TUGAS PERTEMUAN: 9

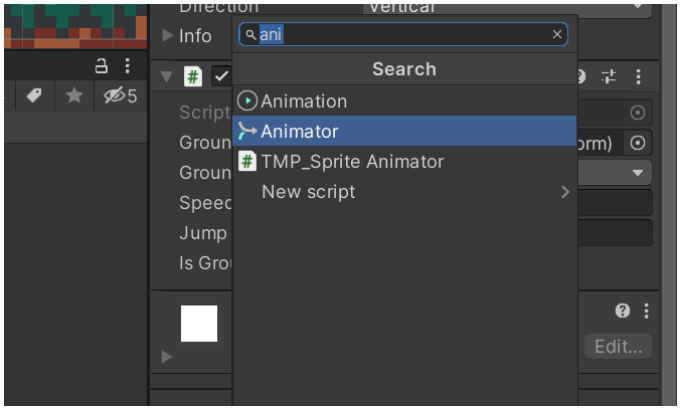
GAME ANIMATION

NIM	:	2118068
Nama	:	Ahmad Bahrul Ilmi
Kelas	:	B
Asisten Lab	:	Devina Dorkas Manuela (2218108)

1.1 Tugas 9 : Menerapkan Game Animation

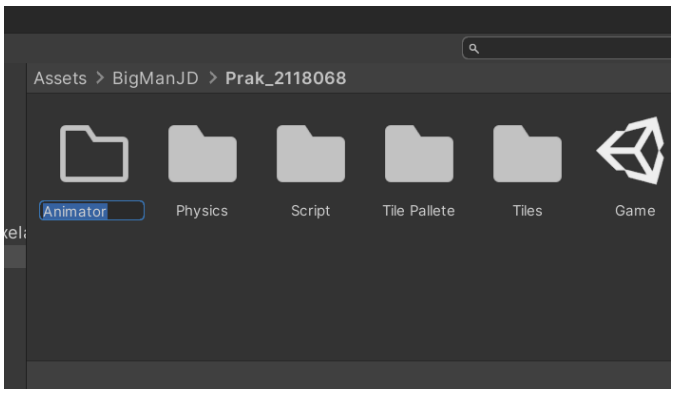
A. Langkah langkah Character Animation

1. Tambahkan komponen *Animator* pada *Inspector* di karakter yang sebelumnya.



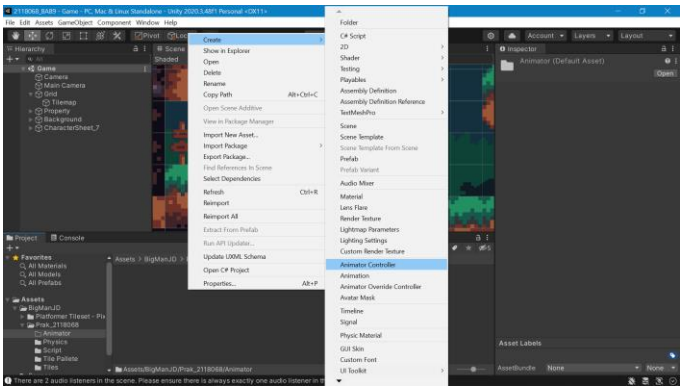
Gambar 9.1 Komponen *animator*

2. Kemudian buat *folder* baru di dalam *folder* Prak_2118068 dengan nama *Animator*.



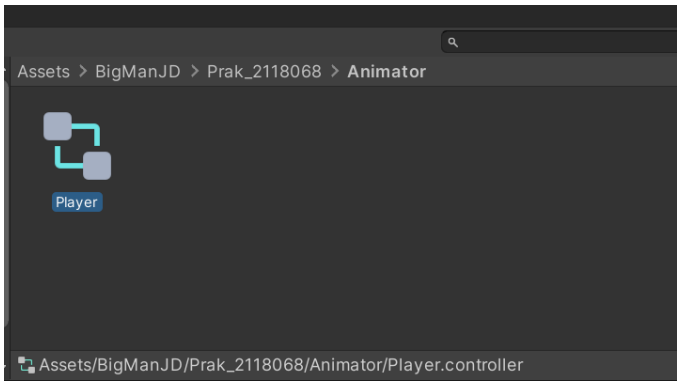
Gambar 9.2 *Folder animator*

3. Pada *folder* tersebut tambahkan *Animator Controller* dengan cara klik kanan pada *folder* pilih *create* lalu *Animator Controller*.



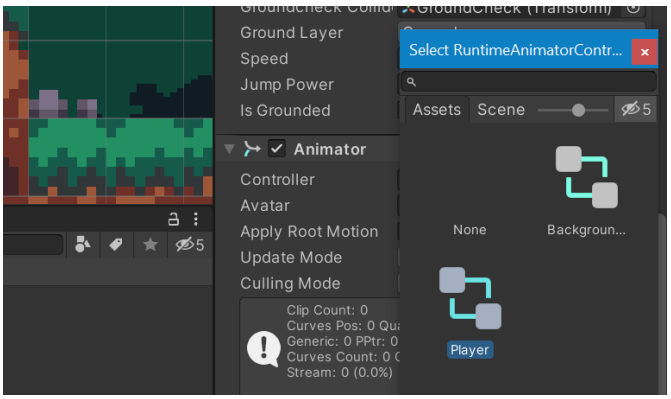
Gambar 9.3 *Animator controller*

4. Beri nama *animator controller* tersebut dengan nama *Player*.



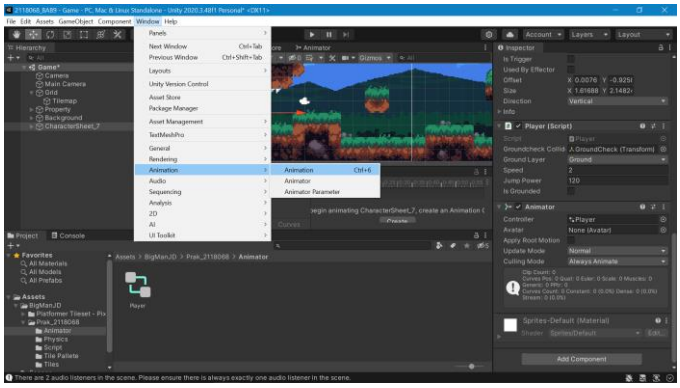
Gambar 9.4 *Animator controller player*

5. Setelah itu klik karakter dan pilih komponen *animator* pada *inspector* lalu ubah *Controller* menjadi *Player*.



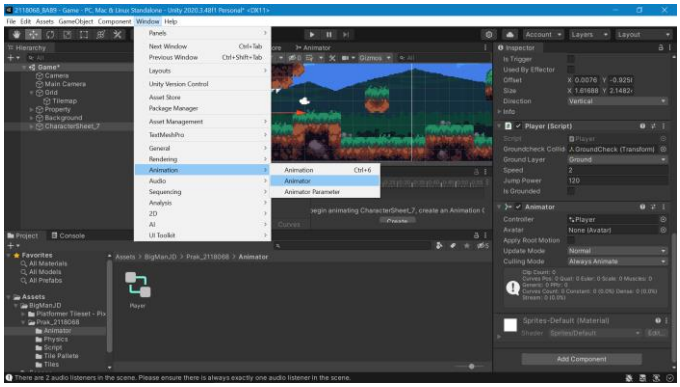
Gambar 9.5 *Controller player*

6. Tambahkan *tab Animation* dengan cara klik *Windows* pilih *Animation* dan *Animation* atau dengan *shortcut* Ctrl+6.



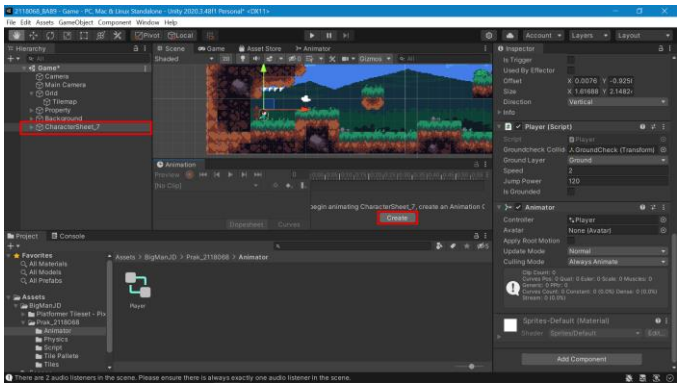
Gambar 9.6 *Tab animation*

7. Dan tambahkan juga *tab Animator* dengan cara klik *Windows* dan *Animation* lalu pilih *Animator*.



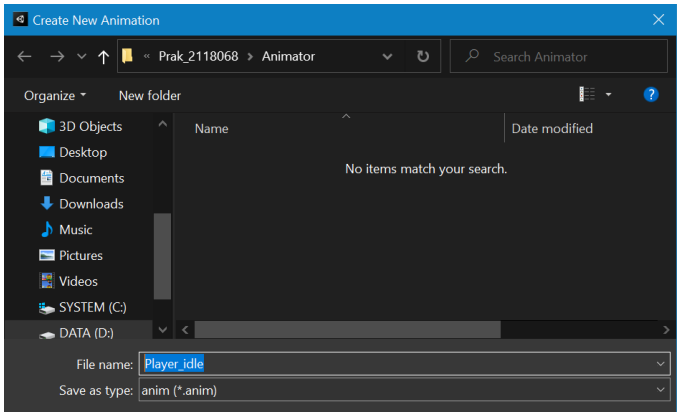
Gambar 9.7 *Tab animator*

8. Setelah itu klik karakter dan pilih *Create* pada *tab Animation*.



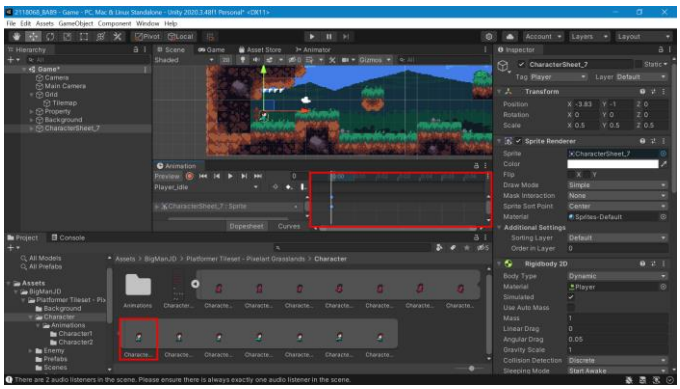
Gambar 9.8 *Create animation*

9. Maka akan muncul jendela untuk menyimpan *file animation*, lalu simpan dengan nama *Player_idle* pada *folder Animator* yang sudah dibuat sebelumnya.



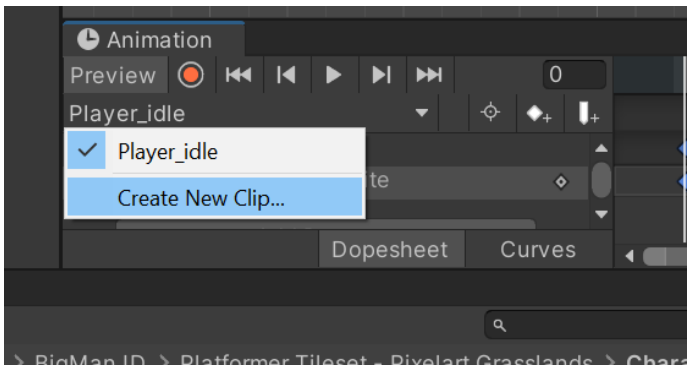
Gambar 9.9 Menyimpan *animation idle*

10. Setelah itu tambahkan karakter *idle* pada *timeline* dengan cara *drag and drop*.



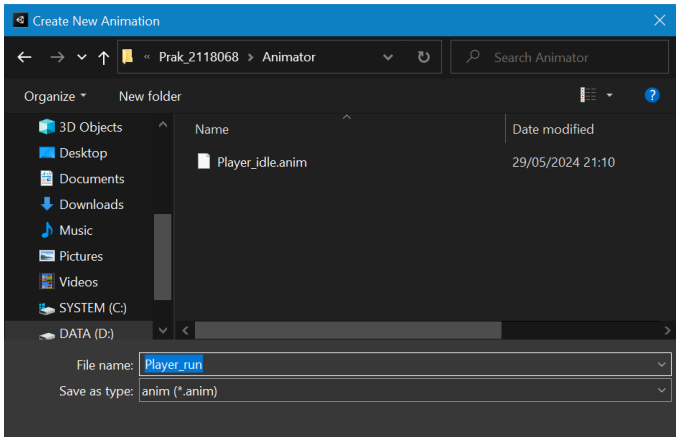
Gambar 9.10 Menambah *player idle*

11. Kemudian buat *animation* baru dengan cara klik *Player_idle* dan pilih *Create New Clip*.



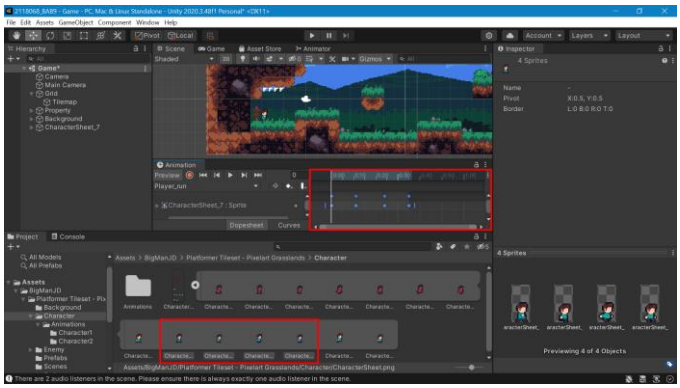
Gambar 9.11 *Create new clip*

12. Simpan dengan nama *Player_run* pada *folder Animator*.



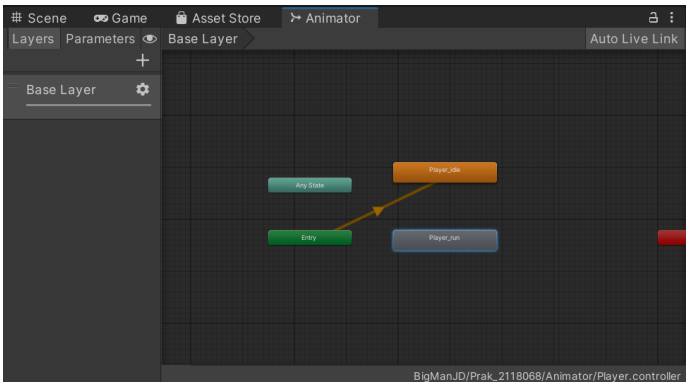
Gambar 9.12 *Menyimpan animation run*

13. Setelah itu tambahkan karakter lari pada *timeline* seperti berikut dan atur juga durasinya dengan menggeser garis *keyframe* pada *timeline*.



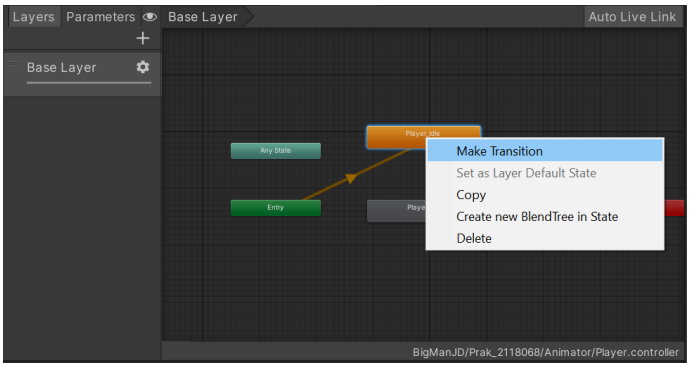
Gambar 9.13 Menambah *palyer run*

14. Pilih *tab animation* untuk memberi transisi pada karakter dari *idle* ke *run*.



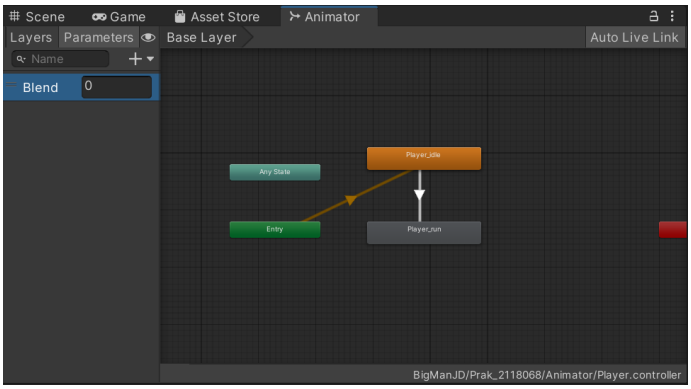
Gambar 9.14 *Tab animator* karakter

15. Beri transisi dengan cara klik kanan pada *player_idle* dan *Make Transition* lalu klik pada *player_run*.



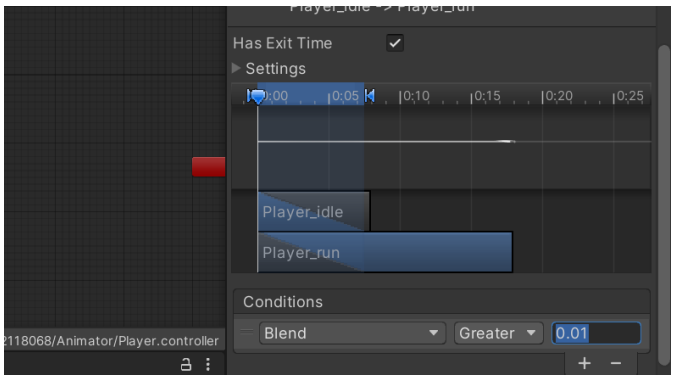
Gambar 9.15 Menambah transisi karakter

16. Kemudian pilih *tab Parameters* dan buat tipe data baru dengan klik *icon plus* dan pilih *float* dan beri nama *Blend*.



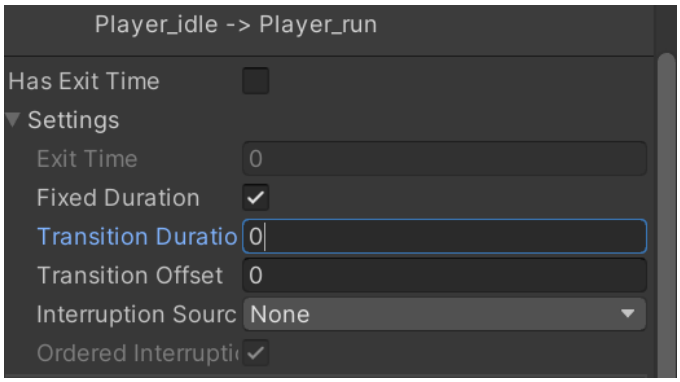
Gambar 9.16 Parameter *blend*

17. Klik panah transisi kemudian pada *inspector* ubah *Condition* menjadi *Blend* dan *Greater* nya menjadi 0.01.



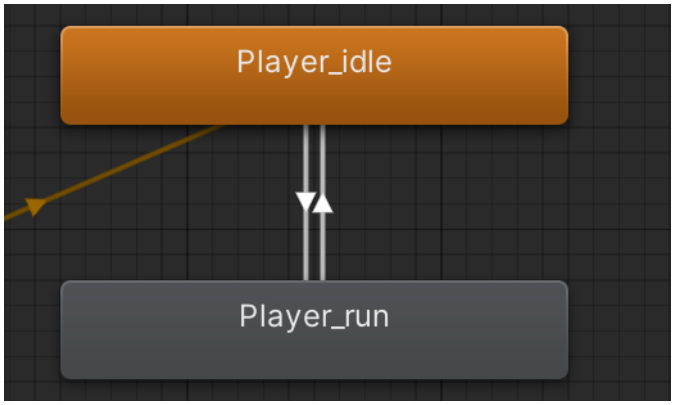
Gambar 9.17 *Condition idle run*

18. Dan pilih *settings unchecklist Has Exit Time* dan ubah *Transition Duration* menjadi 0.



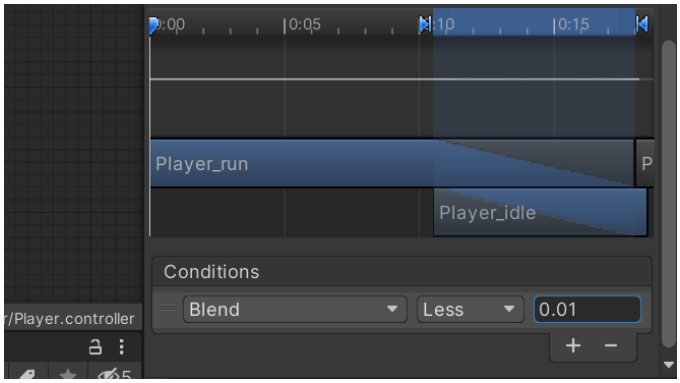
Gambar 9.18 *Settings idle run*

19. Buat transisi baru lagi dari *player_run* ke *player_idle* dengan cara yang sama seperti sebelumnya.



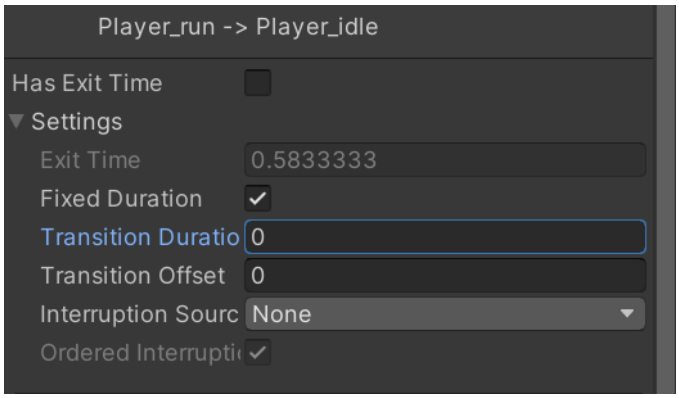
Gambar 9.19 Transisi *run* ke *idle*

20. Ubah *Conditions* menjadi *Blend* dan ubah *Operator* manejasi *Less*.



Gambar 9.20 *Condition run idle*

21. Dan *unchecklist Has Exite Time* kemudian ubah *Transition Duration* menjadi 0.



Gambar 9.21 *Settings run idle*

22. Buka *script Player* dan tambahkan variabel baru dengan nama *animator*.

```
public Animator animator;
```

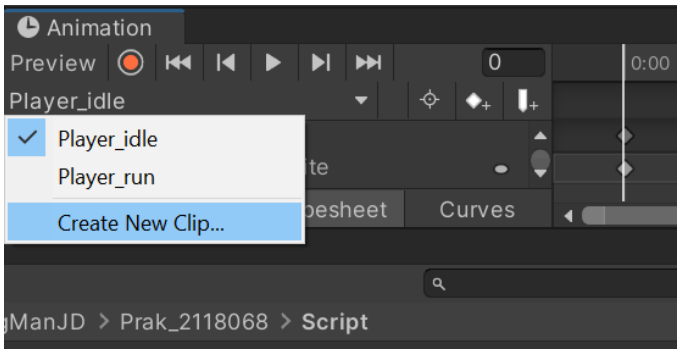
23. Pada fungsi *Awake* tambahkan komponen baru seperti berikut.

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

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

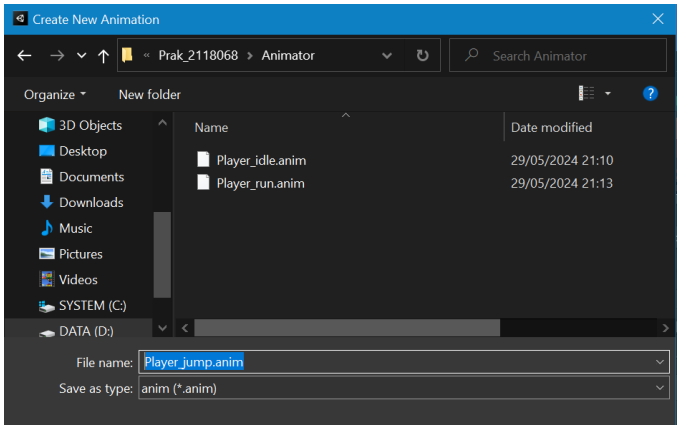
```
animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
```

25. Kemudian tambahkan *clip* baru lagi.



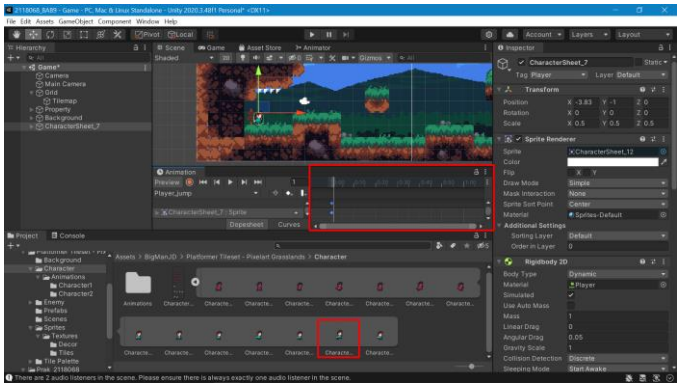
Gambar 9.22 Menambah *clip* baru

26. Dan beri nama *Player_jump* simpan pada *folder Animator*.



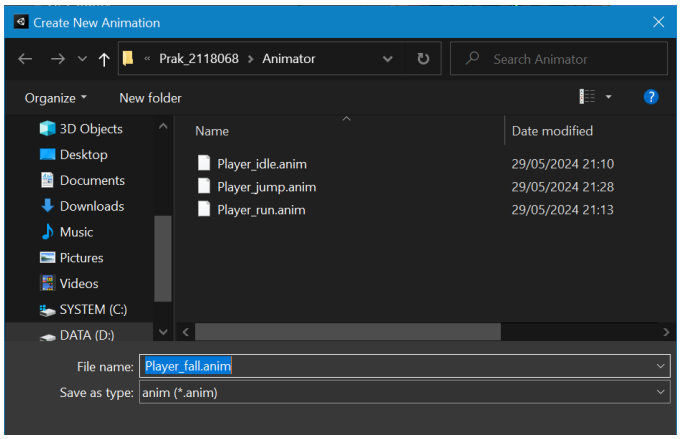
Gambar 9.23 Menyimpan *animation jump*

27. Kemudian tambahkan karakter melompat pada *timeline* seperti berikut.



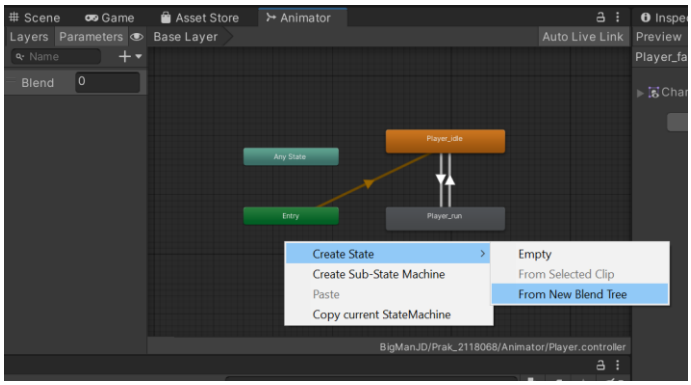
Gambar 9.24 Menambah *player jump*

28. Dan buat *clip* baru lagi dan simpan dengan nama *Player_fall*.



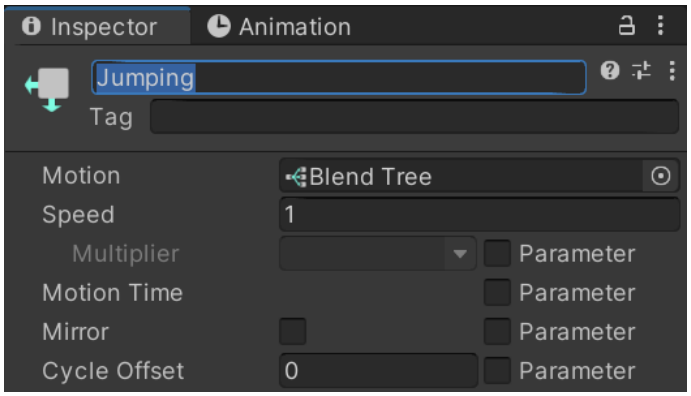
Gambar 9.25 Menimpan *animation fall*

29. Setelah itu pilih *tab Animator* dan buat *State Blend Tree* dengan cara klik kanan pilih *Create State* dan pilih *From New Blend Tree*.



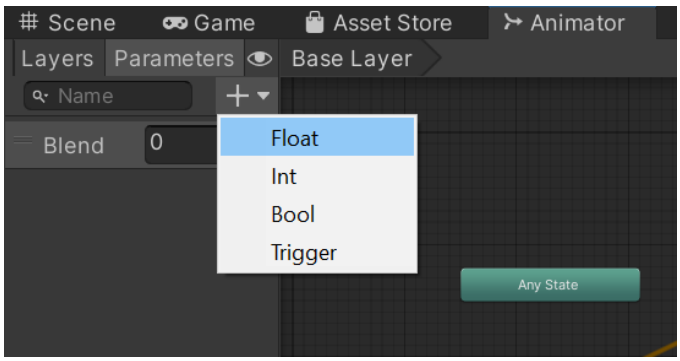
Gambar 9.26 *State blend tree*

30. Pada *inspector* ubah nama *state* menjadi *Jumping*.



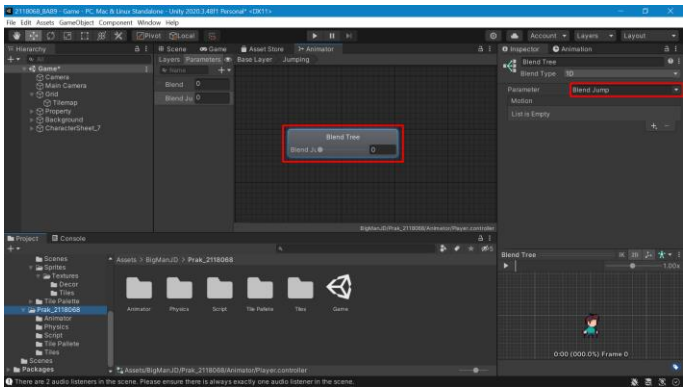
Gambar 9.27 Ubah nama *state*

31. Buat parameter baru lagi dengan tipe data *float* dan beri nama *Blend Jump*.



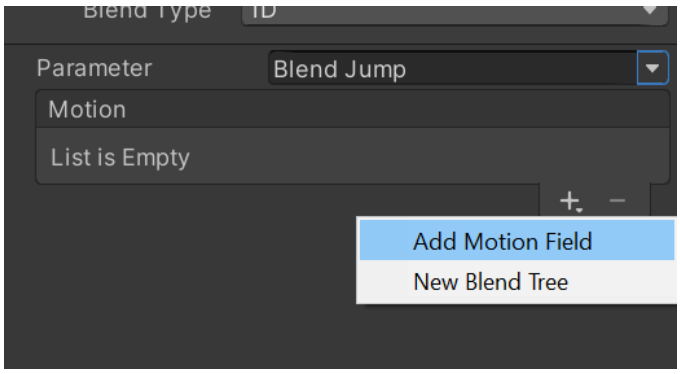
Gambar 9.28 Parameter *blend jump*

32. Klik dua kali pada *Blend Tree* dan ubah parameter pada *inspector* menjadi *Blend Jump*.



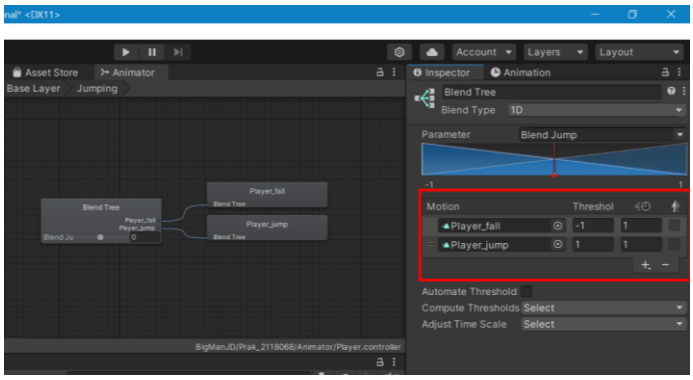
Gambar 9.29 Ubah parameter *blend jump*

33. Buat dua *motion field* pada *blend tree* dengan klik *icon plus*.



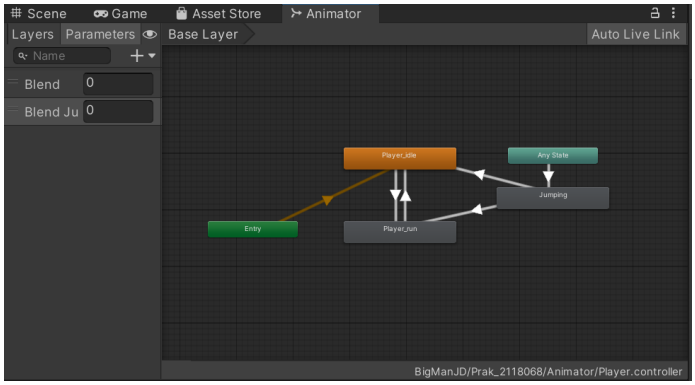
Gambar 9.30 *Add motion field*

34. Dan ubah *motion* menjadi seperti berikut.



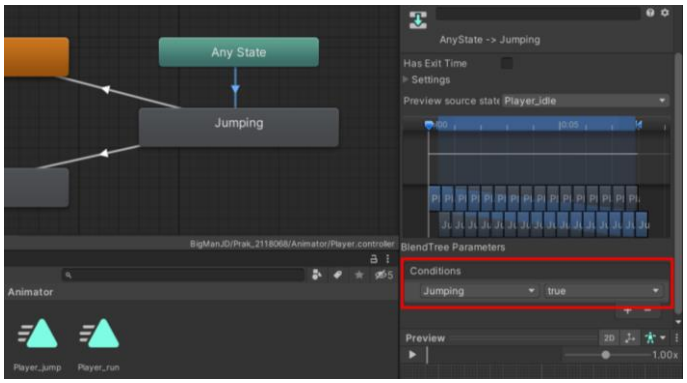
Gambar 9.31 Ubah *motion filed*

35. Kemudian atur *Animator* dengan menambah transisi seperti berikut.



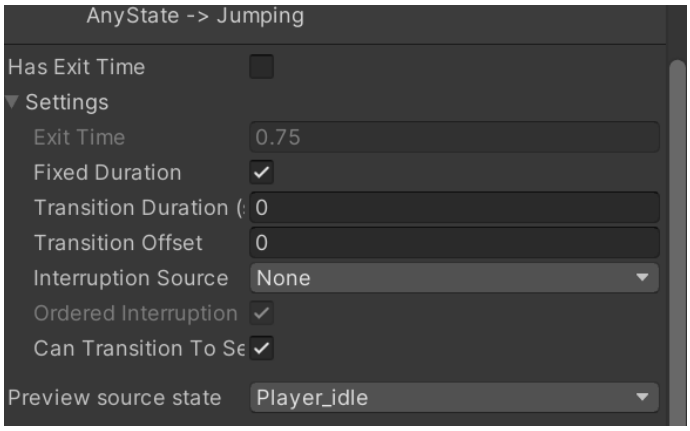
Gambar 9.32 Menambah transisi *jumping*

36. Klik transisi *Any State* ke *Jumping* dan pada *inspector* ubah *Condition* menjadi *Jumping* dan beri operator menjadi *True*.



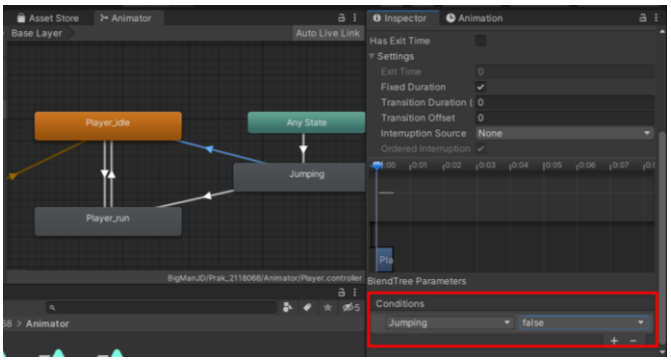
Gambar 9.33 Parameter *jumping*

37. Dan atur juga untuk *settings* transisi *anystate* ke *jumping*.



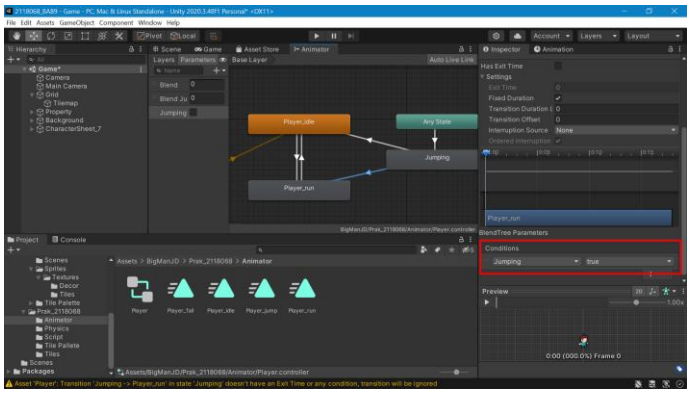
Gambar 9.34 *Settings anystate jumping*

38. Dan untuk transisi *Jumping* ke *Player Idle* ubah *Condition* nya menjadi seperti berikut dan ubah juga *settings* seperti sebelumnya.



Gambar 9.35 Transisi *jumping* ke *idle*

39. Lalu ubah jiga transisi dan juga *Condition* serta *Settings* dari *Jumping* ke *Player run*.



Gambar 9.36 Transisi *jumping* ke *run*

40. Setelah itu buka *script player* dan tambahkan perintah berikut didalam kondisi *IF* pada fungsi *Update*.

```
animator.SetBool("Jumping", true);
```

41. Tambahkan perintah berikut pada fungsi *FixedUpdate*.

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

42. Dan juga perintah berikut pada fungsi *GroundCheck* dibawah kondisi *IF*.

```
Animator.SetBool("Jumping", !isGrounded);
```

43. Jika *project* dijalankan maka sekarang terdapat animasi berlari dan melompat pada karakter.

Gambar 9.37 Hasil *game animation*

B. Link Github

https://github.com/AhmadBahrulIlmi/2118068_PRAK_ANIGAME.git

C. Kuis

1. Kesalahan

```
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);
        }
    }
```

2. Tambahan

```
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:

Kesalahan pada *source code* diatas terdapat pada fungsi *HandleJumpInput* yang memiliki kondisi *if else*, pada kondisi *if* tidak adanya parameter kedua yaitu bernilai *false* atau *true* yang harusnya jika *space* ditekan maka menghasilkan nilai *true*, maka *("isJumping", true)* dan pada kondisi *else if* jika *space* masih ditekan maka kondisinya *("isJumping", true)* kemudian seharusnya ada kondisi lagi yaitu jika *space* tidak ditekan dengan menambahkan kondisi *else ("isJumping", false).*

Lalu pada fungsi *HandleMovementInput* terdapat kesalahan pada kondisi yang ada, pada kondisi *if* yang awalnya (!=1) harusnya (==0) karena kondisi tersebut untuk memberi perintah animasi karakter tidak digerakkan. Maka pada kondisi tersebut memiliki perintah *("isIdle", true)* kemudian *("isWalking", false)* dan untuk *transform.Translate()* harusnya diletakkan pada kondisi jika karakter digerakkan yaitu kondisi *else* karena kondisi (ini merupakan perinah jika karakter digerakkan. Maka untuk kondisi *else* tidak adanya perintah yang *valid*, pada perintah tersebut harusnya terdapat *("isIdle", false) ("isWalking", true)* dan t*ransform.Translate(Vector3.right * move * Time.deltaTime)* dan kenapa *Vector3.right* yang awalnya *Vector3.left*, karena jika *left* maka pergerakan karakter tidak sama dengan arah yang sebenarnya atau terbalik. Kemudian untuk kondisi *if* yang kedua harusnya *move<0* dan *Vector3(-1, 1, 1)* karena jika kurang dari 0 maka karakter menghadap ke kiri dan parameter *vector* nilainya harus sama. Dan untuk kondisi *else if move>0* itu juga parameter *vector3* nya harus bernilai sama maka *Vector3(1, 1, 1)* untuk karakter menghadap ke kanan.