



## TUGAS PERTEMUAN: 9

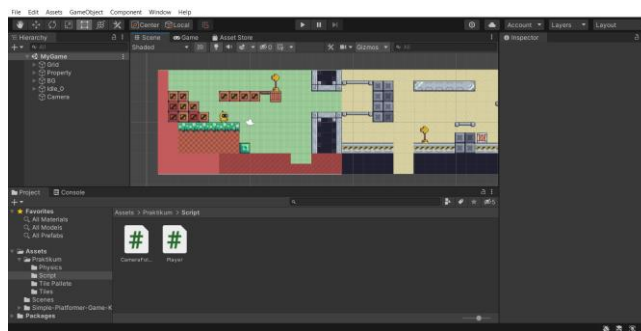
### CAMERA & CHARACTER MOVEMENT

NIM	:	2118028
Nama	:	Novianto Aldo Wibisono
Kelas	:	A
Asisten Lab	:	Aprillia Dwi Dyah S. (2118143)

#### 9.1 Tugas 1 : Membuat Camera & Character Movement

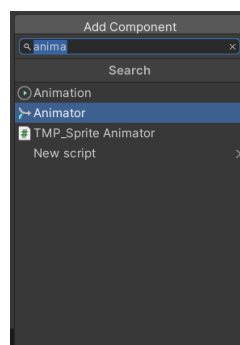
##### A. Mempersiapkan Project

1. Buka *project* Unity sebelumnya yang telah ditambahkan *camera* dan *character movement*.



Gambar 9.1 Membuka File Project

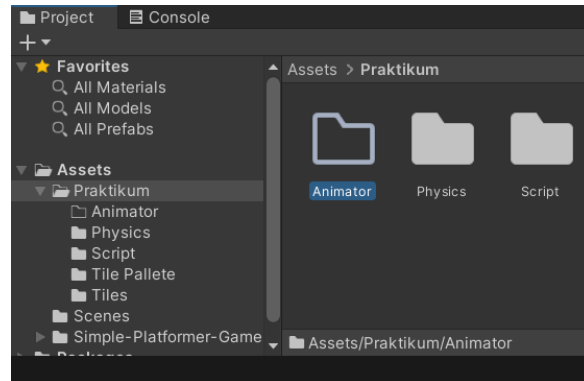
2. Pada tab *Inspector* hirarki karakter, tambahkan komponen *Animator*.



Gambar 9.2 Menambah Komponen Animator

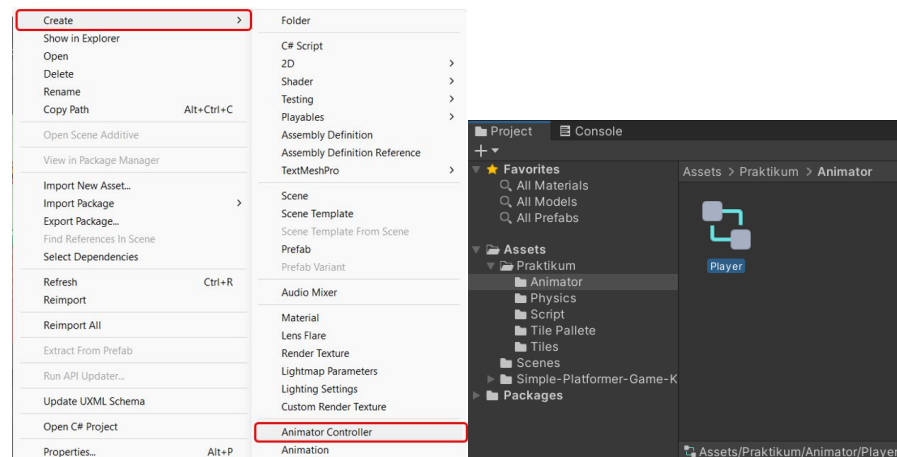


3. Kemudian pada *folder* Praktikum buat *folder* dengan nama Animator.



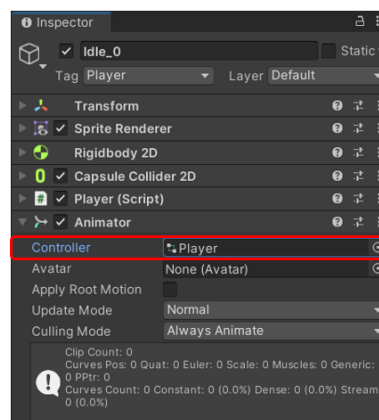
Gambar 9.3 Membuat Folder

4. Lalu buat file *Animator Controller* pada *folder* Animator, ubah namanya menjadi *Player*.



Gambar 9.4 Membuat File Animator Controller

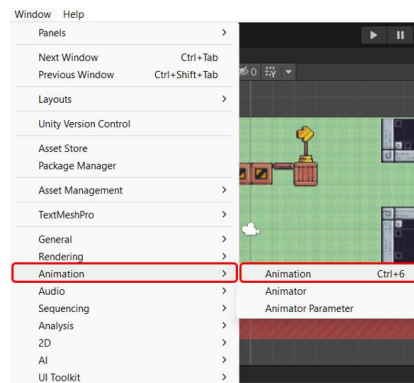
5. Selanjutnya ubah pengaturan *Controller* yang berada di *Component Animator* pada hirarki *player*, ubah menjadi *Player*.



Gambar 9.5 Mengubah Controller

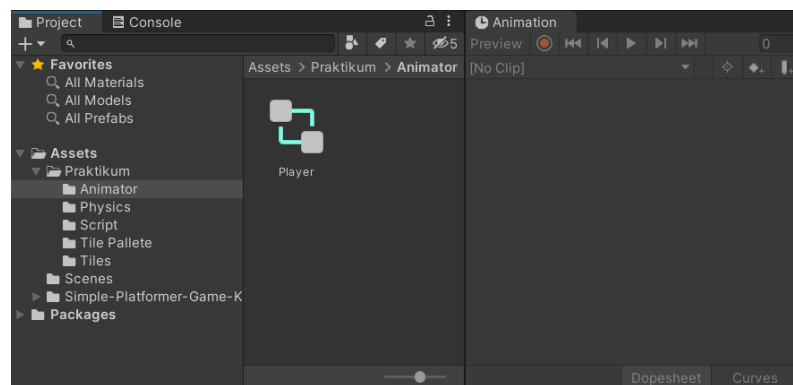


6. Tambahkan panel *Animation* melalui menu *Window*, pilih *Animation* lalu *Animation* atau tekan Ctrl+6.



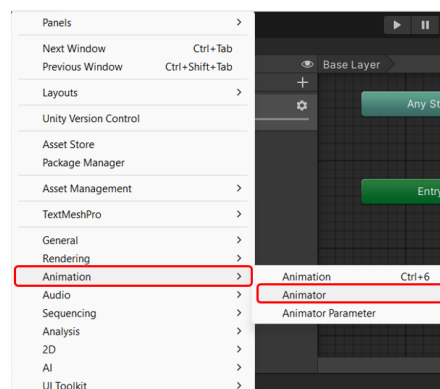
Gambar 9.6 Menambahkan Panel Animation

7. Kemudian geser panel yang baru muncul ke bagian bawah, *split panel Project* dengan *Animation*.



Gambar 9.7 Meletakkan Panel Animation

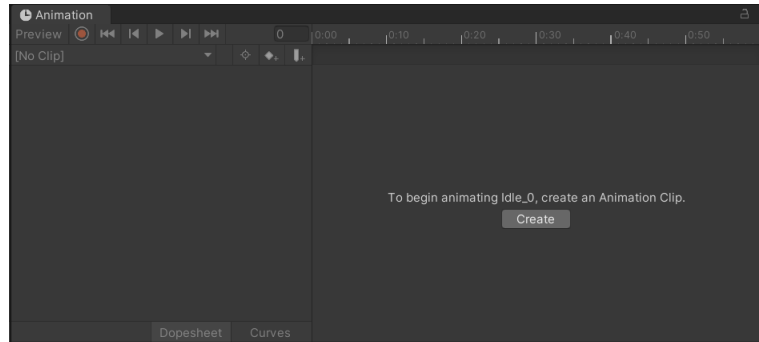
8. Tambahkan panel *Animator*.



Gambar 9.8 Menambah Panel Animator



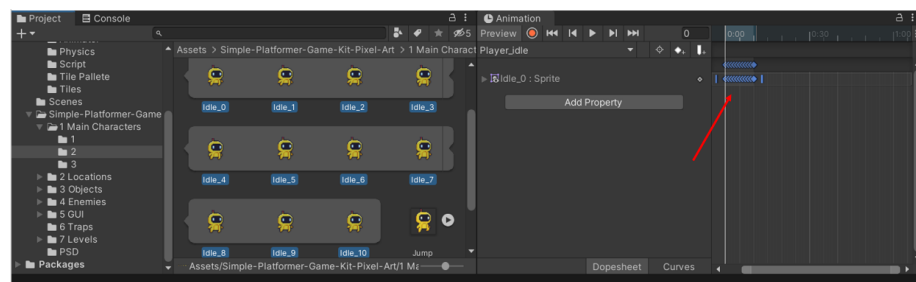
9. Klik `Idle_0` pada hirarki, kemudian pada menu panel *Animation* pilih *Create* dan simpan ke dalam *folder Animator* dengan nama '`Player_idle`' untuk membuat animasi.



Gambar 9.9 Membuat Animasi

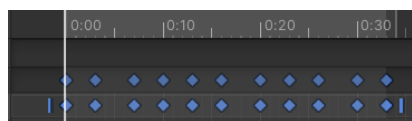
## B. Menambahkan Animasi Idle Dan Run

1. Pada *folder Project* buka *folder Main Character* lalu pilih *folder 2* dan pilih gambar '`Idle_0`' hingga '`Idle_10`', kemudian *drag* ke *tab Animation*.



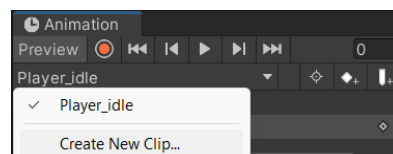
Gambar 9.10 Menambahkan Animasi Idle

2. Tekan `Ctrl+A` pada menu panel *Animation*, lalu geser kotak kecil pada *timeline* hingga *frame 0:33*.



Gambar 9.11 Mengubah Timeline

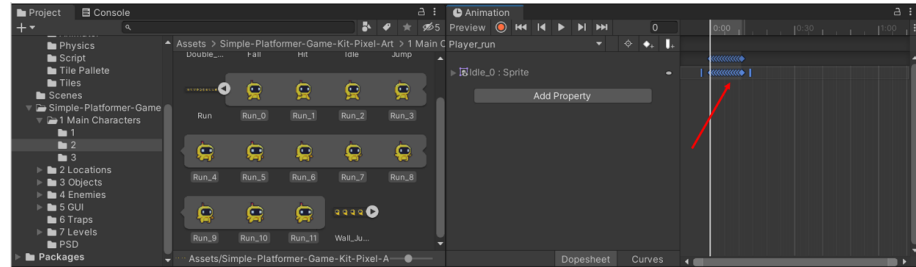
3. Buat animasi baru dengan klik pada `Player_idle`, kemudian pilih *Create New Clip* dan beri nama `Player_run` serta simpan pada *folder Animator*.



Gambar 9.12 Membuat Animasi Baru

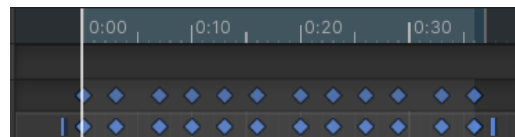


4. Pada *folder Project* buka *folder Main Character* lalu pilih *folder 2* dan pilih gambar 'Run\_0' hingga 'Run\_10', kemudian *drag* ke *tab Animation*.



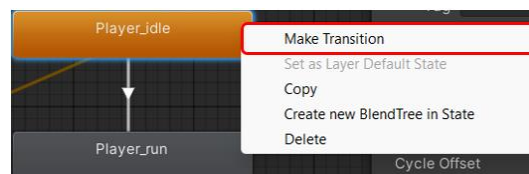
Gambar 9.13 Menambahkan Animasi Run

5. Tekan **Ctrl+A** pada menu panel *Animation*, lalu geser kotak kecil pada *timeline* hingga *frame 0:36*.



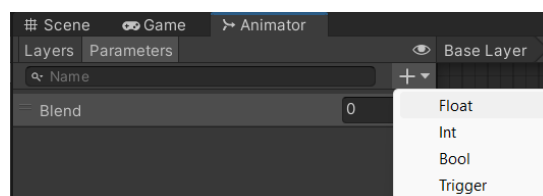
Gambar 9.14 Mengubah Timeline

6. Buat transisi antara *Player\_idle* dan *Player\_run* pada *tab Animator* dengan mengklik kanan *Player\_idle* dan pilih *Make Transition*, lalu tarik ke *Player\_run*.



Gambar 9.15 Membuat Transisi

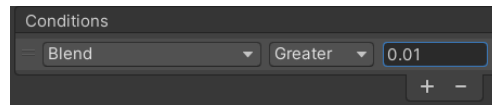
7. Masuk ke *tab parameter*, lalu tambahkan tipe data *float* dengan cara menekan *icon* tambah dan ubah nama menjadi *Blend*.



Gambar 9.16 Membuat Tipe Data Baru

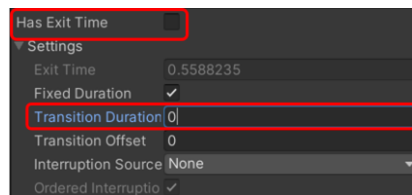


8. Klik panah putih tersebut, kemudian pada bagian *Condition* tambahkan tipe data *Blend* dengan nilai kondisi 0.01.



Gambar 9.17 Memberikan Tipe Data

9. Hilangkan centang pada pengaturan *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0 pada bagian *Settings*.



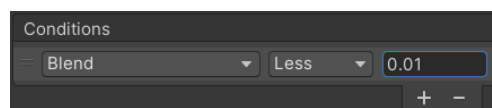
Gambar 9.18 Mengubah Pengaturan Transisi

10. Buat juga transisi dari *Player\_run* ke *Player\_idle* dengan cara klik kanan pada *Player\_run*, lalu pilih *Make Transition*.



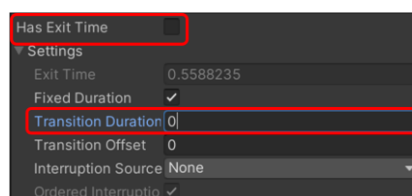
Gambar 9.19 Menambahkan Transisi

11. Kemudian pada transisi yang baru tambahkan tipe data *Blend* pada bagian *Condition* dengan nilai *Less* dan 0.01.



Gambar 9.20 Memberikan Tipe Data

12. Hilangkan centang pada pengaturan *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0 pada bagian *Settings*.



Gambar 9.21 Mengubah Pengaturan Transisi



13. Untuk menyesuaikan animasi, buka *script Player* dan tambahkan baris kode berikut di baris awal *class Player*.

```
public Animator animator;
```

14. Ubah fungsi *Awake* untuk menambahkan *script* komponen Animator.

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

15. Ubah fungsi *FixedUpdate* seperti berikut untuk memicu transisi berjalan.

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

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

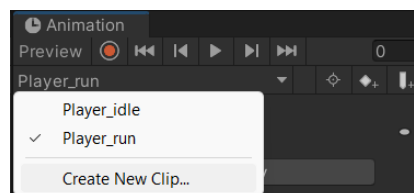
16. Kemudian coba jalankan, maka *player* akan memiliki animasi pada saat berhenti maupun berjalan.



Gambar 9.22 Menjalankan Game

### C. Menambahkan Animasi Jump dan Fall

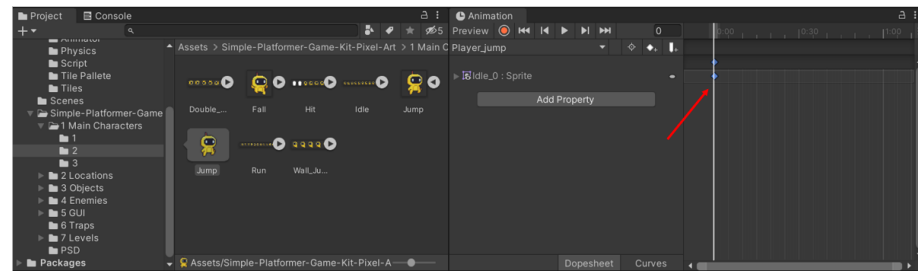
1. Buat animasi baru dengan klik pada *Player\_run*, kemudian pilih *Create New Clip* dan beri nama *Player\_jump* serta simpan pada *folder Animator*.



Gambar 9.23 Membuat Animasi Baru

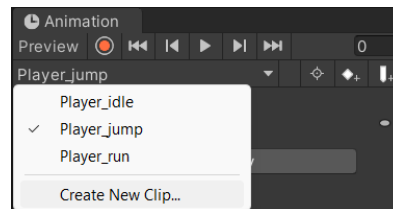


2. Pada *folder Project* buka *folder Main Character* lalu pilih *folder 2* dan pilih gambar 'Jump', kemudian *drag* ke *tab Animation*.



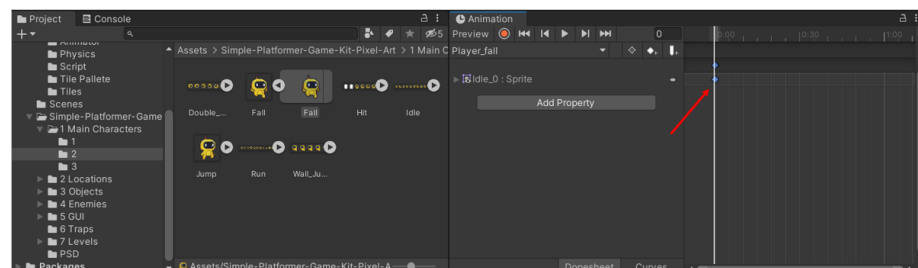
Gambar 9.24 Menambahkan Animasi Jump

3. Buat animasi baru dengan klik pada *Player\_jump*, kemudian pilih *Create New Clip* dan beri nama *Player\_fall* serta simpan pada *folder Animator*.



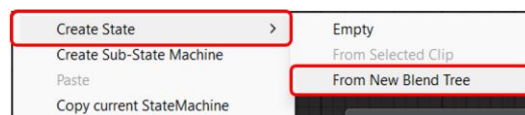
Gambar 9.25 Membuat Animasi Baru

4. Pada *folder Project* buka *folder Main Character* lalu pilih *folder 2* dan pilih gambar 'Fall', kemudian *drag* ke *tab Animation*.



Gambar 9.26 Menambahkan Animasi Fall

5. Kemudian klik kanan di area kosong pada menu *Animator*, pilih *Create State* lalu *From New Blend Tree* untuk menambahkan animasi ketika lompat.

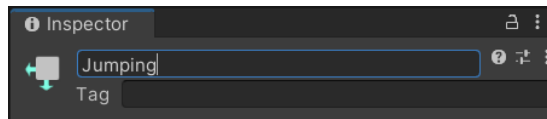


Gambar 9.27 Membuat Blend Tree Baru



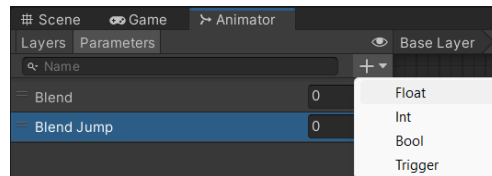


6. Buka menu *Inspector* pada *Blend Tree*, lalu ubah namanya menjadi *Jumping*.



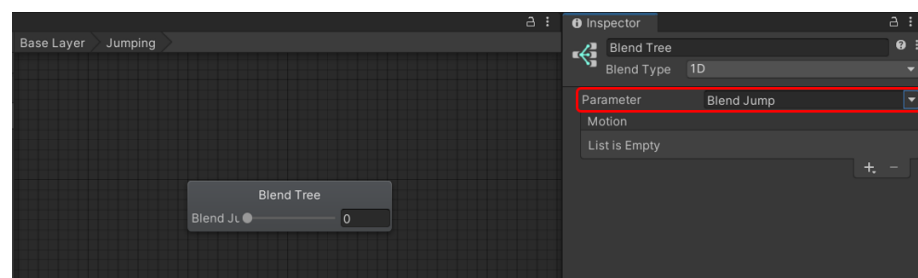
Gambar 9.28 Mengubah Nama Blend Tree

7. Masuk ke *tab* parameter, lalu tambahkan tipe data *float* dengan cara menekan *icon* tambah dan ubah nama menjadi *Blend Jump*.



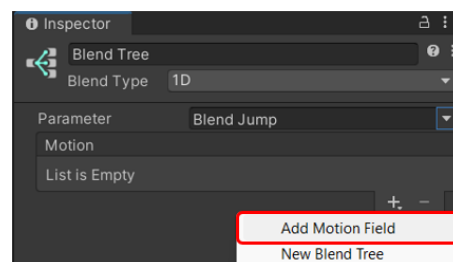
Gambar 9.29 Membuat Tipe Data Baru

8. Klik dua kali pada *Blend Tree Jumping*, lalu klik *Blend Tree* untuk mengubah parameter menjadi *Blend Jump* pada menu *Inspector*.



Gambar 9.30 Megubah Parameter

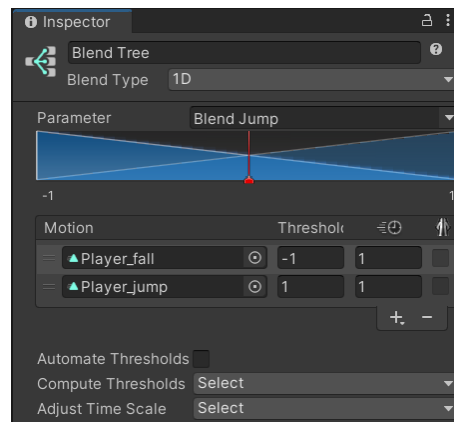
9. Tambahkan dua *Motion Field* dengan menekan *Icon* +, lalu pilih *Add Motion Field*.



Gambar 9.31 Membuat Motion Field

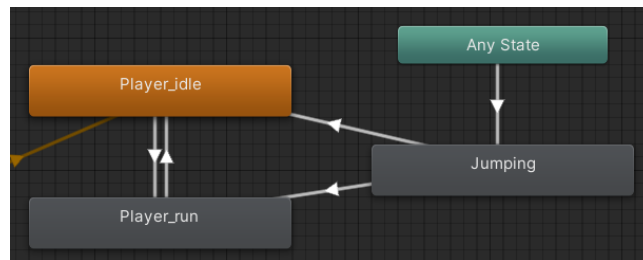


10. Klik *icon* pada bagian *None (Motion)*, lalu sesuaikan pengaturannya seperti gambar berikut.



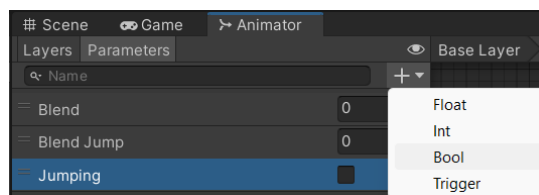
Gambar 9.32 Mengatur Motion Field

11. Kembali ke *Base Layer*, lalu buat transisi seperti pada gambar berikut.



Gambar 9.33 Membuat Transisi

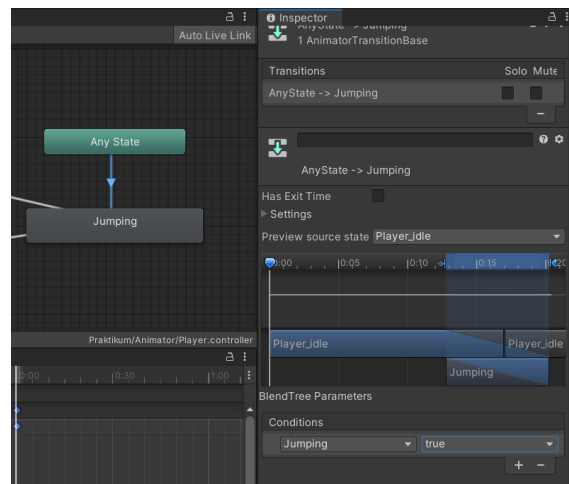
12. Masuk ke *tab* parameter, lalu tambahkan tipe data *bool* dengan cara menekan *icon* tambah dan ubah nama menjadi *Jumping*.



Gambar 9.34 Membuat Tipe Data Baru

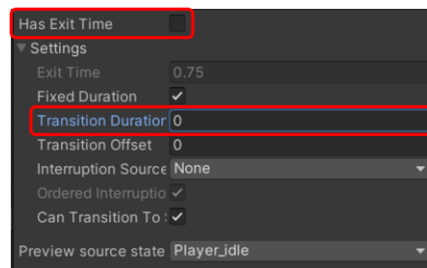


13. Klik panah yang mengarah ke *Jumping*, tambahkan *condition Jumping* dan ubah nilainya menjadi *true* pada menu *Inspector*.



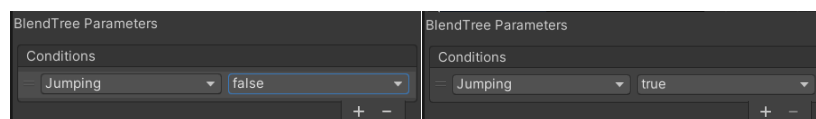
Gambar 9.35 Memberikan Tipe Data

14. Hilangkan centang pada pengaturan *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0 pada bagian *Settings*.



Gambar 9.36 Mengubah Pengaturan Transisi

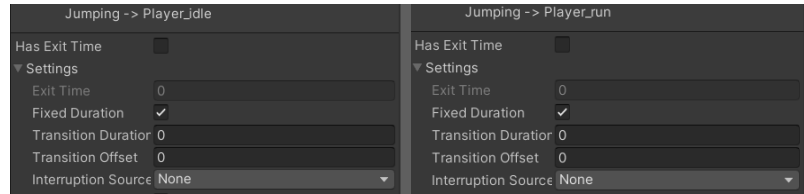
15. Klik panah yang mengarah ke *Player\_idle* dan *Player\_run*, tambahkan *condition Jumping*. Pada arah panah ke *Player\_idle* ubah menjadi *false*, pada arah panah ke *Player\_run* ubah menjadi *true*.



Gambar 9.37 Memberikan Tipe Data



16. Hilangkan centang pada pengaturan *Has Exit Time* dan atur nilai *Transition Duration* menjadi 0 pada bagian *Settings* pada kedua arah panah tersebut.



Gambar 9.38 Mengubah Pengaturan Transisi

17. Buka *script Player*, 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;
}
```

18. Ubah fungsi *FixedUpdate* seperti berikut.

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

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

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

19. Ubah fungsi *GroundCheck* seperti berikut.

```
void GroundCheck()
{
    isGrounded = false;
    Collider2D[] colliders =
    Physics2D.OverlapCircleAll(groundcheckCollider.position
    , groundCheckRadius, groundLayer);
    if (colliders.Length > 0)
        isGrounded = true;

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



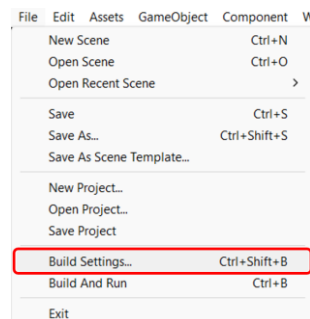
20. Kemudian jika di-play maka karakter sudah menggunakan animasi.



Gambar 9.39 Menjalankan Game

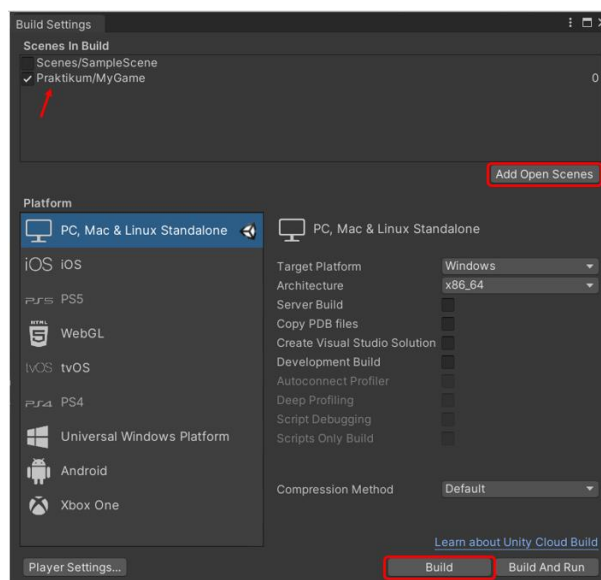
#### D. Render Game

1. Lalu untuk merender *game*, pilih *Build Settings* pada menu *File*.



Gambar 9.40 Menu File

2. Selanjutnya *Add Open Scenes* dan pilih *Scene* pada *project* yang ingin dirender, lalu pilih *Build* dan simpan kedalam *folder* yang diinginkan.



Gambar 9.41 Build Settings

#### E. Link Github Pengumpulan

[https://github.com/Nziaxi/2118028\\_PRAK\\_ANIGAME](https://github.com/Nziaxi/2118028_PRAK_ANIGAME)



## 9.2 Kuis : Memperbaiki Source Code Error

```
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);
    }
}
```

Penjelasan:

Pada fungsi `HandleJumpInput()`, terdapat perlu perbaikan pada saat mengatur 'isJumping' ke 'true' saat tombol spasi ditekan ('`Input.GetKeyDown`'). Lalu, perbaikan pada saat mengatur 'isJumping' ke 'false' saat tombol spasi ditekan ('`Input.GetKeyUp`'). Pada fungsi `HandleMovementInput()`, perlu perbaikan jika 'move' sama dengan 0, karakter dalam keadaan diam ('isIdle' *true*, 'isWalking' *false*). Sementara jika 'move' tidak sama dengan 0, karakter dalam keadaan diam ('isIdle' *false*, 'isWalking' *true*). Skala karakter perlu diubah dengan ukuran yang sama ketika bergerak ke kiri atau ke kanan agar terlihat proporsional.