



TUGAS PERTEMUAN: 9

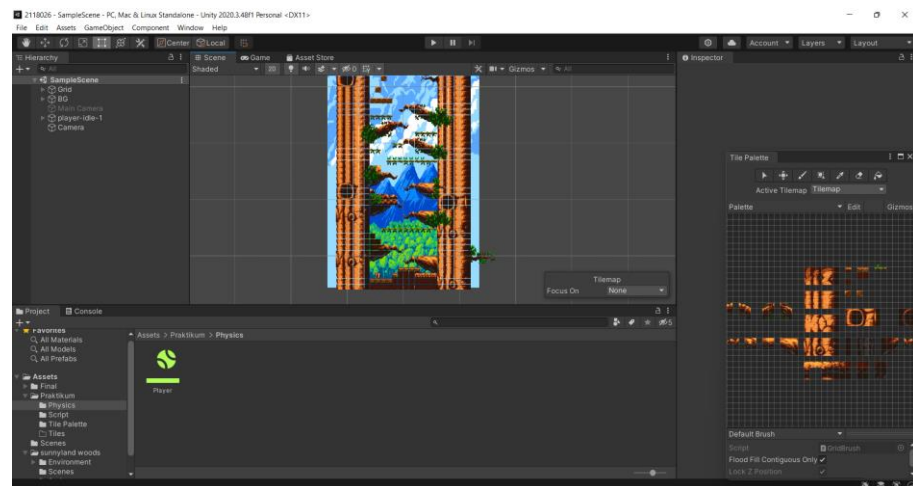
GAME ANIMATION

NIM	:	2118026
Nama	:	Bryan Ifan Etikamena
Kelas	:	B
Asisten Lab	:	Aprillia Dwi Dyah S (2118143)

9.1 Tugas 9 : Membuat Character Animation

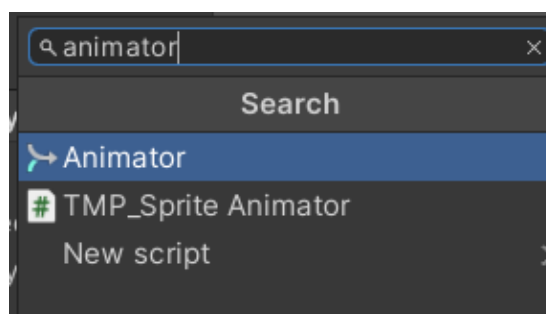
A. Langkah-langkah Membuat Character Animation

1. Buka *file* projek *Unity* sebelumnya pada bab 8 untuk digunakan kembali.



Gambar 9.1 Membuka *Project Unity*

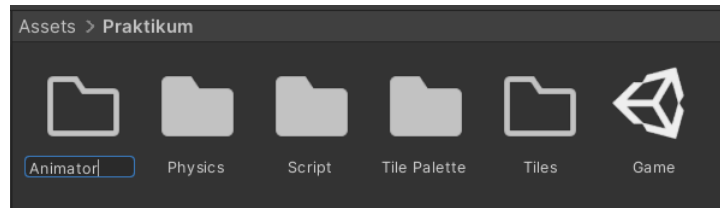
2. Pada karakter klik *inspector* kemudian pilih *Add Component Animator*.



Gambar 9.2 Membuka *Inspector* Karakter

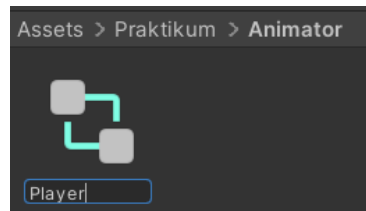


3. Pada *folder* *Praktikum* Buat *Folder* baru “Animator”.



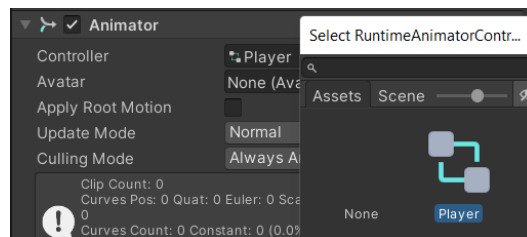
Gambar 9.3 Membuat *Folder Animator*

4. Buat *File Animator Controller* pada *folder Animator*, ubah namanya menjadi *Player*.



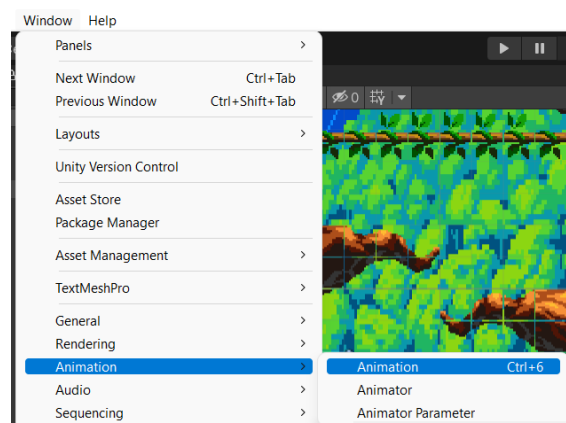
Gambar 9.4 Membuat *File Animator Controller*

5. Klik *player* pada *Hierarchy*, kemudian cari *Component Animator*, pada *setting Controller* ubah menjadi *Player*.



Gambar 9.5 Mengubah *Setting Controller*

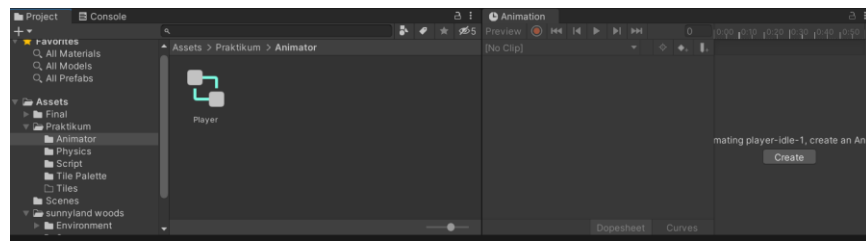
6. Tambahkan menu panel *Animation* di menu *Window*, pilih *Animation > Animation* atau tekan *Shortcut CTRL + 6*.



Gambar 9.6 Menambahkan Panel *Animation*

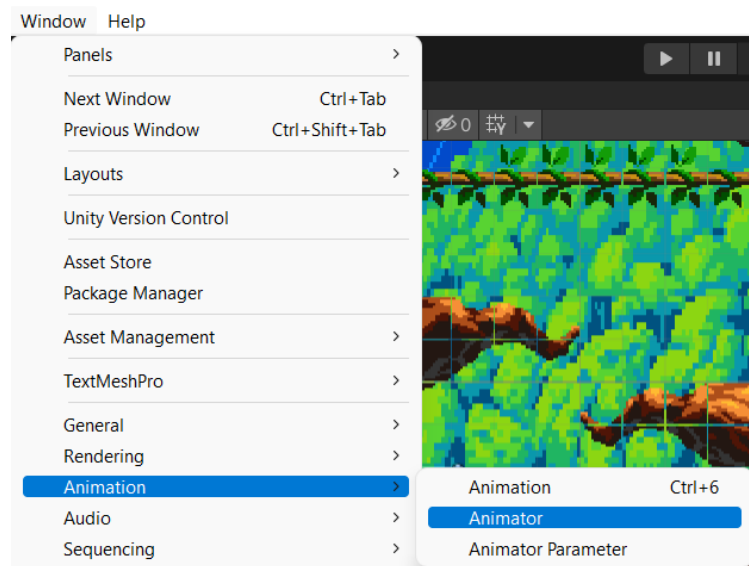


7. Akan muncul menu panel baru, geser panel tersebut di bawah sendiri seperti gambar di bawah ini, *Split Panel Project* dan *Animation*.



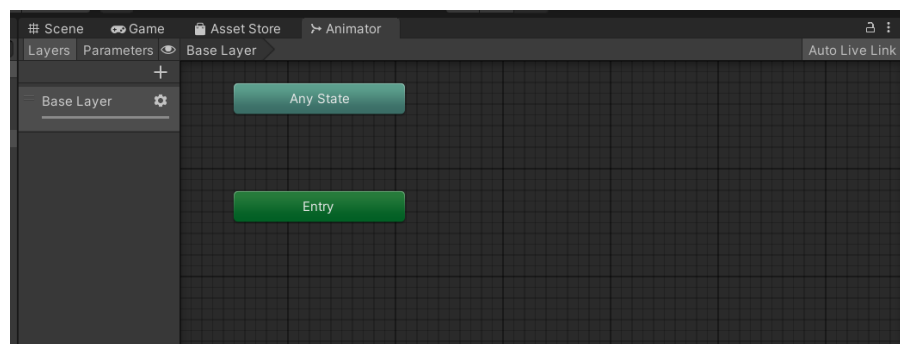
Gambar 9.7 *Split Panel*

8. Tambahkan menu panel *Animator*.



Gambar 9.8 Menambahkan Panel *Animator*

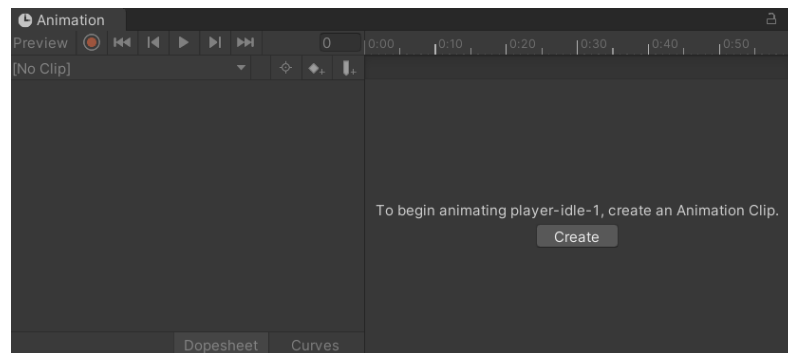
9. Geser Panel tersebut sesuai dengan gambar.



Gambar 9.9 Mengatur Posisi Panel

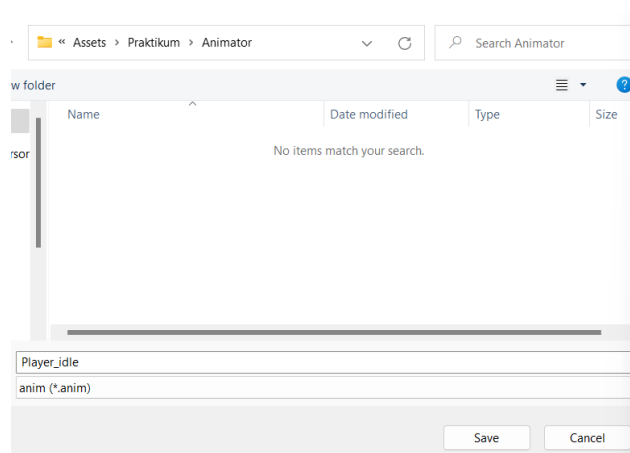


10. Untuk membuat animasi klik *player-idle1* pada *Hierarchy*, kemudian ke menu panel *Animation*, pilih *Create*.



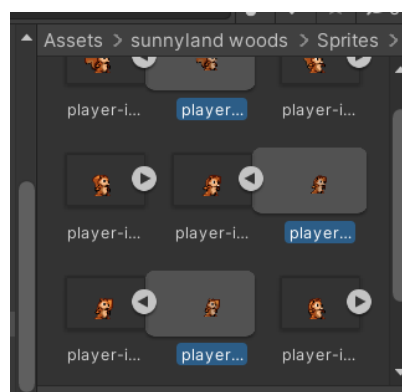
Gambar 9.10 Mencoba *Source Code*

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



Gambar 9.11 Menyimpan *Folder Animation*

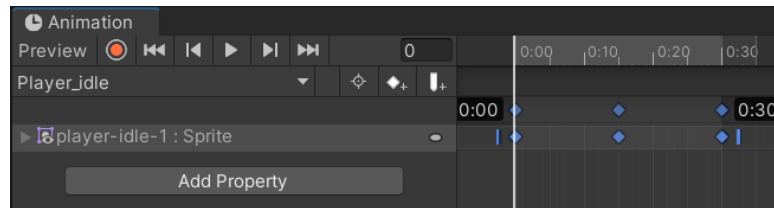
12. Pada menu *Project* buka *folder player* lalu pilih *Idle* dan pilih gambar *player-idle-1*, *player-idle-3* dan *player-idle-4*, kemudian *drag* ke *tab Animation*.



Gambar 9.12 Menambahkan *Player Idle*

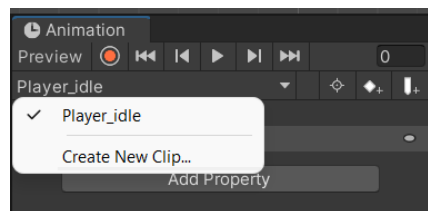


13. Tekan CTRL + A pada menu panel *Animation* geser kotak kecil pada *timeline* sampai *frame* 0:30 agar animasinya tidak terlalu cepat.



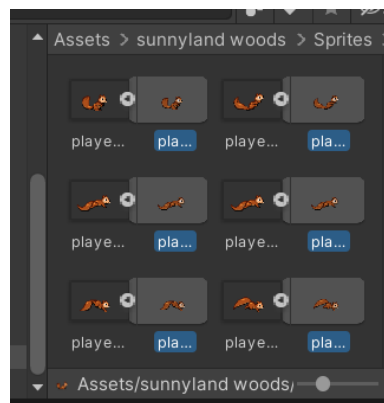
Gambar 9.13 Mengatur *Timeline* Animasi

14. Buat animasi baru, Klik pada “*Player_idle*” kemudian pilih *Create New Clip*, dan beri nama “*Player_run*”, Simpan pada Folder Animator.



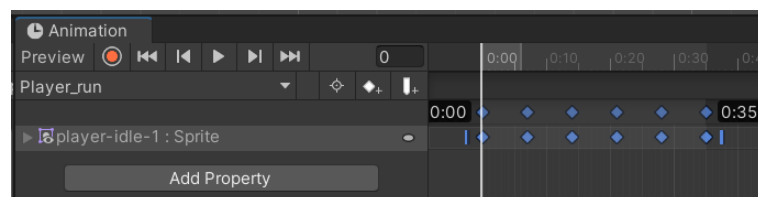
Gambar 9.14 Membuat *Clip* Baru

15. Buka menu *Project* kemudian cari *folder Player > run*, Pilih *player-run-1* sampai *player-run-6*, *drag and drop* pada menu *Animation*.



Gambar 9.15 Menambahkan *Player Run*

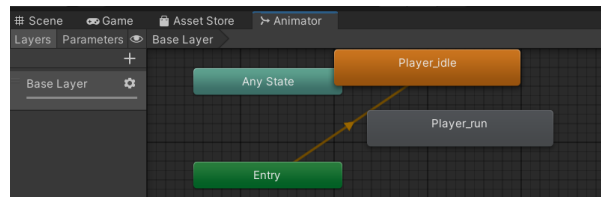
16. Pada panel *timeline* tekan Ctrl+A di *keyboard*, klik bagian kotak kecil di samping *keyframe* terakhir dan geser sampai waktu 0:35.



Gambar 9.16 Mengatur *Timeline* Animasi



17. Pilih ke menu *Animator* yang telah dibuka sebelumnya dan akan tampil seperti berikut.



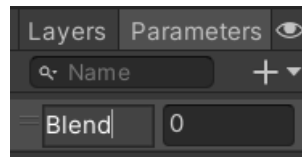
Gambar 9.17 Membuka Menu *Animator*

18. 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*.



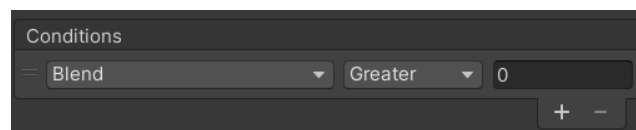
Gambar 9.18 Membuat *Transition*

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



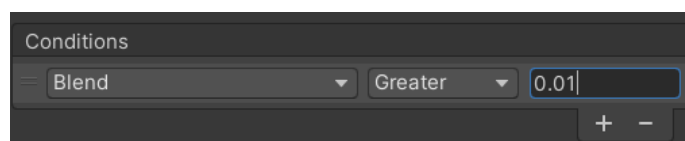
Gambar 9.19 Membuat Parameter

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



Gambar 9.20 Menambahkan Parameter

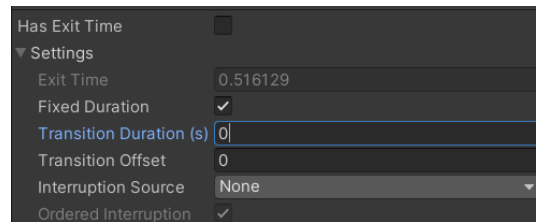
21. Atur nilai *conditions blend* tersebut menjadi 0.01.



Gambar 9.21 Mengatur Nilai *Conditions Blend*

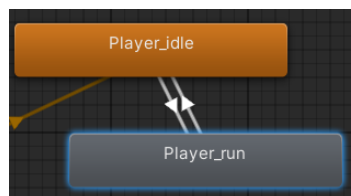


22. Pada bagian *Settings*, hilangkan centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



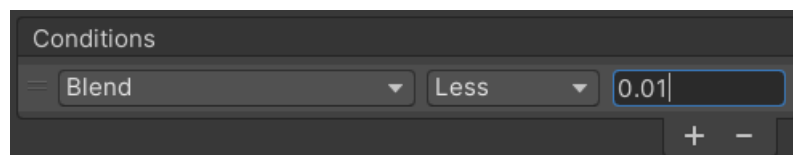
Gambar 9.22 Mengatur *Settings*

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



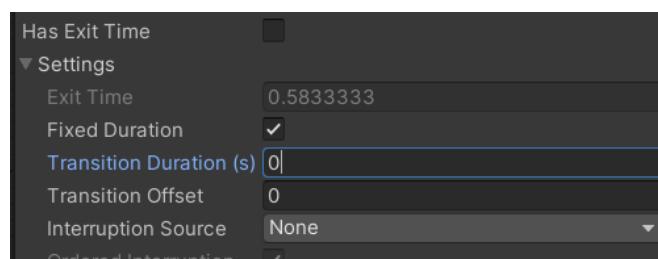
Gambar 9.23 Membuat *Transiton*

24. 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.



Gambar 9.24 Membuat *Physical Material 2D*

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



Gambar 9.25 Mengatur *Settings*

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

```
public Animator animator;
```



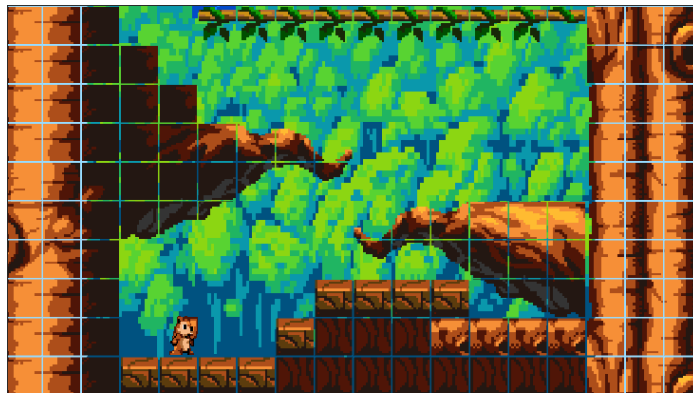
27. Tambahkan *Script* Komponen *Animator* pada *void Awake*.

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

28. Dan pada fungsi *FixedUpdate* tambahkan *source code* berikut pada *void FixedUpdate()*.

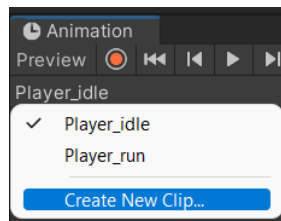
```
animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
```

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



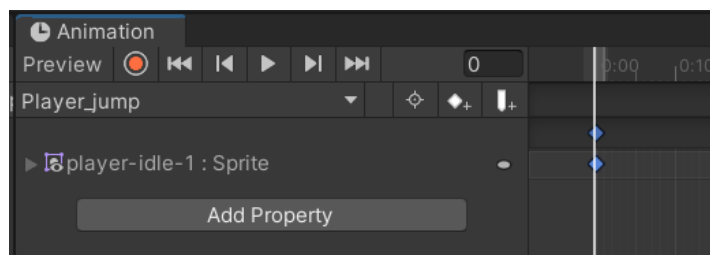
Gambar 9.26 Animasi Karakter

30. Kemudian buat animasi baru tekan tulisan “*Player_run*” kemudian pilih *Create New Clip*, dan beri nama “*Player_jump*”



Gambar 9.27 Membuat *Clip* baru

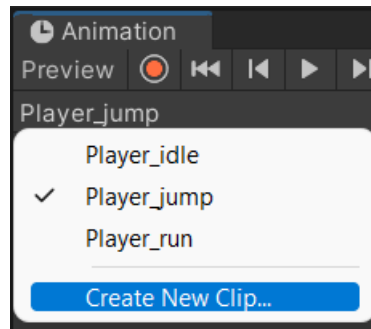
31. Pada *folder player* buka *jump* lalu pilih gambar *player-jump-1*, kemudian *drag* ke *tab Animation*.



Gambar 9.28 Menambahkan *Player Jump*

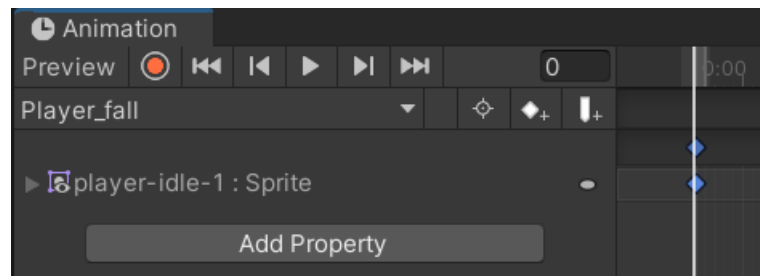


32. Buat animasi baru dengan cara tekan tulisan “*Player_jump*” kemudian pilih *Create New Clip*, dan beri nama “*Player_fall*”.



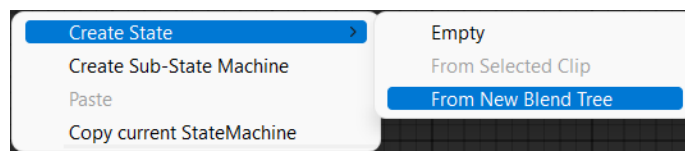
Gambar 9.29 Membuat *Clip* baru

33. Pada tab *Project* buka *folder* karakter lalu pilih *Idle* dan pilih gambar *player-crouch-1*, kemudian drag ke tab *Animation*.



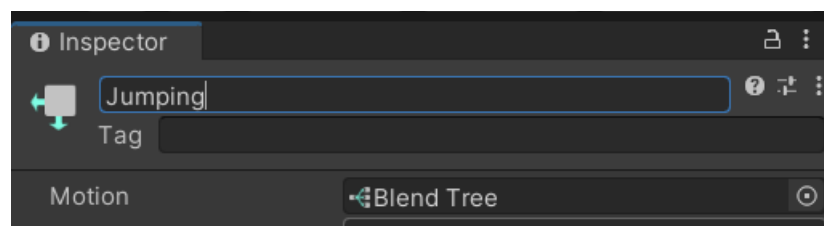
Gambar 9.30 Menambahkan *Player Fall*

34. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu *Animator*, di area kosong, pilih *Create State>From New Blend Tree*.



Gambar 9.31 Membuat *Blend Tree*

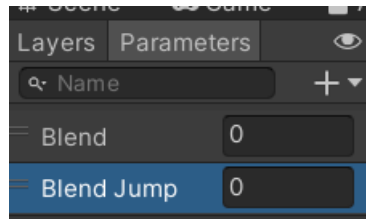
35. Pada *Animator* klik *Blend Tree*, di menu *Inspector*, ubah namanya menjadi *Jumping*.



Gambar 9.32 Mengubah Nama *Blend Tree*

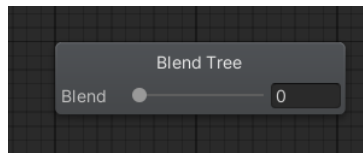


36. Pada menu *Parameters* tambahkan parameter tipe data *Float* tekan icon $+$ dan ubah namanya menjadi “Blend Jump”.



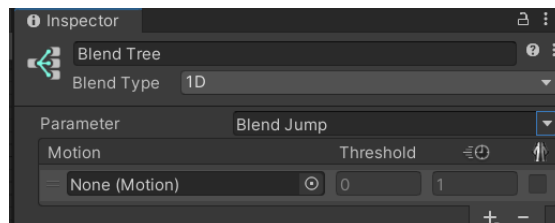
Gambar 9.33 Menambahkan Parameter Baru

37. Pada menu *Animator*, Klik dua kali pada *Blend Tree* “Jumping”, Tekan pada *Blend Tree*.



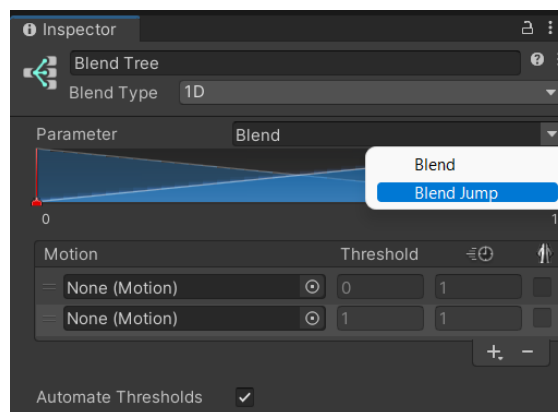
Gambar 9.34 Double Klik Jumping

38. Klik 2X *Blend Tree* “Jumping”, pada *inspector* ubah parameter menjadi “Blend Jump”.



Gambar 9.35 Mengatr Parameter

39. Buka menu *Inspector*, tekan icon $+$ dan pilih *Add Motion Field*.
Tambahkan dua *Motion Field*.



Gambar 9.36 Menambahkan *Motion Field*

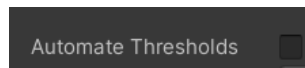


40. Klik bagian *icon None (Motion)*, maka akan muncul *Windows Motion*,
 Tambahkan Sesuai dengan urutan.



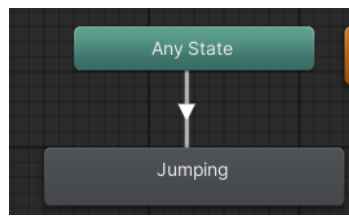
Gambar 9.37 Mengatur *Motion*

41. Hilangkan centang “*Automate Thresholds*” dan atur nilai *Threshold* seperti berikut.



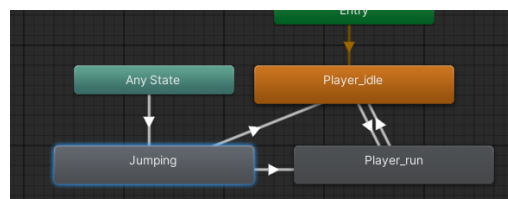
Gambar 9.38 *Uncheck Automate Thresholds*

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



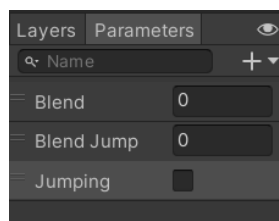
Gambar 9.39 Membuat *Transition*

43. Klik kanan *Jumping*, pilih *Make Transition* dan arahkan panahnya ke *Player_idle* dan *Player_run*.



Gambar 9.40 Membuat *Transition*

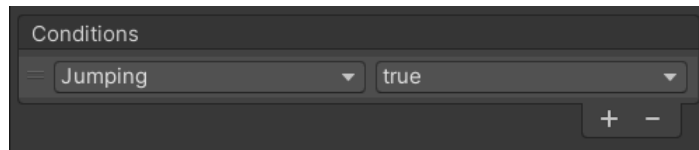
44. Tambahkan parameter transisi dengan tipe data *Bool* tekan *icon +* dan ubah namanya menjadi “*Jumping*”.



Gambar 9.41 Menambahkan Parameter

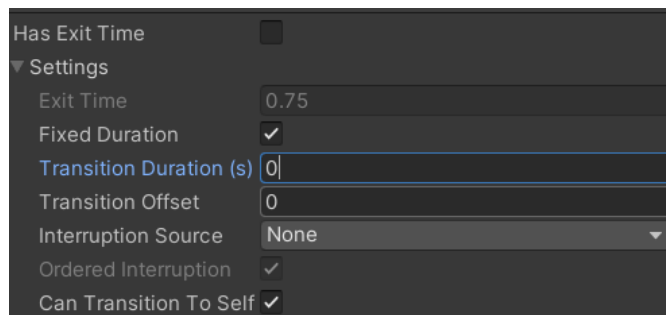


45. Klik panah yang mengarah ke *Jumping*, pada *inspector* tambahkan *condition*, pilih *condition Jumping* dan ubah nilainya menjadi *true*.



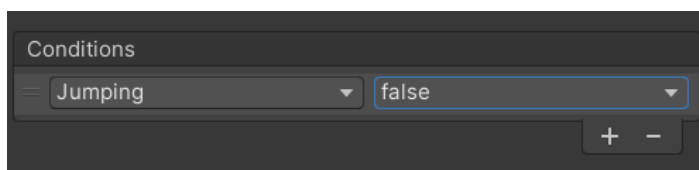
Gambar 9.42 Menambahkan *Condition*

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



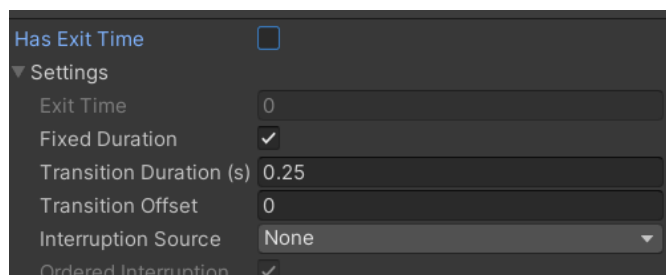
Gambar 9.43 Mengatur *Settings*

47. 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*.



Gambar 9.44 Menambahkan *Condition*

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



Gambar 9.45 Mengatur *Settings*



49. Buka *script Player*, dan tambahkan *source code* berikut pada fungsi *update*.

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

50. Pada Fungsi *FixedUpdate* tambahkan seperti berikut.

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

51. Tambahkan baris kode seperti di bawah ini 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);
}
```

52. Jika di *play* maka karakter sudah bisa bergerak dengan animasi.



Gambar 9.46 *Play Game*



9.2 Kuis Pertemuan 9

```
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);
    }
}
```

Analisis :

Terdapat dua kekurangan nilai parameter di `animator.SetBool("isJumping",);` pada void `HandleJumpInput` dimana harusnya di kedua code itu ditambahkan atau diberi nilai true atau false sehingga menjadi seperti ini `animator.SetBool("isJumping", true);` atau `animator.SetBool("isJumping", false);`. Kemudian pada void `HandleMovementInput` seharusnya untuk mengecek player dalam kondisi idle atau tidak berjalan itu harusnya menggunakan `if (move == 0)` bukan `if`



(move != 1) dan juga harusnya if (move < 0) bukan if (move != 0)
agar sesuai dengan fungsi else ifnya

```
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(-4, 1, 1);
        }
        else if (move > 0)
        {
            transform.localScale = new Vector3(4, 1, 1);
        }
    }
}
```