



TUGAS PERTEMUAN: 9

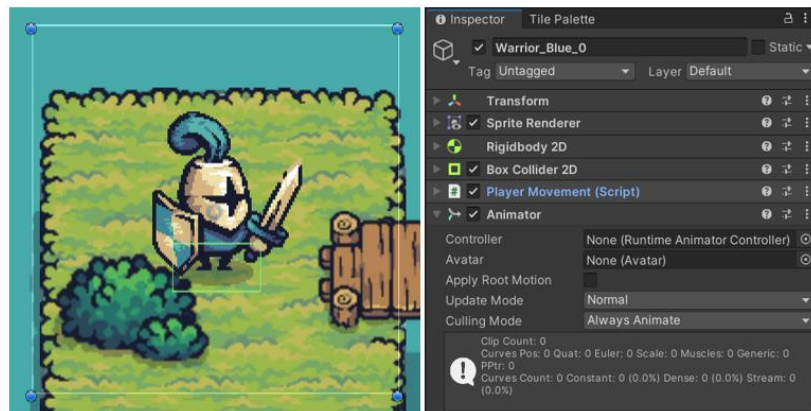
GAME ANIMATION

NIM	:	2118064
Nama	:	Ilham Maulana Prasetyo
Kelas	:	B
Asisten Lab	:	Bagas Anardi Surya W (2118004)

9.1 Tugas 1 : Membuat Character Animation

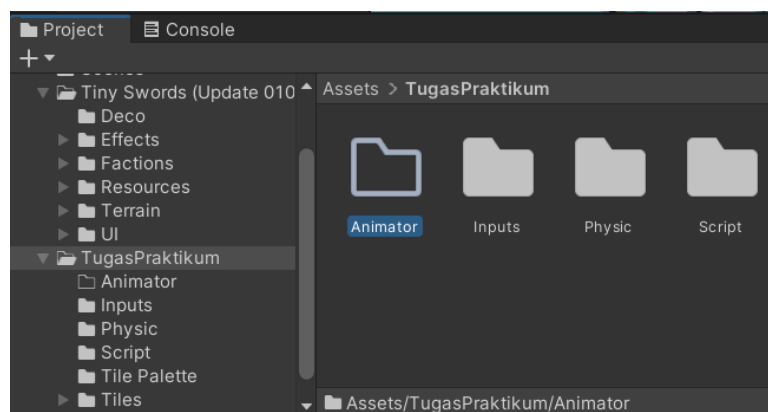
A. Character Animation

1. Pada karakter klik *Inspector* kemudian pilih *Add Component Animator*.



Gambar 9.1 Menambahkan Komponen Animator

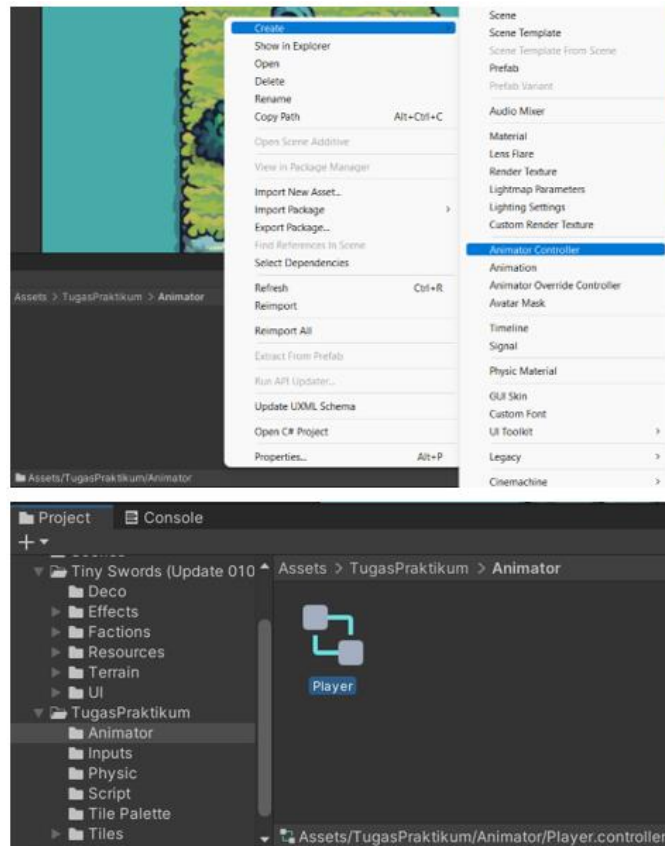
2. Pada folder TugasPraktikum buat folder baru bernama “Animator”.



Gambar 9.2 Membuat Folder Animator

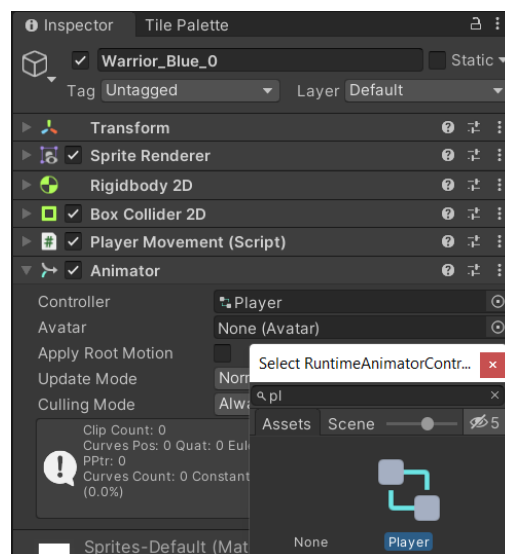


3. Buat file *Animator Controller* pada folder *Animator* , ubah namanya menjadi *Player*.



Gambar 9.3 Membuat *Animator Controller*

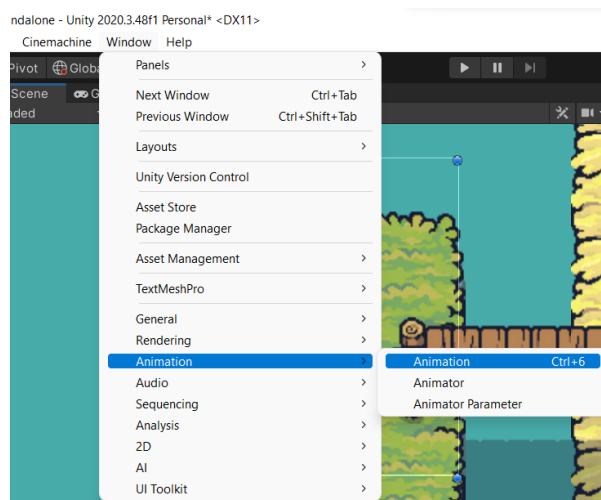
4. Pada *Hierarchy* klik *Player*, kemudian cari *Component Animator* di *Inspector*, pada *setting Controller* ubah menjadi *Player*.



Gambar 9.4 *Setting Controller Player*

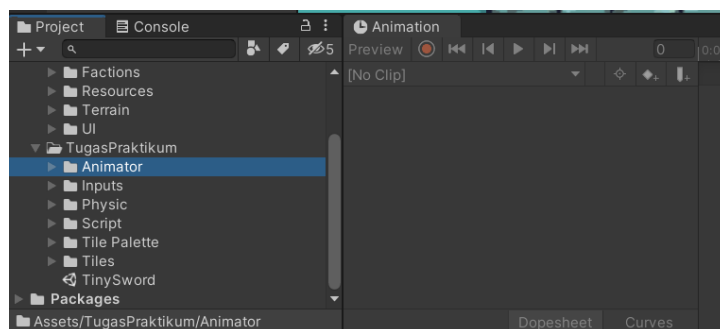


5. Tambahkan menu panel *Animation* di menu *Window*, pilih *Animation* > *Animation* atau tekan Shorcut CTRL + 6.



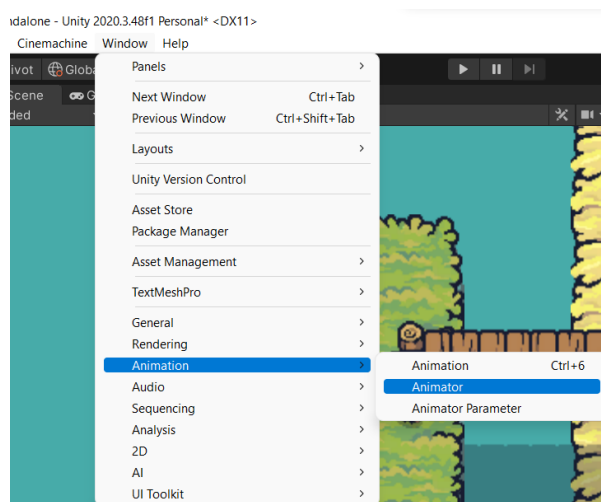
Gambar 9.5 Menambahkan Panel *Animation*

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



Gambar 9.6 *Split Panel Project* dan *Animation*

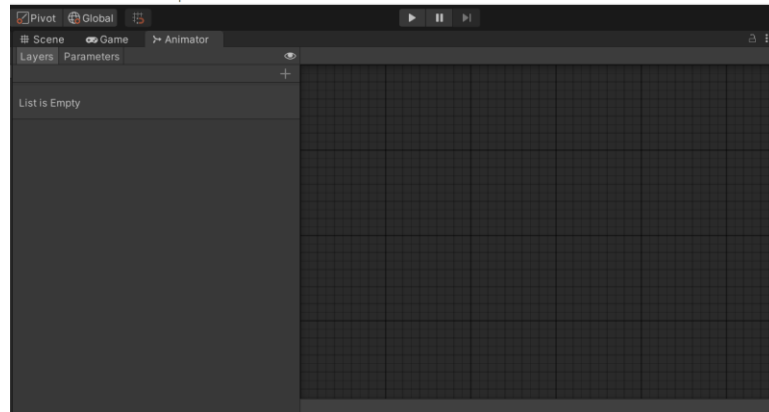
7. Tambahkan menu panel *Animator*.



Gambar 9.7 Menambahkan Panel *Animator*

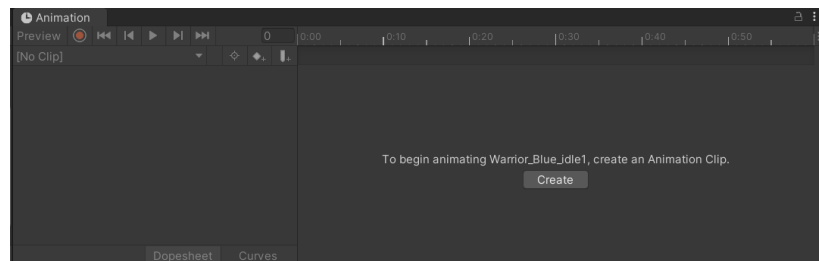


8. Geser panel sesuai gambar berikut.



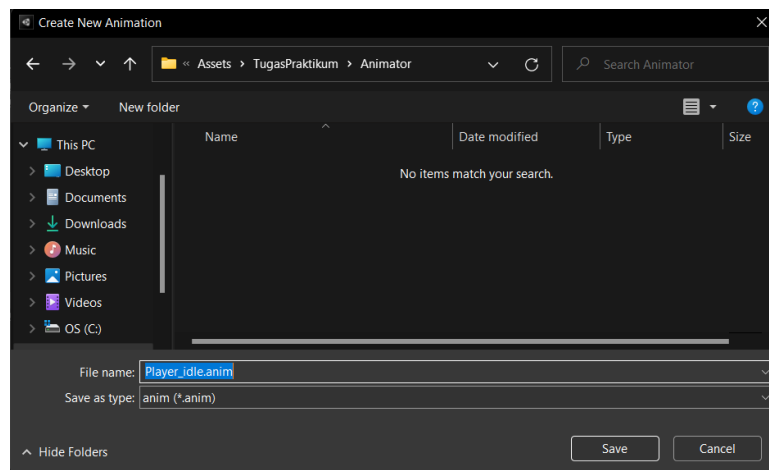
Gambar 9.8 Menyesuaikan Panel *Animator*

9. Untuk membuat animasi klik *Warrior_Blue_idle1* pada *Hierarchy*, kemudian ke menu panel *Animation*, pilih *Create*.



Gambar 9.9 Membuat Animasi

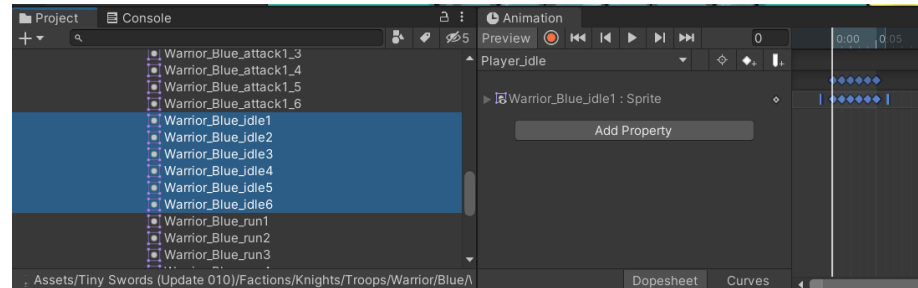
10. Simpan pada folder *Animator* dan beri nama “*Player_idle*”.



Gambar 9.10 Menyimpan File *Player_idle*

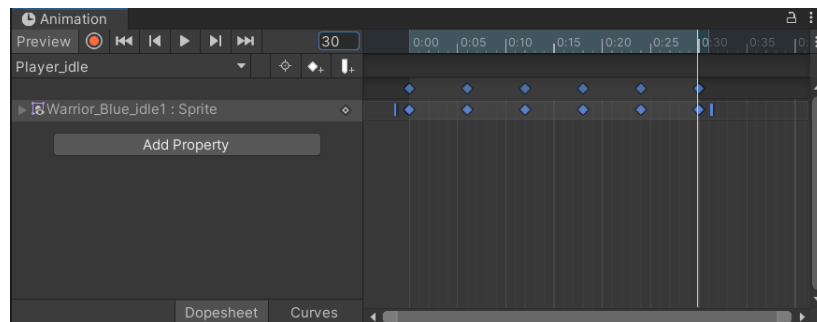


11. Pada menu Project buka folder Warrior lalu pilih Blue dan pilih gambar Warrior_Blue_idle1 sampai Warrior_Blue_idle6, kemudian *drag* ke tab *Animation*.



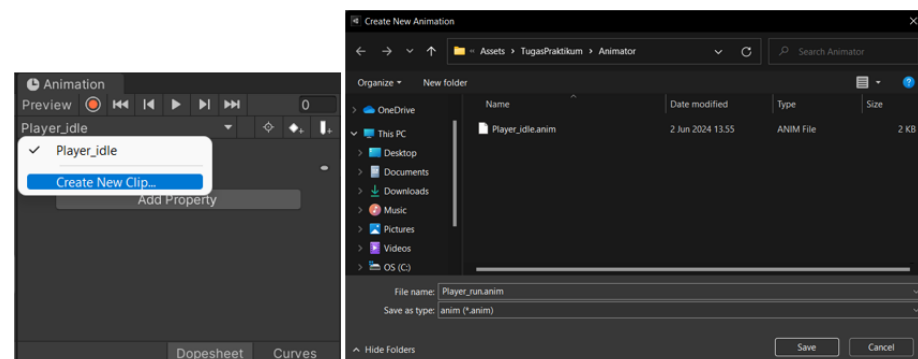
Gambar 9.11 *Drag* Gambar *Player Idle*

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



Gambar 9.12 Geser *Keyframe* *Player_idle*

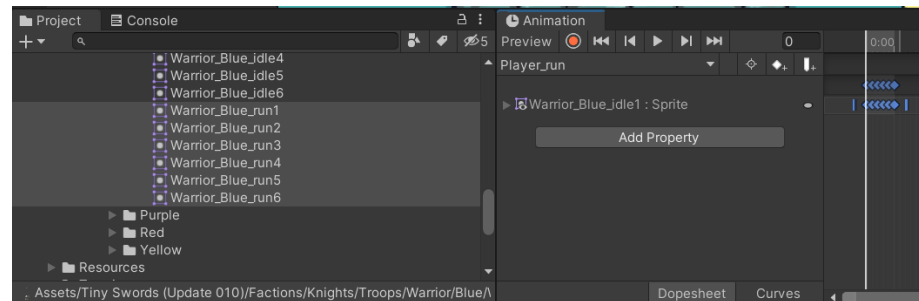
13. Buat animasi baru, klik pada “Player_idle” kemudian pilih *Create New Clip*, dan beri nama “Player_run”, Simpan pada folder *Animator*.



Gambar 9.13 Membuat Animasi *Player_run*

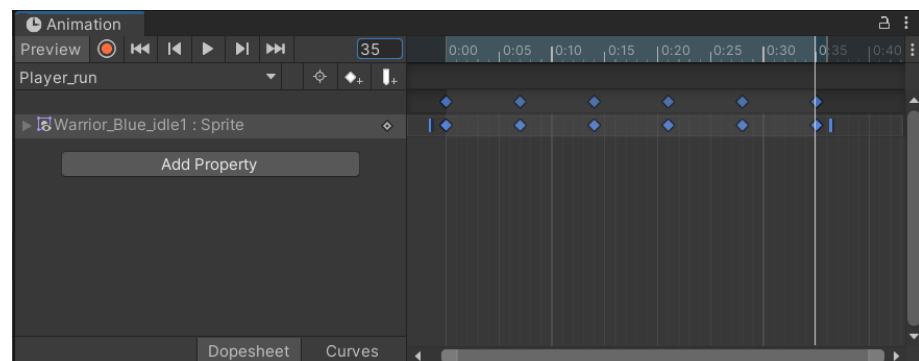


14. Buka menu Project kemudian cari folder Warrior > Blue, Pilih Warrior_Blue_run1 sampai Warrior_Blue_run6, *drag and drop* pada menu *Animation*.



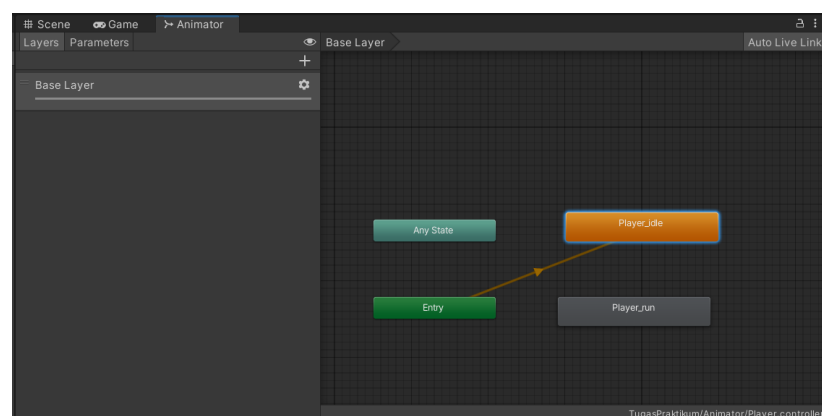
Gambar 9.14 *Drag Gambar Player Run*

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



Gambar 9.15 *Geser Keyframe Player_run*

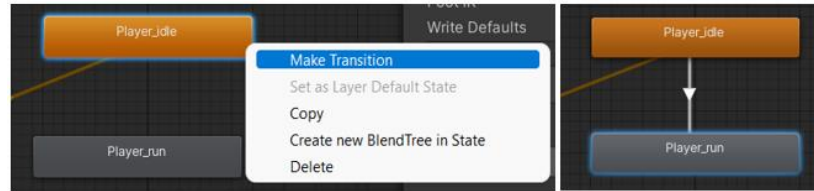
16. Pergi ke menu *Animator* yang telah dibuat sebelumnya dan akan tampil seperti berikut.



Gambar 9.16 *Tampilan Menu Animator*

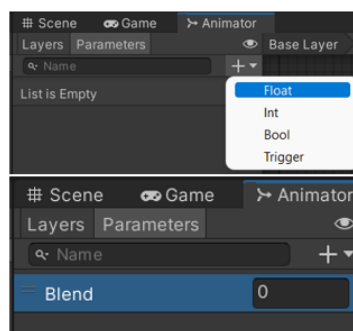


17. Kemudian buat transisi antara `Player_idle` dan `Player_run` dengan cara klik kanan pada `Player_idle` dan pilih *Make Transition* dan tarik ke `Player_run`.



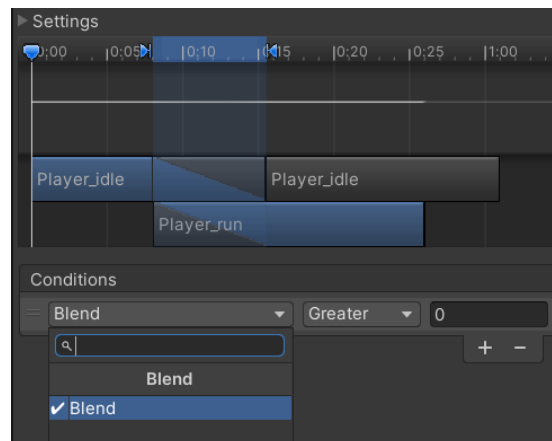
Gambar 9.17 Membuat Transisi

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



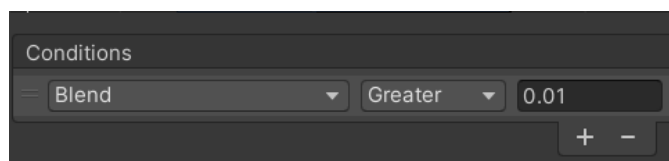
Gambar 9.18 Menambahkan Tipe Data Float

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



Gambar 9.19 Tambah Parameter Blend

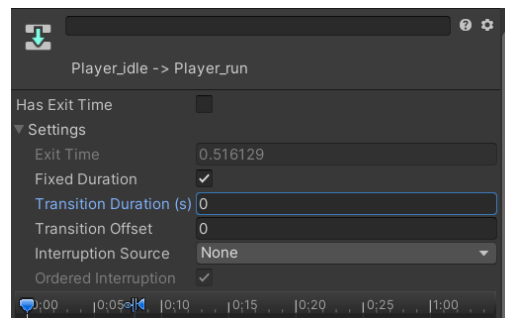
20. Atur nilai *Conditions* blend tersebut menjadi 0.01.



Gambar 9.20 Mengatur Nilai *Conditions* Blend

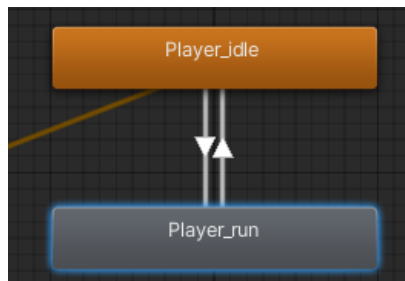


21. Pada bagian *Settings*, hilangkan centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



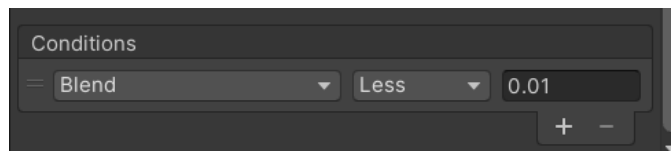
Gambar 9.21 Tampilan *Setting* Animasi

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



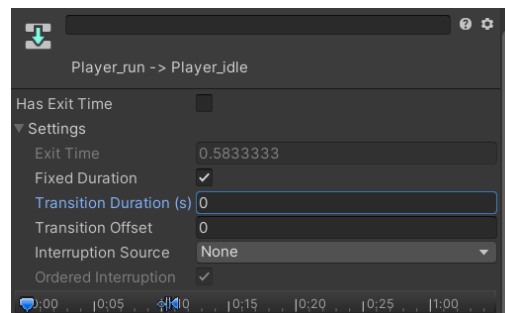
Gambar 9.22 Membuat Transisi

23. Klik panah yang mengarah dari *Player_run* ke *Player_idle*, lalu pada bagian *Conditions*, tambahkan parameter *Blend*, ubah operator dari *Great* menjadi *Less* dan ubah nilai menjadi 0.01



Gambar 9.23 Menambahkan Parameter Transisi

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



Gambar 9.24 Tampilan *Setting* Animasi



25. Agar animasi dapat sesuai ketika berjalan, buka *script* PlayerMovement dan tambahkan *source code* berikut pada *class* PlayerMovement.

```
public Animator animator;
```

```
5 public class PlayerMovement : MonoBehaviour
6 {
7     public Animator animator;
8
9     Rigidbody2D rb;
```

Gambar 9.25 Menambahkan Kode pada *Class* PlayerMovement

26. Tambahkan *script* komponen animator pada void Awake seperti dibawah.

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

```
24 private void Awake()
25 {
26     rb = GetComponent<Rigidbody2D>();
27     animator = GetComponent<Animator>();
28 }
```

Gambar 9.26 Menambahkan Kode pada Fungsi Awake

27. Dan pada fungsi FixedUpdate tambahkan *source code* berikut.

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

```
49 void FixedUpdate()
50 {
51     GroundCheck();
52     Move(horizontalValue, verticalValue, jump);
53
54     animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x) + Mathf.Abs(rb.velocity.y));
55 }
```

Gambar 9.27 Menambahkan Kode pada Fungsi FixedUpdate

28. Jika dijalankan maka player dapat memiliki animasi ketika berhenti ataupun ketika berjalan.



Gambar 9.28 Tampilan Hasil Animasi

B. Link Github

https://github.com/ilham-pras/2118064_PRAK_ANIGAME.git



9.2 Kuis

Menjelaskan source code dibawah ini:

```
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);
    }
}
```

Analisa:

Dalam fungsi HandleJumpInput, baris kode animator.SetBool("isJumping, "); tidak ada nilai yang ditetapkan untuk parameter isJumping yang akan menyebabkan error. Karena itu, diperlukan nilai boolean "*true*" atau "*false*" untuk mengatur nilai parameternya. Lalu pada kondisi percabangan else if() seharusnya diubah ke GetKeyUp untuk deteksi akhir lompat, perbaikannya seperti dibawah.

```
void HandleJumpInput()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        animator.SetBool("isJumping", true);
        rb.AddForce(Vector2.up * jumpForce, ForceMode2D.Impulse);
    }
}
```



```
else if (Input.GetKeyUp(KeyCode.Space))
{
    animator.SetBool("isJumping", false);
}
```

Lalu pada fungsi `HandleMovementInput`, kondisi percabangan pertama “`move != 1`” salah dan pernyataannya juga kurang. Kondisi ini seharusnya “`move == 0`” untuk mengecek apakah pemain sedang tidak bergerak, perbaikannya seperti dibawah.

```
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);
}
```

Untuk kondisi percabangan kedua “`move != 1`” juga salah, ini seharusnya “`move < 0`” untuk mencerminkan pemain menghadap kiri. Selain itu, nilai `Vector3` juga harus sama agar skala karakter tidak kacau, perbaikannya seperti dibawah.

```
if (move < 0)
{
    transform.localScale = new Vector3(-1, 1, 1);
}
else if (move > 0)
{
    transform.localScale = new Vector3(1, 1, 1);
}
```