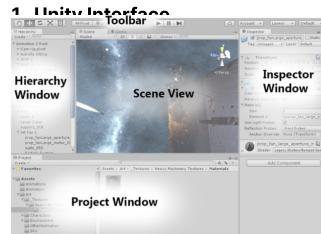


## PENGANTAR UNITY



### 2. Scene view

- Tempat memilih dan memposisikan Game Objects, seperti scenery, karakter, kamera, cahaya, dsb.
- Tempat memilih dan memodifikasi Game Objects
- Navigasi & setting scene
- Move, rotate, Scale, & RectTransform

### 2. Game view

- Hasil render dari kamera
- Merepresentasikan apa yang dilihat oleh pemain
- Seting ukuran layar

### 3. Hierarchy Window

- Berisi daftar seluruh Game Object pada scene yang sedang dipilih

### 4. Project Window

- Tempat mengakses dan mengelola Aset

### 5. Project Window

- Menampilkan informasi detil dari Game Object yang sedang dipilih

### 6. Toolbar window

- Terdiri atas 7 kontrol dasar: Transform tools, Transform Gizmo Toggles, Play/Pause/Step Buttons, Cloud Button, Account Drop-down, Layers Drop-down, LayoutDrop-down

## PENGANTAR #C

**1. Struktur script unity**  
using System.Collections.Generic;  
using UnityEngine;

```
public class Coba : MonoBehaviour {
    void Start(){}
    void Update(){}
}
```

Blok kode method utama (main method)

Blok kode Kelas Nama Kelas harus sama dengan nama file

### 2. Method utama unity

- **Void start ()** → hanya akan dieksekusi sekali pada saat script pertama kali dijalankan
- **void Awake()** → seperti method Void Start(), namun akan dieksekusi sebelum method Void Start()
- **void Update()** → dieksekusi secara terus menerus sesuai dengan durasi frame
- **void FixedUpdate()** → seperti method Void Update(), namun memiliki waktu durasi eksekusi yang lebih singkat
- **void LateUpdate()** → seperti method Void Update(), namun memiliki waktu durasi eksekusi yang lebih panjang

### 3. Deklarasi variabel

- Format deklarasi variabel: `<tipeData> <nama>;`
- Format deklarasi dan inisialisasi variabel: `<tipeData> <nama> = <nilai>;`
- Format manipulasi variabel: `<nama> = <nilai>`

### 4. Atribut variabel

- **[SerializeField]** → menampilkan variabel atau objek di Inspektor
- **[Header(<Teks Judul>)]** → memberikan judul pada proses pengkategorian atribut variabel di Inspektor Panel
- **[Range(<min>, <max>)]** → membatasi rentang bilangan bulat maupun bilangan desimal (<min> nilai terkecil dan <max> nilai terbesar)
- **[TextArea(<min>, <max>)]** → memperbesar form dialog teks (<min> jumlah minimal baris dan <max> jumlah maksimum baris).

### 4. Array

- Deklarasi larak: `> <tipe>[] <nama> = <tipe>[<jumlah>];`
- Deklarasi dan inisialisasi larak: `<tipe>[] <nama> = {<elemen>};`
- Format manipulasi larak: `<nama>[<indeks>] = <nilai>`

### 5. Perulangan

- **Perulangan FOR:** `for(<init>; <kondisi>; <perubahan>) {<pernyataan> }`
- **Perulangan WHILE:** `<init> while(<kondisi>) {<pernyataan> <perubahan> }`

### 6. Method

- **Void Method (Tanpa Nilai Balik):** `void <nama>(<argumen>){<pernyataan> }`
- **Non Void Method (Dengan Nilai Balik):** `<tipeData> <nama>(<argumen>){<pernyataan> return <nilaiBalik> }`

### 7. Overloading method

→ nama fungsi sama, parameternya beda

### TIMER

#### 1. FPS (frame per second)

- Jumlah frame/detik
- dipengaruhi oleh kecepatan komputer dan kec.render
- bersifat dinamis → diset 1 menit bisa saja menjadi lebih dr 1 menit krn tergantung komputer & kec.render
- selisih nilai yg tdk akurat menyebabkan → satuan waktu tdk akurat & pengalaman bermain mengalami perubahan
- Hal tsb dapat diatas dengan → Fungsi `Time.deltaTime` merupakan fungsi untuk menampilkan lama waktu eksekusi tiap frame

### DOCUMENT DESIGN GAME

- dokumen yang dihasilkan pada tahap pre-production dalam Game Development Life Cycle (GDLC)
- Dokumen tsb mendeskripsikan setiap aspek dari gim yang akan dibuat

#### 1. Concept document

- untuk meng-explore ide pengembangan gim dg detail
- Sering digunakan sebagai proposal dalam suatu organisasi
- Dikembangkan oleh desainer gim
- short sales pitch yang terdiri dari 1 s.d. 3 halaman
- Tidak menampilkan ilustrasi secara detail
- Harus memiliki bagian-bagian berikut: Intro;Description;Key Features;Genre, spin, flavor; Platform(s) / market data; Dapat juga memiliki bagian berikut: Background /License info; Concept art

#### 2. High concept

- Kalimat kunci yang mendeskripsikan gim
- Konsep ringkas & cepat

#### 3. Proposal Document

- untuk melakukan penawaran
- Ditujukan kepada pihak publisher gim / pihak ketiga
- konten detail: 5 - 50 halaman
- Sales oriented, Big picture, Polished

#### 4. Game design document

- functional spec → menjelaskan desain gim seperti apa yang akan dibuat (WHAT)
- Mendeskripsikan player's experience dan interaction secara detail
- Artistic feel, Owned by the game designer, A living document, "The Bible"
- Harus mencakup:Game mechanics, User Interface, Visuals, Audio, Story (if any), Level Specs

### APA ITU UNITY?

- Video Game Engine

- Royalty Free, Multi-Platform, & So Many Active Creators

- Real-time Solution

- Berisi architectural vision dan technology yang digunakan

- Engineering detail, Production detail, Owned by tech director or chief engineer
- 10 s.d. 100 pages

#### Tipe Asset pada Unity:

- Image files, 3D Model Files, Meshes & Animations, Audio Files.

### TEXT INPUT

using UnityEngine.UI;

```
public class ScTextInput : MonoBehaviour {
    //Game objects
    Image imikan;
    Text txSoal;
    InputField txJawaban;

    //Attributes
    int indeks = 0;
    float jarakPindah;
    [SerializeField] [TextArea(2, 5)] string[] paketSoal = new string[1];
```

### Fungsi acak soal

```
void AcakSoal() {
    for (int i=0; i < paketSoal.Length-1; i++) {
        int acakan = Random.Range(0, paketSoal.Length);

        string simpanan = paketSoal[i];
        paketSoal[i] = paketSoal[acakan];
        paketSoal[acakan] = simpanan;
    }
}
```

### Fungsi tampil soal

```
void TampilSoal() {
    txSoal.text = paketSoal[indeks];
    txJawaban.text = "";
    txJawaban.Select();
}
```

### Inisialisasi

```
void Start() {
    imikan =
        GameObject.Find("Imikan").GetComponent<Image>();
    txSoal =
        GameObject.Find("TxSoal").GetComponent<Text>();
    txJawaban =
        GameObject.Find("TxJawaban").GetComponent<InputField>();
```

```
;
```

### Menentukan jarak pindah dengan menghitung total (860 px)

```
jarakPindah = 860 / paketSoal.Length;
```

```
AcakSoal();
TampilSoal();
}
```

### Fungsi Update() adalah fungsi utama yang bekerja secara terus menerus

```
void Update() {
    //Menyimpan posisi ikan saat ini
    if (txJawaban.text == paketSoal[indeks]) {
        Vector2 posBaru = new Vector2(posikan.x + jarakPindah, posikan.y);
        //Meng-update posisi ikan
        imikan.rectTransform.anchoredPosition = posBaru;
    }
}
```

```
if (indeks == paketSoal.Length - 1) {
    txSoal.text = "Permainan telah berakhir";
    txJawaban.text = "";
}
```

```
} else {
    indeks += 1;
    TampilSoal();
}}
```

```
• Harus mencakup: Tooling, Art/music/sound/production pipeline, Technology detail (Platform & portability issues, Networking or special tech, Server details), Software engineering info (Major design elements, Key areas of technical risk, Alternatives to risky or expensive sections)
```

### 4. Game design document

- functional spec → menjelaskan desain gim seperti apa yang akan dibuat (WHAT)

- Mendeskripsikan player's experience dan interaction secara detail

- Artistic feel, Owned by the game designer, A living document, "The Bible"

- Harus mencakup:Game mechanics, User Interface, Visuals, Audio, Story (if any), Level Specs

### APA ITU UNITY?

- Video Game Engine

- Royalty Free, Multi-Platform, & So Many Active Creators

- Real-time Solution

- Berisi architectural vision dan technology yang digunakan

- Engineering detail, Production detail, Owned by tech director or chief engineer

- 10 s.d. 100 pages

### ASSET WORKFLOW

- Asset adalah item-item yang digunakan dalam project gim

- Asset dapat dibuat dalam Unity maupun luar Unity untuk kemudian di-import ke dalam Unity

### KEYBOARD INPUT

using UnityEngine.UI;

```
public class ScKeyboardInput : MonoBehaviour {
    //Game objects
    Image imikan;
    //RectTransform adalah variabel yang menyatakan posisi dari suatu objek di Canvas (Screen Space)
    RectTransform poskan;
```

### Attributes

```
int xlkan = 0;
```

```
int ylkan = 0;
```

```
//Vector2 adalah kelas objek yang menyimpan nilai X dan Y (untuk menyimpan posisi)
```

```
Vector2 posAwal;
[SerializeField] float jarakPerpindahan = 100;
```

### Fungsi update posisi

```
void updatePosisi() {
    float updateX = posAwal.x + (xlkan * jarakPerpindahan);
    float updateY = posAwal.y + (ylkan * jarakPerpindahan);
    poskan.anchoredPosition = new Vector2(updateX, updateY);
}
```

### Inisialisasi

```
void Start() {
    //Pertama-tama kita instansiasi objek Image dulu
```

```
imikan =
    GameObject.Find("Imikan").GetComponent<Image>();
```

```
//Lalu kemudian kita instansiasi objek posisi dari Image tersebut
```

```
poskan =
    imikan.GetComponent<RectTransform>();
```

```
//anchoredPosition adalah nilai posisi suatu objek di Canvas (Screen Space)
```

```
posAwal = poskan.anchoredPosition;
```

```
void Update() {
    //Input.GetKeyDown(KeyCode.RightArrow) merupakan fungsi yang menghasilkan nilai true apabila kita menekan tombol tertentu
```

```
if (Input.GetKeyDown(KeyCode.RightArrow)) {
    // 11 adalah jumlah kotak (area pada background maksimal (terdapat 12 kotak (0 - 11))
    if (xlkan < 11) {
        xlkan += 1;
    }
}
```

```
//Scale berfungsi untuk menentukan arah ikan (menghadap ke kanan atau ke kiri)
```

```
poskan.localScale = new Vector2(1, 1);
```

```
updatePosisi();
}
```

```
//ikan akan kembali ke pojok kiri jika sudah mentok di pojok kanan
```

```
else {
    xlkan = -1;
}
```

```
}
```

```
else if
```

```
(Input.GetKeyDown(KeyCode.LeftArrow)) {
    if (xlkan > 0) {
        xlkan -= 1;
    }
}
```

```
poskan.localScale = new Vector2(-1, 1);
```

```
updatePosisi();
}
```

```
//ikan akan kembali ke pojok kanan jika sudah mentok di pojok kanan
```

```
else {
    xlkan = 11;
}
```

```
}
```

```
//Untuk pergerakan naik dan turun tidak perlu ada perubahan arah ikan
```

```
else if (Input.GetKeyDown(KeyCode.UpArrow)) {
    if (ylkan < 5) {
        ylkan += 1;
        updatePosisi();
    }
}
```

```
//ikan akan kembali ke bawah jika sudah mentok di pojok atas
```

```
else {
    ylkan = -1;
}
```

```
}
```

```
else if (Input.GetKeyDown(KeyCode.DownArrow)) {
    if (ylkan > 0) {
        ylkan -= 1;
        updatePosisi();
    }
}
```

```
//ikan akan kembali ke atas jika sudah mentok di pojok bawah
```

```
else {
    ylkan = 6;
}
```

```
}
```