PADUAN PENGGUNAAN APLIKASI PETA VIRTUAL 3D GEDUNG JURUSAN FISIKA UNIVERSITAS LAMPUNG BERBASIS ANDROID

1. Tampilan Menu Utama

Tampilan menu utama memiliki beberapa tombol.

Tombol 

. digunakan untuk masuk ke dalam *game mode.*

Tombol  (*Setting*) digunakan untuk membuka menu kualitas gambar yang akan ditampilkan saat aplikasi dijalankan.

Tombol  (*About*) digunakan untuk membuka menu informasi dari kegunaan aplikasi dan informasi nama pembuat gedung.

Tombol  *(Tutorials)* digunakan untuk membuka menu yang berisi tutorial atau panduan menggunakan aplikasi.

Tombol 

(*Exit*) digunakan untuk keluar dari aplikasi peta virtual.

2. Tampilan *Game Mode*

Untuk membuka tampilan *game mode*, tekan tombol 

pada menu utama.

Menu ini merupakan tampilan game mode dari aplikasi peta virtual gedung jurusan fisika. Tampilan menu utama memiliki beberapa tombol.

Tombol  *(Analog)* digunakan untuk menggerakan karakter atau *player.*

Tombol  *(Run)* digunakan untuk menggerakkan *player* dengan kemampuan berjalan sendiri tanpa menggunakan *analog*

Tombol  *(Interaction)* digunakan untuk membuka pintu dan menghidupkan/ mematikan lampu ketika *player* mendekati pintu atau saklar lampu.

Tombol  *(Jump)* digunakan untuk *player* melompat keatas.

Tombol  *(Information)* digunakan untuk mengetahui informasi dari ruangan dimana *player* berada.

Tombol  *(Pause)* digunakan untuk masuk ke dalam menu *pause.*

Tombol  *(Back)* digunakan untuk kembali ke menu utama.

3. Tampilan Menu *Pause*

Tombol 

digunakan untuk kembali ke *game mode*.

Tombol 

digunakan untuk membuka menu *navigation*.

Tombol 

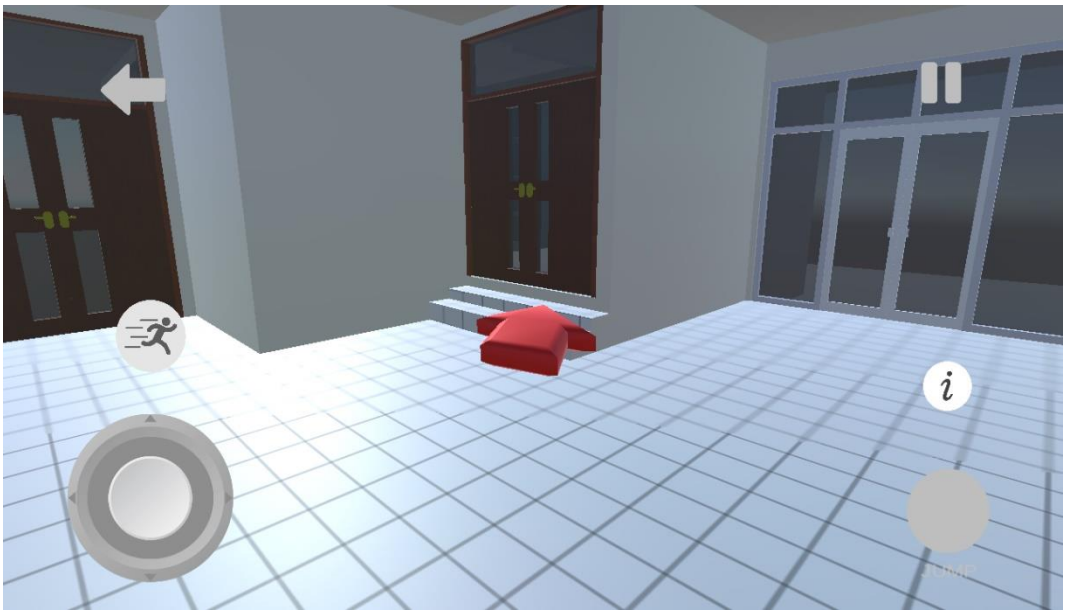
digunakan untuk membuka menu *teleport*.

4. Menu *Navigation*

Untuk membuka menu navigation, tekan tombol 

pada menu *pause*.

Menu *navigation* berisi nama-nama dari ruangan yang dapat menunjukkan arah dari suatu ruangan.



Jika tombol 

di-*klik* maka akan muncul panah berwarna merah yang akan menunjukkan arah ruangan Administrasi. Panah akan hilang ketika pengguna telah sampai ditujuan dan jika panah tidak hilang artinya pengguna belum sampai di tempat tujuannya, oleh karena itu misi *navigation* harus selesai jika anak panah ingin dihilangkan.

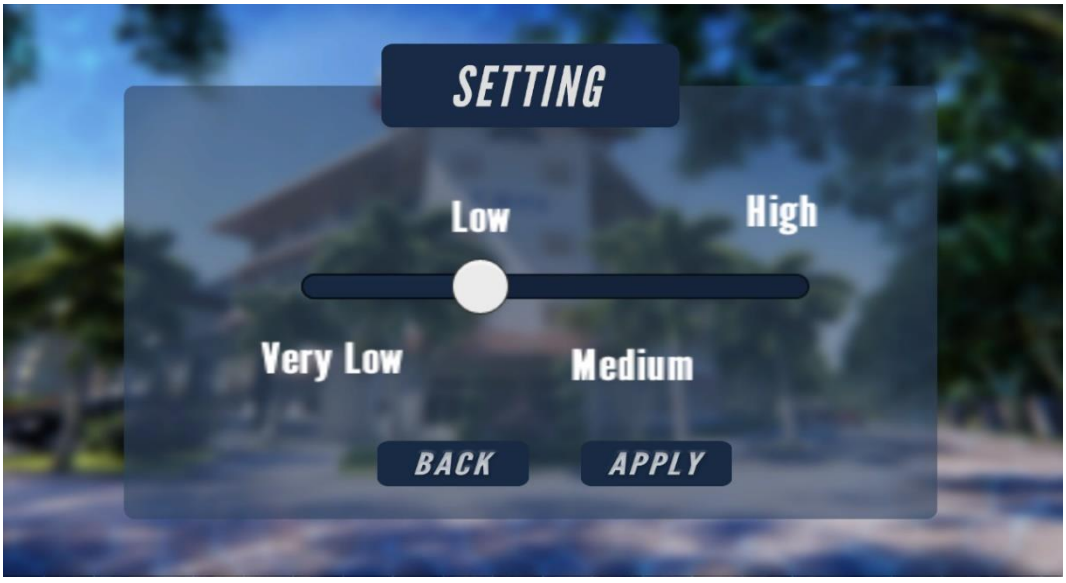
5. **Menu *Teleport***

Untuk membuka menu navigation, tekan tombol 

pada menu *pause*.

Menu *teleport* menampilkan nama-nama dari ruangan dan berguna untuk memindahkan *player* dari satu tempat ke tempat lainnya yang berada didalam lingkup peta virtual dengan cepat sesuai pilihan tempat yang ingin dituju oleh pengguna.

6. **Menu *Setting***



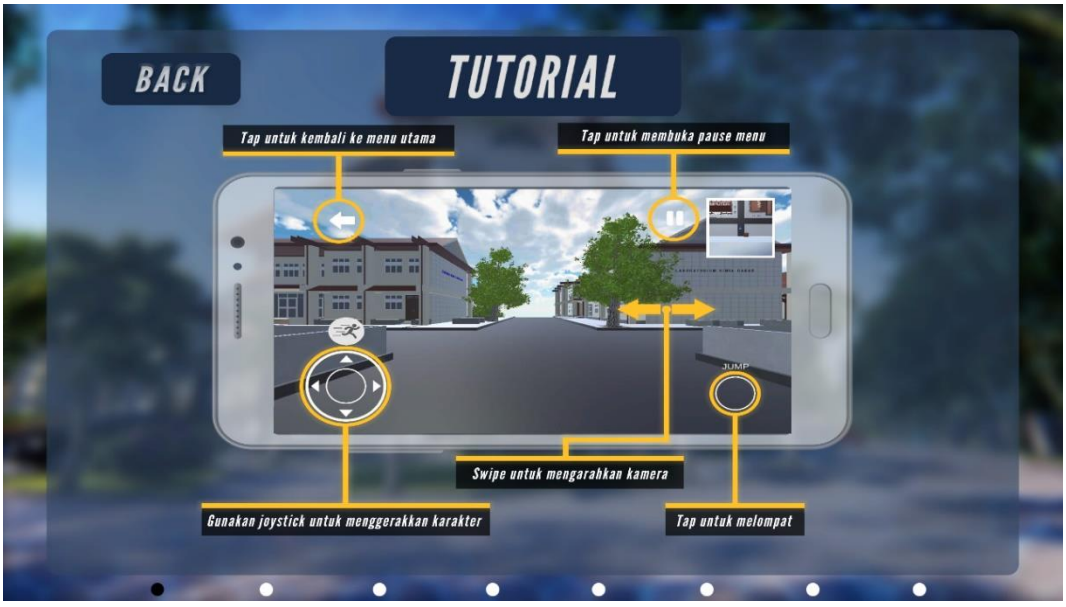
Menu *setting* menampilkan menu untuk menentukan kualitas gambar yang akan ditampilkan saat aplikasi dijalankan. Pada tampilan ini terdapat 4 level kualitas yang

dimulai dari tombol paling kiri yaitu tampilan *very low* atau rendah sampai tombol paling kanan yaitu *high* atau tinggi. Perbedaan yang ditampilkan dari tiap-tiap level yaitu semakin tinggi level nya maka semakin baik gambar yang akan ditampilkan

7. **Menu *About***

Menu *about* menampilkan informasi tentang pengembang aplikasi.

8. **Menu *Tutorials***



Menu *tutorials* menampilkan tutorial atau panduan menggunakan aplikasi. Tutorial yang di tampilakan berupa tampilan aplikasi, tombol interaksi, menu *pause*, menu *teleport* serta menu *navigation.*

Daftar Kode Program Unity 3D

(Bahasa Pemrograman C#)

1. Kode Program Main Menu Aplikasi

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

public class Main_menu : MonoBehaviour {

	public GameObject helpPanel, settingPanel, aboutPanel, exitPanel, loadingImg;

	void Start() {
		Load ();
	}

	public void DeleteSave(){
		PlayerPrefs.DeleteKey ("MyPositionX");
		PlayerPrefs.DeleteKey ("MyPositionY");
		PlayerPrefs.DeleteKey ("MyPositionZ");
		PlayerPrefs.DeleteKey ("MyRotationX");
		PlayerPrefs.DeleteKey ("MyRotationY");
		PlayerPrefs.DeleteKey ("MyRotationZ");
	}

	public void Load ()
	{
		if (PlayerPrefs.HasKey ("QualityIndex")) {
			int QualityLevel = PlayerPrefs.GetInt ("QualityIndex");
			QualitySettings.SetQualityLevel (QualityLevel);
		}
		else
		{
			QualitySettings.SetQualityLevel (1);
		}
	}

	public void Play(){
		DeleteSave ();
		SceneManager.LoadScene("MIPA");
		Time.timeScale = 1;
		loadingImg.SetActive(true);
	}
```

```
	public void help(){
		helpPanel.SetActive (true);
	}

	public void about(){
		aboutPanel.SetActive (true);
	}

	public void Setting(){
			settingPanel.SetActive (true);
		}

	public void exit(){
		exitPanel.SetActive (true);
	}

	public void unAbout(){
			aboutPanel.SetActive (false);
		}

	public void unHelp(){
			helpPanel.SetActive (false);
		}

	public void unSetting(){
			settingPanel.SetActive (false);
		}

	public void noExit(){
			exitPanel.SetActive (false);
		}

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

2. Kode Program Pengaturan Grafis

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

public class Setting : MonoBehaviour {
	public Slider Slider;
	public GameObject Success, Setting;

	void Start() {
		//GetQuality ();
	}

	void Update() {
		GetQuality ();
	}


	public void Save ()
	{
		PlayerPrefs.SetInt ("QualityIndex", (int)Slider.value);
		PlayerPrefs.Save ();
		Success.SetActive (true);
	}

	public void backToMenu ()
	{
		Success.SetActive (false);
		Setting.SetActive (false);
	}

	public void SetQuality ()
	{
		QualitySettings.SetQualityLevel ((int)Slider.value);

	}

	public void GetQuality ()
	{
		int QuaLevel = QualitySettings.GetQualityLevel();
		Slider.value = QuaLevel;

	}
}
```

3. Kode Program Player

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityStandardAssets.Characters.FirstPerson;

public class Controller_FPS : MonoBehaviour {

	public FixedJoystick MoveJoystick;
	public Button mybutton;
	public Sprite imageA;
	public Sprite imageB;
	public FixedButton JumpButton;
	public FixedTouchField TouchField;


	// Use this for initialization
	void Start () {

	}


	// Update is called once per frame
	void Update () {
		var fps = GetComponent<FirstPersonController>();

		fps.RunAxis = MoveJoystick.inputVector;
		fps.JumpAxis = JumpButton.Pressed;
		fps.m_MouseLook.LookAxis = TouchField.TouchDist;

		if (mybutton.image.overrideSprite == imageB) {
			transform.position += transform.forward*2 *
Time.deltaTime;
		}

	}
}
```

4. Kode Program Joystick

```
using UnityEngine;
using UnityEngine.EventSystems;

public class FixedJoystick : Joystick
{
	[Header("Fixed Joystick")]

	Vector2 joystickPosition = Vector2.zero;
	private Camera cam = new Camera();

	void Start()
	{
		joystickPosition = RectTransformUtility.WorldToScreenPoint(cam, background.position);
	}

	public override void OnDrag(PointerEventData eventData)
	{
		Vector2 direction = eventData.position - joystickPosition;
		inputVector = (direction.magnitude > background.sizeDelta.x / 2f) ? direction.normalized : direction / (background.sizeDelta.x / 2f);
		handle.anchoredPosition = (inputVector * background.sizeDelta.x / 2f) * handleLimit;
	}

	public override void OnPointerDown(PointerEventData eventData)
	{
		OnDrag(eventData);
	}

	public override void OnPointerUp(PointerEventData eventData)
	{
		inputVector = Vector2.zero;
		handle.anchoredPosition = Vector2.zero;
	}
}
```

5. Kode Program Arah Player

```
using UnityEngine;
using UnityEngine.EventSystems;

public class FixedTouchField : MonoBehaviour, IPointerDownHandler, IPointerUpHandler
{
	[HideInInspector]
	public Vector2 TouchDist;
	[HideInInspector]
	public Vector2 PointerOld;
	[HideInInspector]
	protected int PointerId;
	[HideInInspector]
	public bool Pressed;

	void Start(){

	}

	void Update()
	{
		if (Pressed)
		{
			if (PointerId >= 0 && PointerId < Input.touches.Length)
			{
				TouchDist = Input.touches[PointerId].position - PointerOld;
				PointerOld = Input.touches[PointerId].position;
			}
			else
			{
				TouchDist = new Vector2(Input.mousePosition.x, Input.mousePosition.y) - PointerOld;
				PointerOld = Input.mousePosition;
			}
		}
		else
		{
			TouchDist = new Vector2();
		}
	}

	public void OnPointerDown(PointerEventData eventData)
	{
		Pressed = true;
		PointerId = eventData.pointerId;
		PointerOld = eventData.position;
```

```
	}

	public void OnPointerUp(PointerEventData eventData)
	{
		Pressed = false;
	}

}
```

6. Kode Program Player Jalan Cepat

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


[RequireComponent(typeof(Button))]
public class disable : MonoBehaviour {
	public Button mybutton;
	public Sprite imageA;
	public Sprite imageB;
	public bool aktif;
	private int counter = 0;

	public void Start () {
		mybutton = GetComponent<Button>();
	}

	public void TombolAktif() {
		aktif = true;
	}

	public void changeButton () {
		counter++;
		if (counter % 2 == 0) {
			mybutton.image.overrideSprite = imageA;

		} else {
			mybutton.image.overrideSprite = imageB;
		}
	}
}
```

7. Kode Program Player Melompat

```
using UnityEngine;
using UnityEngine.EventSystems;

public class FixedButton : MonoBehaviour, IPointerUpHandler, IPointerDownHandler
{
	[HideInInspector]
	public bool Pressed;


	void Start(){
	}

	void Update(){
	}

	public void OnPointerDown(PointerEventData eventData)
	{
		Pressed = true;
	}

	public void OnPointerUp(PointerEventData eventData)
	{
		Pressed = false;
	}
}
```

8. Kode Program Tombol Interaksi

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

public class PindahButton : MonoBehaviour,IPointerDownHandler,IPointerUpHandler {

	public bool aktif;

	IEnumerator LateCall()
	{
		yield return new WaitForSeconds(1f);
		aktif = false;
	}

	public void TombolAktif() {
```

```
                aktif = true;
                StartCoroutine (LateCall());
        }

        public void TombolNonAktif() {
                aktif = false;
        }

        public void OnPointerDown(PointerEventData eventData)
        {
                aktif = true;
        }

        public void OnPointerUp(PointerEventData eventData)
        {
                aktif = false;
        }
}
```

9. Kode Program Navigasi

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

public class Navigation_Fisika : MonoBehaviour {
        int dumping = 6;

        //deklarasi objek target
        public Transform target;
        public Transform Administrasi;
        public Transform Kajur;
        public Transform KPS2;
        public Transform Dosen1;
        public Transform Dosen2;
        public Transform Elektronika;
        public Transform Sidang;
        public Transform Ruang21;
        public Transform Ruang22;
        public Transform Ruang23;
        public Transform Baca;
        public Transform Mushola;
        public Transform Ruang31;
        public Transform Ruang33;
        public Transform Rapat;
        public Transform PintuMasuk;
```

```
	private float x;
	private float y;
	private float z;
	public GameObject arrow, Button, textButton, naviPanel, pausePanel;

	// Update is called once per frame
	void Update () {
		if (arrow.activeSelf == true) {
			Quaternion rotation = Quaternion.LookRotation
(target.position - arrow.transform.position);
			arrow.transform.rotation = Quaternion.Slerp
(arrow.transform.rotation, rotation, Time.deltaTime * dumping);
			x = target.position.x - arrow.transform.position.x;
			y = target.position.y - arrow.transform.position.y;
			z = target.position.z - arrow.transform.position.z;
		}
		if ((x>-3 && x<3)  && (y>-1 && y<1) && (z>-3 && z<3)) {
			arrow.SetActive (false);
		}
	}
	// fungsi navigasi ke arah target, buat sesuai gedung atau ruangan

	public void navAdministrasi(){
		goTarget (Administrasi);
	}
	public void navKajur(){
		goTarget (Kajur);
	}
	public void navKPS2(){
		goTarget (KPS2);
	}
	public void navDosen1(){
		goTarget (Dosen1);
	}
	public void navDosen2(){
		goTarget (Dosen2);
	}
	public void navElektronika(){
		goTarget (Elektronika);
	}
	public void navSidang(){
		goTarget (Sidang);
	}
	public void navRuang21(){
		goTarget (Ruang21);
	}
	public void navRuang22(){
		goTarget (Ruang22);
```

```
	}
	public void navRuang23(){
		goTarget (Ruang23);
	}
	public void navBaca(){
		goTarget (Baca);
	}
	public void navMushola(){
		goTarget (Mushola);
	}
	public void navRuang31(){
		goTarget (Ruang31);
	}
	public void navRuang33(){
		goTarget (Ruang33);
	}
	public void navRapat(){
		goTarget (Rapat);
	}
	public void navPintuMasuk(){
		goTarget (PintuMasuk);
	}

	// fungsi menu
	public void menu(){
		Button.SetActive (true);
		textButton.SetActive (true);
		naviPanel.SetActive (false);
		pausePanel.SetActive (false);
		Time.timeScale = 1;
	}

	//fungsi target
	public void goTarget(Transform targetNum){
		arrow.SetActive (true);
		target = targetNum;
		menu();
	}
}
```

10. Kode Program Teleport

```
using UnityEngine;
using System.Collections;


public class Teleport_Fisika : MonoBehaviour {
	public Transform player ;
	public GameObject Button, textButton, telePanel, pausePanel;

	void Start () {
	}

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

	public void Administrasi(){
		teleport(-6, 3, -28);
		}

	public void Kajur(){
		teleport(-5, 2, -39);
		}

	public void KPS2(){
		teleport(-2, 2, -42);
		}

	public void Dosen1(){
		teleport(1, 2, -43);
		}

	public void Dosen2(){
		teleport(-38, 4, -26);
		}

	public void Elektronika(){
		teleport(-21, 2, -41);
		}

	public void Sidang(){
		teleport(-23, 2, -41);
		}

	public void Ruang21(){
		teleport(-38, 7, -26);
		}
```

```
	public void Ruang22(){
		teleport(-24, 7, -38);
		}

	public void Ruang23(){
		teleport(-10, 7, -35);
		}

	public void Baca(){
		teleport(-14, 7, -39);
		}

	public void Mushola(){
		teleport(-27, 7, -38);
		}
	public void Ruang31(){
		teleport(-38, 13, -26);
		}

	public void Ruang33(){
		teleport(-10, 13, -35);
		}

	public void Rapat(){
		teleport(-17, 13, -38);
		}

	public void PintuMasuk(){
		teleport(-10, 3, -29);
		}


	// Fungsi Teleport
	public void teleport(float xPosition, float yPosition, float zPosition){
		player.position = new Vector3 (xPosition, yPosition, zPosition);
		Button.SetActive (true);
		textButton.SetActive (true);
		telePanel.SetActive (false);
		pausePanel.SetActive (false);
		Time.timeScale = 1f;
		}
}
```

11. Kode Program Membuka dan Menutup Pintu

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

public class Pintu : MonoBehaviour {

	Animator anim;
	public GameObject textInfo;
	public PindahButton pindahButton;
	public GameObject btnInteraksi;
	public GameObject panelInfo;

	void Start(){
		anim = GetComponent<Animator>();
	}

	IEnumerator LateCall()
	{
		yield return new WaitForSeconds(6f);
		anim.enabled = true;
	}

	void OnTriggerEnter(Collider a){
		if (a.name == "Player") {
			textInfo.SetActive(true);
			btnInteraksi.SetActive (true);
			panelInfo.SetActive (true);
		}
	}

	void OnTriggerExit(Collider a){
		if (a.name == "Player") {
			panelInfo.SetActive (false);
			textInfo.SetActive(false);
			btnInteraksi.SetActive (false);
			StartCoroutine (LateCall());
		}
	}

	void OnTriggerStay(Collider a) {
		if (a.name == "Player") {
			if (pindahButton.aktif == true) {
				anim.SetTrigger("OpenDoor");
			}
		}
	}
```

```
        void pauseAnimationEvent(){
                anim.enabled = false;
        }
}
```

12. Kode Program Pintu Terkunci

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

public class Door_Locked : MonoBehaviour {

        public GameObject textButton;
        public PindahButton pindahButton;
        public GameObject btnInteraksi;
        public GameObject roomInfo;
        public GameObject lockedInfo;

        void OnTriggerEnter(Collider a){
                if (a.name == "Player") {
                        textButton.SetActive(true);
                        btnInteraksi.SetActive (true);
                        roomInfo.SetActive(true);
                }
        }

        void OnTriggerExit(Collider a){
                if (a.name == "Player") {
                        textButton.SetActive(false);
                        btnInteraksi.SetActive (false);
                        roomInfo.SetActive(false);
                        lockedInfo.SetActive(false);
                }
        }

        void OnTriggerStay(Collider a) {
                if (a.gameObject.name == "Player") {
                        if (pindahButton.aktif == true) {
                                lockedInfo.SetActive(true);
                        }
                }
        }
}
```

13. Kode Program Info Ruangan

```
using UnityEngine;
using System.Collections;

public class RoomInformation : MonoBehaviour {

	public GameObject info;
	public GameObject button;
	int collide;

	void OnTriggerEnter(Collider a){
		if (a.name == "Player") {
			Debug.Log ("Enter", gameObject);
			button.SetActive (true);
		}
	}

	void OnTriggerExit(Collider a){
		if (a.name == "Player") {
			Debug.Log ("Exit", gameObject);
			button.SetActive (false);
			info.SetActive (false);
		}
	}

	void OnTriggerStay(Collider a) {
		if (a.name == "Player") {
			if (Input.GetKeyDown (KeyCode.E)) {
				if (info.activeSelf) {
					info.SetActive (false);
				} else {
					info.SetActive (true);  }
			}
		}
	}
}
```

14. Kode Program Menghidupkan dan Mematikan Lampu

```
using UnityEngine;
using System.Collections;
public class Lamp : MonoBehaviour {
	public GameObject textOn;
	public GameObject textOff;
	public PindahButton pindahButton;
	public GameObject btnInteraksi;
	public GameObject lamp;

	void OnTriggerEnter(Collider a){
		if (a.name == "Player") {
			btnInteraksi.SetActive (true);
			if (lamp.activeSelf) {
				textOff.SetActive (true);
				} else {
					textOn.SetActive (true);
				}
		}
	}

	void OnTriggerExit(Collider a){
		if (a.name == "Player") {
			btnInteraksi.SetActive (false);
			textOn.SetActive (false);
			textOff.SetActive (false);
		}
	}

	void OnTriggerStay(Collider a) {
		if (a.name == "Player") {
			if (pindahButton.aktif ==  true) {
				if (lamp.activeSelf) {
					lamp.SetActive (false);
					textOff.SetActive (false);
					textOn.SetActive (true);

				} else {
					lamp.SetActive (true);
					textOff.SetActive (true);
					textOn.SetActive (false);
				}
			}
		}
	}
}
```