TUGAS PERTEMUAN: 9

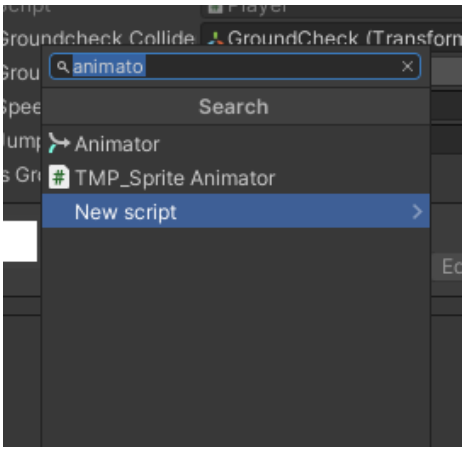
GAME ANIMATION

NIM	:	1918082
Nama	:	Dimas Fariski Setyawan Putra
Kelas	:	E
Asisten Lab	:	M. Rafi Faddilani (2118114)
Baju Adat	:	-
Referensi	:	-

1.1 Tugas 1 : Membuat Character Animation

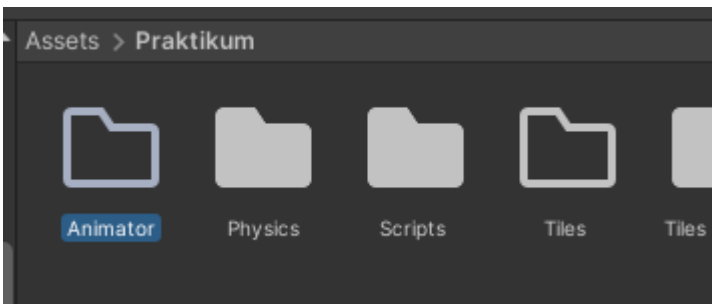
A. Membuat Character Animation

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



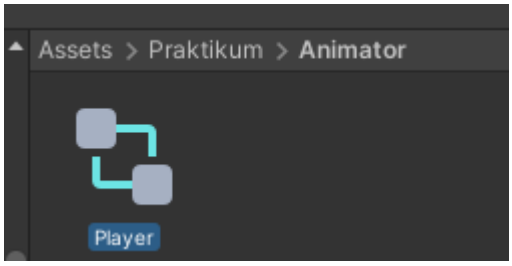
Gambar 1.1 Menambahkan Komponen

2. Buat foler baru,”Animator”



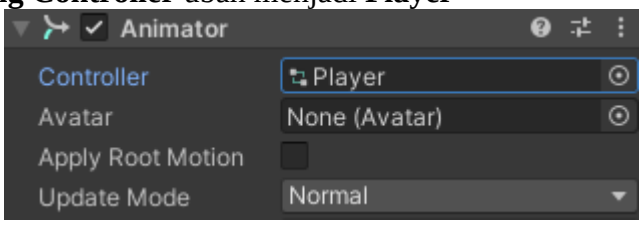
Gambar 1.2 Membuat Folder

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



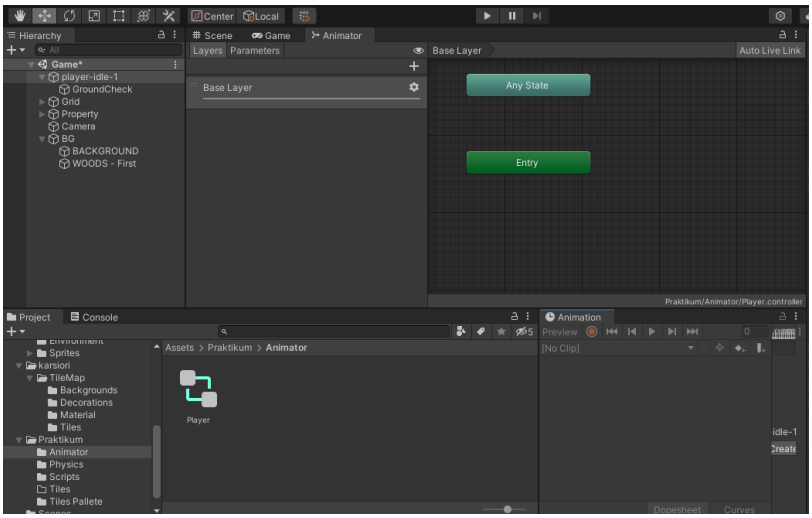
Gambar 1.3 Membuat Animator Controller

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



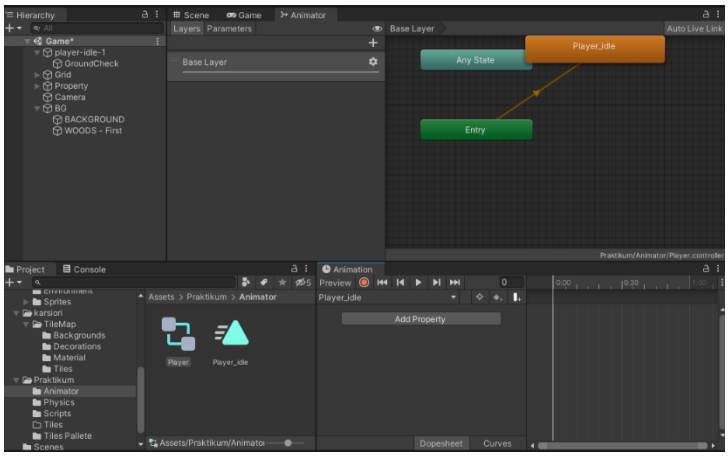
Gambar 1.4 Mengubah Hirarki Animator

5. Tambahkan menu panel **Animation** di menu **Window**, pilih **Animation** > **Animation** atau tekan **Shorcut CTRL + 6,** dan tambahkan panel animator



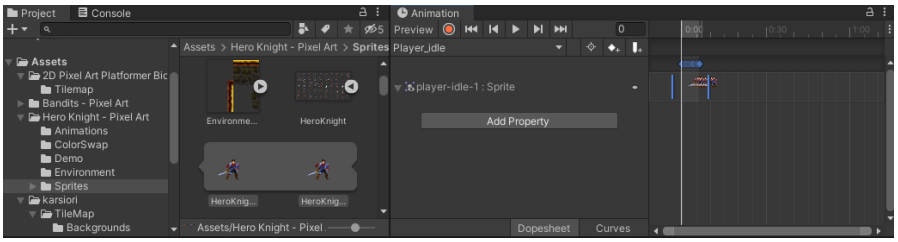
Gambar 1.5 Membuka Panel Animator dan Animation

6. Untuk membuat animasi klik **player-idle1** pada **Hierarchy**, kemudian ke menu panel **Animation**, pilih **Create** beri nama file “Player_idle”



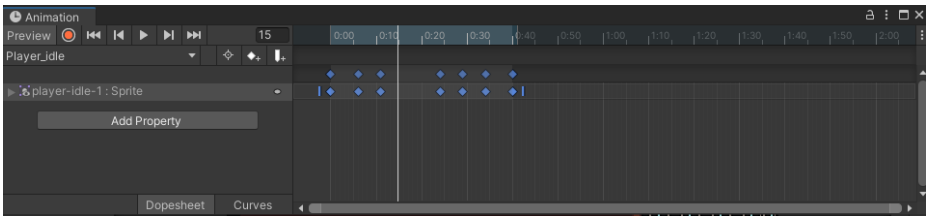
Gambar 1.6 Membuat Klip Animasi

7. Drag and drop heroknight-0 sampai heroknight-6 ke animation



Gambar 1.7 Memasukkan Animasi

8. Tekan CTRL + A pada menu **panel Animation** geser kotak kecil pada timeline sampai frame 0:40 agar animasinya tidak terlalu cepat

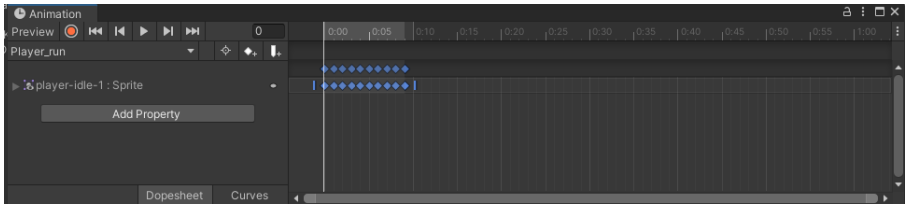


Gambar 1.8 Mengubah Timeframe Animasi

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

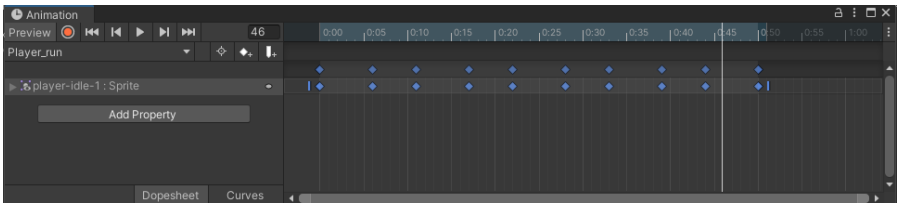
Gambar 1.9 Membuat Klip Animasi

10. Drag and drop heroknight-8 sampai heroknight-17 ke animation



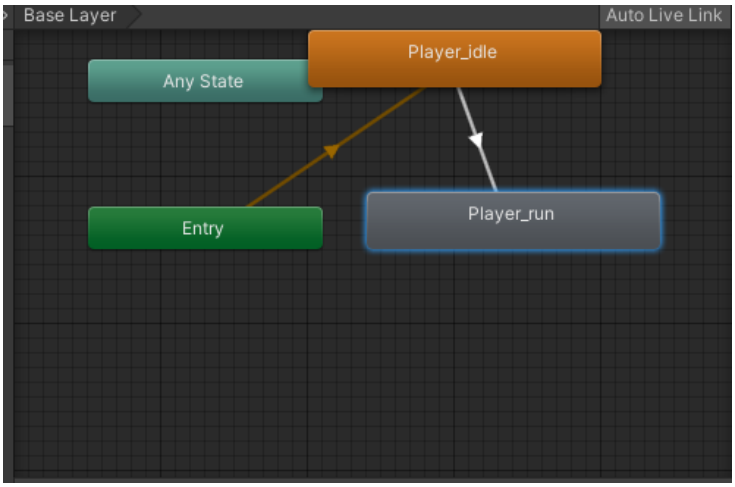
Gambar 1.10 Memasukkan Animasi

11. Tekan CTRL + A pada menu **panel Animation** geser kotak kecil pada timeline sampai frame 0:50 agar animasinya tidak terlalu cepat



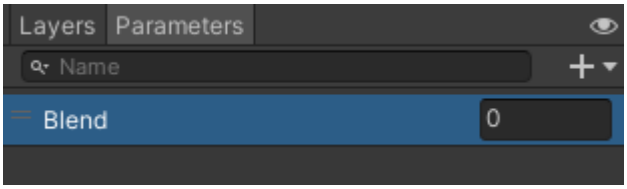
Gambar 1.11 Mengubah Timeframe Animasi

12. 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



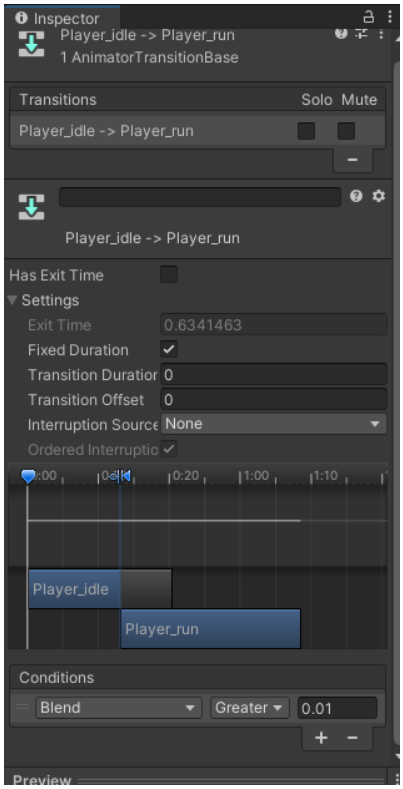
Gambar 1.12 Mebuat Transisi

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



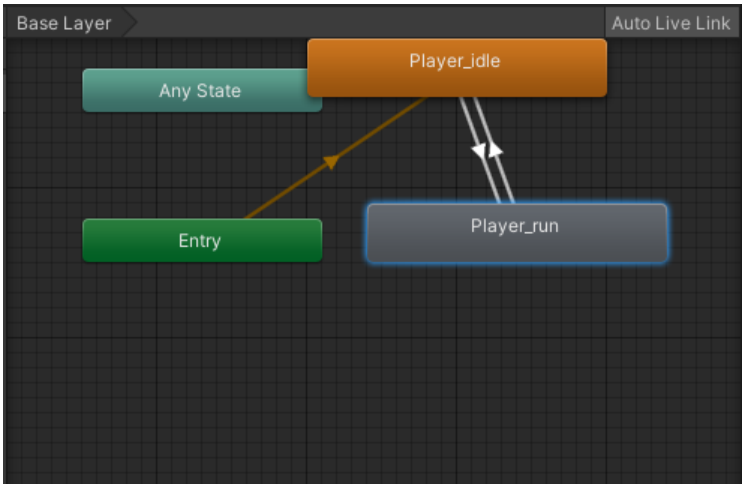
Gambar 1.13 Menambahkan Parameter Blend

14. Buat konfigurasi pada inspector seperti berikut



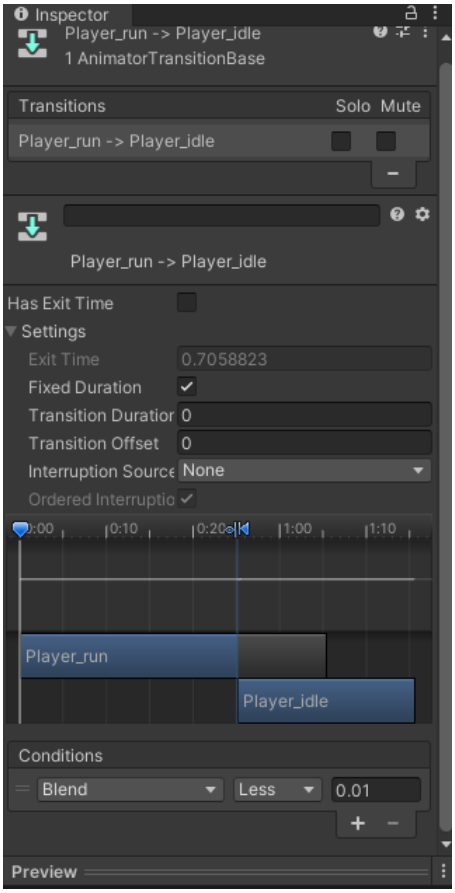
Gambar 1.14 Melakukan Konfigurasi Animasi

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



Gambar 1.15 Menyambungkan Transisi

16. Buat konfigurasi pada transisi player_run dan player_idle seperti berikut



Gambar 1.16 Melakukan Konfigurasi Animasi

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

Player.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    public Animator animator;
    Rigidbody2D rb;
    [SerializeField] Transform groundcheckCollider;
    [SerializeField] LayerMask groundLayer;
    const float groundCheckRadius = 0.2f; // +
    [SerializeField] float speed = 1;
    float horizontalValue;
    [SerializeField] float jumpPower = 300f;
    bool jump;
    [SerializeField] bool isGrounded; // +
    bool facingRight;
    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
```

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

    void Update()
    {
        horizontalValue                                    =
Input.GetAxisRaw("Horizontal");
        if (Input.GetButtonDown("Jump"))
            jump = true;
        else if (Input.GetButtonUp("Jump"))
            jump = false;
    }

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

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

    void Move(float dir, bool jumpflag)
    {
        if (isGrounded && jumpflag)
        {
            isGrounded = false;
            jumpflag = false;
            rb.AddForce(new Vector2(0f, jumpPower));
        }
        #region gerak kanan kiri



        float   xVal   =   dir   *   speed   *   100   *
Time.fixedDeltaTime;
        Vector2  targetVelocity  =  new  Vector2(xVal,
rb.velocity.y);
        rb.velocity = targetVelocity;

        if (facingRight && dir < 0)
        {
            // ukuran player
            transform.localScale  =  new  Vector3(-1,  1,
1);
            facingRight = false;
        }

        else if (!facingRight && dir > 0)
        {
```

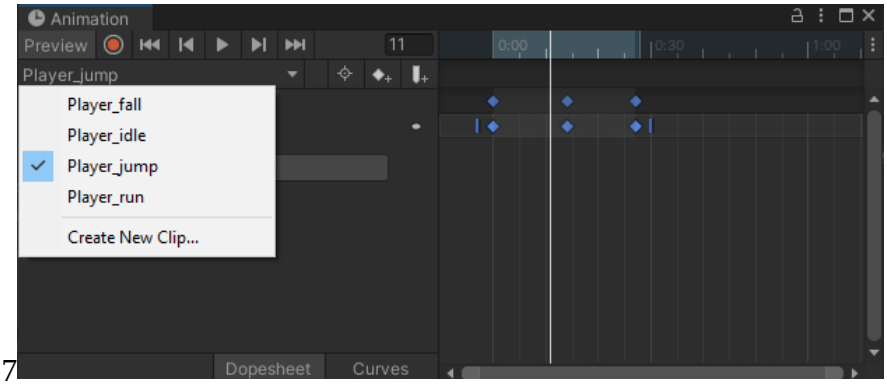
```
                // ukuran player
                transform.localScale = new Vector3(1, 1, 1);
                facingRight = true;
            }

            #endregion
        }
}
```

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

Gambar 1.17 Hasil Aimasi Idle dan Run

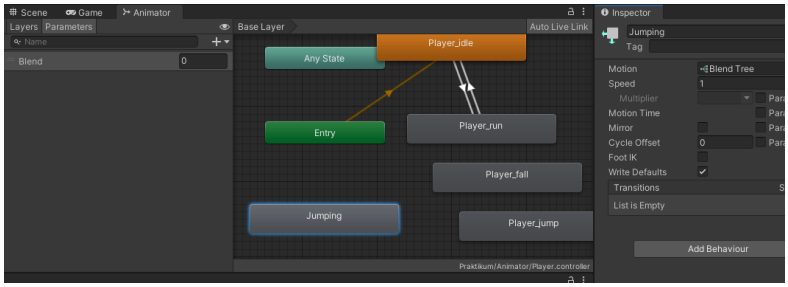
19. Buat klip baru, “Player_jump” dan “Player_fall”. Dan import gambar kedalam masing-masing klip



7

Gambar 1.18 Membuat Animasi Jatuh dan Lompat

20. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong , pilih Create State>From New Blend Tree dan ubah Namanya “Jumping”

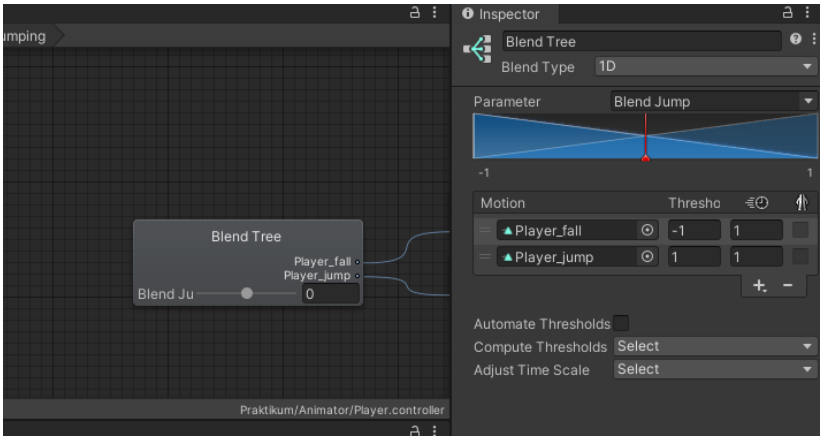


Gambar 1.19 Membuat Blend Tree

21. Pada menu Parameteres tambahkan parameter tipe data Float tekan icon + dan ubah namanya menjadi “Blend Jump”

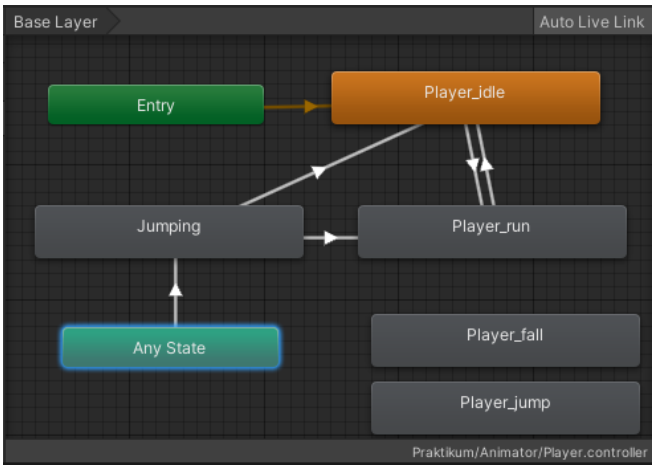
Gambar 1.20 Menambahkan Parameter Blend Jump

22. Pada menu Animator, Klik dua kali pada Blend Tree “Jumping”, Tekan pada Blend Tree dan konfigurasi seperti berikut



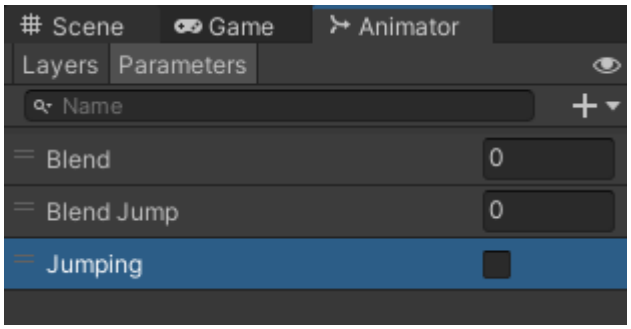
Gambar 1.21 Konfigurasi Blend Tree

23. Kembali ke Base Layer, klik kanan Any State, pilih Make Transition dan arahkan panahnya seperti berikut



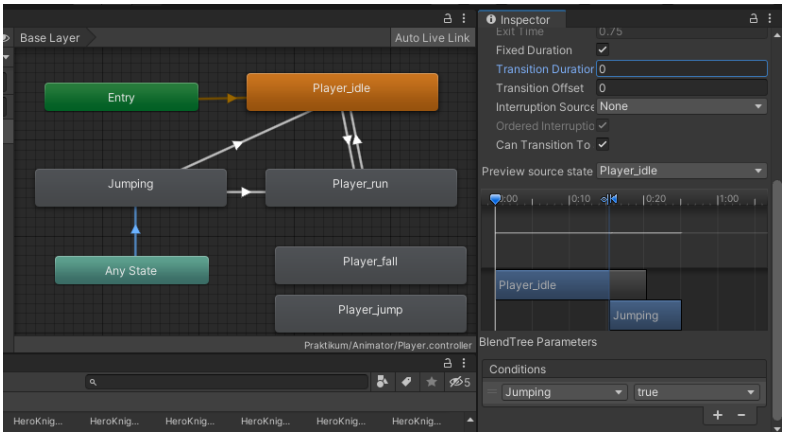
Gambar 1.22 Menyambungkan Transisi

24. Buat parameter baru



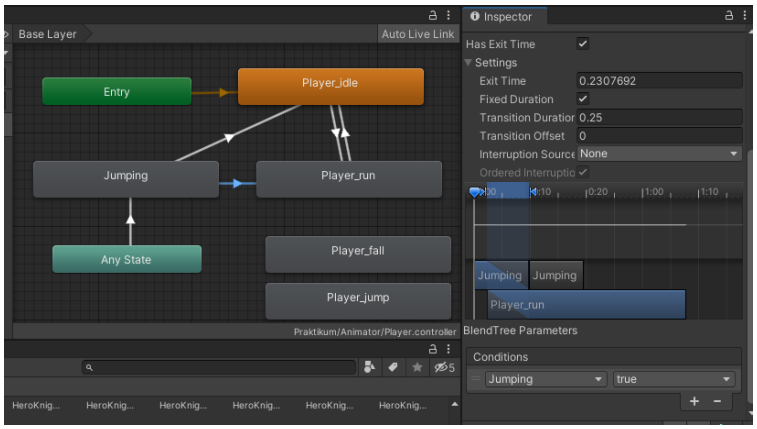
Gambar 1.23 Membuat Parameter Jumping

25. Klik panah yang mengarah ke Jumping, pada inspector tambahkan condition, pilih condition Jumping dan ubah nilainya menjadi true dan konfigurasikan seperti berikut



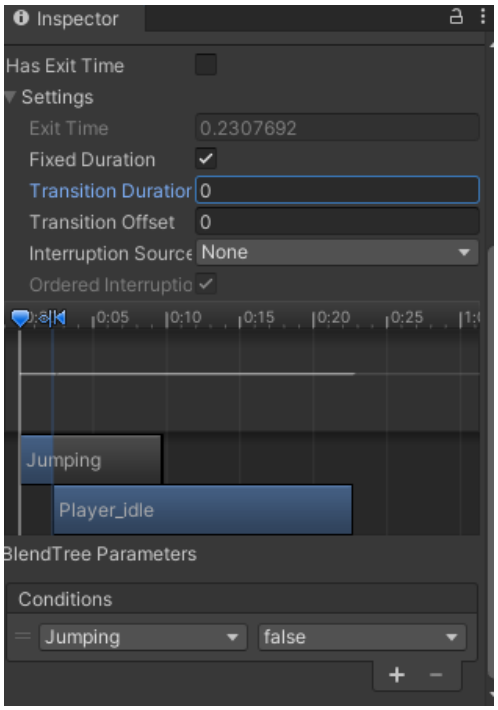
Gambar 1.24 Menambahkan Kondisi Transisi

26. 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 1.25 Konfigurasi Transisi

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



Gambar 1.26 Konfigurasi Transisi

28. Buka script Player, dan ubah source code berikut

Player.cs

`using System.Collections;`
`using System.Collections.Generic;`
`using UnityEngine;`

```
public class Player : MonoBehaviour
{
    public Animator animator;
    Rigidbody2D rb;
    [SerializeField] Transform groundcheckCollider;
    [SerializeField] LayerMask groundLayer;
    const float groundCheckRadius = 0.2f; // +
    [SerializeField] float speed = 1;
    float horizontalValue;
    [SerializeField] float jumpPower = 300f;
    bool jump;
    [SerializeField] bool isGrounded; // +
    bool facingRight;
    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();
    }

    void Update()
    {
        horizontalValue                                    =
Input.GetAxisRaw("Horizontal");
        if (Input.GetButtonDown("Jump")){
            animator.SetBool("Jumping", true);
            jump = true;
        }
        else if (Input.GetButtonUp("Jump"))
        {
            jump = false;
        }

    }

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

    void GroundCheck()
    {
        isGrounded = false;
        Collider2D[]             colliders             =
Physics2D.OverlapCircleAll(groundcheckCollider.position
, groundCheckRadius, groundLayer);
        if (colliders.Length > 0)
        {
            isGrounded = true;
        }
        animator.SetBool("Jumping", !isGrounded);
    }

    void Move(float dir, bool jumpflag)
    {
        if (isGrounded && jumpflag)
```

```
        {
            isGrounded = false;
            jumpflag = false;
            rb.AddForce(new Vector2(0f, jumpPower));
        }
        #region gerak kanan kiri



        float xVal = dir * speed * 100 *
Time.fixedDeltaTime;
        Vector2 targetVelocity = new Vector2(xVal,
rb.velocity.y);
        rb.velocity = targetVelocity;

        if (facingRight && dir < 0)
        {
            // ukuran player
            transform.localScale = new Vector3(-1, 1,
1);
            facingRight = false;
        }

        else if (!facingRight && dir > 0)
        {
            // ukuran player
            transform.localScale = new Vector3(1, 1, 1);
            facingRight = true;
        }

        #endregion
    }
}
```

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

Gambar 1.27 Hasil Tampilan

B. Link Github Pengumpulan

https://github.com/a-blue-moon/animasi-dan-game/tree/main/BAB%209

C. KUIS

```
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", false);
        }
    }

    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);
        }
    }
```

Analisa:

Pada source code di atas kurang nya parameter kedua saat memanggil fungsi “animator.setBool()”. Parameter kedua ini berfungsi untuk mengubah status animasi melompat (“isJumping”). Agar tidak terjadi error, tambahkan parameter kedua berupa nilai Boolean, dan disesuaikan dengan kondisi yang tepat.