

LAMPIRAN

A. Script Main Menu

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
public class MainMenu : MonoBehaviour
{
    public Button backButton;
    public Button quitButton;
    // Referensi ke tombol BackButton

    private void Start()
    {
        backButton.gameObject.SetActive(false);
    }
    public void PlayGame()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void GoToSettingMenu()
    {
        SceneManager.LoadScene("SettingMenu");
    }

    public void GoToMainMenu()
    {
        SceneManager.LoadScene("MainMenu");
    }

    public void QuitGame()
    {
        Application.Quit();
    }

    public void ShowBackButton()
    {
        backButton.gameObject.SetActive(true);
    }

    public void HideBackButton()
```

```

    {
        backButton.gameObject.SetActive(false);
    }

    public void ShowQuitButton()
    {
        quitButton.gameObject.SetActive(true);
    }

    public void HideQuitButton()
    {
        quitButton.gameObject.SetActive(false);
    }
}

```

B. Script Character

```

using UnityEngine;
using UnityEngine.InputSystem;

namespace StarterAssets
{
    [RequireComponent(typeof(CharacterController))]
    #if ENABLE_INPUT_SYSTEM &&
    STARTER_ASSETS_PACKAGES_CHECKED
    [RequireComponent(typeof(PlayerInput))]
    #endif
    public class Character : MonoBehaviour
    {
        private ThirdPersonController _thirdPersonController;
        public InputAction attackAction;
        [Header("Controls")]
        public float playerSpeed = 5.0f;
        public float crouchSpeed = 2.0f;
        public float sprintSpeed = 7.0f;
        public float jumpHeight = 0.8f;
        public float gravityMultiplier = 2;
        public float rotationSpeed = 5f;
        public float crouchColliderHeight = 1.35f;
        [Header("Animation Smoothing")]
        [Range(0, 1)]
        public float speedDampTime = 0.1f;
        [Range(0, 1)]
        public float velocityDampTime = 0.9f;
        [Range(0, 1)]
        public float rotationDampTime = 0.2f;
    }
}

```

```

[Range(0, 1)]
public float airControl = 0.5f;
public StateMachine movementSM;
public CombatState combatting;
public AttackState attacking;
public StandingState standing;
[HideInInspector]
public float gravityValue = -9.81f;
[HideInInspector]
public float normalColliderHeight;
[HideInInspector]
public CharacterController controller;
[HideInInspector]
public PlayerInput playerInput;
[HideInInspector]
public Transform cameraTransform;
[HideInInspector]
public Animator animator;
[HideInInspector]
public Vector3 playerVelocity;

// Start is called before the first frame update
private void Start()
{
    _thirdPersonController =
FindObjectOfType<ThirdPersonController>();
    controller = GetComponent<CharacterController>();
    animator = GetComponent<Animator>();
    playerInput = GetComponent<PlayerInput>();
    cameraTransform = Camera.main.transform;

    movementSM = new StateMachine();

    attacking = new AttackState(this, movementSM);
    combatting = new CombatState(this, movementSM);

    movementSM.Initialize(standing);

    normalColliderHeight = controller.height;
    gravityValue *= gravityMultiplier;
}

public void AccessStarterAssetsInputs()
{
    // Panggil metode DoSomethingWithStarterAssetsInputs dari
ThirdPersonController

```

```

        _thirdPersonController.DoSomethingWithStarterAssetsInputs();
    }

    private void Update()
    {
        movementSM.currentState.HandleInput();

        movementSM.currentState.LogicUpdate();
    }

    private void FixedUpdate()
    {
        movementSM.currentState.PhysicsUpdate();
    }
}

```

C. Script StarterAssetsInputs

```

using UnityEngine;

#if ENABLE_INPUT_SYSTEM &&
STARTER_ASSETS_PACKAGES_CHECKED

using UnityEngine.InputSystem;

#endif

namespace StarterAssets
{
    public class StarterAssetsInputs : MonoBehaviour
    {
        [Header("Character Input Values")]

        public Vector2 move;

        public Vector2 look;

        public bool jump;

        public bool sprint;

        public bool attack; // New parameter to handle attack input
    }
}

```

```

[Header("Movement Settings")]

public bool analogMovement;

[Header("Mouse Cursor Settings")]

public bool cursorLocked = true;

public bool cursorInputForLook = true;

// Add a reference to the Animator component

public Animator animator;

public void OnMove(InputValue value)
{
    MoveInput(value.Get<Vector2>());
}

public void OnLook(InputValue value)
{
    if (cursorInputForLook)
    {
        LookInput(value.Get<Vector2>());
    }
}

public void OnJump(InputValue value)
{
    JumpInput(value.isPressed);
}

public void OnSprint(InputValue value)
{
    SprintInput(value.isPressed);
}

```

```

}

public void OnAttack()
{
    // Set the attack flag to true when left mouse button is pressed
    attack = true;
}

public void MoveInput(Vector2 newMoveDirection)
{
    move = newMoveDirection;
    // If attack flag is true, set move parameter to activate combat mode
    if (attack)
    {
        animator.SetFloat("move", 1f);
    }
}

public void DoSomething()
{
    Debug.Log("Doing something from StarterAssetsInputs!");
    // Add your logic or code here
}

public void LookInput(Vector2 newLookDirection)
{
    look = newLookDirection;
}

public void JumpInput(bool newJumpState)

```

```

{
    jump = newJumpState;
}

public void SprintInput(bool newSprintState)
{
    sprint = newSprintState;
}

private void Update()
{
    // Check for left mouse button down
    if (Mouse.current.leftButton.wasPressedThisFrame)
    {
        // Set the attack flag to true when left mouse button is pressed
        animator.SetTrigger("attack");
    }

    // Check for 'R' key press and trigger the "drawWeapon" animation
    if (Keyboard.current.rKey.wasPressedThisFrame)
    {
        // Trigger the "drawWeapon" animation
        animator.SetTrigger("drawWeapon");
    }

    // Check for 'Q' key press and trigger the "sheathWeapon" animation
    if (Keyboard.current.qKey.wasPressedThisFrame)
    {
        // Trigger the "sheathWeapon" animation

```



```

        animator.SetTrigger("sheathWeapon");

        // Optionally, you may want to reset the "drawWeapon" trigger
        animator.ResetTrigger("drawWeapon");
    }

    if (attack && (Keyboard.current.wKey.isPressed ||
Keyboard.current.aKey.isPressed || Keyboard.current.sKey.isPressed ||
Keyboard.current.dKey.isPressed))
    {
        animator.SetTrigger("move");
    }
}
}
}
}

```

D. Script StandingState

```

using UnityEngine;
using StarterAssets;

public class StandingState : State
{
    float gravityValue;

    bool crouch;

    Vector3 currentVelocity;

    bool grounded;

    bool sprint;

    float playerSpeed;

    Vector3 cVelocity;
}

```

```

    public StandingState(Character _character, StateMachine _stateMachine) :
    base(_character, _stateMachine)
    {
        character = _character;
        stateMachine = _stateMachine;
    }
    public override void Enter()
    {
        base.Enter();
        crouch = false;
        sprint = false;
        input = Vector2.zero;
        velocity = Vector3.zero;
        currentVelocity = Vector3.zero;
        gravityVelocity.y = 0;
        playerSpeed = character.playerSpeed;
        grounded = character.controller.isGrounded;
        gravityValue = character.gravityValue;
    }
    public override void PhysicsUpdate()
    {
        base.PhysicsUpdate();
        gravityVelocity.y += gravityValue * Time.deltaTime;
        grounded = character.controller.isGrounded;
        if (grounded && gravityVelocity.y < 0)

```

```

    {
        gravityVelocity.y = 0f;
    }

    currentVelocity = Vector3.SmoothDamp(currentVelocity, velocity, ref
cVelocity, character.velocityDampTime);

    character.controller.Move(currentVelocity * Time.deltaTime * playerSpeed
+ gravityVelocity * Time.deltaTime);

    if (velocity.sqrMagnitude > 0)
    {
        character.transform.rotation =
Quaternion.Slerp(character.transform.rotation,
Quaternion.LookRotation(velocity), character.rotationDampTime);
    }
}

public override void Exit()
{
    base.Exit();

    gravityVelocity.y = 0f;

    character.playerVelocity = new Vector3(input.x, 0, input.y);

    if (velocity.sqrMagnitude > 0)
    {
        character.transform.rotation = Quaternion.LookRotation(velocity);
    }
}
}

```

E. Script HealtCurrent

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

public class HealtCurrent : MonoBehaviour
{
    [SerializeField] private float _maksimalDarah = 10;
    private float _darahSekarang;
    [SerializeField] private HealthBarPlayer _penunjukDarah;
    [SerializeField] private SoundEffect soundEffect;
    public Canvas layarGameOver;
    // Tambahkan deklarasi GameObject projectile dan GameObject player
    public GameObject projectile;
    public GameObject player;
    private void Start()
    {
        _darahSekarang = _maksimalDarah;
        _penunjukDarah.PerbaruiHealthBar(_maksimalDarah, _darahSekarang);
    }
    private void OnTriggerEnter(Collider other)
    {
        // Ubah kondisi menjadi apabila GameObject dengan tag "Bullet" menyentuh
        GameObject "Player"
```

```

if (other.gameObject.tag == "Projectile")
{
    _darahSekarang -= 1; // Kurangi darah saat terjadi trigger dengan player
    _penunjukDarah.PerbaruiHealthBar(_maksimalDarah, _darahSekarang);

    if (_darahSekarang <= 0)
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex
+ 1); // Load the next scene
    }
}

void MatikanInputKeyboard()
{
    // Matikan semua input dari keyboard
    if (Input.anyKeyDown)
    {
        Input.ResetInputAxes();
    }
}
}

```

F. Script MissionComplete

```

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

```

```

using UnityEngine.UI;

public class missionCompletion : MonoBehaviour

{
    public GameObject player;
    public GameObject baju;
    public Text information;
    public GameObject land;
    /* private Renderer bajuRenderer;*/
    private string originalInformationText;
    private void Start()
    {
        originalInformationText = information.text;
        information.enabled = false;
        baju.SetActive(false); // Mulai dengan objek baju terlihat
    }
    private void /*IEnumerator*/ OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Player"))
        {
            Debug.Log("Player touched baju");
            information.enabled = true;
            information.text = "Kembali ke Dita";
            land.SetActive(true);
            Debug.Log("Succes Land Activated");
            baju.SetActive(false);
        }
    }
}

```

```

        /* yield return new WaitForSeconds(1f); // Tunggu selama 5 detik*/
    }
}
}

```

G. Script HealthBarPlayer

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class HealthBarPlayer : MonoBehaviour
{
    public void PerbaruiHealthBar(float maksimalDarah, float darahSekarang)
    {
        _isiHealthBar.fillAmount = darahSekarang / maksimalDarah;
    }
}

```