LAMPIRAN

1. Script Menu Utama

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.XR;
public class MenuUtamaTombol : MonoBehaviour
{
    private void Awake()
    {
        XRSettings.enabled = false;
    }
    public void MulaiGame()
    {
        SceneManager.LoadScene("BelajarScene");
    }
    public void BukaInfo()
    {
        SceneManager.LoadScene("MenuInfo");
    }
```

```
    public void KeluarGame()
    {
        Application.Quit();
    }
}
```

2. **Script Kontroller**

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

public class KarakterKontroller : MonoBehaviour
{
    public float kecepatan = 10f;

    public Transform lantai1;
    public Transform depanLantai1;
    public Transform lantai2;
    public Transform depanLantai2;

    private bool maju = false;
    private bool mundur = false;
    private bool kanan = false;
    private bool kiri = false;
    private bool lihatkanan = false;
    private bool lihatkiri = false;

    // Start is called before the first frame update
    void Awake()
    {
        XRSettings.enabled = true;
    }

    // Update is called once per frame
    void Update()
    {

        Rigidbody rb = GetComponent<Rigidbody>();
```

```
if (Input.GetKeyDown(KeyCode.W))
{
   maju = true;
}
if (Input.GetKeyUp(KeyCode.W))
{
   maju = false;
}
if (Input.GetKeyDown(KeyCode.S))
{
   mundur = true;
}
if (Input.GetKeyUp(KeyCode.S))
{
   mundur = false;
}
if (Input.GetKeyDown(KeyCode.A))
{
   kiri = true;
}
if (Input.GetKeyUp(KeyCode.A))
{
   kiri = false;
}
if (Input.GetKeyDown(KeyCode.D))
{
   kanan = true;
}
if (Input.GetKeyUp(KeyCode.D))
{
   kanan = false;
}
if (Input.GetKeyDown(KeyCode.Q))
{
   lihatkiri = true;
}
if (Input.GetKeyUp(KeyCode.Q))
{
   lihatkiri = false;
}
if (Input.GetKeyDown(KeyCode.E))
{
   lihatkanan = true;
}
if (Input.GetKeyUp(KeyCode.E))
{
```

```
            lihatkanan = false;
        }

        if (maju)
        {
            transform.Translate(new Vector3(0, 0, 1) * kecepatan *
Time.deltaTime);
        }

        if (mundur)
        {
            transform.Translate(new Vector3(0, 0, -1) * kecepatan *
Time.deltaTime);
        }

        if (kiri)
        {
            transform.Translate(new Vector3(-1, 0, 0) * kecepatan *
Time.deltaTime);
        }

        if (kanan)
        {
            transform.Translate(new Vector3(1, 0, 0) * kecepatan *
Time.deltaTime);
        }

        if (lihatkiri)
        {
            Quaternion deltaRotation = Quaternion.Euler(Vector3.down *
kecepatan * 10 * Time.deltaTime);
            rb.MoveRotation(rb.rotation * deltaRotation);
        }

        if (lihatkanan)
        {
            Quaternion deltaRotation = Quaternion.Euler(Vector3.up * kecepatan
* 10 * Time.deltaTime);
            rb.MoveRotation(rb.rotation * deltaRotation);
        }

        if (rb.velocity.magnitude > 5f)
        {

        }
```

```
        if (Input.GetAxis("Horizontal") != 0)
        {
            transform.Translate(new Vector3(Input.GetAxis("Horizontal"), 0, 0) *
kecepatan * Time.deltaTime);
        }
        if (Input.GetAxis("Vertical") != 0)
        {
            transform.Translate(new Vector3(0, 0, Input.GetAxis("Vertical")) *
kecepatan * Time.deltaTime);
        }

        if (Input.GetAxis("Mouse X") != 0)
        {
            transform.Translate(new Vector3(Input.GetAxis("Mouse X"), 0, 0) *
kecepatan * Time.deltaTime);
        }
        if (Input.GetAxis("Mouse Y") != 0)
        {
            transform.Translate(new Vector3(0, 0, Input.GetAxis("Mouse Y")) *
kecepatan * Time.deltaTime);
        }
    }

    private void OnTriggerEnter(Collider other)
    {
        if (other.name == "TeleportKeLantai1")
        {
            this.GetComponent<SphereChanger>().ChangeSphere(lantai1);
        }
        if (other.name == "TeleportKeDepanLantai1")
        {
            this.GetComponent<SphereChanger>().ChangeSphere(depanLantai1);
        }
        if (other.name == "TeleportKeLantai2")
        {
            this.GetComponent<SphereChanger>().ChangeSphere(lantai2);
        }
        if (other.name == "TeleportKeDepanLantai2")
        {
            this.GetComponent<SphereChanger>().ChangeSphere(depanLantai2);
        }
    }
}
```

3. **Kontrol Object**

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

public class Kamera : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

3. Pindah Tempat

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

public class PindahTempat : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

4. Sphere Changer

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

public class SphereChanger : MonoBehaviour {



    //This object should be called 'Fader' and placed over the camera
    GameObject m_Fader;

    //This ensures that we don't mash to change spheres
    bool changing = false;


    void Awake()
    {

        //Find the fader object
        m_Fader = GameObject.Find("Penghilang");

        //Check if we found something
        if (m_Fader == null)
            Debug.LogWarning("No Fader object found on camera.");

    }




    public void ChangeSphere(Transform nextSphere)
    {

        //Start the fading process
        StartCoroutine(FadeCamera(nextSphere));

    }
```

```
IEnumerator FadeCamera(Transform nextSphere)
{

    //Ensure we have a fader object
    if (m_Fader != null)
    {
        //Fade the Quad object in and wait 0.75 seconds
        StartCoroutine(FadeIn(0.5f,
m_Fader.GetComponent<Renderer>().material));
        yield return new WaitForSeconds(0.75f);

        //Change the camera position
        Camera.main.transform.parent.position = nextSphere.position;

        //Fade the Quad object out
        StartCoroutine(FadeOut(0.5f,
m_Fader.GetComponent<Renderer>().material));
        yield return new WaitForSeconds(0.75f);
    }
    else
    {
        //No fader, so just swap the camera position
        Camera.main.transform.parent.position = nextSphere.position;
    }


}


IEnumerator FadeOut(float time, Material mat)
{
    //While we are still visible, remove some of the alpha colour
    while (mat.color.a > 0.0f)
    {
        mat.color = new Color(mat.color.r, mat.color.g, mat.color.b,
mat.color.a - (Time.deltaTime / time));
        yield return null;
    }
}

IEnumerator FadeIn(float time, Material mat)
{
    //While we aren't fully visible, add some of the alpha colour
```

```
        while (mat.color.a < 1.0f)
        {
            mat.color = new Color(mat.color.r, mat.color.g, mat.color.b,
mat.color.a + (Time.deltaTime / time));
            yield return null;
        }
    }


}
```