

```

PINDAH SCENE
using UnityEngine.SceneManagement;
public class pindahScene : MonoBehaviour {
    public void PindahScene(string tujuan) {
        SceneManager.LoadScene(tujuan);
    }
}

MITOS DAN FAKTA
//untuk inisiasi objek UI
using UnityEngine.UI;
//untuk
EventSystem.current.SetSelectedGameObjec t(null)
using UnityEngine.EventSystems;

public class mesinGame : MonoBehaviour {
//digunakan untuk memunculkan di inspector
untuk kelas
[System.Serializable]
class paketSoal {
    [SerializeField] [TextArea(2, 5)] public string pertanyaan;
    [SerializeField] public bool jawaban;
}

//paketSoal[] adl array yang menyimpan soal
[SerializeField] paketSoal[] soal = new paketSoal[1];

//mengenalkan text dan button ke inspector
[SerializeField] Text TxSoal;
[SerializeField] Text TxSkor;
[SerializeField] Button bFakta;
[SerializeField] Button bMitos;

int soalKe = 0; //menyimpan urutan soal
int skor = 0; //menyimpan total skor

void acakSoal() {
//Terdapat perulangan sebanyak array soal
for (int i = 0; i < soal.Length; i++) {
//Tiap iterasi, nilai dari soal[i] akan disimpan pada objek simpanan
    paketSoal simpanan = soal[i];
//mengacak soal
    int acakan = Random.Range(0, soal.Length); //soal yang diacak akan disimpan di soal[i]
    soal[i] = soal[acakan];
//soal hasil acakan dimasukkan ke simpanan
    soal[acakan] = simpanan;
}
}

public void fTombol(bool fakta) {
//apakah jawaban pemain sama dengan yang tersimpan di atribut 'jawaban'
//dari pertanyaan saat ini yang diindeks oleh 'soalKe'
    if (soal[soalKe].jawaban == fakta) {
        skor += 20;
        TxSkor.text = skor.ToString();
    }
//memeriksa apakah masih ada pertanyaan selanjutnya
    if (soalKe < soal.Length - 1)
    {
//menampilkan pertanyaan baru yg di indeks 'soalKe'
        soalKe += 1;
        TxSoal.text =
soal[soalKe].pertanyaan;
    }
//jika sudah tidak ada pertanyaan selanjutnya
    else {
        TxSoal.text = "Selamat Atas Raihan Skor Anda!";
        bFakta.gameObject.SetActive(false);
//menonaktifkan button bFakta
        bMitos.gameObject.SetActive(false);
//menonaktifkan button bMitos
    }
    Debug.Log(skor);
//agar tidak ada yg ter select
    EventSystem.current.SetSelectedGameObje ct(null);
}

// Start is called before the first frame update
void Start()
{
//memanggil fungsi acak soal
    acakSoal();
//menampilkan soal yang telah diacak
    TxSoal.text = soal[soalKe].pertanyaan;
//menampilkan skor
    TxSkor.text = skor.ToString();
}

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

PERPINDAHAN IKAN
public class Perpindahan : MonoBehaviour {
// Atribut
    [SerializeField] float perpindahan;
    [SerializeField] float kecepatan;

    void Update() {
        if (Input.GetKey(KeyCode.RightArrow)) {
//mengontrol visual objek ke kanan atau ke kiri
            GetComponent<SpriteRenderer>().fl ipX = false;
//membuat vector dengan perubahan sumbu x sejauh perpindahan
            Vector3 posisiPerpindahan = new Vector3(posisiPerpindahan, 0f, 0f);
//jika posisi objek < 16
            if (transform.position.x < 16) {
//ni mengubah posisi objek dengan menambahkan vektor posisi
                Perpindahan * kec * Time.deltaTime.
//ni digunakan untuk menggerakkan objek ke kanan dengan kecepatan tertentu.
                transform.position +=
posisiPerpindahan * kecepatan *
Time.deltaTime;
            }
            } else {
//mengatur ulang posisi ke (-16,0,0)
            Vector3 posisi = new Vector3(-16, 0, 0);
            transform.position = posisi;
        }
        if (Input.GetKey(KeyCode.LeftArrow)) {
            GetComponent<SpriteRenderer>().fl ipX = true;
            Vector3 posisiPerpindahan = new Vector3(perpindahan, 0f, 0f);
            if (transform.position.x > -16) {
                transform.position -=
posisiPerpindahan * kecepatan *
Time.deltaTime;
            } else {
                Vector3 posisi = new Vector3(16, 0, 0);
                transform.position = posisi;
            }
        }
    }
}

TIMER
using UnityEngine.UI;
using System;
public class Timer : MonoBehaviour {
//Atribut
//atribut yg mengindikasikan timer sedang aktif atau tidak
    bool aktif;
    //menyimpan jml waktu tersisa dlm detik
    float waktu;
    //mengelola waktu dengan format TimeSpan
    TimeSpan angka;
    //durasi timer berjalan, di atur di inspector
    [SerializeField] float durasi;
}

void tulisTeks(Text teks, int nilai){
    if (nilai < 10)
    {
        teks.text = "0" + nilai;
    } else {
        teks.text = nilai.ToString();
    }
}

void timerBerjalan() {
//mengurangkan waktu dengan Time.deltaTime (hitung mundur)
    waktu -= Time.deltaTime;
    if (waktu > 0) {
        angka =
TimeSpan.FromSeconds(waktu);
        tulisTeks(tMenit, angka.Minutes);
        tulisTeks(tDetik, angka.Seconds);
        tulisTeks(tSub, angka.Milliseconds / 10);
    } else {
//untuk menghentikan timer, karena waktu > 0 sudah tdk terpenuhi
        aktif = false;
    }
}

void Start() {
//saat game dimulai, timer tdk langsung berjalan
    aktif = false;
//waktu diatur dari durasi
    waktu = durasi;
//konversi ke TimeSpan
    angka = TimeSpan.FromSeconds(waktu);
    tulisTeks(tMenit, angka.Minutes);
    tulisTeks(tDetik, angka.Seconds);
    tulisTeks(tSub, angka.Milliseconds / 10);
}

void Update() {
//jika timer aktif maka memanggil timerBerjalan()
    if (aktif == true) {
        timerBerjalan();
    }
}

public void bStartStop() {
//pembalikkan nilai atribut dari true jadi false
//dan sebaliknya
    aktif = !aktif;
}

public void bReset() {
//timer akan dimulai lagi dari awal
    Start();
}

BUTTON INPUT
using UnityEngine.EventSystems;
using UnityEngine.UI;

public class ScButtonInput : MonoBehaviour {
//Game objects
    Image imSoal;
    Text txSkor;
    Button btnA, btnB, btnC, btnD;
    Text txBtnA, txBtnB, txBtnC, txBtnD;

//Attributes
    int soalKe = 0;
    int skor = 0;
    [SerializeField] int skorPerSoal = 20;

//Kelas soal
[System.Serializable] class soal {
    [SerializeField] public Sprite gambarSoal;
    [SerializeField] public string teksA;
    [SerializeField] public string teksB;
    [SerializeField] public string teksC;
    [SerializeField] public string teksD;
    [SerializeField] public string jawaban;
}

//Array kelas soal
[SerializeField] soal[] paketSoal = new soal[1];

//Fungsi menampilkan soal & jawaban
void TampilSoal() {
    imSoal.sprite = paketSoal[soalKe].gambarSoal;
    txBtnA.text = paketSoal[soalKe].teksA;
    txBtnB.text = paketSoal[soalKe].teksB;
    txBtnC.text = paketSoal[soalKe].teksC;
    txBtnD.text = paketSoal[soalKe].teksD;
    txSkor.text = skor.ToString();
}

//Inisialisasi (Intansiasi game object hanya dapat dilakukan di fungsi Start)
void Start() {
//Alternatif cara instansiasi komponen Text tanpa melalui Inspector
    txSkor =
GameObject.Find("TxSkor").GetComponent<Text>();
    imSoal =
GameObject.Find("ImSoal").GetComponent<Image>();
    btnA =
GameObject.Find("BtnA").GetComponent<Button>();
    btnB =
GameObject.Find("BtnB").GetComponent<Button>();
    btnC =
GameObject.Find("BtnC").GetComponent<Button>();
    btnD =
GameObject.Find("BtnD").GetComponent<Button>();
//Alternatif cara instansiasi anak komponen Button (Text) tanpa Inspector
    txBtnA =
btnA.GetComponentInChildren<Text>();
    txBtnB = btnB.GetComponentInChildren<Text>();
    txBtnC = btnC.GetComponentInChildren<Text>();
    txBtnD = btnD.GetComponentInChildren<Text>();
//Alternatif memberikan fungsi tombol melalui script
    btnA.onClick.AddListener(delegate () {
        FunTombol('A');
    });
    btnB.onClick.AddListener(delegate () {
        FunTombol('B');
    });
    btnC.onClick.AddListener(delegate () {
        FunTombol('C');
    });
    btnD.onClick.AddListener(delegate () {
        FunTombol('D');
    });
}

TampilSoal();
}

//Fungsi Tombol
public void FunTombol(char pilihan) {
    if (pilihan == 'A') {
//Sesuai dengan jawaban
        if (paketSoal[soalKe].teksA ==
paketSoal[soalKe].jawaban) {
            skor += skorPerSoal;
        }
    } else if (pilihan == 'B') {
//Sesuai dengan jawaban
        if (paketSoal[soalKe].teksB ==
paketSoal[soalKe].jawaban) {
            skor += skorPerSoal;
        }
    } else if (pilihan == 'C') {
//Sesuai dengan jawaban
        if (paketSoal[soalKe].teksC ==
paketSoal[soalKe].jawaban) {
            skor += skorPerSoal;
        }
    } else if (pilihan == 'D') {
//Sesuai dengan jawaban
        if (paketSoal[soalKe].teksD ==
paketSoal[soalKe].jawaban) {
            skor += skorPerSoal;
        }
    }
}

//Masih ada soal
if (soalKe < paketSoal.Length - 1) {
    soalKe += 1;
}
//Soal sudah habis (disable komponen)
else {
//Menyembunyikan (hide) game object
    imSoal.gameObject.SetActive(false);
    btnA.gameObject.SetActive(false);
    btnB.gameObject.SetActive(false);
    btnC.gameObject.SetActive(false);
    btnD.gameObject.SetActive(false);
}

PENGANTAR UNITY

- Asset store → jual beli asset gim
- Asset package → kumpulan file dan data suatu project gim, Tersimpan sesuai dengan struktur direktori pada project asalnya
- Standard Asset → Kumpulan asset yang umum; Jenis-jenisnya : (2D, Cameras, Characters, CrossPlatformInput, Effects, Environment, ParticleSystems, Prototyping, Utility, Vehicles.); Tersimpan sebagai Package.

```

Keterangan	Type					
Bilangan bulat 8 bit (-128 ... 127)	byte					
Bilangan bulat 16 bit (-32.768 ... 32.767)	short					
Bilangan bulat 32 bit (-2.147.483.648 ... 2.147.483.647)	int					
Bilangan bulat 64 bit (-2 ⁶³ ... 2 ⁶³ -1)	long					
Bilangan desimal 32 bit	float					
Nilai boolean 1 bit (true atau false)	boolean					
Karakter tunggal (ditulis di dalam tanda baca petik)	char					
Karakter tunggal atau jamak (ditulis di dalam tanda baca petik)	string					