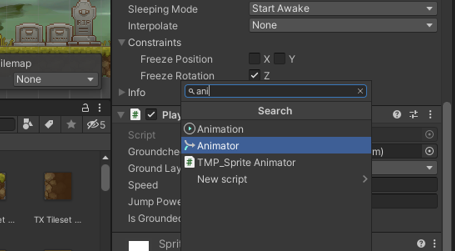
# 9 GAME ANIMATION

|  |  |  |
| --- | --- | --- |
| **NIM** | : | 2118061 |
| **Nama** | : | Adi Julia Saputra |
| **Kelas** | : | Informatika B |
| **Asisten Lab** | : | Bagas Anardi (2118004) |

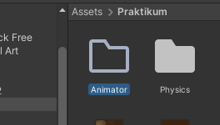
## Tugas 9 : Game Animation

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



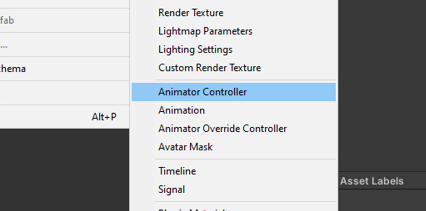
### 9.1 *Add Component Animator*

1. Pada *folder* Praktikum buat *Folder* baru “*Animator*”.



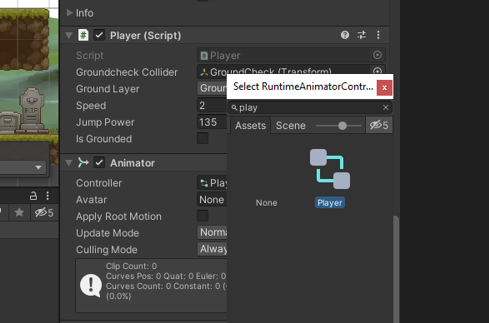
### 9.2 *New* *Folder*

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



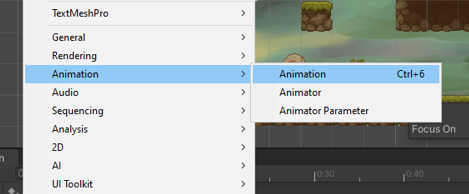
### 9.3 *Animator Controller*

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



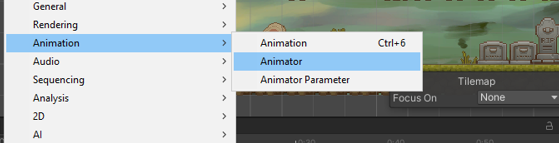
### 9.4 *Setting Controller*

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



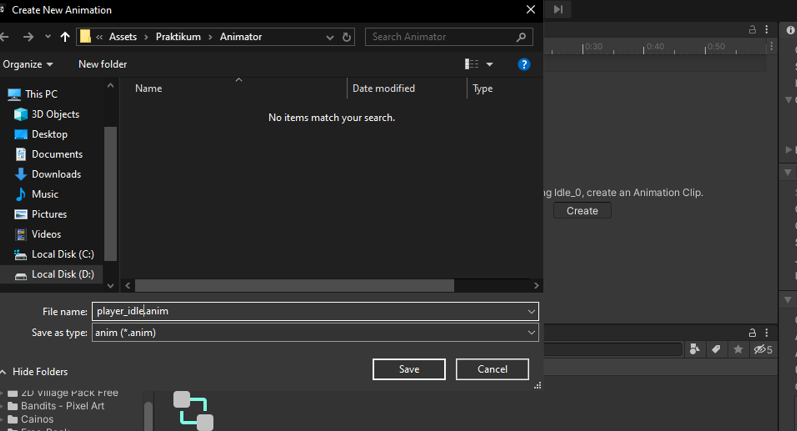
### 9.5 *Window Animation*

1. Tambahkan menu panel *Animator.*



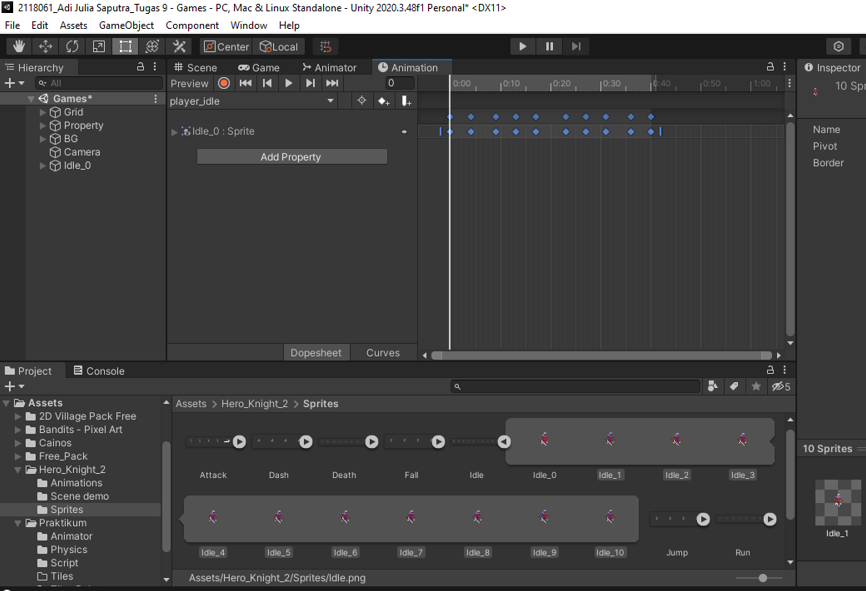
### 9.6 *Window Animator*

1. Untuk membuat animasi klik *player* pada *Hierarchy*, kemudian ke menu panel *Animation*, pilih *Create*. Simpan pada *folder* *Animator* dan beri nama “Player\_idle”.



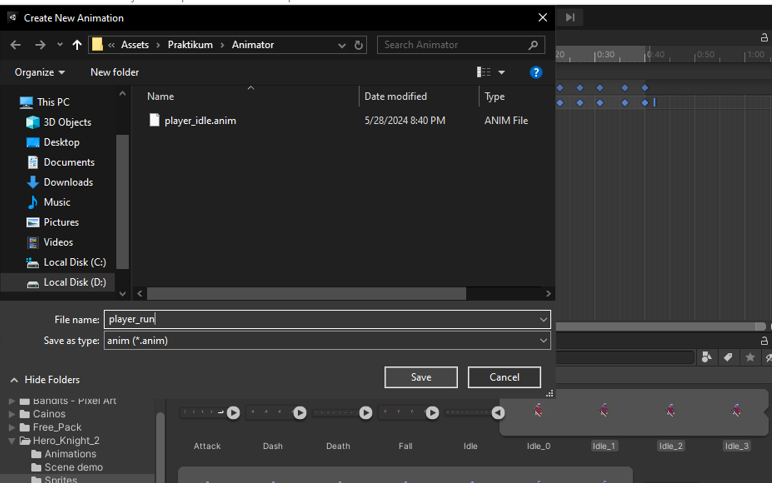
### 9.7 *Player Idle*

1. Pada menu *Project* buka *folder* *player* lalu pilih *Idle* dan pilih semua gambar *idle*, kemudian *drag* ke *tab* *Animation*. Tekan CTRL + A pada menu panel *Animation* geser kotak kecil pada *timeline* sampai *frame* 0:30 agar animasinya tidak terlalu cepat.



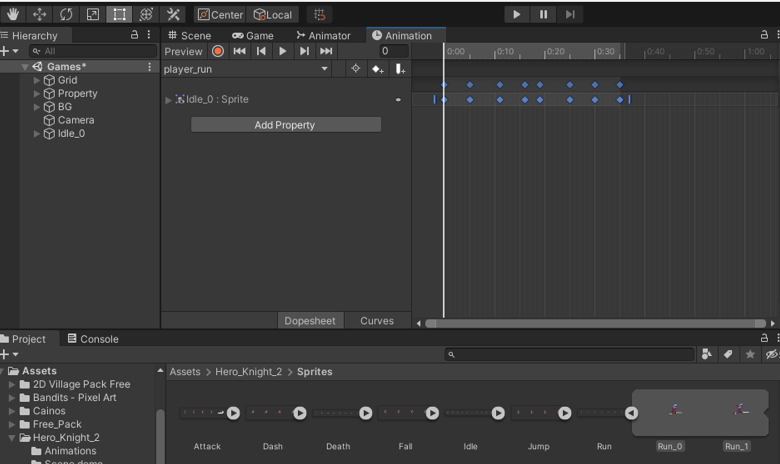
### 9.8 *Player idle*

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



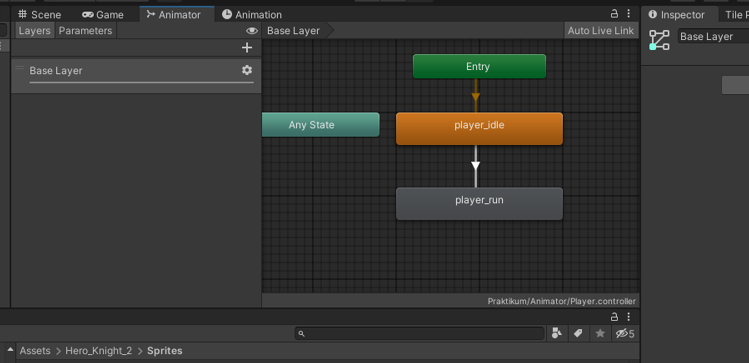
### 9.9 *Player run*

1. Buka menu *Project* kemudian cari *folder* *Player* > *run*, Pilih gambar *player run*, *drag and drop* pada menu *Animation*. Pada panel *timeline* tekan Ctrl+A di *keyboard*, klik bagian kotak kecil disamping *keyframe* terakhir dan geser sampai waktu 0:35.



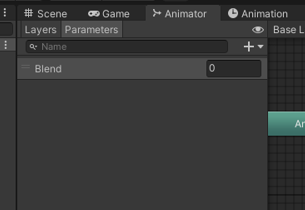
### 9.10 *Player run*

1. Pilih ke menu *Animator* yang telah dibuka sebelumnya dan akan tampil seperti berikut. 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*.



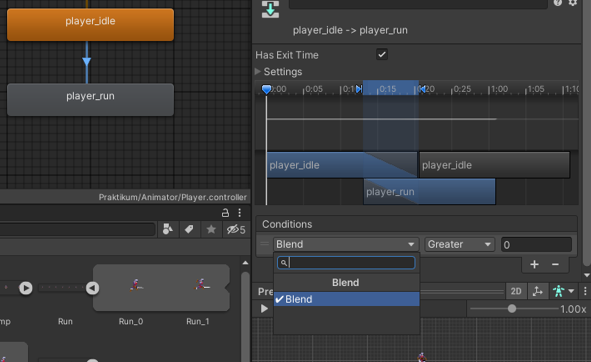
### 9.11 *Make Transition*

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



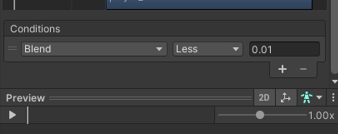
### 9.12 *Blend*

1. Klik panah putih tersebut, pada bagian *conditions* klik *icon* tambah kemudian atur menjadi “*Blend*”*.*



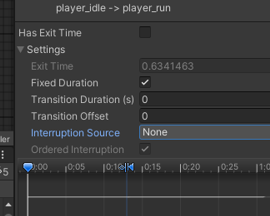
### 9.13 *Blend Setting*

1. Atur nilai *conditions blend* tersebut menjadi 0.01*.*



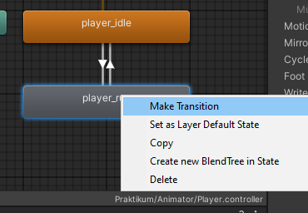
### 9.14 *Conditions Blend*

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



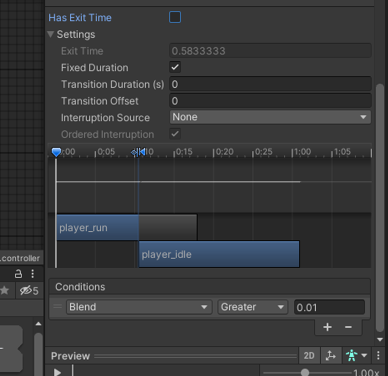
### 9.15 *Settings*

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



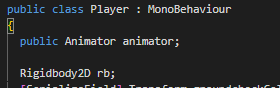
### 9.16 *Make Transition*

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



### 9.17 *Blend Settings*

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



### 9.18 *Source Code*

1. Tambahkan Script Komponen Animator.

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

1. Dan pada fungsi FixedUpdate rubah menjadi source code berikut***e***.

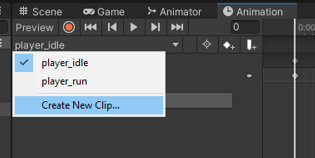
|  |
| --- |
| void FixedUpdate()    {    GroundCheck();    Move(horizontalValue, jump);    animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));    animator.SetFloat("Blend Jump", rb.velocity.y);    } |

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



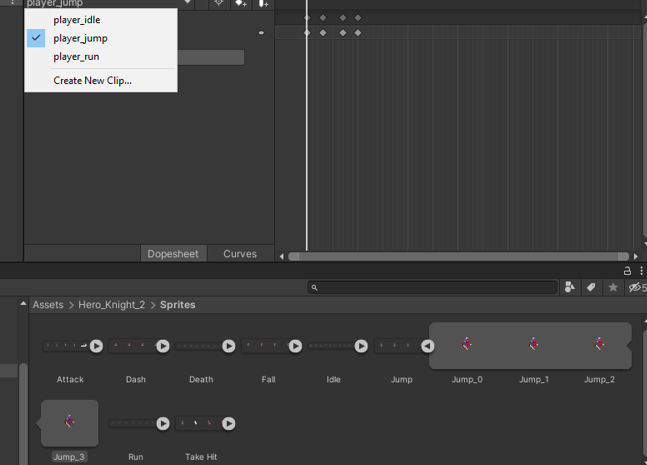
### 9.19 Animasi Berjalan

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



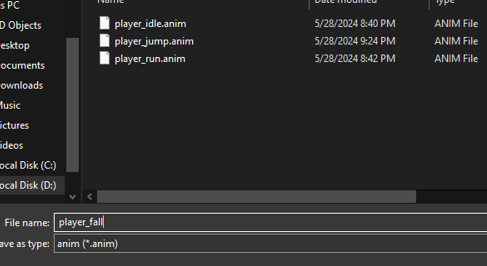
### 9.20 *Create New Clip*

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



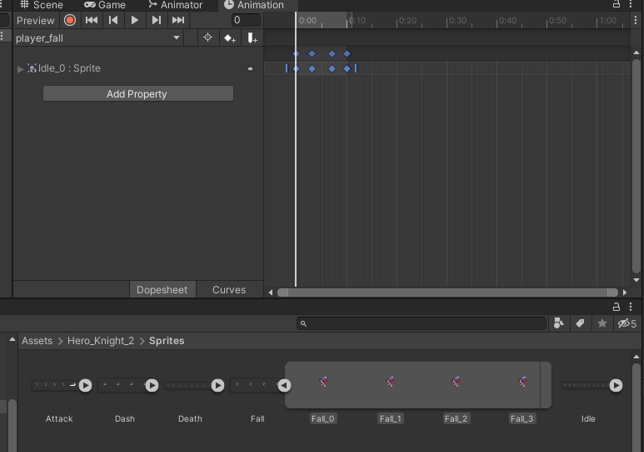
### 9.21 *Tab Animation*

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



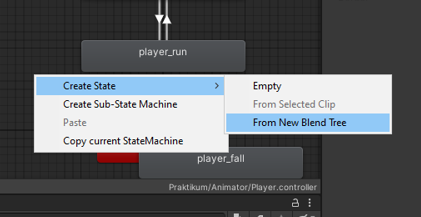
### 9.22 *Create New Clip*

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



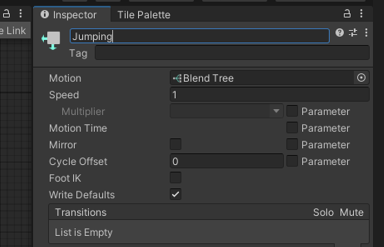
### 9.23 *Tab Animation*

1. Kemudian untuk menambahkan animasi ketika melompat. Klik kanan pada menu *Animator*, di area kosong, pilih *Create State>From New Blend Tree*.



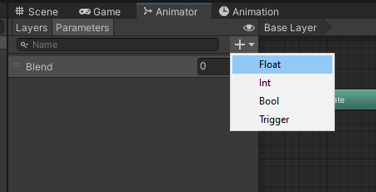
### 9.24 *Blend Tree*

1. Pada *Animator* *klik* *Blend Tree*, di *menu* *Inspector*, ubah namanya menjadi *Jumping.*



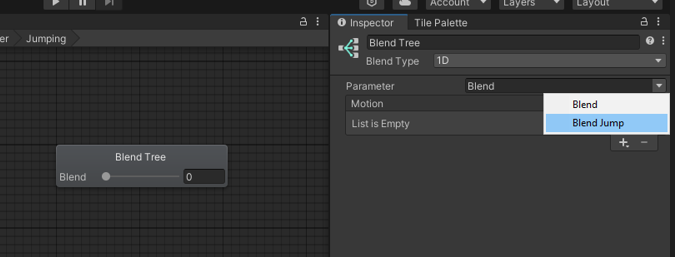
### 9.25 *Jumping*

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



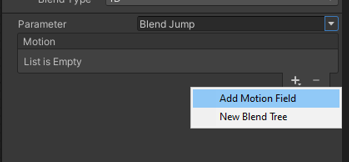
### 9.26 *Blend Jump*

1. Pada menu *Animator*, Klik dua kali pada *Blend Tree* “*Jumping*”, Tekan pada *Blend Tree*. Klik 2X *Blend* *Tree* “*Jumping*”, pada *inspector* ubah parameter menjadi “*Blend Jump*”



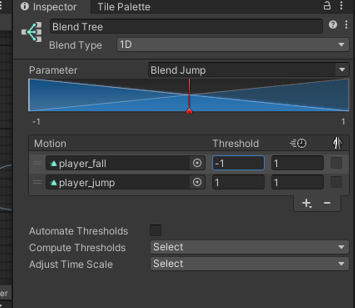
### 9.27 *Blend Jump*

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



### 9.28 *Setting Camera*

1. Klik bagian *icon None* (*Motion*), maka akan muncul *Windows Motion*, Tambahkan Sesuai dengan urutan. Hilangkan centang “*Automate Thresholds*” dan atur nilai *Threshold* seperti berikut :



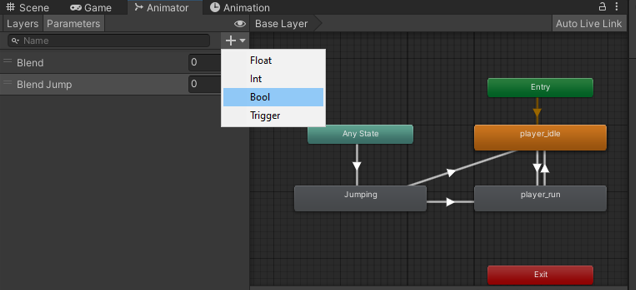
### 9.29 *Blend Tree*

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



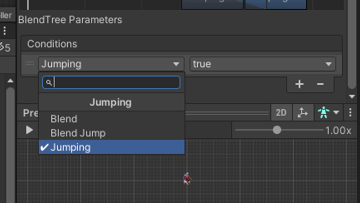
### 9.30 *Make Transition*

1. Klik kanan *Jumping*, pilih *Make Transition* dan arahkan panahnya ke *Player\_idle* dan *Player\_run*. Tambahkan parameter transisi dengan tipe data *Bool* tekan *icon* + dan ubah namanya menjadi “*Jumping*”.



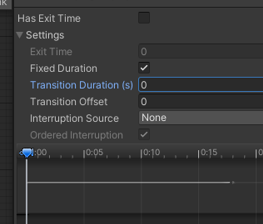
### 9.31 *Camera Setting*

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



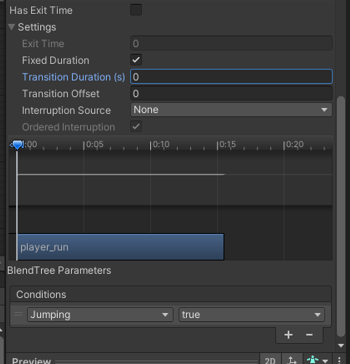
### 9.32 *Jumping Condition*

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



### 9.33 *Settings*

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



### 9.34 *Setting Inspector*

1. Buka *script Player*, dan rubah *source code* pada *void update* pada bagian percabangan *if*.

|  |
| --- |
| if (Input.GetButtonDown("Jump")){          animator.SetBool("Jumping", true);          jump = true;      } |

1. Pada Fungsi *FixedUpdate* tambahkan seperti berikut.

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

1. Rubah *source code* pada *method* *GroundCheck*.

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

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



### 9.35 Aplikasi Dijalankan

1. **Repository GitHub**

[2119061\_PRAK\_ANIGAME](https://github.com/AdiJulia/2118061_PRAK_ANIGAME)

1. **Kuis Pertemuan 9**

*Source Code error* ditandai dengan warna merah.

|  |
| --- |
| 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);  }} |

Perbaikan menjadi :

|  |
| --- |
| void HandleJumpInput()  {  if (Input.GetKeyDown(KeyCode.Space))  {  animator.SetBool("isJumping", true); // Perbaikan: Menambahkan nilai boolean true  rb.AddForce(Vector2.up \* jumpForce, ForceMode2D.Impulse);  }  else if (Input.GetKey(KeyCode.Space))  {  animator.SetBool("isJumping", true); // Perbaikan: Menambahkan nilai boolean true  }  else  {  animator.SetBool("isJumping", false); // Perbaikan: Menambahkan else untuk mengatur isJumping ke false jika tidak ditekan  }  }  void HandleMovementInput()  {  float move = Input.GetAxis("Horizontal");  if (move == 0) // Perbaikan: Menyatakan idle state saat move adalah 0  {  animator.SetBool("isIdle", true);  animator.SetBool("isWalking", false); // Perbaikan: Mengatur isWalking ke false saat idle  }  else  {  animator.SetBool("isIdle", false);  animator.SetBool("isWalking", true); // Perbaikan: Mengatur isWalking ke true saat bergerak  transform.Translate(Vector3.left \* move \* Time.deltaTime);  }  if (move < 0)  {  transform.localScale = new Vector3(-1, 1, 1); // Perbaikan: Mengatur scale saat bergerak ke kiri  }  else if (move > 0)  {  transform.localScale = new Vector3(1, 1, 1); // Perbaikan: Mengatur scale saat bergerak ke kanan  }  } |

Analisis :

Pada metode HandleJumpInput, terdapat kesalahan pada pemanggilan animator.SetBool("isJumping", ); yang tidak memberikan nilai boolean sebagai argumen kedua. Seharusnya nilai boolean (true atau false) harus ditambahkan untuk menunjukkan status animasi isJumping. Selain itu, kondisi else if (Input.GetKey(KeyCode.Space)) juga memiliki kesalahan yang sama karena tidak memberikan nilai boolean pada SetBool.

Pada metode HandleMovementInput, kondisi if (move != 1) seharusnya diperbaiki menjadi move == 0 untuk mendeteksi keadaan idle ketika tidak ada pergerakan horizontal. Pada kondisi ini, animasi isIdle harus diatur ke true dan animasi isWalking harus diatur ke false. Sebaliknya, ketika terdapat pergerakan (move != 0), animasi isIdle harus diatur ke false dan isWalking diatur ke true. Selain itu, terdapat inkonsistensi pada pengaturan transform.localScale. Pada kondisi if (move != 0), skala diatur ke (-4, 1, 1), yang tidak sesuai dengan kondisi di else if (move > 0) di mana skala diatur ke (1, 2, 1). Kedua kondisi ini seharusnya konsisten dalam mengatur skala objek untuk pergerakan ke kiri dan ke kanan.