



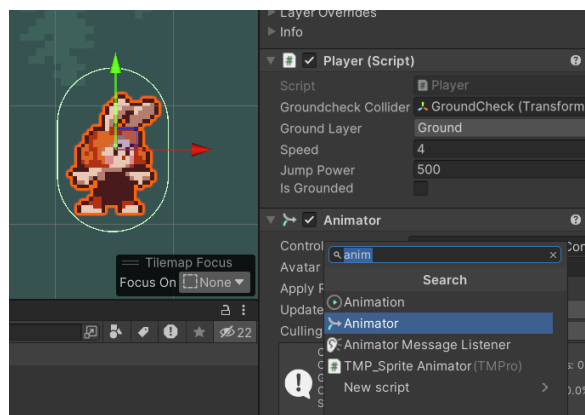
TUGAS PERTEMUAN: 8 CAMERA & CHARACTER MOVEMENT

NIM	:	2118077
Nama	:	Abdul Wahid
Kelas	:	B
Asisten Lab	:	MARIA AVRILIANA SURAT LELAONA (2218096)
Baju Adat	:	Pangasi Baju Adat sunda (jawa barat)
Referensi	:	https://akcdn.detik.net.id/community/media/visual/2022/04/20/pakaian-adat-jawa-barat-5_43.jpeg?w=700&q=90

9.1 Tugas 9 : Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

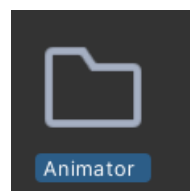
A. Membuat Pergerakan Player

1. Pada karakter klik inspector kemudian pilih Add Component Animator



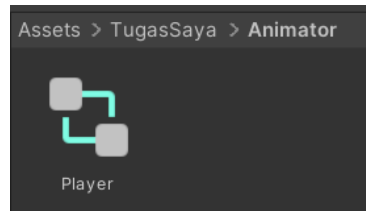
Gambar 9.1 Add Component Animator

2. Pada folder TugasSaya Buat Folder baru “Animator”



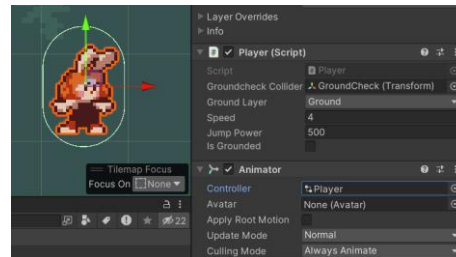
Gambar 9.2 Folder Animator

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



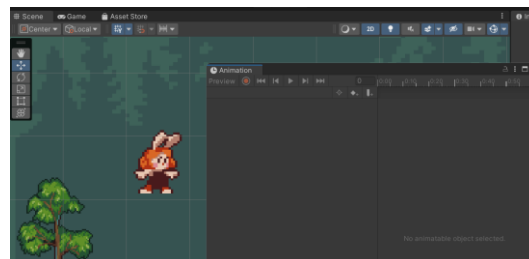
Gambar 9.3 Animator Player

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



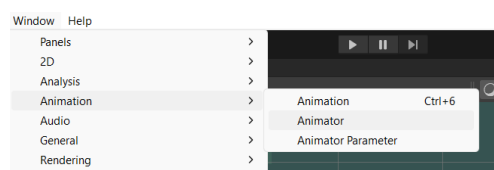
Gambar 9.4 Controller Player

5. Tambahkan menu panel Animation di menu Window, pilih Animation > Animation atau tekan Shortcut CTRL + 6



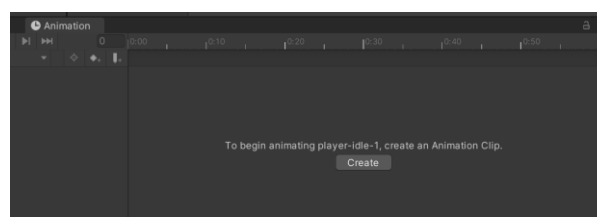
Gambar 9.5 Animation Window

6. Tambahkan menu panel Animator



Gambar 9.6 panel Animator

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



Gambar 9.7 menu panel Animation

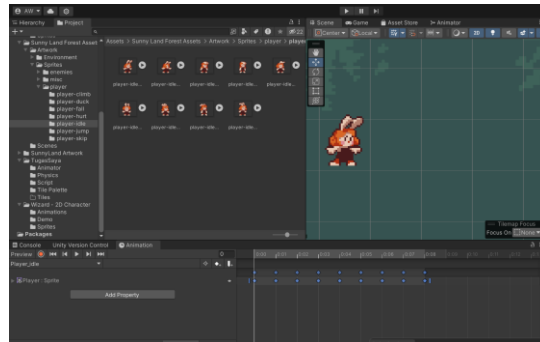


8. Simpan pada folder Animator dan beri nama “Player_idle”



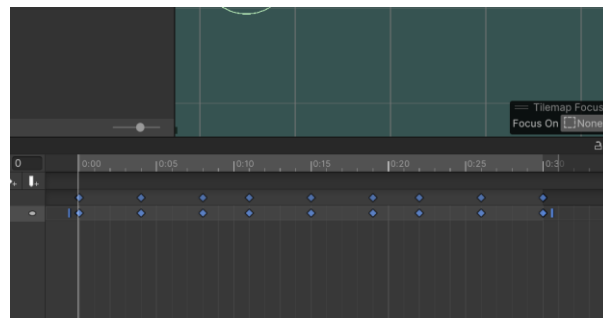
Gambar 9.8 folder Animator Player_idle

9. Pada menu Project buka folder player lalu pilih Idle dan pilih gambar player-idle-1 sampai player-idle-9, kemudian drag ke tab Animation



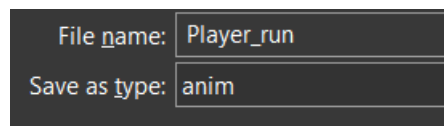
Gambar 9.9 player-idle

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



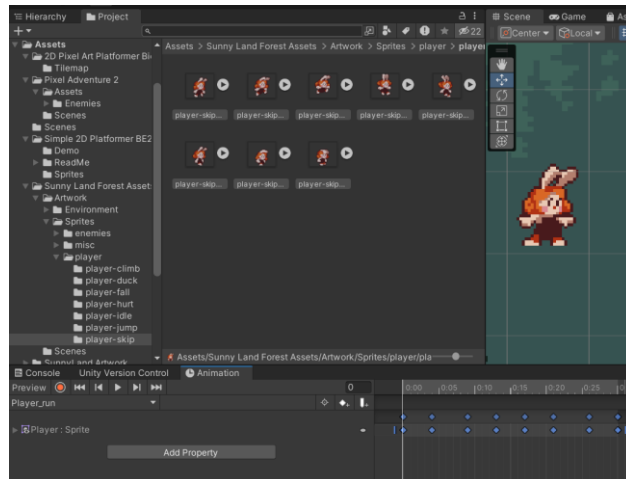
Gambar 9.10 Frame player-idle

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



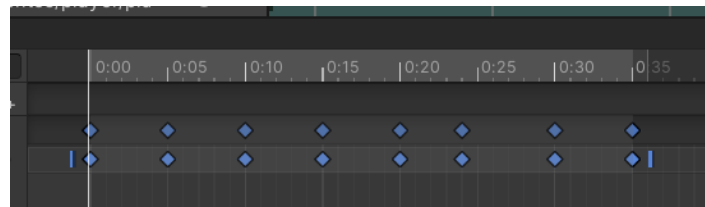
Gambar 9.11 Folder Player_run

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



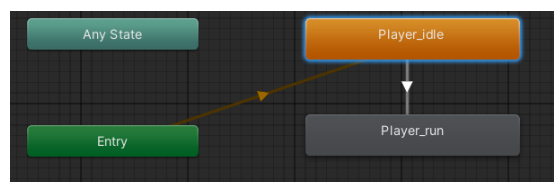
Gambar 9.12 Player_run

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



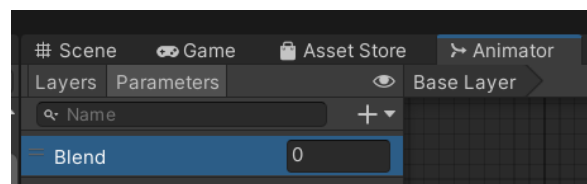
Gambar 9.13 Frame Player_run

14. 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



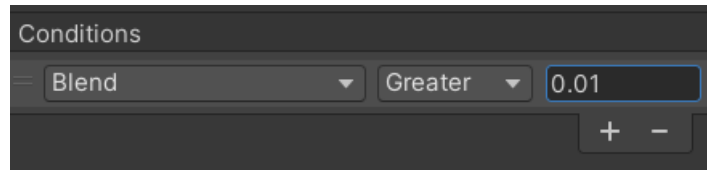
Gambar 9.14 Make Transition player_idle dan player_run

15. Masuk ke tab parameter, tambahkan tipe data bdengan cara tekan icon tambah dan ubah namanya menjadi "Blend"



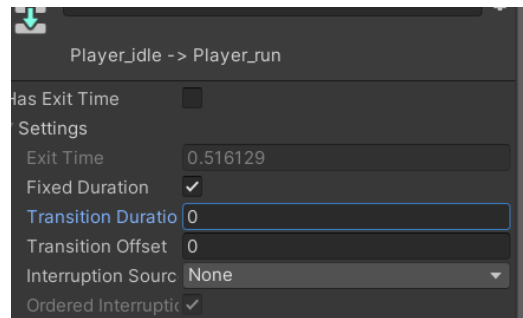
Gambar 9.15 Blend

16. Klik panah putih tersebut, pada bagian conditions klik icon tambah kemudian atur menjadi "Blend" dan atur nilai conditions blend tersebut menjadi 0.01



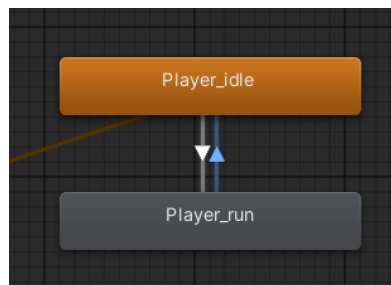
Gambar 9.16 Conditions Blend

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



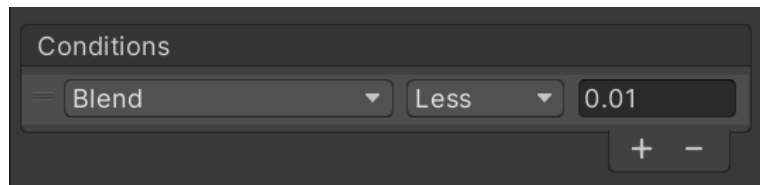
Gambar 9.17 Setting Player idle player run

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



Gambar 9.18 Make Transition player_run ke player_idle

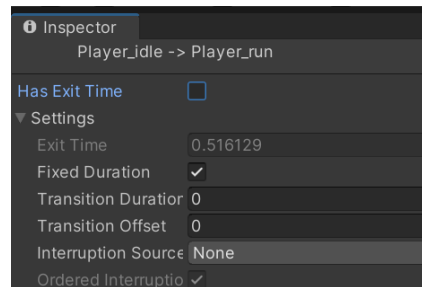
19. Tambahkan parameter transisi dengan tipe data Float. Klik ikon tambah dan rename menjadi "Blend". Setelah itu, ubah operator dari Greater menjadi Less dan atur nilainya menjadi 0.01.



Gambar 9.19 Conditions player_run ke player_idle



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



Setting player_run ke player_idle

21. Tambahkan Script

```
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;
    [SerializeField] float jumpPower = 200; // ++

    float horizontalValue;

    [SerializeField] bool isGrounded; // +
    bool facingRight = true; // Awal menghadap kanan
    bool jump; //++

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

    // Update is called once per frame
    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 (dir > 0 && !facingRight)
    {
        Flip();
    }
    else if (dir < 0 && facingRight)
    {
        Flip();
    }
    #endregion
}

void Flip()
{
    facingRight = !facingRight;
    Vector3 theScale = transform.localScale;
    theScale.x *= -1;
    transform.localScale = theScale;
}
}
```



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



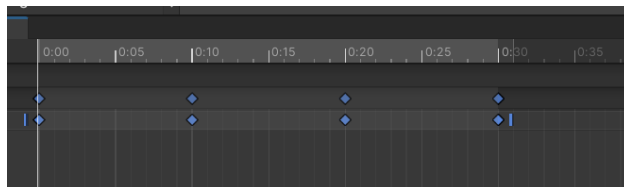
Gambar 9.20 Run player berjalan

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



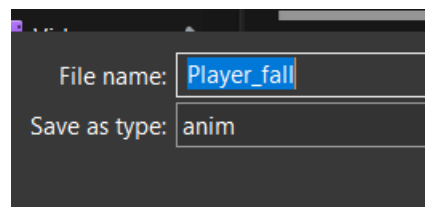
Gambar 9.21 Animasi Player jump

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



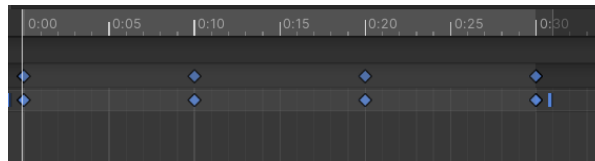
Gambar 9.22 Frame Player jump

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



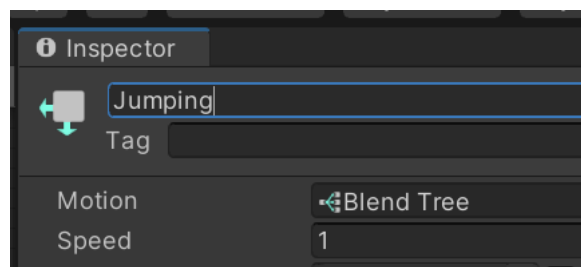
Gambar 9.23 Animasi Player_fall

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



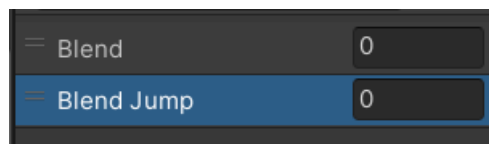
Gambar 9.24 Frame Player_fall

27. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu Animator, di area kosong , pilih Create State>From New Blend Tree dan pada Animator klik Blend Tree, di menu Inspector, ubah namanya menjadi Jumpig



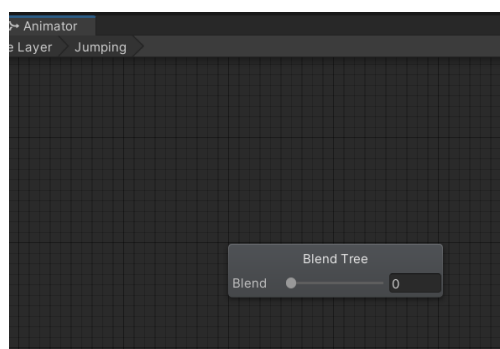
Gambar 9.25 From New Blend Tree

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



Gambar 9.26 Blend Jump

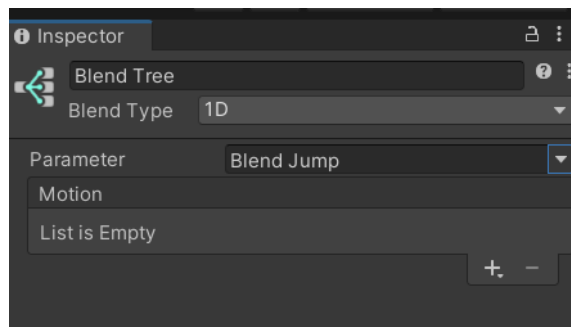
29. Pada menu Animator, Klik dua kali pada Blend Tree “Jumping”, Tekan pada Blend Tree



Gambar 9.27 Blend Tree Jumping

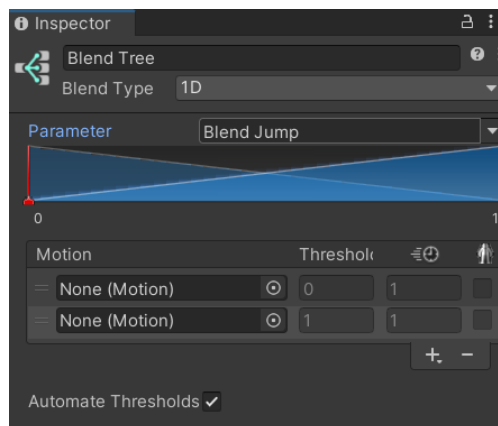


30. Klik 2X Blend Tree “Jumping”, pada inspector ubah parameter menjadi “Blend Jump”



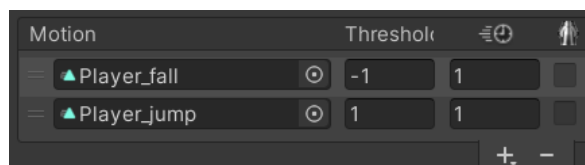
Gambar 9.28 Parameter Blend Jump

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



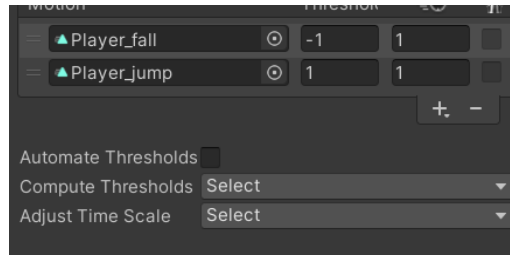
Gambar 9.29 Add Motion Field

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



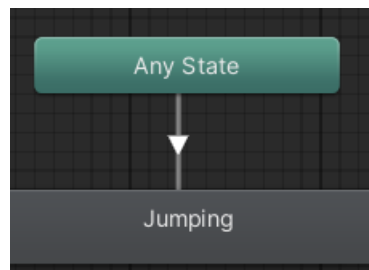
Gambar 9.30 Motion

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



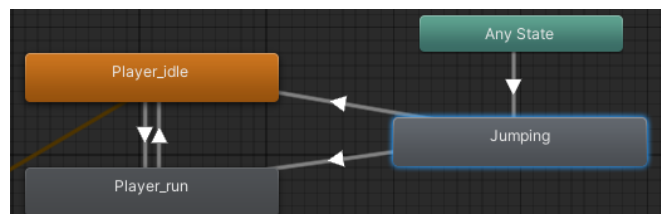
Gambar 9.31 Threshold

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



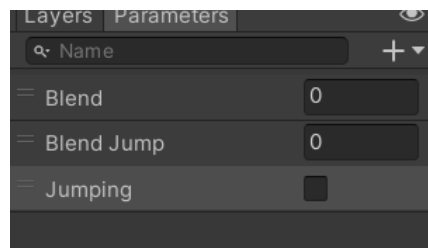
Gambar 9.32 Make Transition Jumping

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



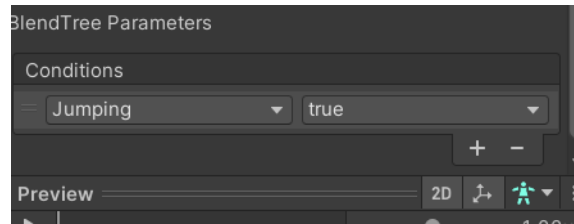
Gambar 9.33 Make Transition Player_idle dan Player_run

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



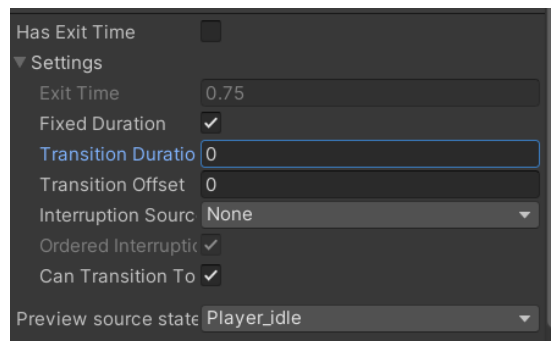
Gambar 9.34 Bool Jumping

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



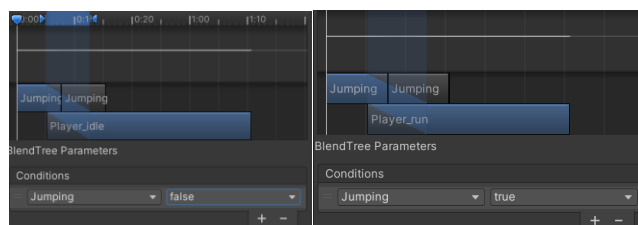
Gambar 9.35 Jumping True

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



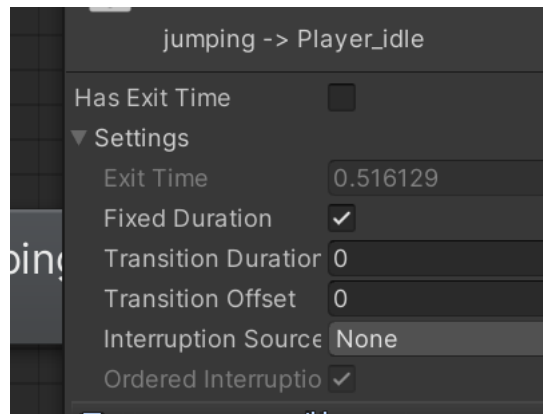
Setting nilai Transition Duration dan Has Exit Time

39. 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 9.36 Condition Player_idle dan Player_run

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



Gambar 9.37 Jumping player idle

41. Tambahkan Script

```
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;
    [SerializeField] float jumpPower = 200; // ++

    float horizontalValue;

    [SerializeField] bool isGrounded; // +
    bool facingRight = true; // Awal menghadap kanan
    bool jump; //++

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

    // Update is called once per frame
    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 (dir > 0 && !facingRight)
    {
        Flip();
    }
    else if (dir < 0 && facingRight)
    {
        Flip();
    }
    #endregion
}

void Flip()
{
    facingRight = !facingRight;
    Vector3 theScale = transform.localScale;
    theScale.x *= -1;
    transform.localScale = theScale;
}
}
```

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



Gambar 9.38 Hasil running

9.2 KUIS

```
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);
    }
}
```

Analisa Source Code :



Soutce Code Diatas manipulasi skala yang tidak konsisten dapat mengakibatkan perubahan ukuran karakter yang tidak diinginkan. Pastikan untuk hanya mengubah komponen x dari localScale untuk membalik arah tanpa mengubah skala asli karakter. Inisialisasi facingRight sebagai true dan gunakan metode Flip untuk membalik arah dengan mengubah x dari localScale. Debugging yang teliti dan uji coba yang berulang sangat penting untuk memastikan kode berjalan sesuai yang diinginkan.