



TUGAS PERTEMUAN: 9

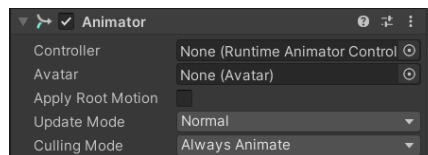
GAME ANIMATION

NIM	:	2118126
Nama	:	Prita Patricia Lakzmi
Kelas	:	D
Asisten Lab	:	Wisando Berlian P. (2218095)

9.1 Tugas 9 : Membuat Character Game Animation

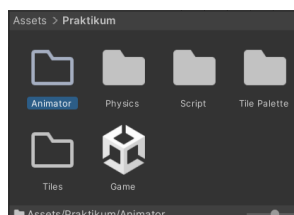
A. Pergerakan Player

1. Buka *Project Unity* pada 2118126_BAB8 sebelumnya yang telah dibuat sebelumnya. Lalu pada *Hierarchy player-id* Add Component dengan nama *Animator* pada *inspector*.



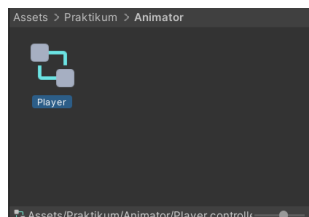
Gambar 9.1 Add Component Animator

2. Buat 1 buah *folder* baru dengan nama “Animator” di dalam *folder* Praktikum.



Gambar 9.2 Membuat Folder Animator

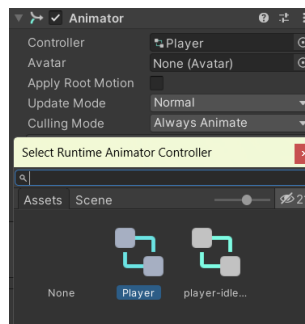
3. Buat 1 buah *file Animator Controller* dengan nama “Player” pada *folder* Animator.



Gambar 9.3 Membuat Animator Controller

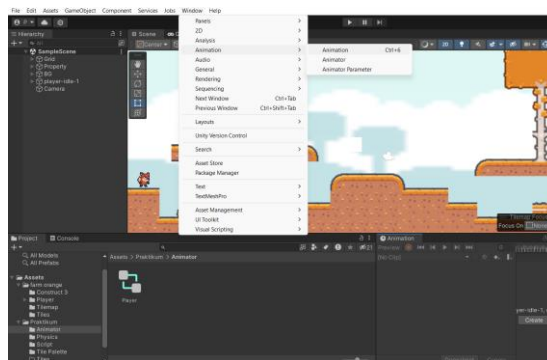


4. Pada *player-idle* di dalam *component Animator* ubah *Controller* menjadi *Player*.



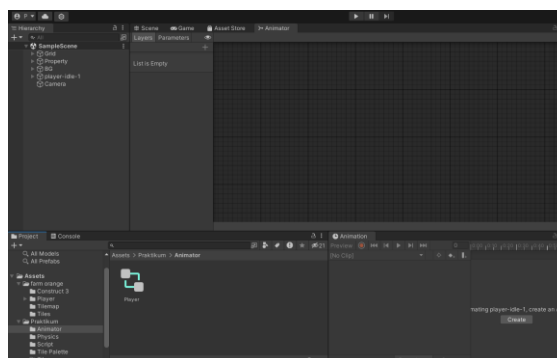
Gambar 9.4 Select Player Controller

5. Tambah menu panel *animator* dengan cara klik menu *Window* pilih *Animation > Animation*. Kemudian posisikan panel dengan posisi *split Panel Project* dan *Animation* bersampingan.



Gambar 9.5 Menambah Panel Animation

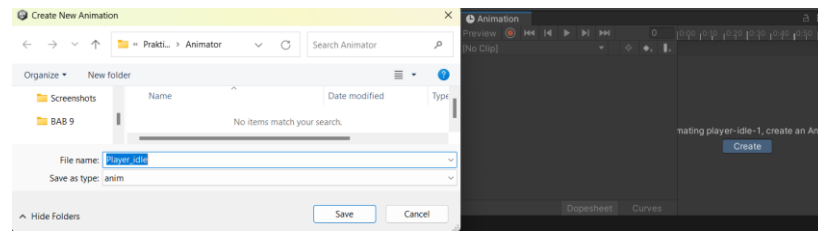
6. Tambahkan menu panel *Animator*, dengan cara menu *Window* pilih *Animation > Animator*.



Gambar 9.6 Menambah Panel Animator

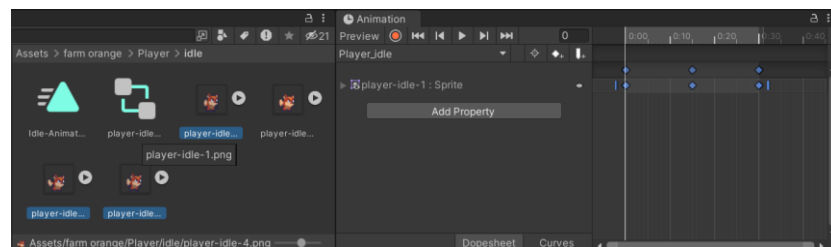


7. Buat animation pada *Hierarchy player-idle* dan pada panel *Animation* klik *button Create* dan beri nama “*Player_idle*”.



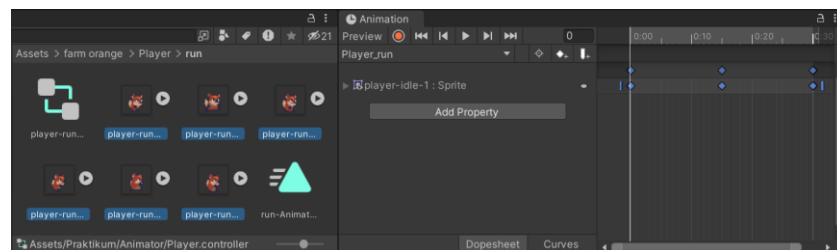
Gambar 9.7 Create New Clip Player_idle

8. Drag and drop player-idle 1, 3, dan 4 pada *timeline frame 0* hingga 30.



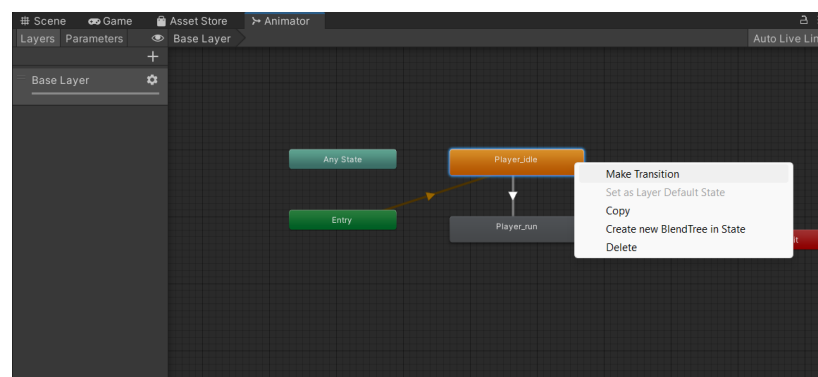
Gambar 9.8 Drag And Drop Player_idle ke dalam Frame

9. Selanjutnya drag and drop player-run pada *timeline frame 0* hingga 30.



Gambar 9.9 Drag And Drop Player_run ke Dalam Frame

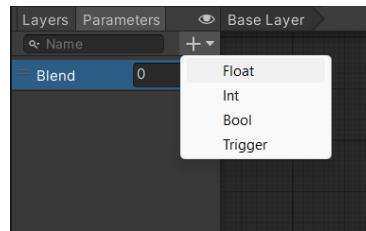
10. Pada menu *Animator Make Transition* pada *Player_idle* ke *Player_run*.



Gambar 9.10 Make Transition Player_idle ke Player_run

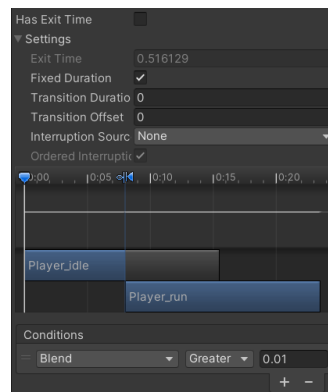


11. Masuk ke tab parameter, tambah 1 buah parameter dengan tipe data *Float* dan *rename* menjadi “*Blend*”.



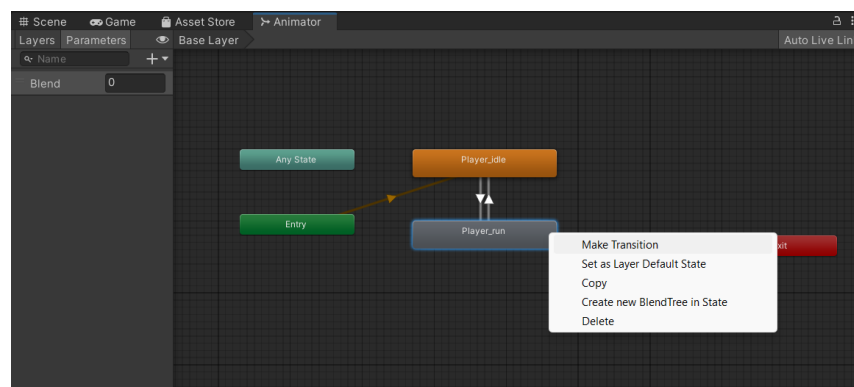
Gambar 9.11 *Parameter Blend*

12. Klik tanda panah *Make Transition* pada *Player_idle* ke *Player_run*. Kemudian pada bagian *inspector* tambah 1 *conditions* dan ubah *conditions* menjadi *Blend*, atur nilai menjadi 0.01 dan sesuaikan *Settings* seperti gambar dibawah ini.



Gambar 9.12 *Inspector Player_idle ke Player_run*

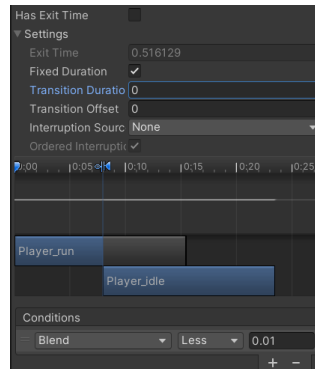
13. *Make Transition* dari *Player_run* ke *Player_idle*.



Gambar 9.13 *Make Transition Player_run ke Player_idle*



14. Klik tanda panah *Make Transition* pada *Player_run* ke *Player_idle*. Kemudian pada bagian *inspector* tambah 1 *conditions* dan ubah *conditions* menjadi *Blend*, atur nilai menjadi 0.01 dan sesuaikan *Settings* seperti gambar dibawah ini.



Gambar 9.14 *Inspector Player_run ke Player_idle*

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

```
public Animator animator;
```

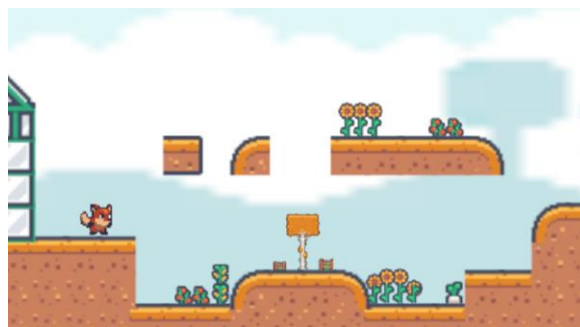
16. Tambahkan *script* komponen *Animator* pada *fungsi void Awake()*.

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

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

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

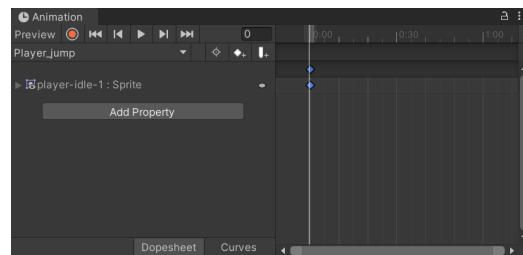
18. Tekan *play* untuk melihat apakah *player* telah memiliki animasi ketika *player* berhenti berjalan.



Gambar 9.15 Tampilan Saat Di *Play*

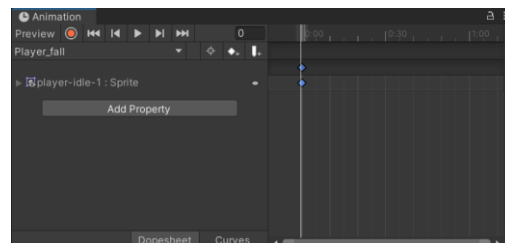


19. *Create New Clip* untuk membuat animasi *Player_jump*. Kemudian *drag and drop player-jump* kedalam *timeline frame*.



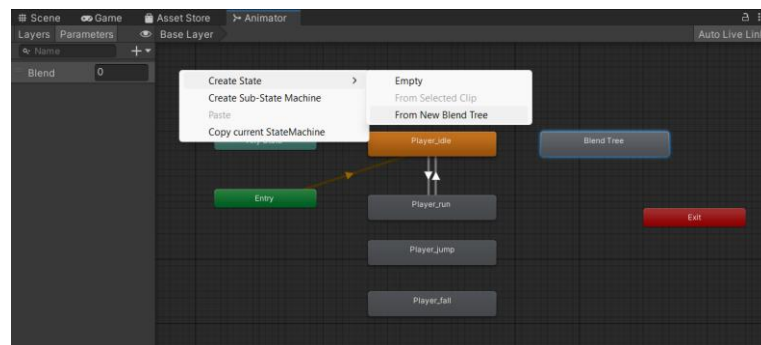
Gambar 9.16 *Create New Clip Player_jump*

20. *Create New Clip* untuk membuat animasi *Player_fall*. Kemudian *drag and drop player-fall* kedalam *timeline frame*.



Gambar 9.17 *Create New Clip Player_fall*

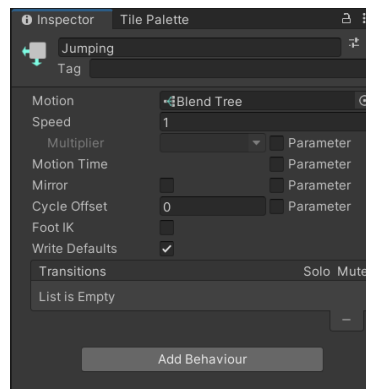
21. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu *animator* pada area kosong dan pilih *Create State > From New Blend Tree*.



Gambar 9.18 Membuat *Blend Tree*

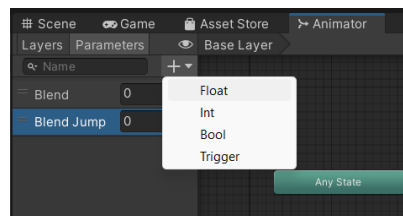


22. Pada panel *Animator* klik *Button Blend Tree*. Di dalam menu *inspector* ubah namanya menjadi *Jumping*.



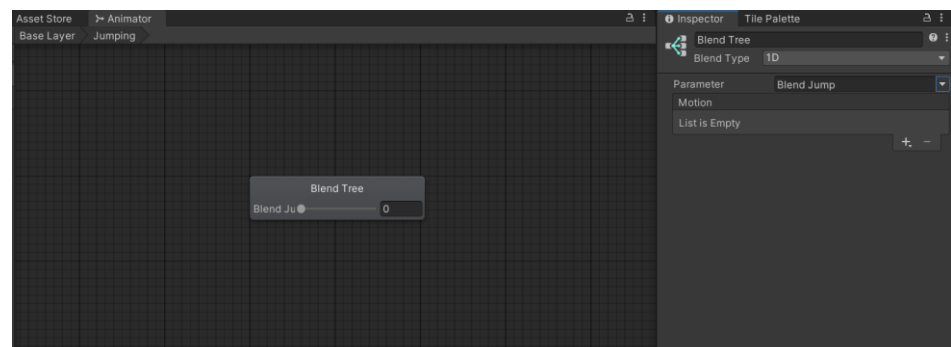
Gambar 9.19 *Inspector Blend Tree*

23. Buat 1 buah parameter baru dengan tipe data *Float* dan *rename* menjadi *Blend Jump*.



Gambar 9.20 Parameter *Blend Jump*

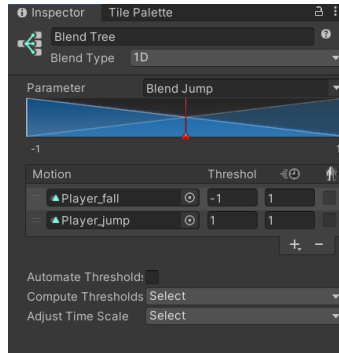
24. Klik 2x pada *Blend Tree* di dalam panel *Animator*. Jika sudah maka tampilannya akan seperti gambar dibawah ini.



Gambar 9.21 *Blend Tree*

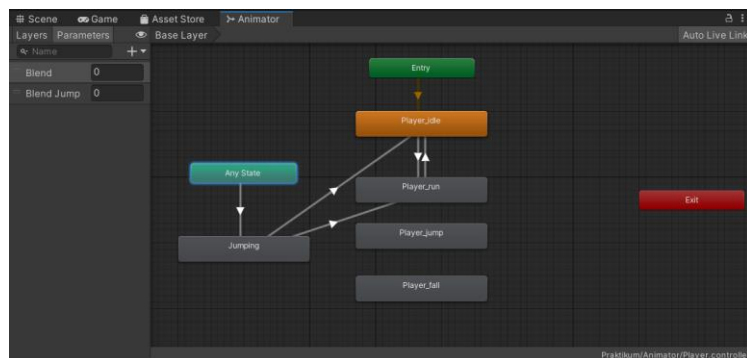


25. Pada inspector *Blend Jump* ubah parameter menjadi '*Blend Jump*'. Dan *Add Motion Field* sebanyak 2x lalu sesuaikan *motion* seperti gambar dibawah ini dan ubah motion menjadi *Player_fall* dan *Player_jump*.



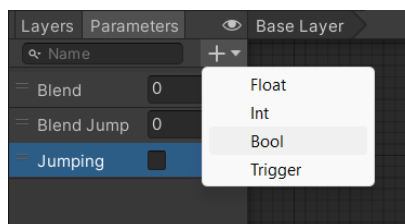
Gambar 9.22 *Inspector Blend Jump*

26. *Make Transition* pada *Any State* ke *Jumping*. Dan *Make Transition* dari *Jumping* ke *Player_idle* dan *Player_run*.



Gambar 9.23 *Inspector Any State ke Jumping*

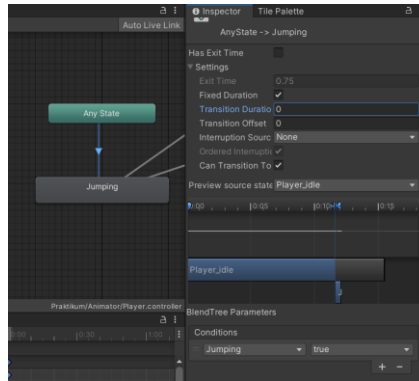
27. Buat 1 buah parameter dengan tipe data *Bool* dan ubah namanya menjadi "*Jumping*".



Gambar 9.24 *Parameter Jumping*

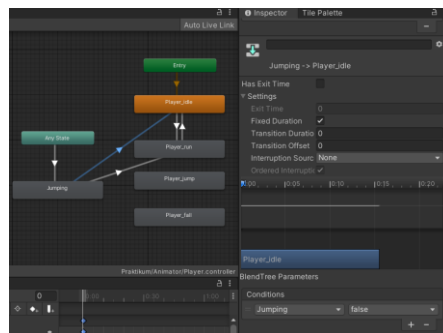


28. Klik tanda panah *Make Transition* dari *Any State* ke *Jumping*. Dan tambahkan *Condition Jumping* pada *inspector* dan ubah nilai nya menjadi *true*.



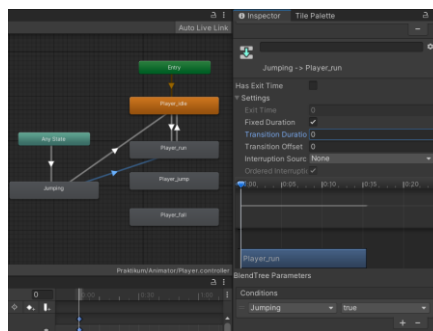
Gambar 9.25 *Inspector Any State ke Jumping*

29. Klik tanda panah *Make Transition* dari *Jumping* ke *Player_idle*. Dan tambahkan *Condition Jumping* pada *inspector* dan ubah nilai nya menjadi *false*.



Gambar 9.26 *Inspector Jumping ke Player_idle*

30. Klik tanda panah *Make Transition* dari *Jumping* ke *Player_run*. Dan tambahkan *Condition Jumping* pada *inspector* dan ubah nilai nya menjadi *true*.



Gambar 9.27 *Inspector Jumping ke Player_run*



31. 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;
    }
}
```

32. Pada fungsi *FixedUpdate* tambahkan *source code* yang dilabel kuning.

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

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

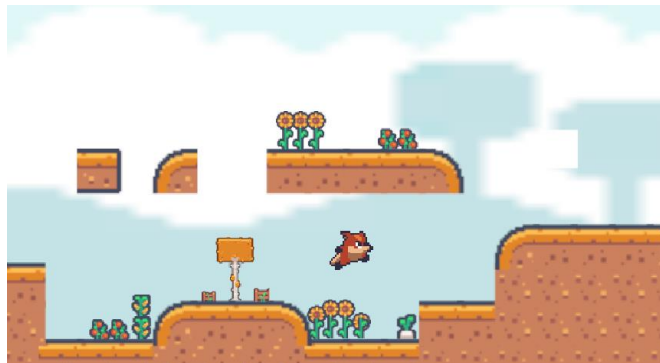
    animator.SetFloat("Blend Jump", rb.velocity.y);
}
```

33. Tambahkan baris *code* 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);
}
```

34. Tekan *button play* untuk melihat apakah karakter sudah dapat bergerak dengan animasi.



Gambar 9.28 Tampilan Saat di *Play*



B. 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);
        }
    }
}
```

Penjelasan :

Pada fungsi *HandleJumpInput*, *Boolean* diberi nilai *true* jika memenuhi kondisi yaitu ketika tombol *Space* ditekan, jika tidak maka diberi nilai *false*.

Memperbaiki logika di bagian fungsi *HandleMovementInput* agar *isIdle* diatur menjadi '*true*' hanya ketika '*move == 0*'. Dan menambahkan '*animator.SetBool("isWalking", true)*' ketika '*move*' tidak sama dengan 0.