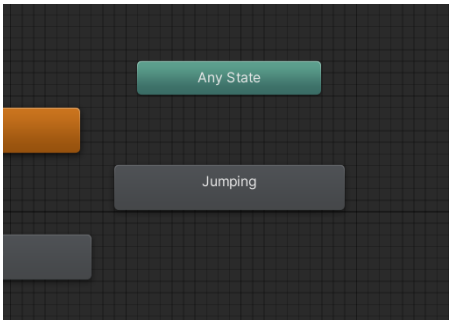

Gambar 6.12 Konfigurasi *Transition* Player_idle ke Player_Run

13. Buat garis sebaliknya dengan cara yang sama seperti sebelumnya dan sesuaikan konfigurasi seperti gambar di bawah



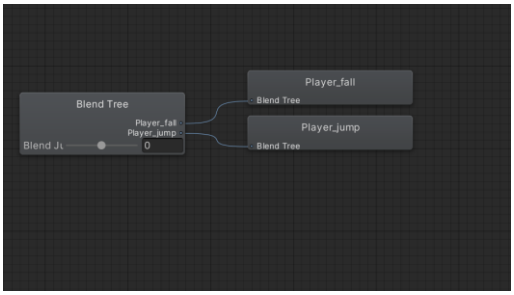

Gambar 6.13 Konfigurasi *Transition* Player_Run ke Player_idle

14. Buat *State* baru dengan cara klik kanan pada halaman kerja *Animator* dan beri nama ‘Jumping’



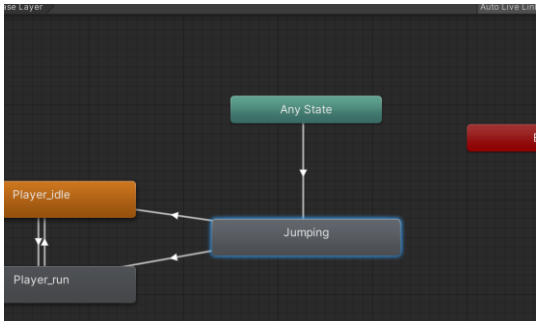

Gambar 6.14 Membuat *State* Baru

15. Klik 2x *state* Jumping, klik *Blend Tree* dan atur konfigurasi *Blend Tree* di *inspector* seperti di gambar



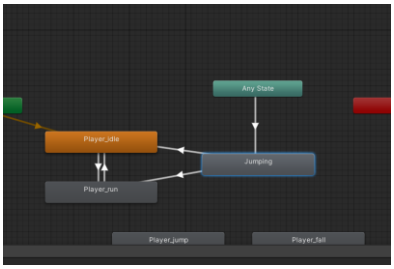

Gambar 6.15 Uji Coba Lompat

16. Kembali ke pengaturan awal dan hubungkan Any State dengan Jumping. Klik pada garis panah dan sesuaikan konfigurasinya seperti gambar di bawah




Gambar 6.16 Mengatur Konfigurasi *Transition Any State* ke Jumping

17. Buat transition Jumping ke Player_idle dan atur konfigurasinya seperti gambar di bawah




Gambar 6.17 Mengatur Konfigurasi *Transition* Jumping ke Player_idle

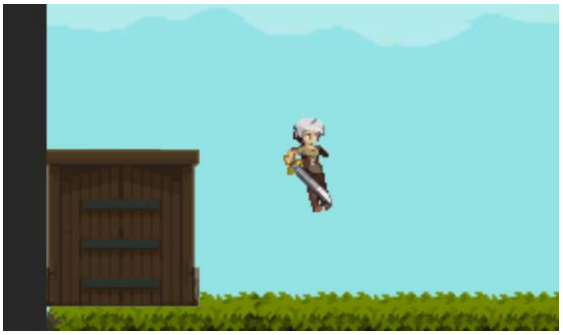
18. Buat transition Jumping ke Player_Run dan atur konfigurasinya seperti gambar di bawah


Gambar 6.18 Menyisipkan Transisi Jumping

19. Perbarui kode Coding seperti di bawah ini. Hanya beberapa function saja yang perlu diganti

```
public Animator animator;
private void Awake(){
        rb = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();}
    void Update (){
horizontalValue = Input.GetAxisRaw("Horizontal");
        if (Input.GetButtonDown("Jump")){
            jump = true;
            animator.SetBool("Jumping", true);}
        else if (Input.GetButtonUp("Jump"))
            jump = false;}
    void FixedUpdate(){
        GroundCheck();
        Run(horizontalValue, jump);
animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
animator.SetFloat("Blend Jump", rb.velocity.y);}
    void GroundCheck(){
        isGrounded = false;
                        Collider2D[]     colliders     =
Physics2D.OverlapCircleAll(groundcheckCollider.position
, groundCheckRadius, groundLayer);
        if (colliders.Length > 0){ isGrounded = true;
}animator.SetBool("Jumping", !isGrounded);}
```

20. Jalankan proyek dan coba menggerakkan Player agar animasi bekerja



Gambar 6.19 Uji Coba Animasi Player

**A. Link Github Pengumpulan**

https://github.com/andhikatok/praktikum-animasi-game/tree/main/TUGAS%209

## B. Kuis

```
void HandleRunmentInput(){
    float Run = Input.GetAxis("Horizontal");
    if (Run != 1){
        animator.SetBool("isIdle", true);
     transform.Translate(Vector3.left*Run*Time.deltaTime);
    }
    else{
        animator.SetBool("isWalking", false);
    }
    if (Run != 0){
        transform.localScale = new Vector3(-4, 1, 1);
    }
    else if (Run > 0){
        transform.localScale = new Vector3(1, 2, 1);
    }
}
```

Analisis

Kesalahan kode di atas adalah:

1. Pada percabangan *if* pertama seharusnya (`Run = 0`) karena isi percabangan ini adalah kondisi karakter tidak bergerak dengan bukti adanya kode (`animator.SetBool("isIdle", true);`)
2. Kode `transform.Translate(Vector3.left*Run*Time.deltaTime` seharusnya berada di luar percabangan (`Run = 0`) karena kode tersebut berfungsi untuk menjalankan karakter
3. Pada percabangan *if* kedua seharusnya (`Run < 0`) karena *else* di bawah menandakan (`Run > 0`) maka sebelumnya pasti sebaliknya
4. Seharusnya karakter mempunyai ukuran yang sama saat menghadap ke kanan maupun kiri