



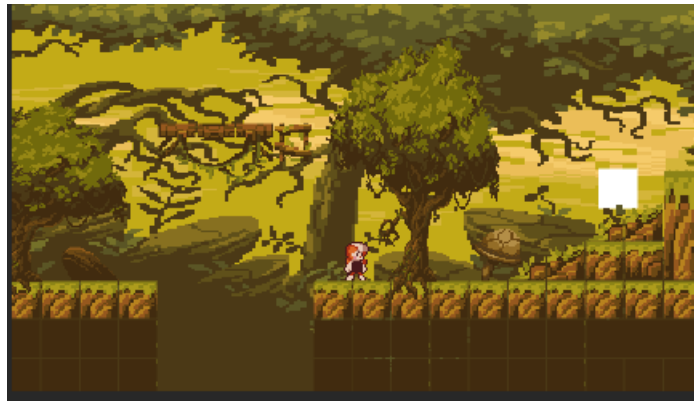
TUGAS PERTEMUAN: 9

GAME ANIMATION

NIM	:	2118021
Nama	:	Irfani Muhamad AL Rizqi
Kelas	:	A
Asisten Lab	:	NATASYA OCTAVIA(2118034)

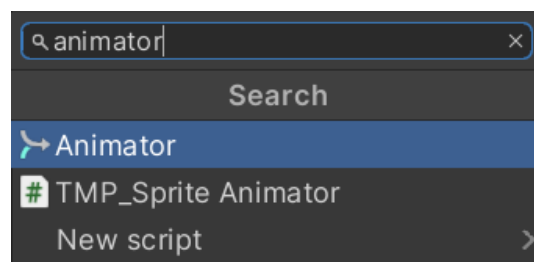
1 Tugas 1 : Membuat Character Animation

1. Buka kembali proyek sebelumnya yang sudah dapat digerakkan dengan *keyboards*.



Gambar 9.1 Membuka Proyek Unity

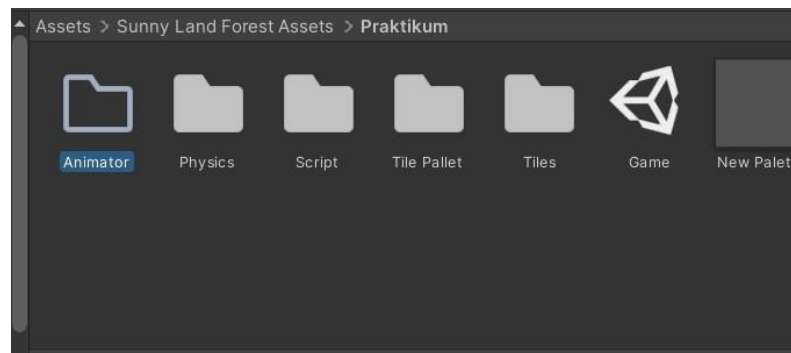
2. Kemudian klik karakternya lalu pada *inspector* tambahkan *component Animator*.



Gambar 9.2 Menambahkan Component Animator



3. Tambahkan *folder* baru pada *folder* “Praktikum” berikan nama *folder* dengan nama “*Animator*”.



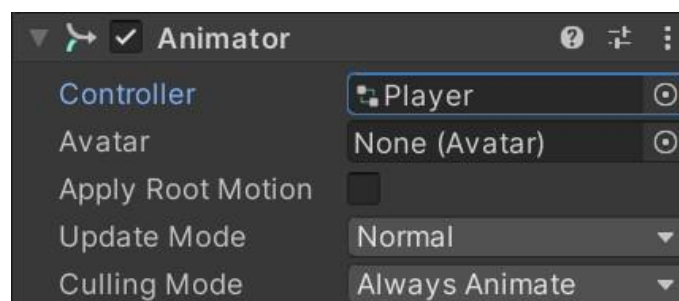
Gambar 9.3 Menambahkan Folder Baru

4. Tambahkan *file* “*Animator Controller*” pada *folder* “*Animator*” dengan cara klik kanan pada *folder* lalu *Create > Animator Controller*, ubah nama menjadi “*Player*”.



Gambar 9.4 Menambahkan File *Animator Controller*

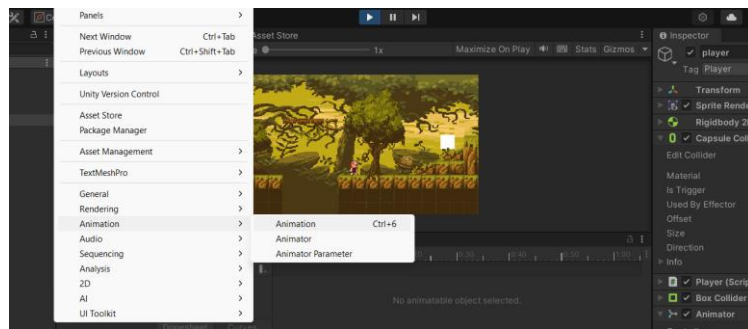
5. Lalu klik pada karakter bagian *Component Animator*, pada *Controller* ubah menjadi *Player*.



Gambar 9.5 Setting *Animator Controller*

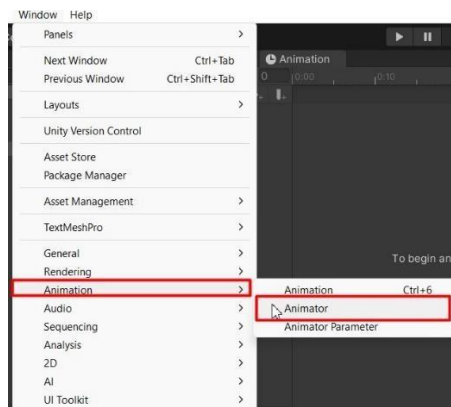


6. Tambahkan *menu panel Animation* dengan cara klik *menu Window > Animation > Animation*.



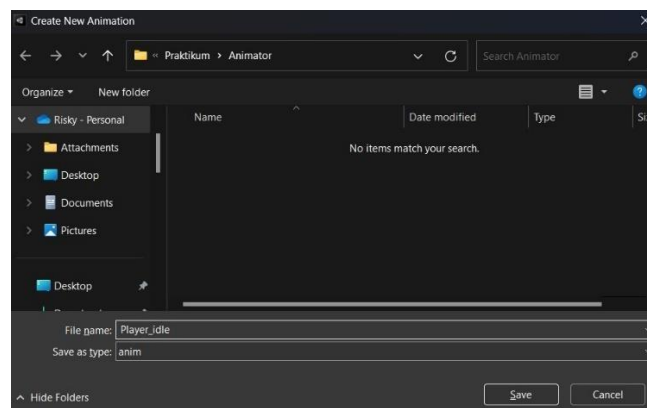
Gambar 9.6 Menambahkan *Windows Animation*

7. Tambahkan lagi *Window* baru dengan cara klik *menu Window > Animation > Animator*.



Gambar 9.7 Menambahkan *Windows Animator*

8. Klik *Create* pada *Animation* lalu simpan pada folder *animator* dan beri nama “*Player_idle*”.



Gambar 9.8 Menyimpan File

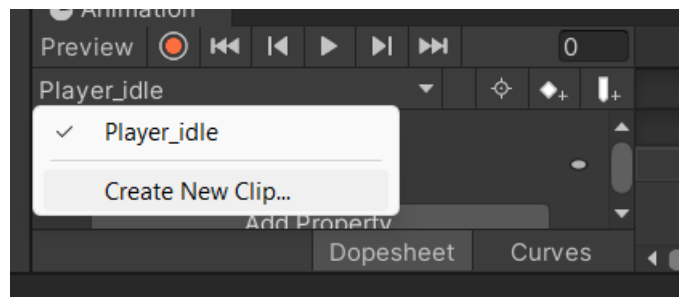


9. Lalu tekan Ctrl + A pada *tab Animation* kemudian geser kotak ke *frame* 40.



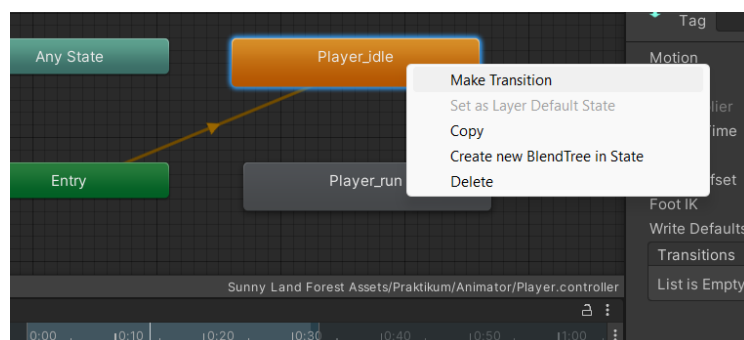
Gambar 9.9 Menambahkan Frame

10. Tambahkan animasi bari dengan klik pada “Player_idle” lalu pilih *Create New Clip*, beri nama “Player_run” dan simpan pada *folder Animator*.



Gambar 9.10 Menambahkan Animasi

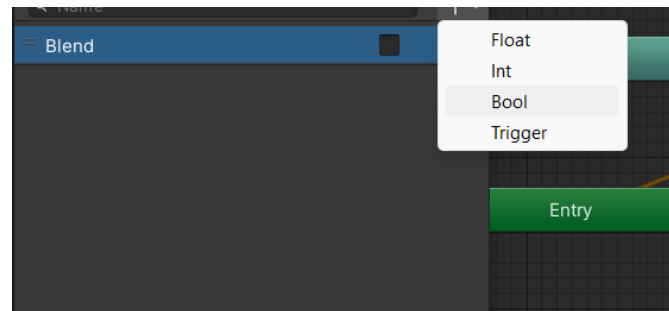
11. Lalu pada *menu Animator*, buat transisi antara Player_idle dan Player_run dengan klik kanan pada Player_idle pilih *Make Transition* lalu tarik ke Player_run.



Gambar 9.11 Membuat Transisi

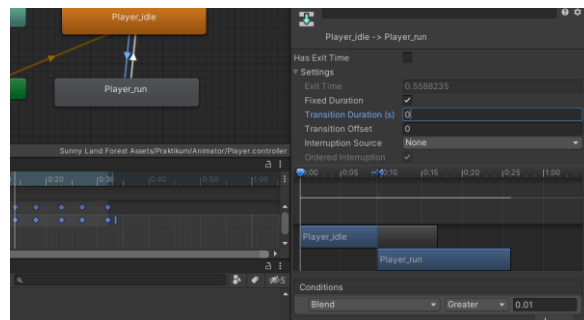


12. Lalu pada *tab parameter* tambahkan tipe data *Float* dengan menekan *icon* tambah dan rubah namanya menjadi “Blend”.



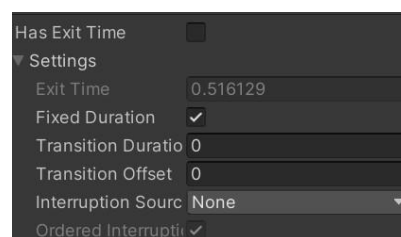
Gambar 9.12 Menambahkan *Parameter*

13. Klik panah putih antara *Player_idle* dan *Player_run*, Kemudian pada bagian *conditions* klik *icon* tambah atur menjadi “Blend” dan beri nilai *conditions* menjadi 0.01.



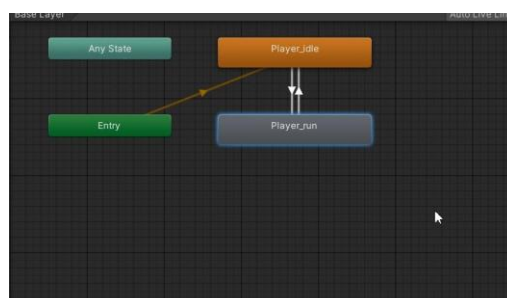
Gambar 9.13 Menambahkan *Conditions*

14. Pada bagian *settings* hilangkan centang pada *Has Exit Time* dan beri nilai *Transition Duration* menjadi 0.



Gambar 9.14 Mengatur *Settings*

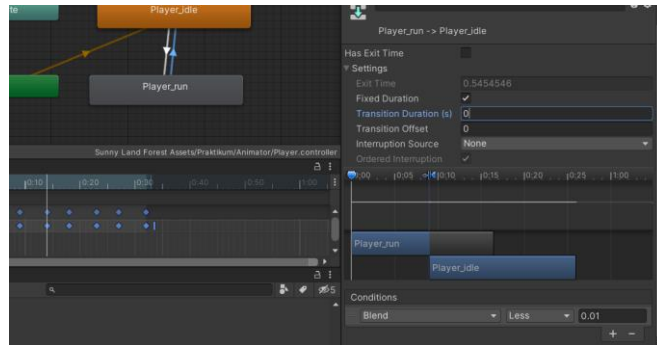
15. Lalu buat transisi kembali dari *Player_run* ke *Player_idle* dengan cara yang sama.



Gambar 9.17 Membuat *Transisi*

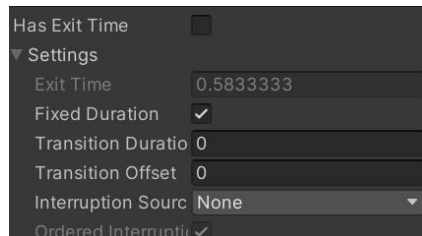


16. Tambahkan lagi *parameter* dengan tipe *data float*, ubah nama menjadi “Blend” lalu rubah operator menjadi *Less* dan berikan nilai 0.01.



Gambar 9.16 Menambahkan *Parameter* Baru

17. Pada bagian *settings* hilangkan juga centang pada *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0.



Gambar 9.17 Mengatur *Setting*

18. Lalu rubah *code* pada *class Player* dengan *code* dibawah ini.

```
Public Animator animator;
```

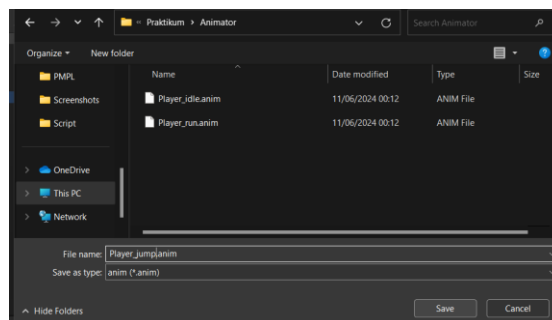
19. Tambahkan juga *code* dibawah ini pada *void Awake()*.

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

20. Tambahkan *code* dibawah ini pada *void FixedUpdate()*.

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

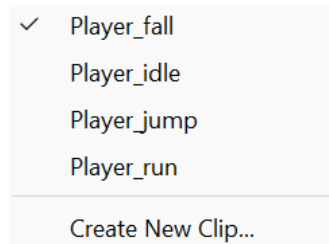
21. Kemudian tambahkan animasi baru dengan nama “Player_jump” dan letakkan pada folder *animator*.



Gambar 9.18 Menambahkan Animasi

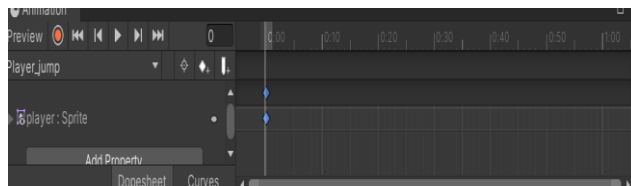


22. Kemudian buat animasi baru lagi dengan nama “Player_fall” dan simpan pada *folder Animator*.



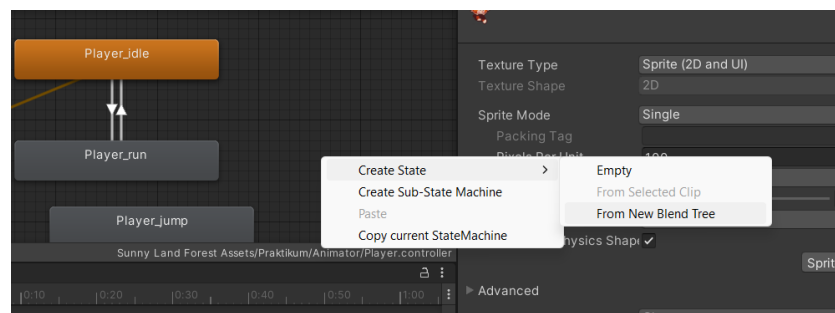
Gambar 9.19 Menambahkan Animasi Fall

23. Lalu tambahkan *asset player fall* pada *tab Animation*.



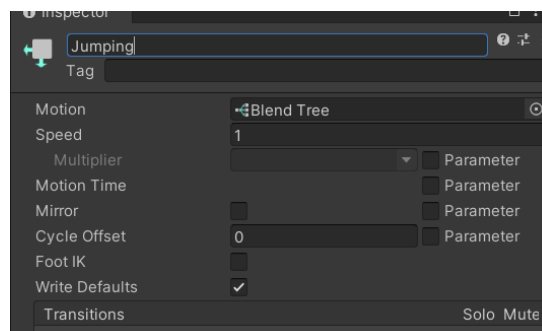
Gambar 9.20 Menambahkan Asset Fall

24. Lalu klik kanan pada *menu Animator* lalu pilih *Create State > From New Blend Tree*.



Gambar 9.21 Membuat Animasi Jump

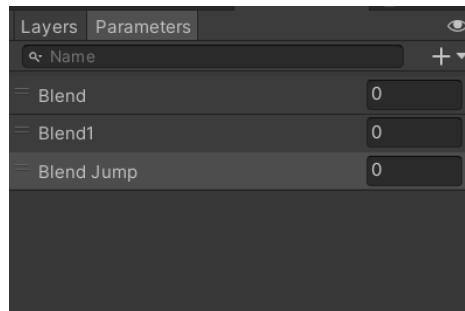
25. Lalu klik *Blend Tree*, rubah namanya pada *inspector* menjadi *Jumping*.



Gambar 9.22 Merubah Nama Blend Tree

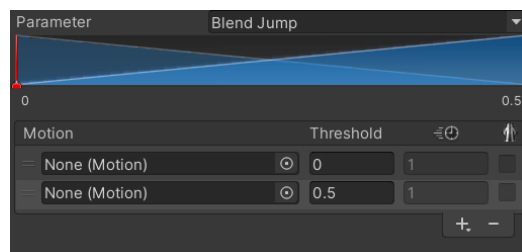


26. Tambahkan *parameter* baru dengan tipe data *Float* lalu rubah namanya menjadi “*Blend Jump*”.



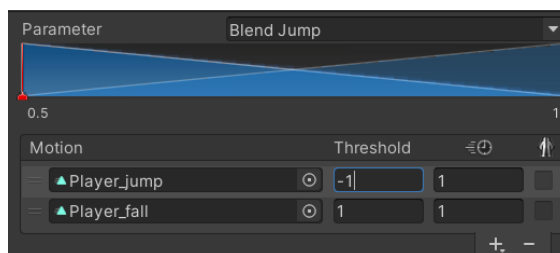
Gambar 9.23 Meletakkan *Script CameraFollow*

27. Tambahkan dua *Motion Field* dengan menekan *icon* tambah dan pilih *Add Motion Field*.



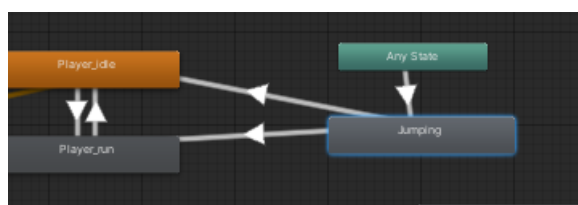
Gambar 9.24 Menambahkan *Motion Field*

28. Pada bagian *None* tambahkan “*Player_fall*” dan “*Player_jump*”, Kemudian hilangkan centang pada “*Automate Thresholds*” dan berikan nilai seperti dibawah.



Gambar 9.25 Menambahkan Nilai *Threshold*

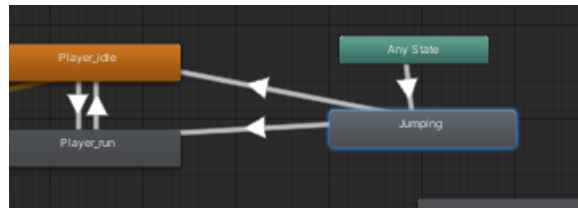
29. Pada *Base Layer* klik kanan pada *Any State* pilih *Make Transition* dan arahkan pada *Jumping*.



Gambar 9.26 Menghubungkan Dengan *Jumping*

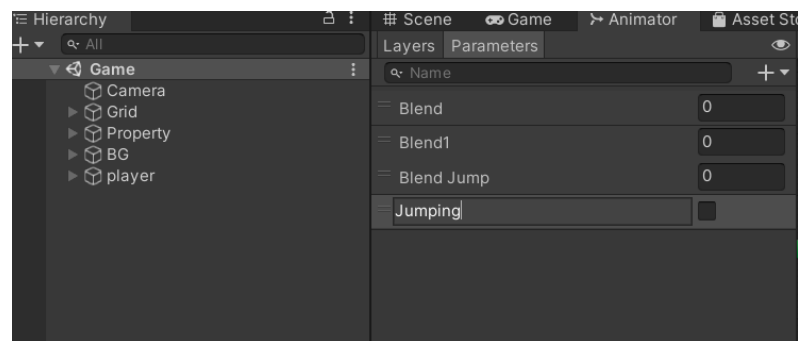


30. Lalu hubungkan *Jumping* dengan *Player_idle* dan *Player_run*.



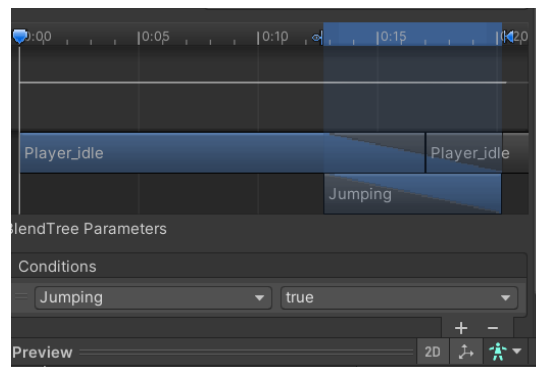
Gambar 9.27 Menghubungkan *Jumping*

31. Tambahkan *parameter* dengan tipe data *Bool* dan rubah namanya menjadi *Jumping*.



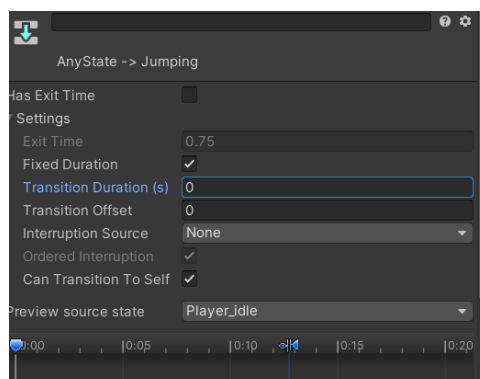
Gambar 9.28 Menambahkan Parameter *Jumping*

32. Klik panah yang mengarah ke *Jumping*, pada *inspector* tambahkan *condition* *Jumping* dan rubah nilainya menjadi *True*.



Gambar 9.29 Menambahkan *Conditions* *Jumping*

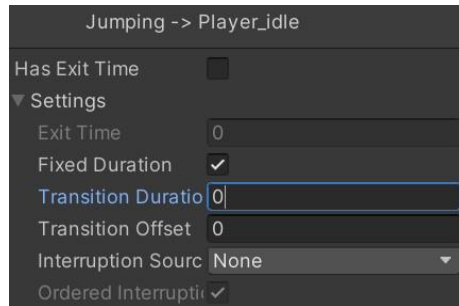
33. Lalu pada *Settings* hapus centang pada *Has Exit Time* dan rubah nilai *Transition Duration* menjadi 0.



Gambar 9.30 Mengatur *Settings*



34. Lalu pada *Setting* Player_idle dan Player_run berikan nilai 0 untuk *Transition Duration* dan hapus centang pada *Has Exit Time*.



Gambar 9.31 Mengatur Setting

35. Lalu pada *script Player* tambahkan *code* berikut pada fungsi *Update*.

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

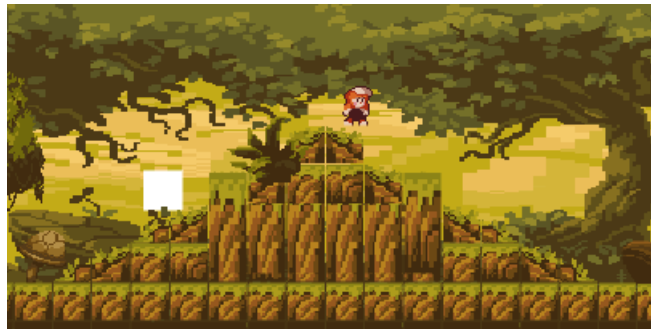
36. Tambahkan *code* berikut pada pada fungsi *FixedUpdate*

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

37. Kemudian tambahkan *code* berikut pada *method* *GroundCheck()*.

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

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



Gambar 9.32 Hasil runing



2 Kuis

A. Kesalahan

```
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);
    }
}
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);
    }
}
```

B. 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);
    }
}
```



Analisa

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

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 memberiperintah 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 perintah jika karakter digerakkan).