



TUGAS PERTEMUAN: 9

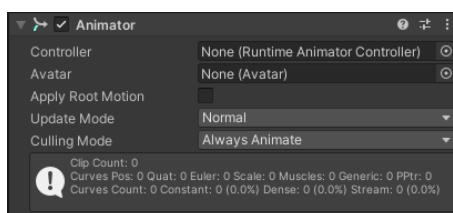
GAME ANIMATION

NIM	:	2118089
Nama	:	Shiva Divanti Natasya
Kelas	:	B
Asisten Lab	:	Nur Aria Hibnastiar

8.1 Tugas 1 : Character Animation

A. Membuat Character Animation

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



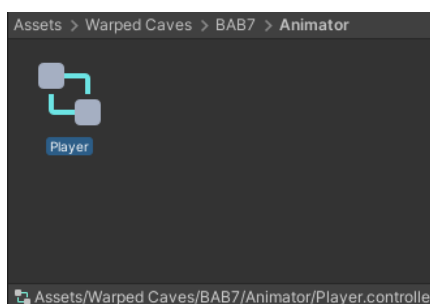
Gambar 9.1 Add Component Animator

2. Pada folder BAB7 buat folder baru “Animator”.



Gambar 9.2 Membuat Folder Animator

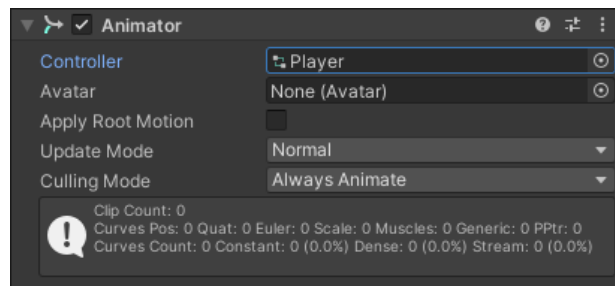
3. Buat *File Animator Controller* pada folder *Animator* , ubah namanya menjadi *Player*.



Gambar 9.3 Membuat File Animator

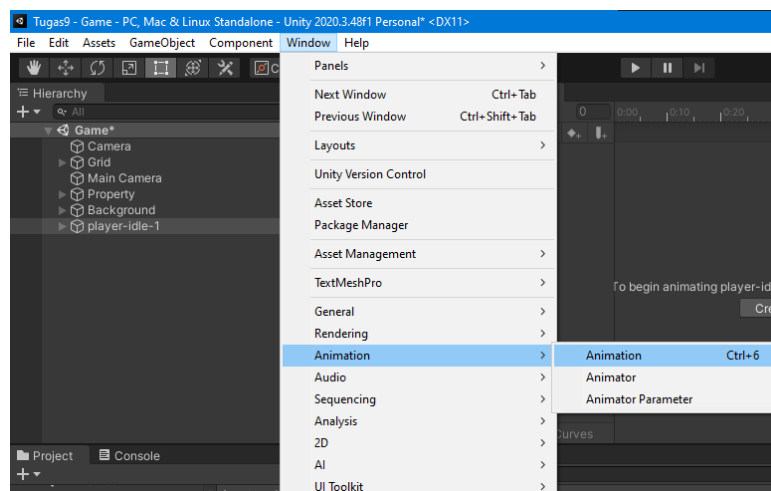


4. Klik player pada *Hierarchy*, kemudian cari *Component Animator*, pada setting *Controller* ubah menjadi *Player*.



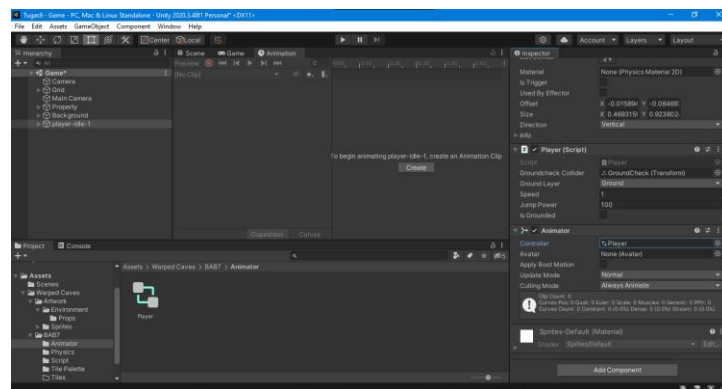
Gambar 9.4 Mengubah Menjadi *Player*

5. Tambahkan menu panel *Animation* di *menu Window*, pilih *Animation > Animation*.



Gambar 9.5 Menambahkan *Menu Panel Animation*

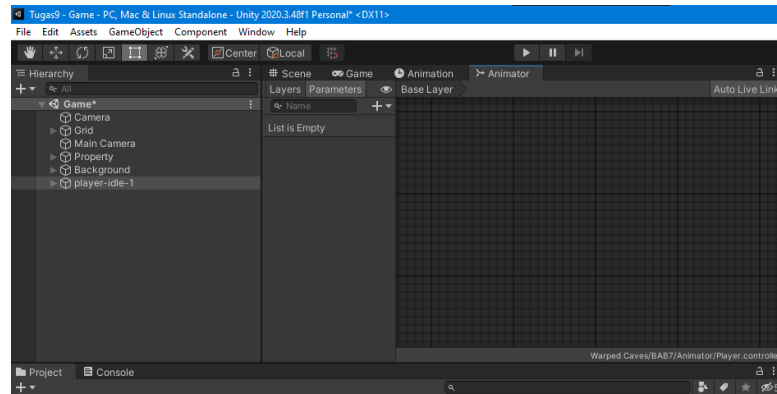
6. Akan muncul menu panel baru, geser panel tersebut dibawah sendiri seperti gambar dibawah ini, *Split Panel Project* dan *Animation*.



Gambar 9.6 *Split Projek*

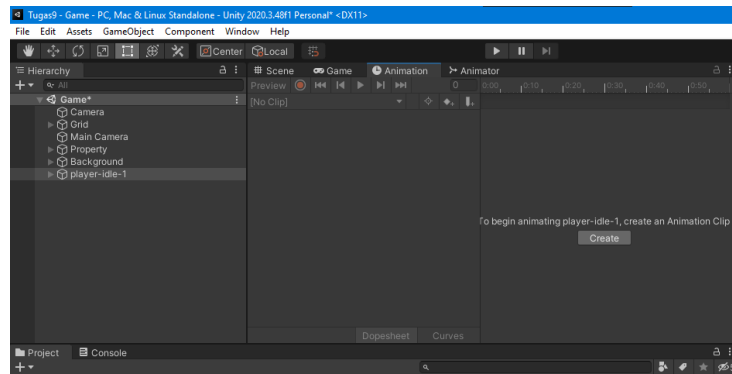


7. Tambahkan *menu* panel Animator, lalu geser Panel tersebut.



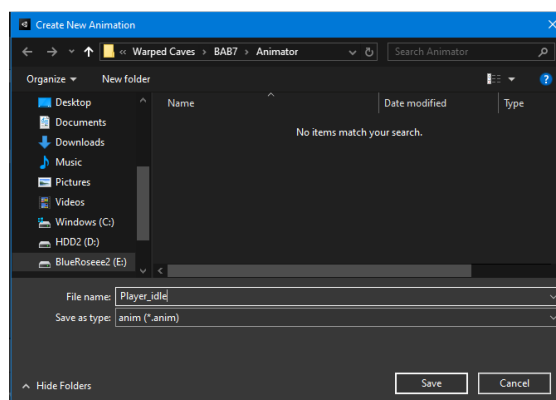
Gambar 9.7 Panel *Animator*

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



Gambar 9.8 Pilih *Create* Untuk Membuat Animasi

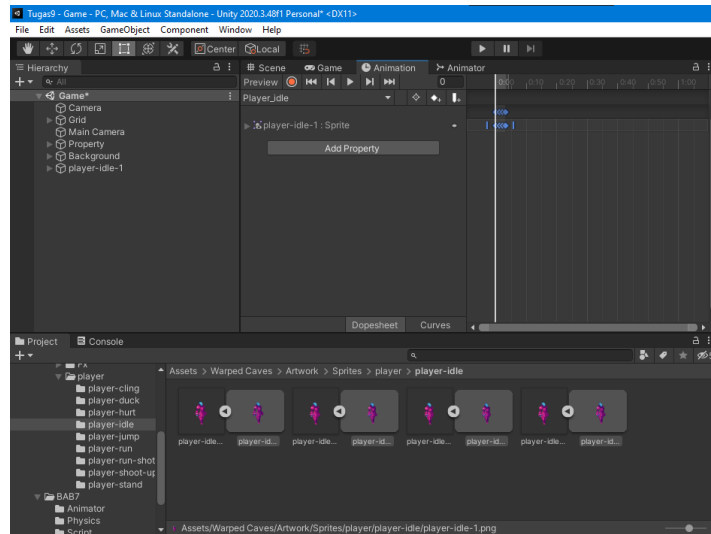
9. Simpan pada folder *Animator* dan beri nama “Player_idle”.



Gambar 9.9 Simpan Di Folder *Animator*

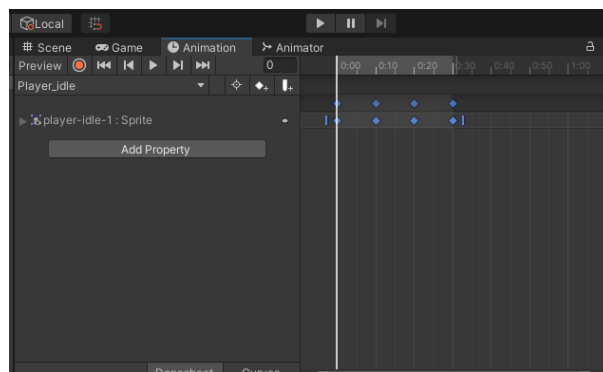


10. Pada menu *Project* buka folder *player* lalu pilih *Idle* dan pilih gambar *player-idle-1*, *player-idle-3* dan *player-idle-4*, kemudian drag ke tab *Animation*.



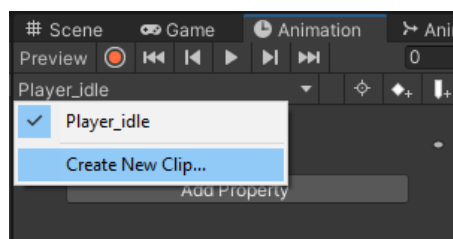
Gambar 9.10 Pilih *Player Idle*

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



Gambar 9.11 Buat *Timeline* Sampai Frame 0:30

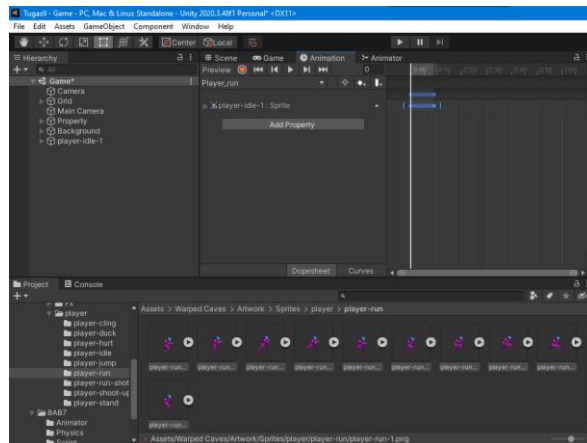
12. Buat animasi baru, Klik pada “*Player_idle*” kemudian pilih *Create New Clip*, dan beri nama “*Player_run*”, Simpan pada Folder *Animator*.



Gambar 9.12 Buat Animasi Baru *Player_run*

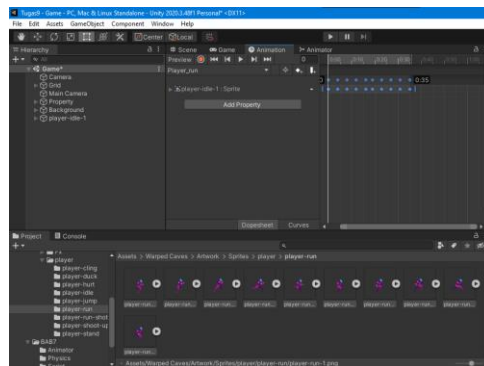


13. Buka menu *Project* kemudian cari folder *Player > run*, Pilih *player-run-1* sampai *player-run-6*, *drag and drop* pada menu *Animation*.



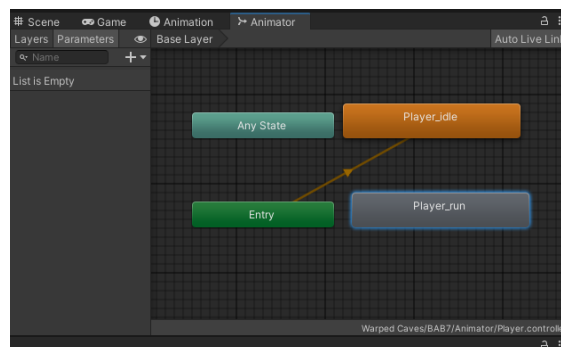
Gambar 9.13 Pilih *Player Run*

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



Gambar 9.14 Buat *Timeline 0:35*

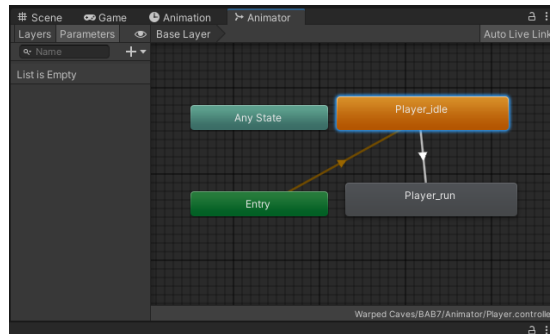
15. Pilih ke *menu Animator* yang telah dibuka sebelumnya dan akan tampil seperti berikut.



Gambar 9.15 *Menu Animator*

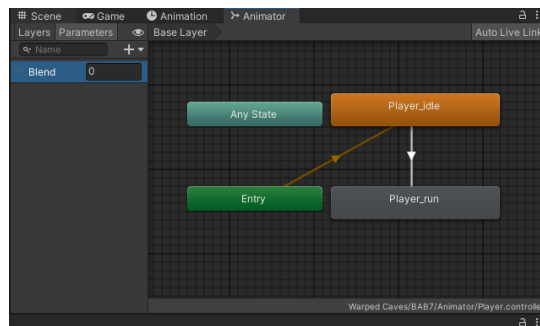


16. Kemudian buat transisi antara `player_idle` dan `player_run` dengan cara klik kanan pada `player_idle` dan pilih `Make Transition` dan tarik ke `player_run`.



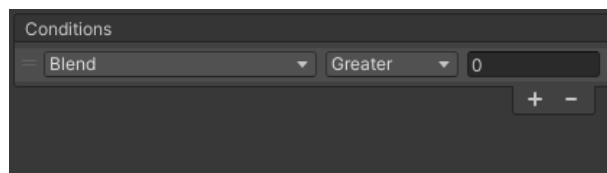
Gambar 9.16 Buat Transisi Antara `Player_Idle` Dan `Player_Run`

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



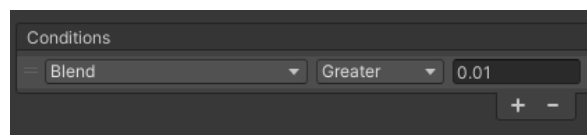
Gambar 9.17 Buat Tipe Data Blend

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



Gambar 9.18 Pilih Kondisi Blend

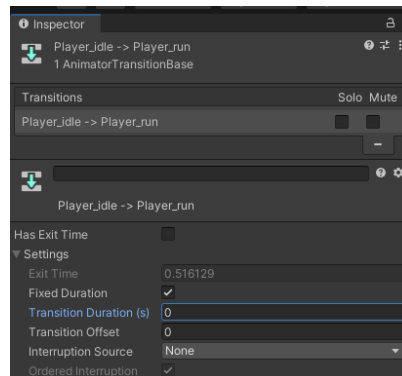
19. Atur nilai *conditions* blend tersebut menjadi 0.01.



Gambar 9.19 Atur Nilai Kondisi

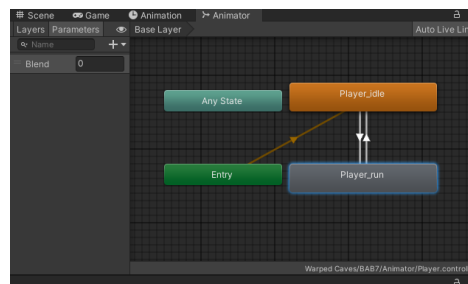


20. Pada bagian *Settings*, hilangkan centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



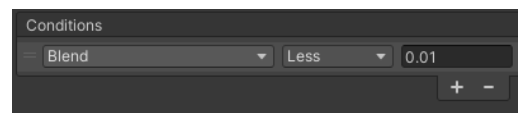
Gambar 9.20 Atur *Settings*

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



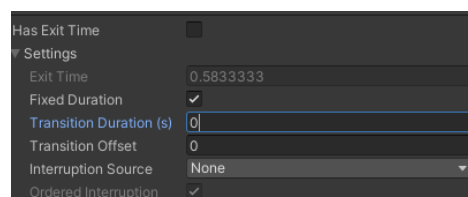
Gambar 9.21 Buat Transisi Dari *Player_Run* Ke *Player_Idle*

22. 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 9.22 Atur Parameter dan Operator

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



Gambar 9.23 Mengatur *Settings*



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

```
public Animator animator;
```

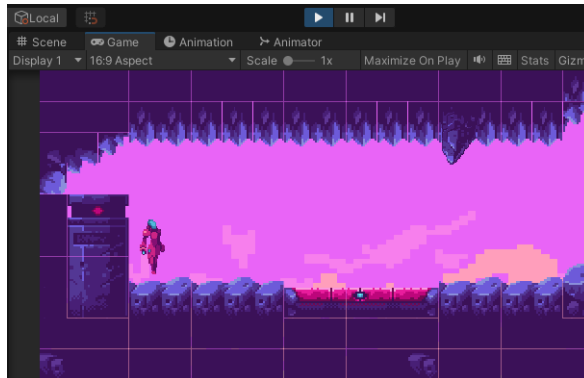
25. Tambahkan *Script* Komponen Animator.

```
private void Awake()  
{  
    rb = GetComponent<Rigidbody2D>();  
    animator = GetComponent<Animator>();  
}
```

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

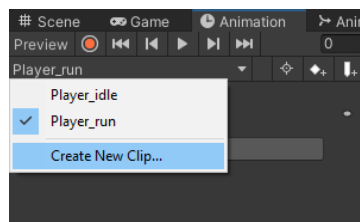
```
void FixedUpdate()  
{  
    GroundCheck();  
    Move(horizontalValue, jump);  
  
    animator.SetFloat("Blend",  
    Mathf.Abs(rb.velocity.x));  
}
```

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



Gambar 9.24 Tampilan Animasi Berjalan dan Berhenti

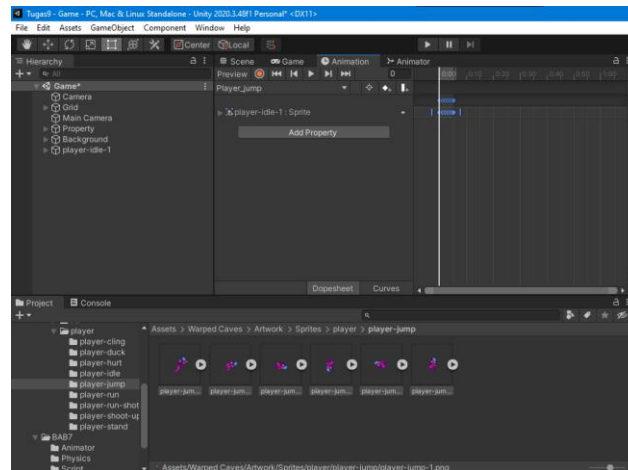
28. Kemudian buat animasi baru tekan tulisan “Player_run” kemudian pilih *Create New Clip*, dan beri nama “Player_jump”.



Gambar 9.25 Buat Animasi Lompat

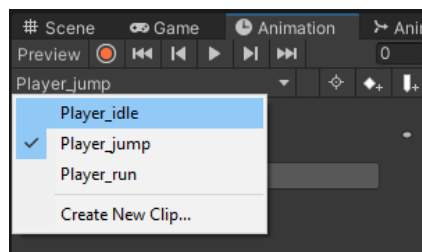


29. Pada folder *player* buka *jump* lalu pilih gambar *player-jump-1*, kemudian *drag* ke tab *Animation*.



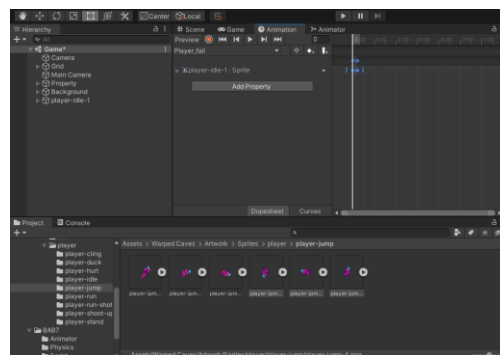
Gambar 9.26 Pilih Gambar *Jump*

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



Gambar 9.27 Buat Animasi Jatuh

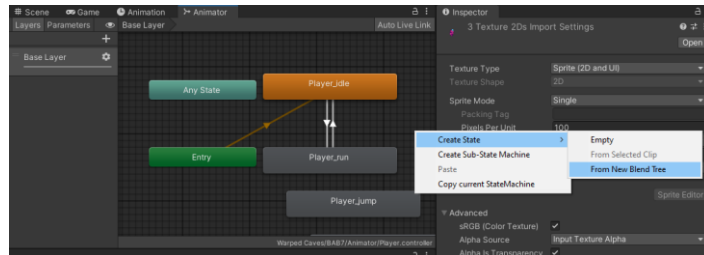
31. Pada tab *Project* buka folder karakter lalu pilih *Idle* dan pilih gambar *player-fall*, kemudian *drag* ke tab *Animation*.



Gambar 9.28 Pilih Gambar *Fall*

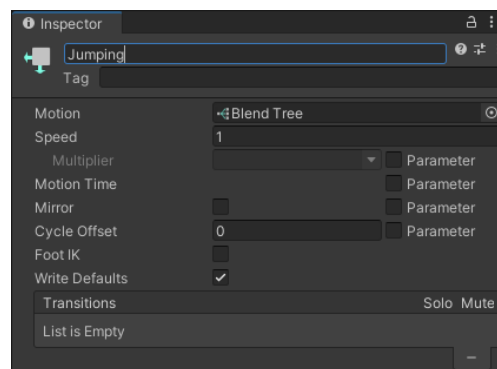


32. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu *Animator*, di area kosong , pilih *Create State>From New Blend Tree*.



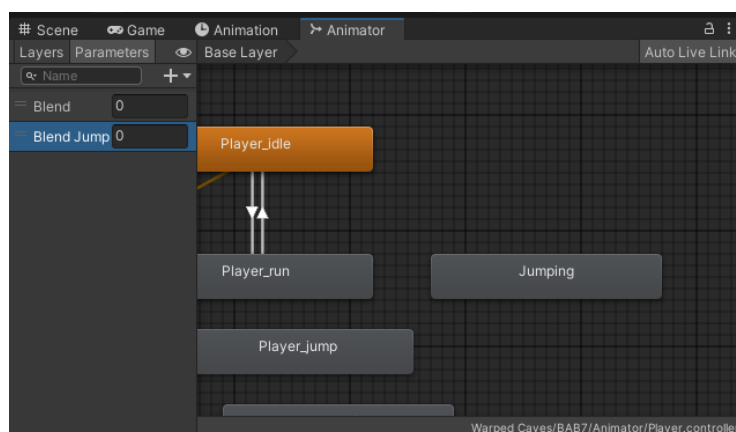
Gambar 9.29 Menambahkan Animasi Melompat

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



Gambar 9.30 Ubah Nama Menjadi *Jumping*

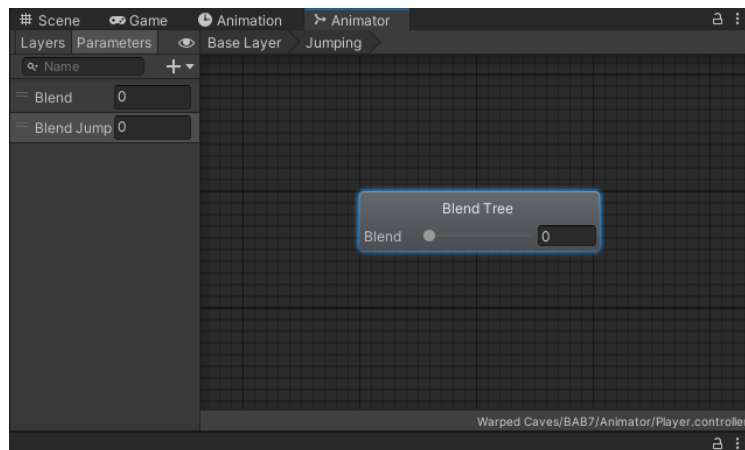
34. Pada menu *Parameters* tambahkan parameter tipe data *Float* tekan icon + dan ubah namanya menjadi “Blend Jump”.



Gambar 9.31 Buat Tipe Data *Blend Jump*

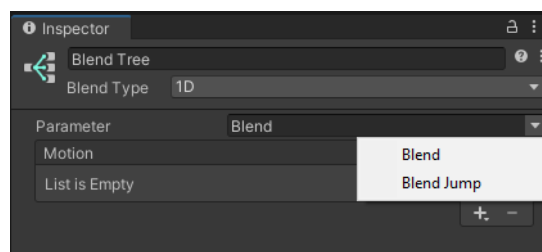


35. Pada *menu Animator*, Klik dua kali pada *Blend Tree “Jumping”*, Tekan pada *Blend Tree*.



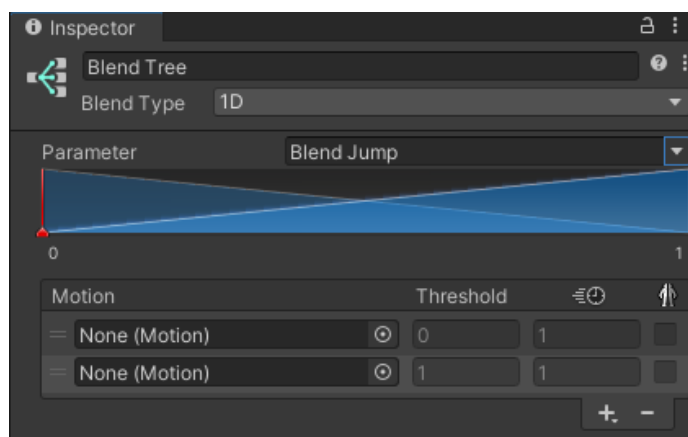
Gambar 9.32 Klik Dua Kali *Blend Tree*

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



Gambar 9.33 Mengubah Parameter

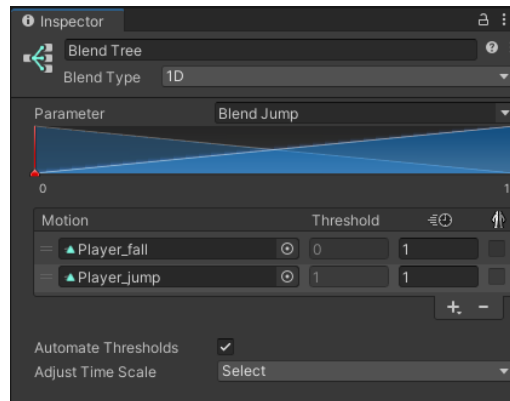
37. Buka *menu Inspector*, tekan icon + dan pilih *Add Motion Field*. Tambahkan dua *Motion Field*.



Gambar 9.34 Tambahkan Dua *Motion Field*

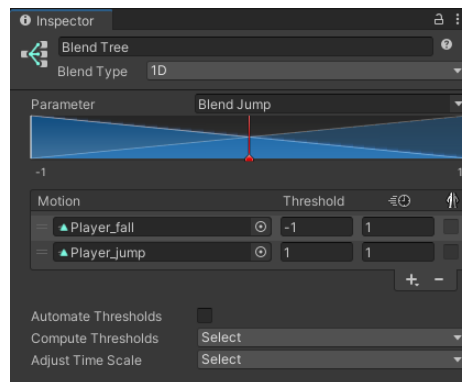


38. Klik bagian *icon None (Motion)*, maka akan muncul *Windows Motion*,
Tambahkan Sesuai dengan urutan.



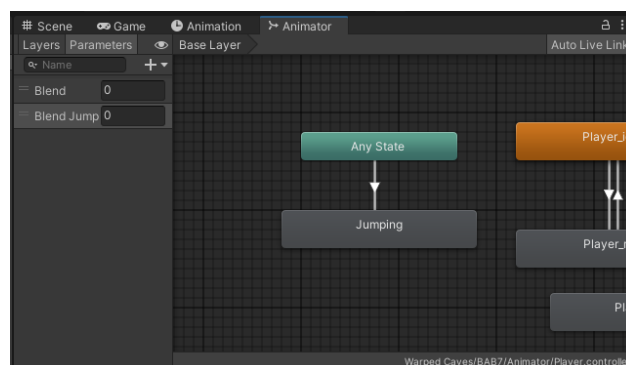
Gambar 9.35 Atur *Motion*

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



Gambar 9.36 Atur Nilai *Threshold*

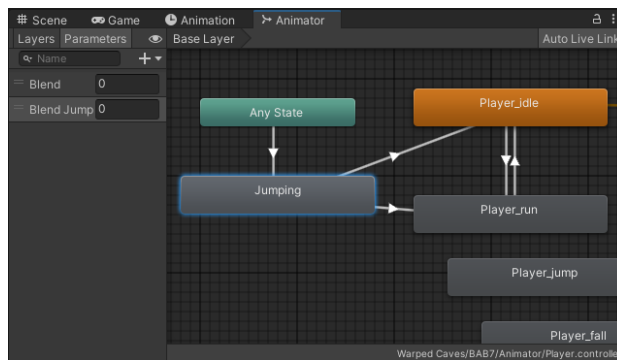
40. Kembali ke *Base Layer*, klik kanan *Any State*, pilih *Make Transition* dan arahkan panahnya ke *Jumping*.



Gambar 9.37 Buat Transisi pada *Any State*

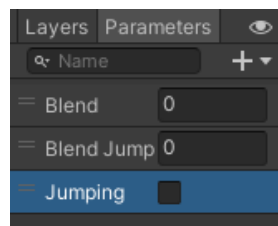


41. Klik kanan *Jumping*, pilih *Make Transition* dan arahkan panahnya ke *Player_idle* dan *Player_run*.



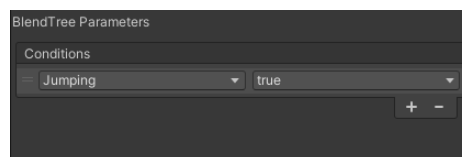
Gambar 9.38 Buat Transisi ke *Player_idle* dan *Player_run*

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



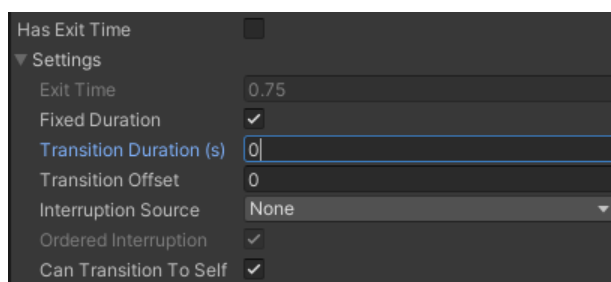
Gambar 9.39 Atur Parameter Transisi

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



Gambar 9.40 Atur Kondisi *Jumping*

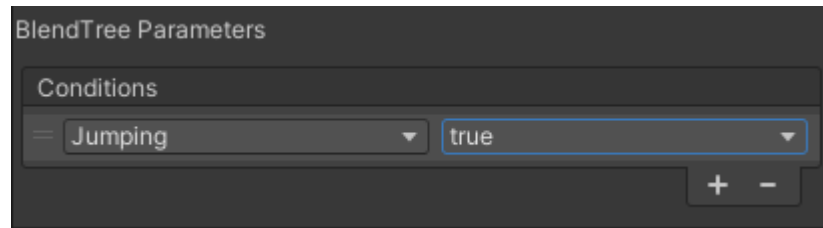
44. Klik *Settings* dan ubah nilai *Transition Duration* menjadi 0 dan hilangkan centang *Has Exit Time*.



Gambar 9.41 Atur di Bagian *Settings*

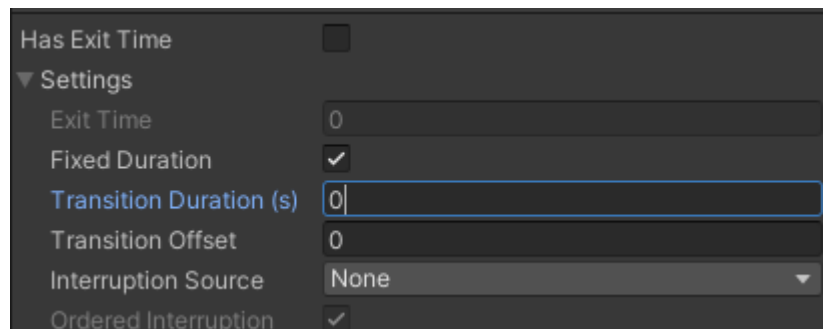


45. Klik panah yang mengarah ke `Player_idle` dan `Player_run`, pada *inspector* tambahkan *condition*, pilih *condition Jumping*, pada arah panah ke `player_idle` ubah menjadi `false`, pada arah panah ke `player_run` ubah menjadi `true`.



Gambar 9.42 Atur Kondisi Pada Setiap *Inspector*

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



Gambar 9.43 Ubah Nilai *Transition Duration*

47. 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;  
    }  
    else if (Input.GetButtonUp("Jump"))  
    {  
        jump = false;  
    }  
}
```

48. Pada Fungsi `FixedUpdate` tambahkan seperti berikut.

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

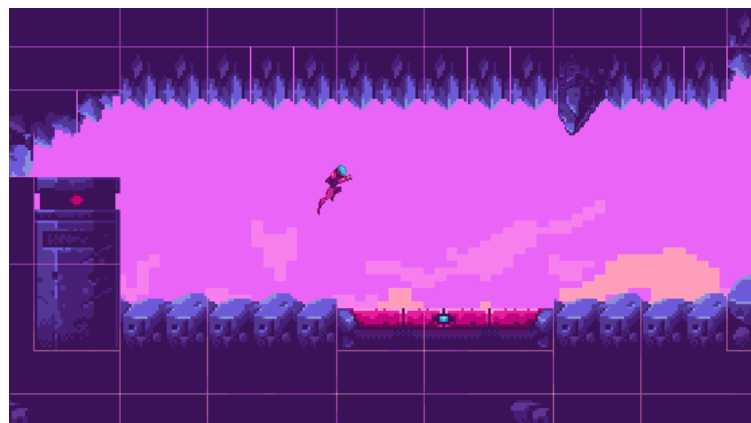


```
        animator.SetFloat("Blend",  
Mathf.Abs(rb.velocity.x));  
  
        animator.SetFloat("Blend Jump", rb.velocity.y);  
    }
```

49. Tambahkan baris kode seperti dibawah 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);  
}
```

50. Jika di play maka karakter sudah bisa bergerak dengan animasi.



Gambar 9.44 Tampilan Animasi Akhir

B. Kuis

1. Kuis CameraFollow

```
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);  
    }  
}
```



```
void HandleMovementInput()
{
    float move = Input.GetAxis("Horizontal");

    if (move != 0)
    {
        animator.SetBool("isIdle", false);
        animator.SetBool("isWalking", true);
        transform.Translate(new Vector3(move, 0, 0) *
Time.deltaTime);
    }
    else
    {
        animator.SetBool("isIdle", true);
        animator.SetBool("isWalking", false);
    }

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

Analisa :

Pada “HandleJumpInput()” ditambahkan nilai *true* ke “animator.SetBool(“isJumping”, *true*);” mengubah kondisi *elseif* menjadi “else if (Input.GetKeyUp(KeyCode.Space))” untuk mengatur nilai *false* ketika di lepas pada “isJumping”. Pada “HandleMovementInput()” mengubah kondisi “if (move != 1)” menjadi “if (move != 0)” untuk menangani gerakan di kedua arah dan menggunakan “new Vector3(move, 0, 0)” agar karakter bergerak sesuai arah input *horizontal* secara langsung. Lalu pada baris “transform.Translate()” menggunakan *Vector3.right* bukan “Vector3.left” dikarenakan untuk menggerakkan karakter ke arah *horizontal* sesuai dengan input pengguna. Selanjutnya penggunaan nilai skala -1 dan 1 digunakan untuk membalik skala objek secara *horizontal* yang membuat objek menghadap arah yang berlawanan saat bergerak kiri dan kanan.

A. Link Github

https://github.com/ShivaDina/2118089_PRAK_ANIGAME