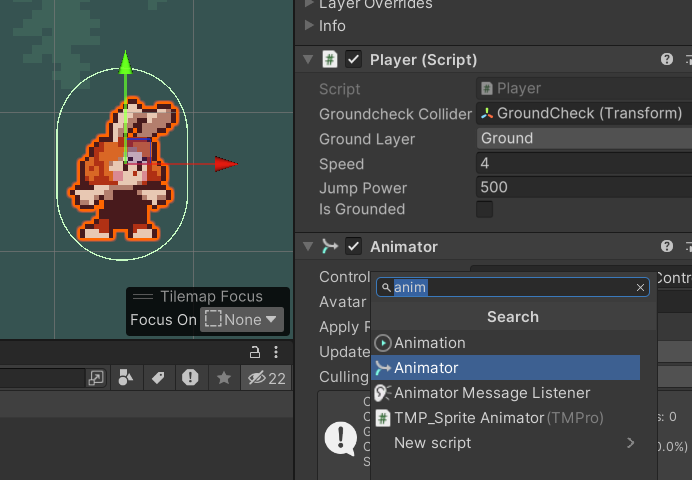
# CAMERA & CHARACTER MOVEMENT

|  |  |  |
| --- | --- | --- |
| **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 |

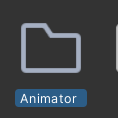
## Tugas 9 : Membuat Character Movement, Detect Ground, Jumping, & Camera Movement

1. **Membuat Pergerakan Player**
2. Pada karakter klik inspector kemudian pilih Add Component Animator



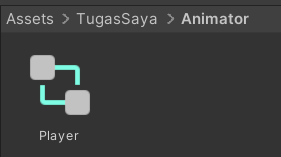
### Add Component Animator

1. Pada folder TugasSaya Buat Folder baru “Animator”



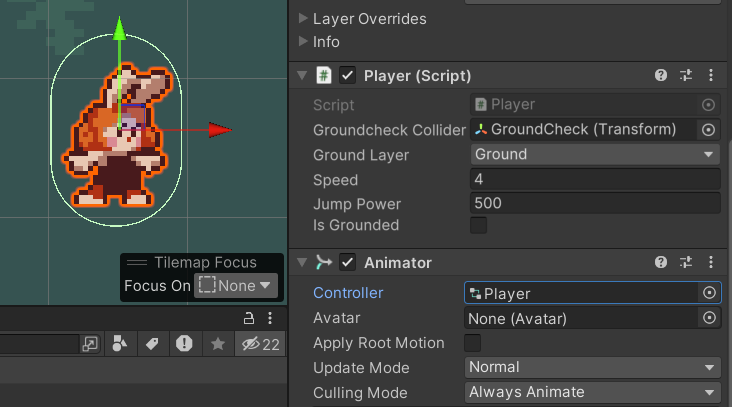
### Folder Animator

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



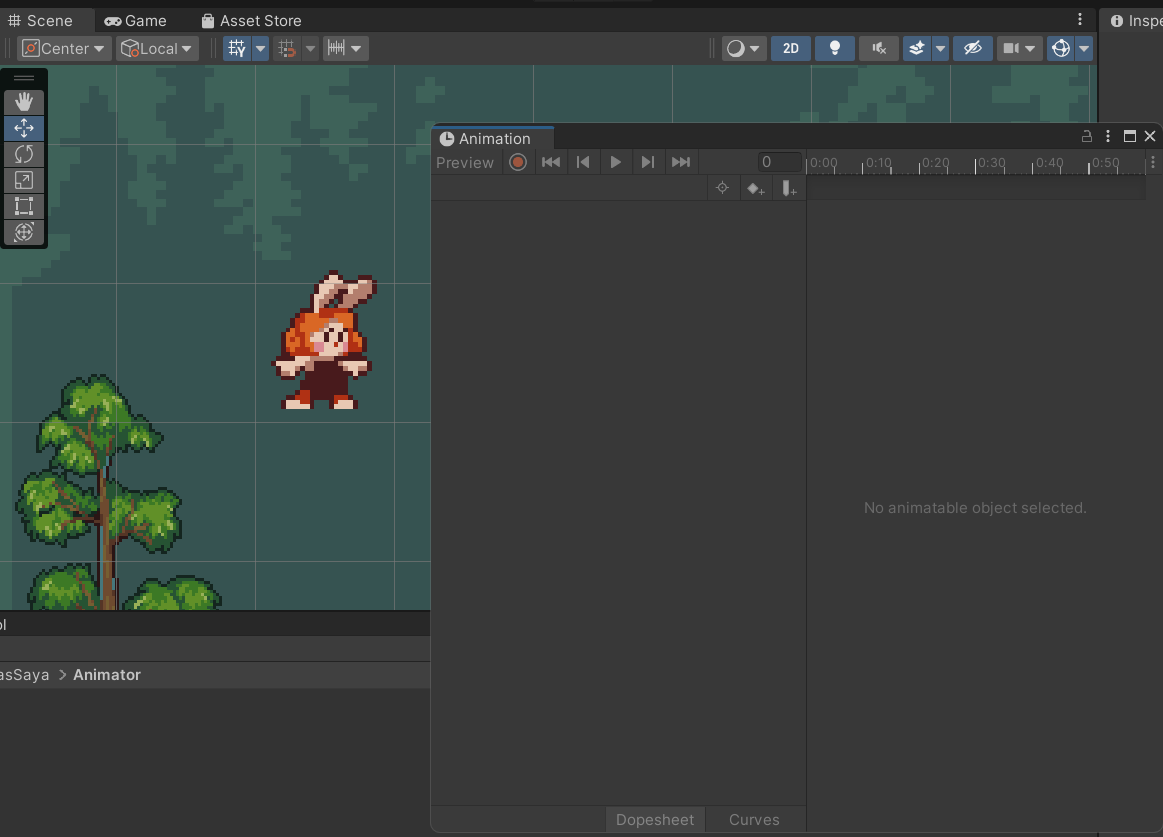
### Animator Player

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



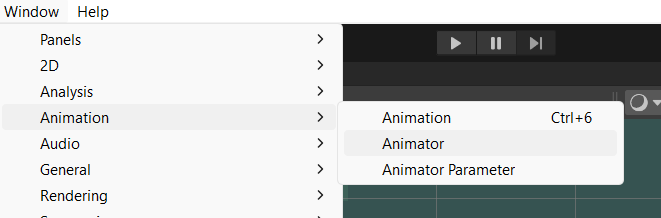
### Controller Player

1. Tambahkan menu panel Animation di menu Window, pilih Animation > Animation atau tekan Shorcut CTRL + 6



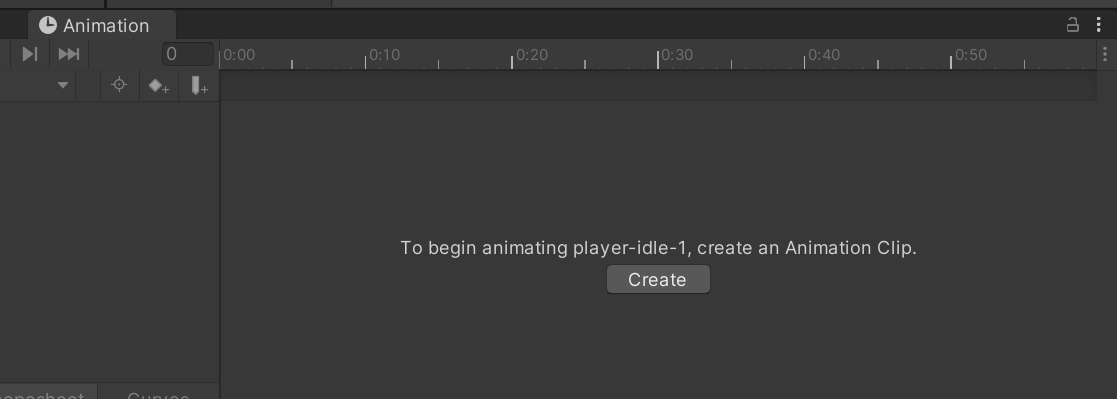
### Animation Window

1. Tambahkan menu panel Animator



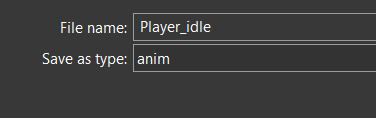
### panel Animator

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



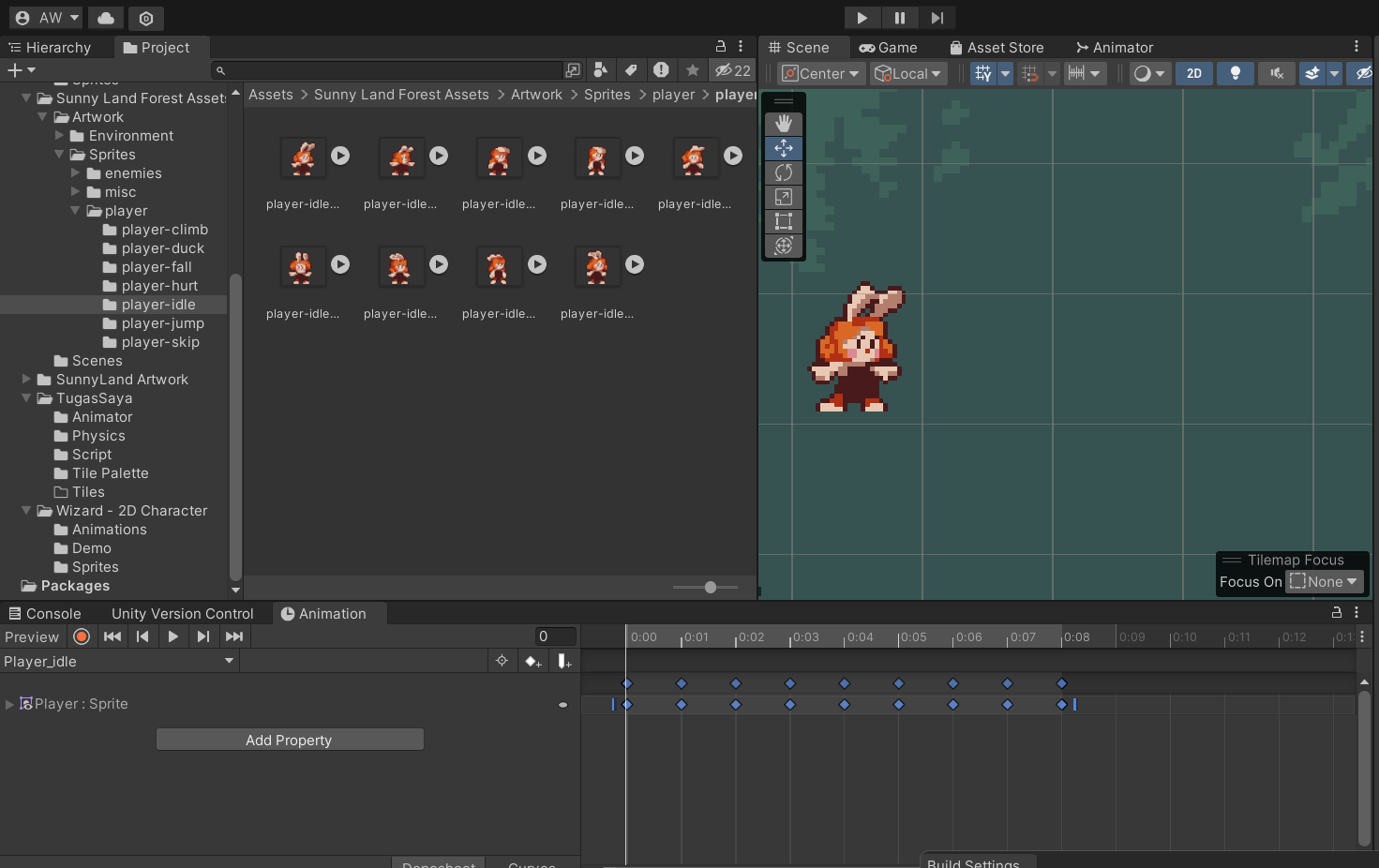
### menu panel Animation

1. Simpan pada folder Animator dan beri nama “Player\_idle”



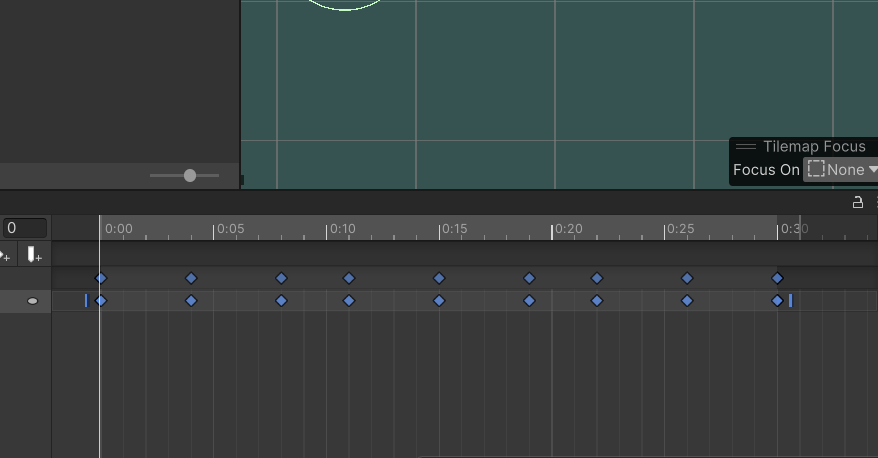
### folder Animator Player\_idle

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



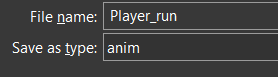
### player-idle

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



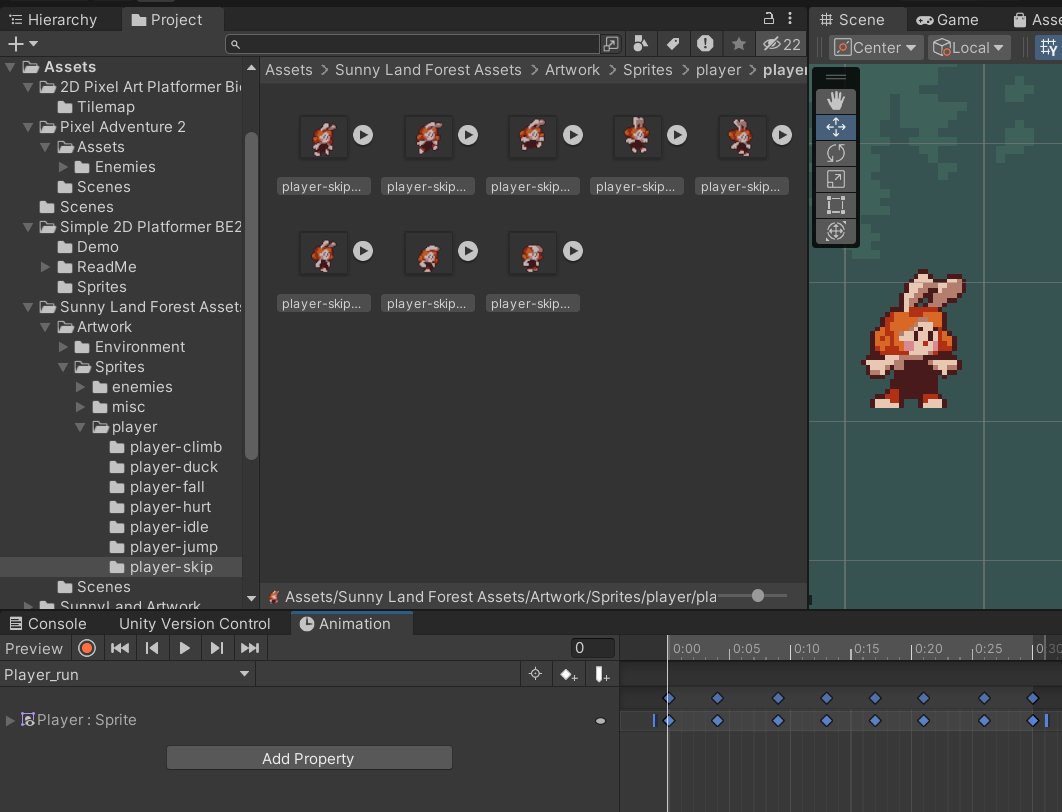
### Frame player-idle

1. Buat animasi baru, Klik pada “Player\_idle” kemudian pilih Create New Clip, dan beri nama “Player\_run”, Simpan pada Folder Animator



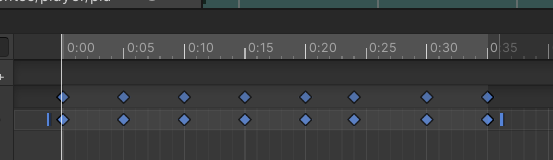
### Folder Player\_run

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



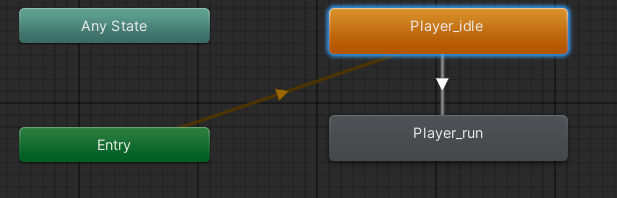
### Player\_run

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



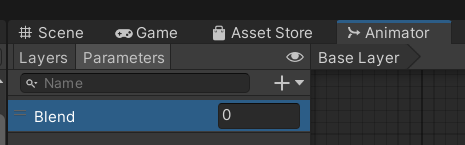
### Frame Player\_run

1. 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



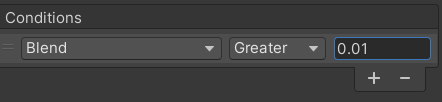
### Make Transition player\_idle dan player\_run

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



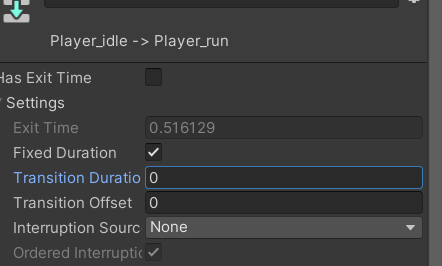
### Blend

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



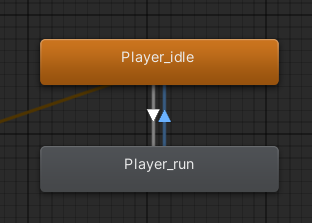
### Conditions Blend

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



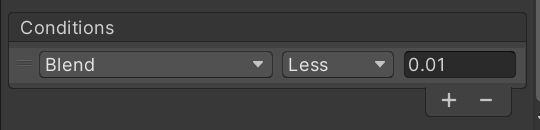
### Setting Player idle player run

1. Buat transisi juga dari player\_run ke player\_idle dengan cara klik kanan pada player\_run dan pilih Make Transition.



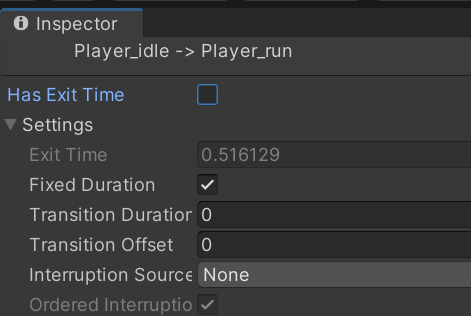
### Make Transition player\_run ke player\_idle

1. 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.



### Conditions player\_run ke player\_idle

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



Setting player\_run ke player\_idle

1. Tambahkan Script

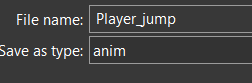
|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Player : MonoBehaviour  {  public Animator animator;    Rigidbody2D rb;  [SerializeField] Transform groundcheckCollider; // +  [SerializeField] LayerMask groundLayer; // +  const float groundCheckRadius = 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;  }  } |

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



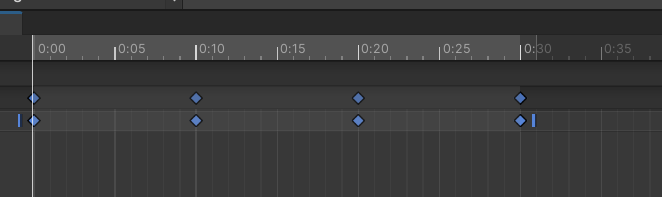
### Run player berjalan

1. Kemudian buat animasi baru tekan tulisan “Player\_run” kemudian pilih Create New Clip, dan beri nama “Player\_jump”



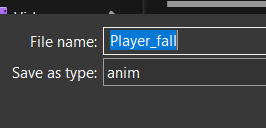
### Animasi Player jump

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



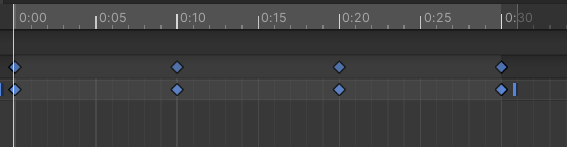
### Frame Player jump

1. Buat animasi baru dengan cara tekan tulisan “Player\_jump" kemudian pilih Create New Clip, dan beri nama “Player\_fall”



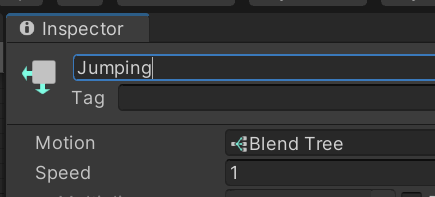
### Animasi Player\_fall

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



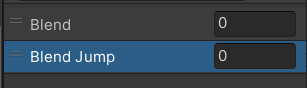
### Frame Player\_fall

1. 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



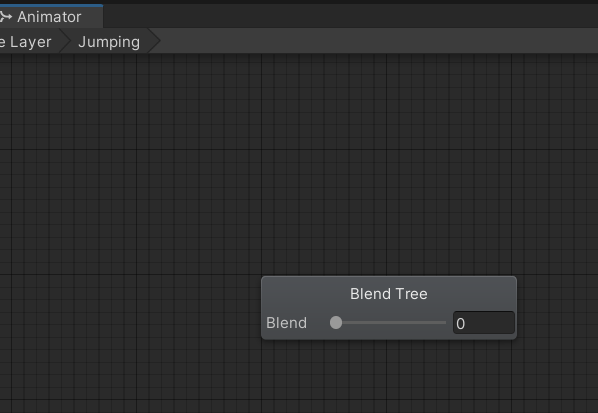
### From New Blend Tree

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



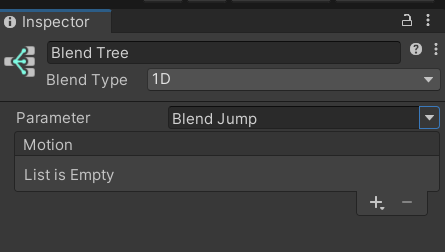
### Blend Jump

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



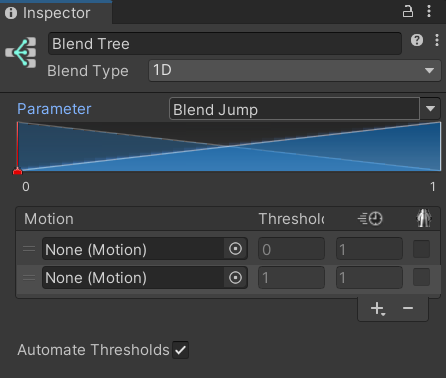
### Blend Tree Jumping

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



### Parameter Blend Jump

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



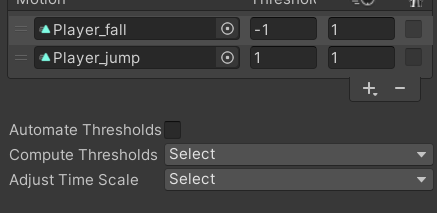
### Add Motion Field

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



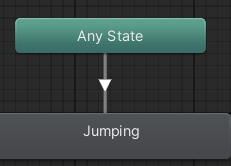
### Motion

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



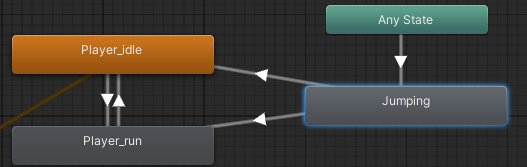
### Threshold

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



### Make Transition Jumping

1. Klik kanan Jumping, pilih Make Transition dan arahkan panahnya ke Player\_idle dan Player\_run



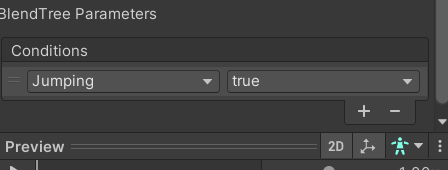
### Make Transition Player\_idle dan Player\_run

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



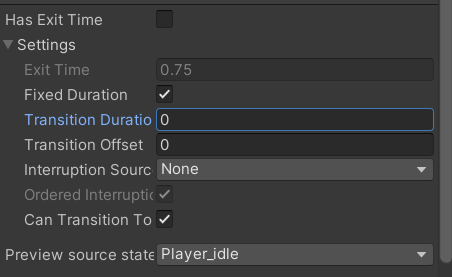
### Bool Jumping

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



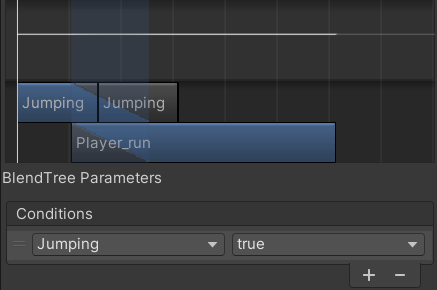
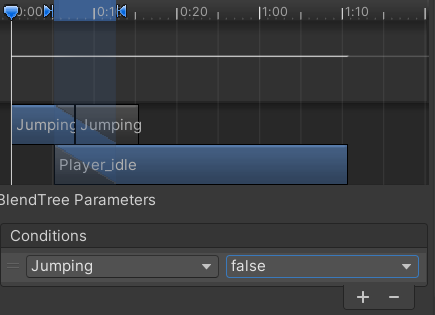
### Jumping True

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



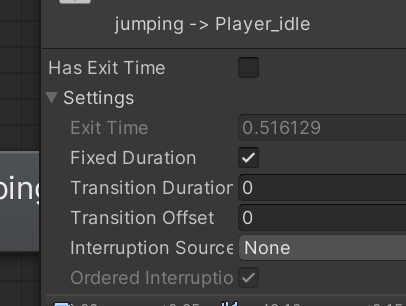
Setting nilai Transition Duration dan Has Exit Time

1. 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



### Condition Player\_idle dan Player\_run

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



### Jumping player idle

1. 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;  }  } |

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



### Hasil running

## 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.