



TUGAS PERTEMUAN: 9

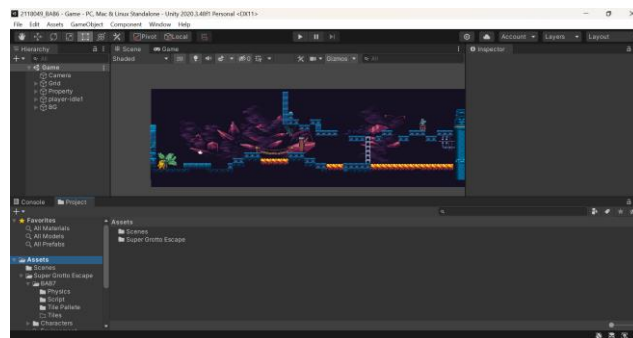
GAME ANIMATION

NIM	:	2118049
Nama	:	Siti Aisyah
Kelas	:	B
Asisten Lab	:	Difa Fisabilillah (2118052)

9.1 Tugas 1 : Character Animation

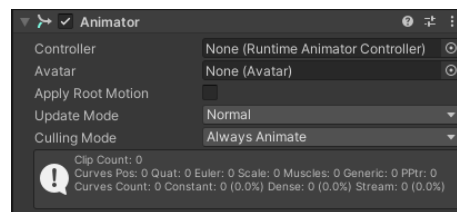
A. Membuat Character Animation

1. Buka proyek unity sebelumnya untuk melanjutkan tahap pembuatan animasi karakter.



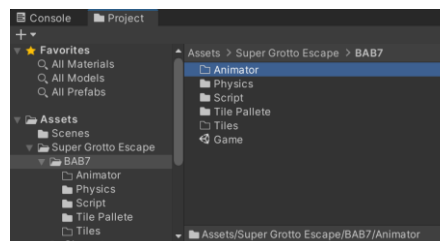
Gambar 9.1 Membuka Projek Unity

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



Gambar 9.2 Add Component Animator

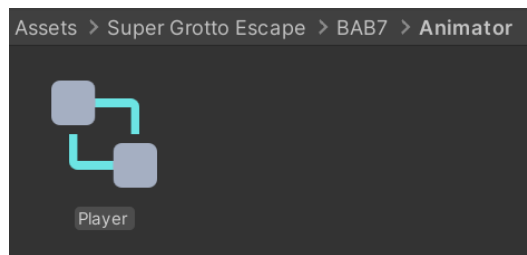
3. Pada folder BAB7 buat folder baru “Animator”.



Gambar 9.3 Membuat Folder Animator

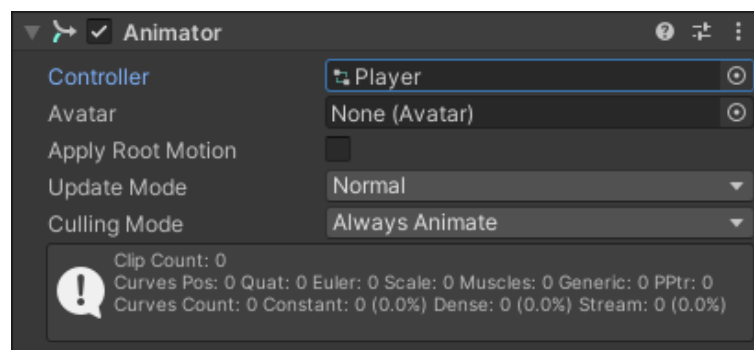


4. Buat *File Animator Controller* pada folder *Animator*, ubah namanya menjadi *Player*.



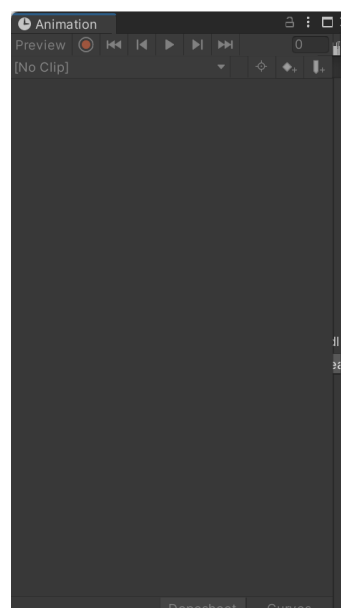
Gambar 9.4 Membuat *File Animator Controller*

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



Gambar 9.5 Mengubah Menjadi *Player*

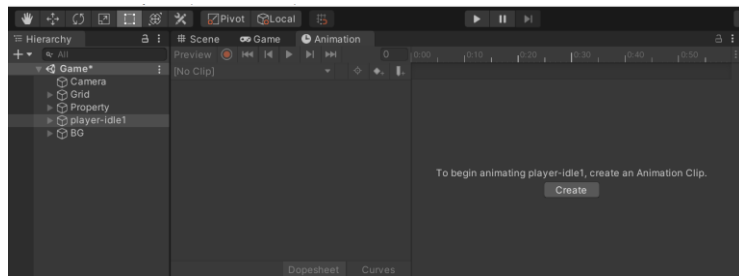
6. Tambahkan menu panel *Animation* di menu *Window*, pilih *Animation*, lalu pilih *Animation* atau Ctrl+6 dan akan muncul panel *Animation*.



Gambar 9.6 Menambahkan *Menu Panel Animation*

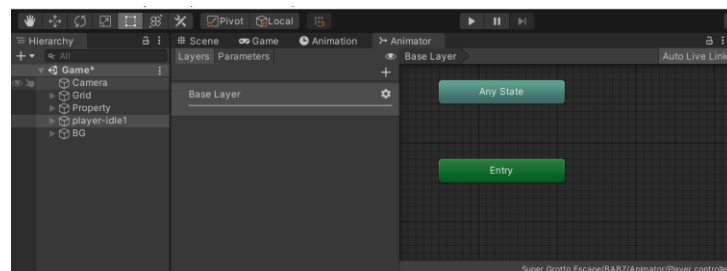


7. Selanjutnya, geser panel tersebut disamping panel game seperti gambar dibawah berikut (*Split Panel Project dan Animation*).



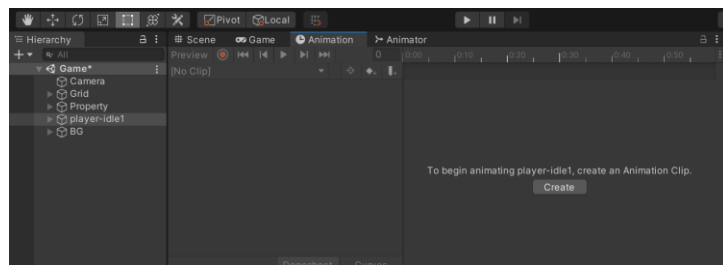
Gambar 9.7 *Split Projek*

8. Tambahkan *menu* panel Animator, lalu geser panel tersebut.



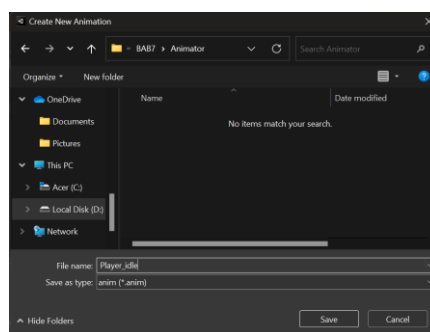
Gambar 9.8 Panel *Animator*

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



Gambar 9.9 Pilih *Create* Untuk Membuat Animasi

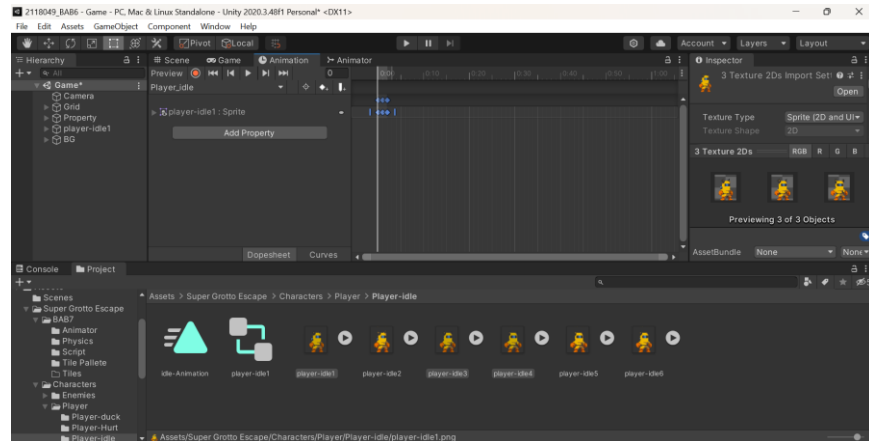
10. Simpan pada folder *Animator* dan beri nama “Player_idle”.



Gambar 9.10 Simpan Di Folder *Animator*

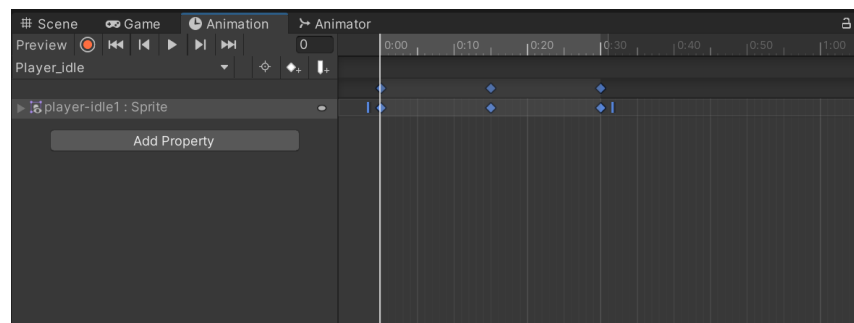


11. Pada menu *Project*, buka folder *player*, lalu pilih *Player-idle* dan pilih gambar *player-idle-1*, *player-idle-3* dan *player-idle-4*, kemudian *drag* ke tab *Animation*.



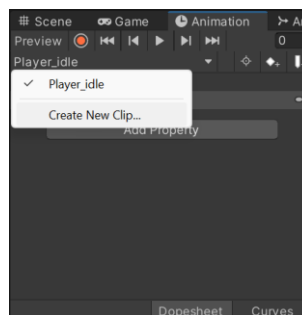
Gambar 9.11 Pilih *Player Idle*

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 Buat *Timeline* Sampai Frame 0:30

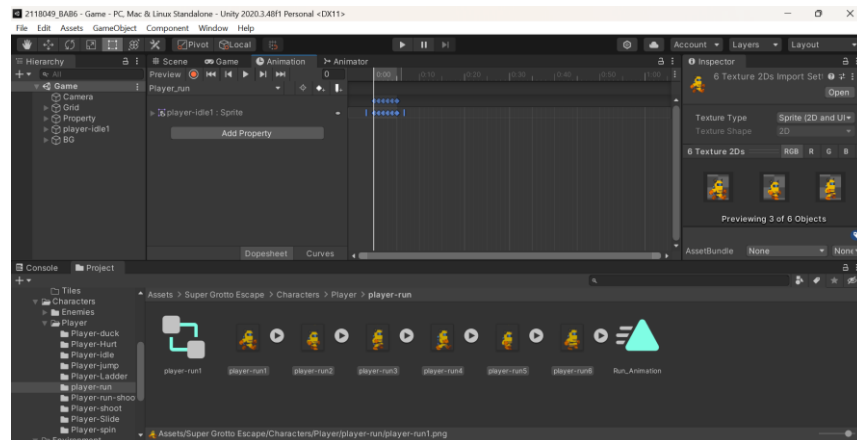
13. Buat animasi baru, klik pada “*Player_idle*” kemudian pilih *Create New Clip*, dan beri nama “*Player_run*”. Kemudian, simpan pada folder *Animator*.



Gambar 9.13 Buat Animasi Baru *Player_run*

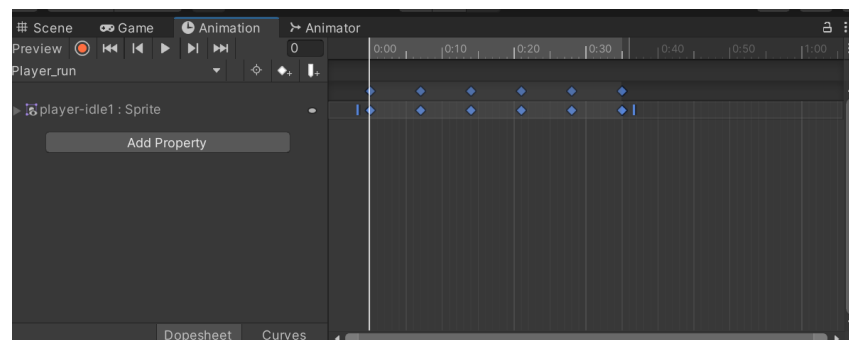


14. Buka menu *Project* kemudian cari folder *Player* lalu *Player_run*, Pilih *player-run-1* sampai *player-run-6*, *drag and drop* pada menu *Animation*.



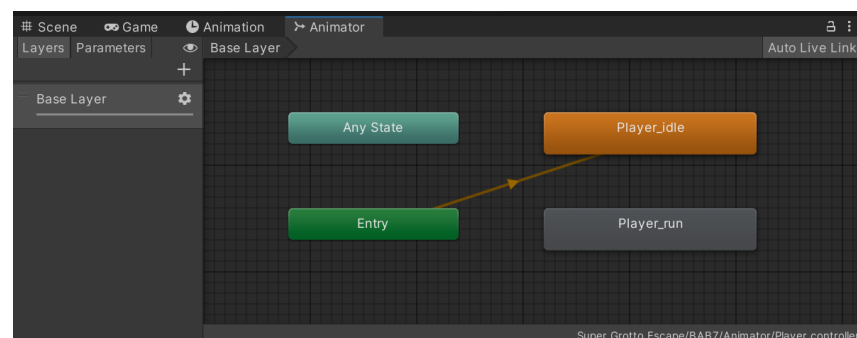
Gambar 9.14 Pilih *Player Run*

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 Buat *Timeline* 0:35

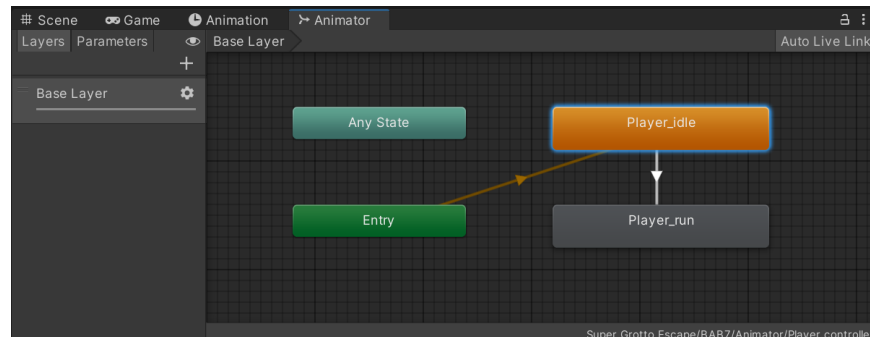
16. Pilih ke *menu Animator* yang telah dibuka sebelumnya dan akan tampil seperti berikut.



Gambar 9.16 *Menu Animator*

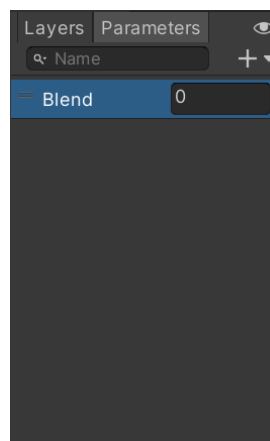


17. Kemudian buat transisi antara `Player_idle` dan `Player_run` dengan cara klik kanan pada `player_idle` dan pilih *Make Transition*. Kemudian, tarik ke `Player_run`.



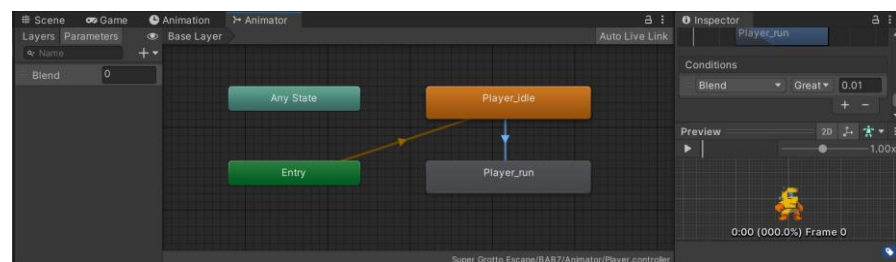
Gambar 9.17 Buat Transisi Antara `Player_Idle` dan `Player_Run`

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



Gambar 9.18 Buat Tipe Data Blend

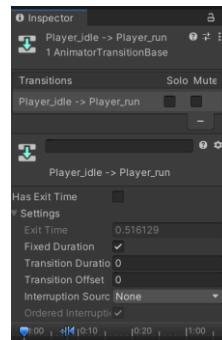
19. Klik panah putih (Palyer_idle ke `Player_run`). Lalu, pada bagian *conditions*, klik *icon* tambah dan atur menjadi “Blend”, serta nilai *conditions* blend menjadi 0.01.



Gambar 9.19 Pilih Kondisi Blend

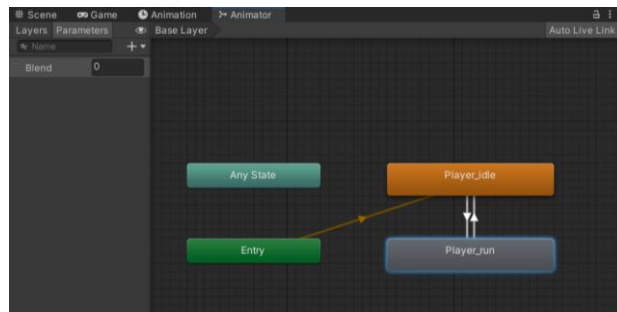


20. Pada bagian *Settings*, hilangkan centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



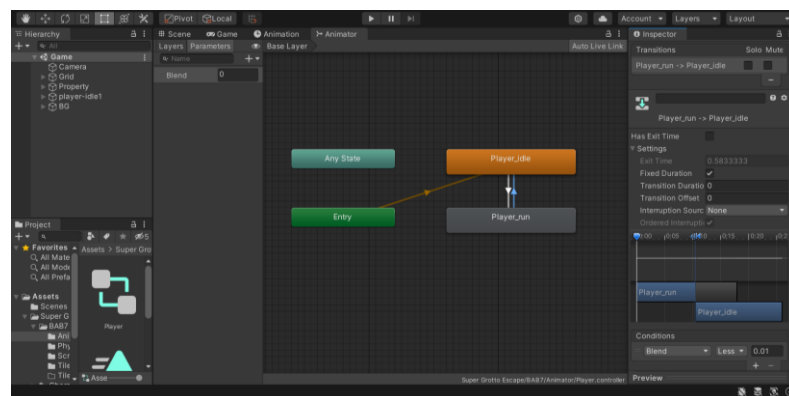
Gambar 9.20 Atur *Settings*

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. Klik panah putih (Player_run ke Palyer_idle). Setelah itu, ubah *conditions* menjadi Blend, lalu ubah operator dari *Greater* menjadi *Less* dan atur nilainya menjadi 0.01. Selanjutnya untuk bagian *setting*, hilangkan centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



Gambar 9.22 Atur *Setting*, Parameter dan Operator



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

```
public Animator animator;
```

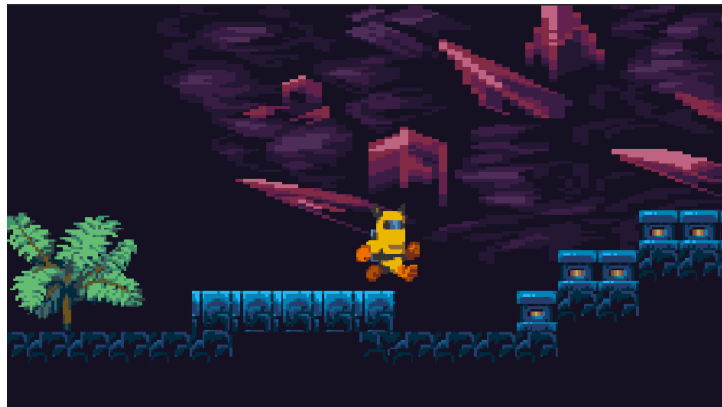
24. Tambahkan *Script* Komponen Animator.

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

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

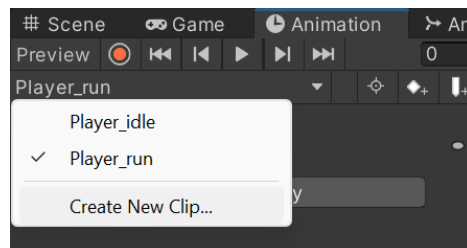
```
void FixedUpdate()  
{  
    GroundCheck();  
    Move(horizontalValue, jump);  
  
    animator.SetFloat("Blend",  
    Mathf.Abs(rb.velocity.x));  
}
```

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



Gambar 9.23 Tampilan Animasi Berjalan dan Berhenti

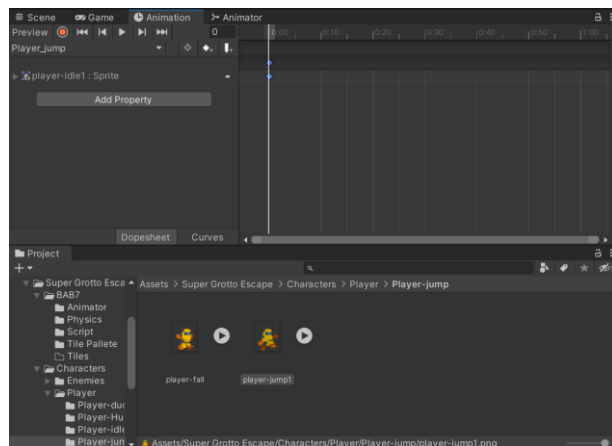
27. Kemudian buat animasi baru dengan cara klik tulisan “Player_run”, kemudian pilih *Create New Clip*, dan beri nama “Player_jump”.



Gambar 9.24 Buat Animasi Lompat

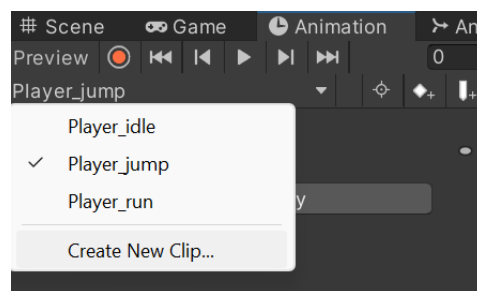


28. Pada folder player, buka folder Player_jump dan pilih gambar player-jump1, kemudian *drag* ke tab *Animation*.



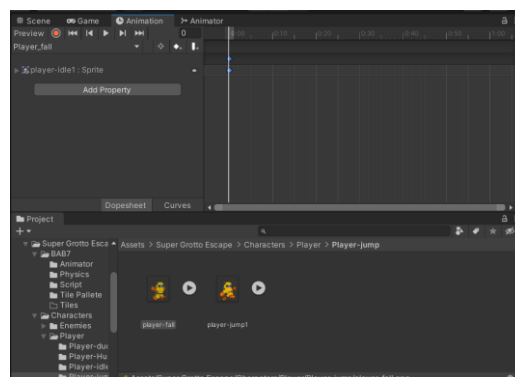
Gambar 9.25 Pilih Gambar *Jump*

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



Gambar 9.26 Buat Animasi Jatuh

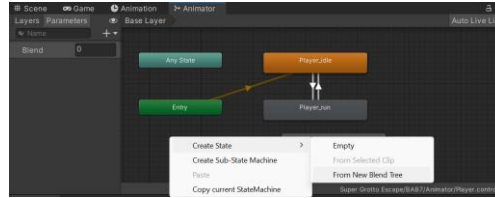
30. Pada folder player, buka folder Player_jump dan pilih gambar player-fall, kemudian *drag* ke tab *Animation*.



Gambar 9.27 Pilih Gambar *Fall*

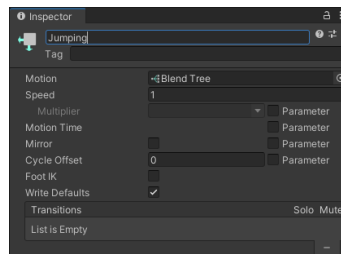


31. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu *Animator*. Pada area kosong, pilih *Create State*, lalu *From New Blend Tree*.



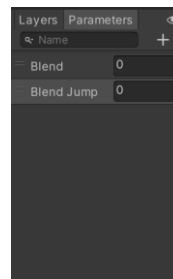
Gambar 9.28 Menambahkan Animasi Melompat

32. Pada *Animator*, klik *Blend Tree*. Kemudian, pada menu *Inspector*, ubah namanya menjadi *Jumping*.



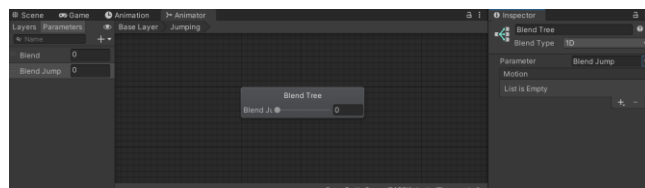
Gambar 9.29 Ubah Nama Menjadi *Jumping*

33. Pada menu *Parameters* tambahkan parameter tipe data *float*, lalu klik *icon* tambah dan ubah namanya menjadi “Blend Jump”.



Gambar 9.30 Buat Tipe Data *Blend Jump*

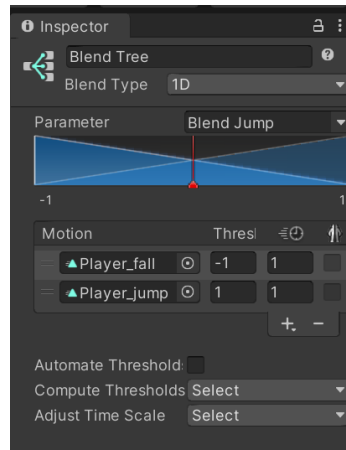
34. Pada menu *Animator*, double click pada *Blend Tree* “*Jumping*”, kemudian klik pada *Blend Tree* dan pada *inspector* ubah parameter menjadi “*Blend Jump*”.



Gambar 9.31 Klik Dua Kali *Blend Tree* dan Mengubah Parameter

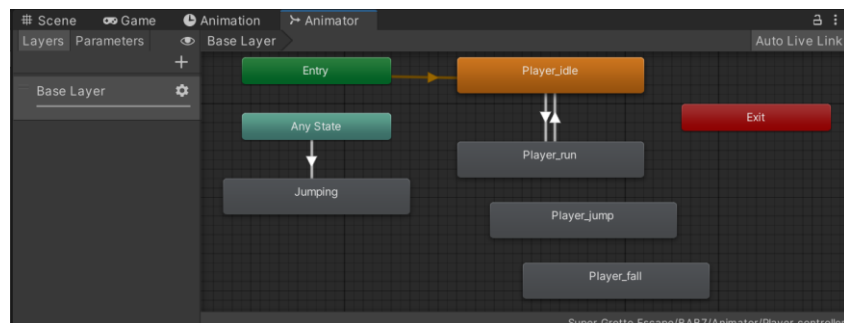


35. Buka menu *Inspector*, tekan *icon* tambah dan pilih *Add Motion Field*.
 Tambahkan dua *Motion Field* dan ubah *motion* menjadi berikut sesuai urutan. Selanjutnya, atur juga nilai *Threshold* dengan menghilangkan centang “*Automate Thresholds*” dan atur nilai seperti berikut.



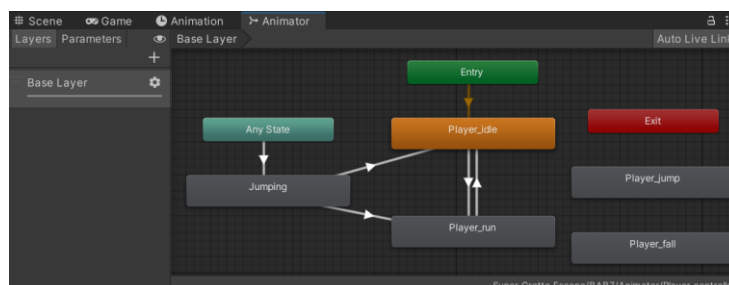
Gambar 9.32 Menambahkan Motion Field *Motion Field*

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



Gambar 9.33 Buat Transisi pada *Any State*

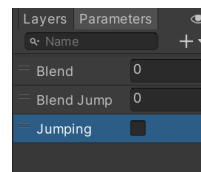
37. Klik kanan *Jumping*, pilih *Make Transition* dan arahkan panahnya ke *Player_idle* dan *Player_run*.



Gambar 9.34 Buat Transisi ke *Player_idle* dan *Player_run*

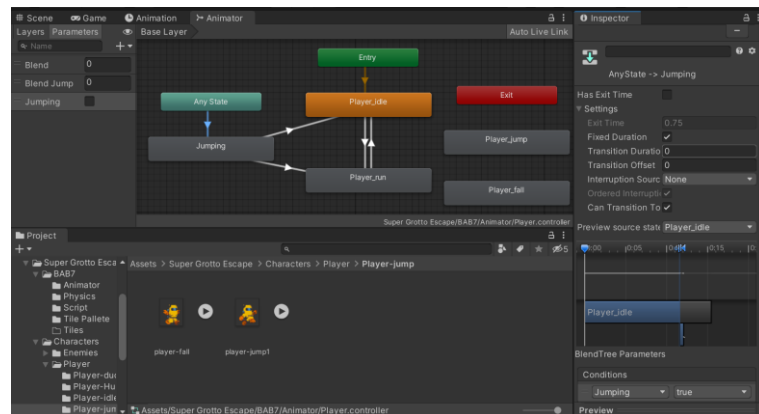


38. Tambahkan parameter transisi dengan tipe data *Bool* tekan *icon* tambah dan ubah namanya menjadi “*Jumping*”.



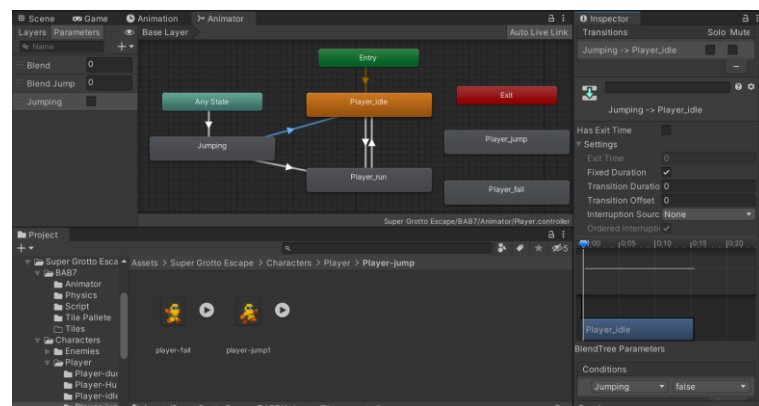
Gambar 9.35 Atur Parameter Transisi

39. Klik panah yang mengarah ke *Jumping*, pada *inspector* tambahkan *condition*, pilih *condition Jumping* dan ubah nilainya menjadi *true*. Kemudian, hilangkan centang *Has Exit Time* dan pada *setting* ubah nilai *Transition Duration* menjadi 0.



Gambar 9.36 Atur Kondisi *Jumping* dan Bagian *Settings*

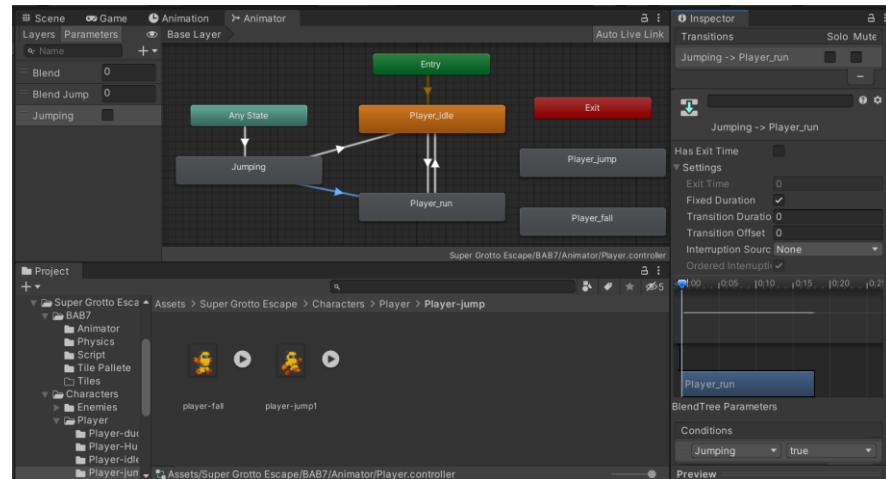
40. Klik panah yang mengarah ke *Player_idle*, pada *inspector* tambahkan *condition*, pilih *condition Jumping* dan ubah nilainya menjadi *false*. Kemudian, hilangkan centang *Has Exit Time* dan pada *setting* ubah nilai *Transition Duration* menjadi 0.



Gambar 9.37 Atur Kondisi Pada *Player_idle*



41. Klik panah yang mengarah ke `Player_run`, pada *inspector* tambahkan *condition*, pilih *condition* `Jumping` dan ubah nilainya menjadi `true`. Kemudian, hilangkan centang *Has Exit Time* dan pada *setting* ubah nilai *Transition Duration* menjadi 0.



Gambar 9.38 Atur Kondisi Pada `Player_run`

42. 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;  
    }  
}
```

43. 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);  
}
```



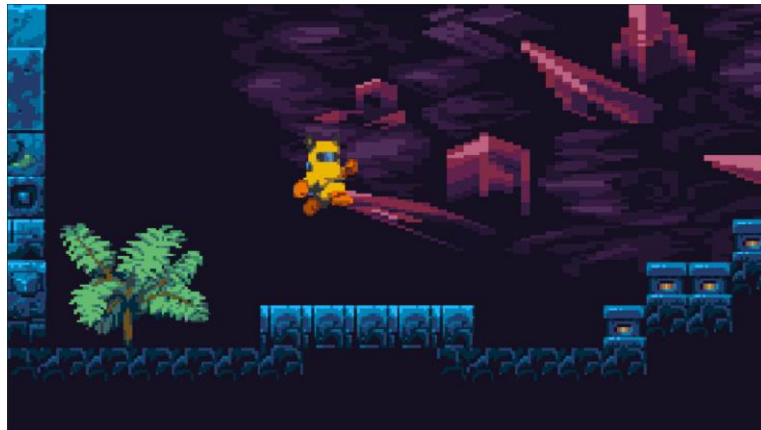
44. Tambahkan baris kode seperti dibawah ini dalam *method* GroundCheck.

```
void GroundCheck()
{
    isGrounded = false;
    Collider2D[] colliders =
    Physics2D.OverlapCircleAll(groundcheckCollider.position
    , groundCheckRadius, groundLayer);

    if (colliders.Length > 0)
    {
        isGrounded = true;
    }

    animator.SetBool("Jumping", !isGrounded);
}
```

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



Gambar 9.39 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 :

Di dalam fungsi “HandleJumpInput()” terdapat beberapa kesalahan dan kekurangan yang perlu diperbaiki. Pada `animator.SetBool("isJumping", true);` ditambahkan nilai *true* untuk mengaktifkan animasi lompat. Kemudian pada *else if* ubah `GetKey` menjadi `GetKeyUp` atau seperti berikut, “*else if* (`Input.GetKeyUp(KeyCode.Space)`)” untuk mengecek apakah tombol “space” baru saja dilepaskan. Selanjutnya, di dalam *else if* tersebut tambahkan nilai *false* (`animator.SetBool("isJumping", false);`) untuk mengatur nilai *false* atau menghentikan animasi lompat. Selanjutnya, di dalam fungsi “HandleMovementInput()” terdapat beberapa kesalahan dan kekurangan yang perlu diperbaiki juga. Pada kondisi “*if* (`move != 1`)” ubah menjadi “*if* (`move != 0`)” untuk mengecek apakah ada gerakan atau tidak. Kemudian, di dalam *if* tersebut terdapat `animator.SetBool("isIdle", true);` yang kondisinya diubah menjadi `animator.SetBool("isIdle", false);` untuk mengatur parameter atau kondisi menjadi *false*. Selanjutnya, menambahkan *source code* `animator.SetBool("isWalking", true);` untuk mengatur parameter “isWalking” menjadi *true* agar dapat mengaktifkan animasi berjalan. Lalu, pada `transform.Translate(new`



`Vector3(move, 0, 0) * Time.deltaTime);` untuk menggerakkan objek dengan kecepatan yang disesuaikan dengan waktu *frame* ke arah horizontal sesuai input. Kemudian, pada *else* menambahkan `animator.SetBool("isIdle", true);` untuk mengatur parameter “isIdle” menjadi *true*. Lalu `if (move != 0)` diubah menjadi `if (move < 0)` yang jika bergerak ke kiri, maka membalikkan skala objek pada sumbu x untuk menghadap kiri. Terakhir pada penggunaan nilai skala -1 dan 1 digunakan agar objek menggunakan ukuran aslinya.

C. Link Github

https://github.com/saasyh/2118049_PRAK_ANIGAME