

Baik  berikut penjelasan **baris demi baris** untuk program `UIManager.cs`, yang merupakan bagian dari aplikasi **MiniPhotoshop**.

Kelas ini berfungsi untuk **mengatur antarmuka pengguna (UI)** seperti tombol, slider (*TrackBar*), dan label, agar dapat diaktifkan, dinonaktifkan, atau direset dengan mudah.

Bagian using dan Namespace

```
using System.Collections.Generic;
using System.Windows.Forms;

namespace MiniPhotoshop.Helpers
{
```

- `using System.Collections.Generic;` → Mengizinkan penggunaan tipe koleksi generik seperti `List<Control>`.
 - `using System.Windows.Forms;` → Menyediakan kelas untuk komponen GUI Windows seperti `Control`, `TrackBar`, dan `Label`.
 - `namespace MiniPhotoshop.Helpers` → Tempat penyimpanan kelas pendukung (helper) agar proyek terorganisir.
-

Deklarasi Kelas dan Variabel

```
public class UIManager
{
    private readonly List<Control> _managedControls;
    private readonly TrackBar _trackBarBrightness;
    private readonly Label _lblBrightnessValue;
    private readonly TrackBar _trackBarBlackWhite;
```

- `public class UIManager` → Kelas publik yang berfungsi sebagai **pengatur elemen UI**.
- `private readonly List<Control> _managedControls;`
Menyimpan daftar kontrol (misalnya tombol atau menu) yang dikelola untuk diaktifkan atau dinonaktifkan bersama.
- `private readonly TrackBar _trackBarBrightness;`
Slider untuk mengatur tingkat kecerahan (brightness).
- `private readonly Label _lblBrightnessValue;`
Label untuk menampilkan nilai kecerahan saat ini.
- `private readonly TrackBar _trackBarBlackWhite;`
Slider tambahan untuk mengatur efek *black & white* atau gradasi.

Semua diberi kata kunci `readonly` karena nilainya diatur saat pembuatan objek dan tidak akan berubah.

Konstruktor

```
public UIManager(List<Control> controlsToManage,
                 TrackBar trackBarBrightness,
                 Label lblBrightnessValue,
                 TrackBar trackBarBlackWhite)
{
    _managedControls = controlsToManage;
    _trackBarBrightness = trackBarBrightness;
    _lblBrightnessValue = lblBrightnessValue;
    _trackBarBlackWhite = trackBarBlackWhite;
}
```

- Konstruktor ini digunakan saat kelas `UIManager` dibuat.
- Parameter:
 - `controlsToManage`: daftar kontrol (misal: tombol Undo, Save, dll) yang akan dikelola.
 - `trackBarBrightness`: slider kecerahan.
 - `lblBrightnessValue`: label yang menampilkan nilai kecerahan.
 - `trackBarBlackWhite`: slider untuk konversi hitam-putih.
- Semua parameter disimpan ke variabel internal kelas agar bisa digunakan oleh fungsi lain.

Fungsi `EnableTools()`

```
public void EnableTools()
{
    SetControlsEnabled(true);
}
```

- Mengaktifkan semua kontrol (tombol, slider, dsb) dengan memanggil fungsi `SetControlsEnabled(true)`.
- Biasanya dipanggil setelah gambar berhasil dimuat atau editor siap digunakan.

Fungsi `DisableTools()`

```
public void DisableTools()
{
    SetControlsEnabled(false);
}
```

- Menonaktifkan semua kontrol agar tidak bisa ditekan atau digeser.

- Biasanya dipanggil ketika tidak ada gambar yang sedang diedit, atau setelah operasi selesai dan UI perlu dikunci sementara.
-

Fungsi `ResetControls()`

```
public void ResetControls()
{
    _trackBarBrightness.Value = 0;
    _lblBrightnessValue.Text = "Brightness: 0";
    _trackBarBlackWhite.Value = 2;
}
```

- Mengatur ulang nilai kontrol UI ke kondisi awal:
 - Brightness diatur ke 0.
 - Label teks diperbarui menjadi “Brightness: 0”.
 - Slider *black & white* dikembalikan ke nilai 2 (kemungkinan nilai tengah default).
 - Fungsi ini biasanya dipanggil saat membuka gambar baru atau mereset editor ke kondisi awal.
-

Fungsi `SetControlsEnabled(bool isEnabled)`

```
private void SetControlsEnabled(bool isEnabled)
{
    foreach (var control in _managedControls)
    {
        control.Enabled = isEnabled;
    }
}
```

- Fungsi bantuan internal yang menerima parameter `isEnabled` (true/false).
 - Melakukan iterasi terhadap semua elemen UI di `_managedControls`.
 - Setiap kontrol (Button, TrackBar, dll) diaktifkan (true) atau dinonaktifkan (false).
 - Digunakan oleh `EnableTools()` dan `DisableTools()` untuk efisiensi (agar tidak perlu menulis ulang logika aktivasi).
-

Kesimpulan

Kelas `UIManager` berfungsi sebagai **pengendali status antarmuka pengguna** (UI Controller) dalam aplikasi MiniPhotoshop.

Secara ringkas:

Fungsi	Keterangan
EnableTools()	Mengaktifkan seluruh kontrol UI
DisableTools()	Menonaktifkan seluruh kontrol UI
ResetControls()	Mengembalikan slider dan label ke nilai awal
SetControlsEnabled()	Fungsi internal untuk mengatur status semua kontrol

Dengan adanya kelas ini, pengelolaan tampilan dan interaksi UI menjadi **terpusat dan mudah dikelola**, sehingga kode utama (misalnya di `MainForm`) tidak perlu berisi banyak baris pengaturan tombol satu per satu.
