Tentu, mari kita bedah kode Form1.cs.

Ini adalah **Main UI Class** (Kelas Antarmuka Utama). Jika kode sebelumnya adalah "mesin" di belakang layar, kode ini adalah "Dashboard" atau "Setir"-nya. Kelas ini menangani klik tombol, pergerakan *mouse*, dan mengatur bagaimana tampilan program berubah.

Berikut penjelasannya baris demi baris, dikelompokkan per fitur:

---

1. Import dan Namespace

C#

```
using MiniPhotoshop.Forms;
using MiniPhotoshop.Helpers;
using MiniPhotoshop.Logic;
// ... (using lainnya)
using System.Windows.Forms;

namespace MiniPhotoshop
{
    public partial class Form1 : Form
    {
```

- using ...: Mengimpor semua alat yang dibutuhkan, baik dari .NET (System...) maupun dari folder logika buatan sendiri (MiniPhotoshop.Logic...).
- public partial class Form1 : Form: Mendefinisikan bahwa Form1 adalah sebuah jendela aplikasi (*Form*). Kata kunci partial berarti kode ini hanya sebagian; sebagian lainnya (kode desain UI) dibuat otomatis oleh Visual Studio di file lain (Form1.Designer.cs).

---

2. Deklarasi Variabel Global (Fields)

C#

```
    private readonly ImageEditorService _editorService;
    private readonly UIManager _toolManager;
    private readonly DragDropManager _dragDropManager;
    private readonly OperationManager _operationManager;
    private Dictionary<RadioButton, PictureBox> _thumbnailMap;
    private bool isColorSelectionMode = false;
```

- _editorService: Objek utama untuk memproses gambar (brightness, grayscale, dll).
- _toolManager: Objek pembantu untuk mematikan/menyalakan tombol (agar tombol edit tidak bisa diklik kalau belum ada gambar).
- _dragDropManager: Objek untuk mengurus fitur *drag & drop* gambar ke thumbnail.
- _operationManager: Objek untuk mengurus operasi aritmatika (tambah, kurang, kali, bagi) antar dua gambar.
- _thumbnailMap: Kamus (*Dictionary*) untuk menghubungkan tombol pilihan (*Radio Button*) dengan kotak gambar kecil (*Thumbnail*).
- isColorSelectionMode: "Bendera" (*flag*) sederhana. Jika true, berarti pengguna sedang dalam mode memilih warna (kursor mouse akan berubah).

---

3. Constructor (Fungsi yang Pertama Dijalankan)

C#

```
    public Form1()
    {
        InitializeComponent();
```

- InitializeComponent(): Fungsi wajib Windows Forms untuk menggambar tombol, label, dan kotak gambar sesuai desain Anda.

C#

```
        _thumbnailMap = new Dictionary<RadioButton, PictureBox>
```

```
        {
            { radioButtonThum1, thumbPictureBox1 },
            // ... dst
        };
```

- Membuat pemetaan: "Kalau radioButtonThum1 dipilih, berarti kita bicara soal gambar di thumbPictureBox1".

C#

```
        _editorService = new ImageEditorService();

        var toolsToManage = new List<Control>
        {
            button2, button3, // ... daftar semua tombol
        };
        _toolManager = new UIManager(toolsToManage, ...);
        _toolManager.DisableTools();
```

- Menyiapkan UIManager. Kita berikan daftar tombol mana saja yang harus dikunci (*disabled*) saat aplikasi baru dibuka (karena belum ada gambar).

C#

```
        // DragDropManager BARU dengan DAFTAR thumbnail
        _dragDropManager = new DragDropManager(..., thumbnails, ...);
        _dragDropManager.RegisterDragDropEvents();
```

- Menyiapkan logika *Drag & Drop*. Ini membuat kotak-kotak thumbnail bisa menerima file gambar yang diseret oleh *mouse*.

C#

```
        // Event Handlers Menggunakan Lambda Expression (=>)
        button5.Click += (s, e) => { if (_editorService.IsImageLoaded) pictureBox1.Image =
_editorService.GetChannel(0); }; // Red
        // ... dst
```

- Ini adalah cara singkat menulis fungsi.
- **Artinya:** "Saat tombol 5 diklik, cek dulu ada gambar tidak? Kalau ada, ubah gambar utama (pictureBox1) jadi Channel Merah (0) dari _editorService."

---

4. Reset UI & Load Gambar

C#

```
    private void ResetUIState()
    {
        isColorSelectionMode = false;
        pictureBox1.Cursor = Cursors.Default;
        _toolManager.ResetControls();
    }
```

- Fungsi bersih-bersih. Membatalkan mode seleksi warna, mengembalikan bentuk kursor mouse jadi panah biasa, dan mereset slider (trackbar).

C#

```
    private void button1_Click(object sender, EventArgs e)
    {
        OpenFileDialog openFileDialog = new OpenFileDialog();
        openFileDialog.Filter = "Image Files|*.bmp;*.jpg;..."; // ...

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            var loadedImage = new Bitmap(openFileDialog.FileName);
            _dragDropManager.LoadImageToThumbnail(loadedImage);
```

```
        }
    }
```

- Membuka jendela dialog "Open File" bawaan Windows.
- Jika pengguna memilih file dan klik OK, gambar dimuat ke dalam Bitmap dan dikirim ke _dragDropManager untuk ditampilkan di thumbnail yang sedang aktif.

---

5. Simpan Gambar (Export ke TXT)

C#

```
    private void button2_Click(object sender, EventArgs e)
    {
        // ... Validasi gambar ada atau tidak ...
        Bitmap bmpToSave = new Bitmap(pictureBox1.Image);

        using (SaveFileDialog sfd = new SaveFileDialog())
        {
            sfd.Filter = "File teks (*.txt) | *.txt";
            // ...
            if (sfd.ShowDialog() == DialogResult.OK)
            {
                // ... Try-Catch block ...
                    FileExporter.ExportPixelDataToTxt(bmpToSave, sfd.FileName);
                // ...
            }
        }
    }
```

- Membuka dialog "Save As".
- Memanggil FileExporter.ExportPixelDataToTxt (kode yang kita bahas sebelumnya) untuk menulis data pixel ke file teks yang dipilih pengguna.

---

6. Restore & Histogram

C#

```
    private void buttonRestore_Click(Object sender, EventArgs e)
    {
        if (!_editorService.IsImageLoaded) return;
        pictureBox1.Image = _editorService.GetRestoredImage();
        ResetUIState();
    }
```

- Mengembalikan gambar ke kondisi awal (sebelum diedit) dengan mengambil salinan asli dari _editorService.

C#

```
    private void TampilkanHistogramChannel(int channel)
    {
        // ... Validasi ...
        Bitmap histBmp = _editorService.GetHistogram(channel, ...);
        pictureBoxHistogram.Image = histBmp;
    }
```

- Fungsi bantuan untuk menu Histogram. Menerima input channel (Merah/Hijau/Biru), meminta grafik histogram dari *Service*, dan menampilkannya di kotak pictureBoxHistogram.

---

7. Fitur Interaktif: Seleksi Warna

C#

```
    private void button8_Click(object sender, EventArgs e)
    {
        // ... Validasi ...
        isColorSelectionMode = true;
        pictureBox1.Cursor = Cursors.Cross;
        // ... Tampilkan pesan instruksi ...
    }
```

- Saat tombol "Seleksi" diklik, kita **TIDAK** langsung memproses gambar.
- Kita hanya menyalakan "mode siap-siap" (isColorSelectionMode = true) dan mengubah kursor jadi tanda tambah (Cursors.Cross) agar user tahu mereka harus mengklik gambar.

C#

```
    private void pictureBox1_MouseClick(object sender, MouseEventArgs e)
    {
        if (!isColorSelectionMode || !_editorService.IsImageLoaded) return;

        Point imagePoint = CoordinateHelper.TranslateMouseClickToImagePoint(pictureBox1,
e.Location);
        pictureBox1.Image = _editorService.ApplyColorSelection(imagePoint);

        isColorSelectionMode = false; // Matikan mode setelah selesai
        pictureBox1.Cursor = Cursors.Default;
    }
```

- Inilah yang terjadi saat user mengklik gambar.
- Jika mode seleksi aktif:
  1. Hitung koordinat klik mouse relatif terhadap gambar asli (karena ukuran PictureBox bisa berbeda dengan ukuran asli gambar).
  2. Panggil ApplyColorSelection di service.
  3. Tampilkan hasilnya dan matikan mode seleksi.

---

8. Fitur Slider & Tombol Edit

C#

```
    private void trackBarBlackWhite_Scroll(object sender, EventArgs e)
    {
        // ...
        int step = trackBarBlackWhite.Value;
        // ... Switch case untuk update label teks (Darkest/Normal/Lightest) ...
        pictureBox1.Image = _editorService.ApplyBinarization(step);
    }
```

- Dijalankan setiap kali slider digeser. Mengambil nilai slider (0-4), lalu memanggil fungsi Binarization.

C#

```
    private void trackBarBrightness_Scroll(object sender, EventArgs e)
    {
        // ...
        int adjustment = trackBarBrightness.Value;
        lblBrightnessValue.Text = "Brightness: " + adjustment.ToString();
        pictureBox1.Image = _editorService.ApplyBrightness(adjustment);
    }
```

- Sama seperti di atas, tapi untuk Kecerahan (Brightness). Nilai slider langsung dikirim sebagai parameter penambah/pengurang kecerahan.

---

9. Operasi Aritmatika (Region Terakhir)

C#

```
    #region Operasi Aritmatika & Logika
    private void tambahToolStripMenuItem_Click(object sender, EventArgs e)
```

```
    {
        pictureBox1.Image = _operationManager.PerformAdd();
    }
    // ... dan seterusnya (kurang, kali, bagi, AND, OR, XOR)
```

- Semua fungsi di bagian ini polanya sama:
- Saat menu diklik -> Panggil fungsi yang sesuai di _operationManager.
- _operationManager (yang tidak ditampilkan kodenya di sini, tapi ada di deklarasi atas) bertugas mengambil dua gambar dari thumbnail yang aktif, melakukan matematika piksel, dan mengembalikan hasilnya untuk ditampilkan di pictureBox1.

Kesimpulan:
Form1.cs bertugas sebagai pengatur lalu lintas. Dia tidak melakukan perhitungan matematika gambar secara langsung (itu tugas _editorService), tetapi dia memastikan input pengguna (klik, geser, drag-drop) diteruskan ke logika yang tepat dan hasilnya ditampilkan kembali ke layar. Catatan, untuk semua kode operasi aritmatika, logika, dan konstanta, itu semua hanya memanggil fungsi atau method yang ada di masing-masing kelas.