Tentu, ini adalah penjelasan mendetail untuk setiap bagian dari kode Form1.cs Anda.

Kode ini membuat aplikasi Windows Forms sederhana yang dapat memuat gambar, menampilkannya, dan melakukan beberapa operasi dasar pengolahan citra seperti melihat *channel* warna (Merah, Hijau, Biru), konversi *grayscale*, dan menampilkan histogram.

Berikut adalah rinciannya per bagian:

Bagian Atas (Namespace & Fields)

C#

```
using System.Drawing.Drawing2D;

namespace MiniPhotoshop
{
    public partial class Form1 : Form
    {
        private Bitmap originalImage = null;
        private int[,,] imageData3D = null;
```

- using System.Drawing.Drawing2D;: Mengimpor *namespace* yang diperlukan untuk operasi menggambar 2D tingkat lanjut, seperti SmoothingMode yang Anda gunakan saat menggambar histogram.
- namespace MiniPhotoshop: Mendefinisikan "wadah" logis untuk kode Anda.
- public partial class Form1 : Form: Mendeklarasikan kelas utama aplikasi Anda, Form1, yang merupakan sebuah *Window* (Formulir).
- private Bitmap originalImage = null;: Ini adalah variabel (disebut juga **field**) untuk menyimpan gambar asli yang dimuat oleh pengguna. Ini digunakan untuk fungsi "Restore" agar Anda bisa kembali ke gambar awal setelah melakukan perubahan.
- private int[,,] imageData3D = null;: Ini adalah *field* kunci Anda. Ini adalah **array 3D** yang dirancang untuk menyimpan data piksel gambar.
  - Dimensi pertama adalah **lebar (x)**.
  - Dimensi kedua adalah **tinggi (y)**.
  - Dimensi ketiga adalah **channel warna (0-3)**.
    - [x, y, 0] = Nilai **Merah (Red)** di piksel (x, y)
    - [x, y, 1] = Nilai **Hijau (Green)** di piksel (x, y)
    - [x, y, 2] = Nilai **Biru (Blue)** di piksel (x, y)
    - [x, y, 3] = Nilai **Grayscale** di piksel (x, y)

Konstruktor (Form1())

C#

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

        // event handler untuk tombol Show Red, Green, Blue
        button5.Click += (s, e) => ShowOnlyColorChannel(0); // Red
        button6.Click += (s, e) => ShowOnlyColorChannel(1); // Green
        button7.Click += (s, e) => ShowOnlyColorChannel(2); // Blue

        // event handler menu histogram
        openToolStripMenuItem.Click += (s, e) => TampilkanHistogramChannel(0); // Histogram R
        saveToolStripMenuItem.Click += (s, e) => TampilkanHistogramChannel(1); // Histogram G
        histogramBToolStripMenuItem.Click += (s, e) => TampilkanHistogramChannel(2); //
Histogram B
        histogramGrToolStripMenuItem.Click += (s, e) => TampilkanHistogramChannel(3); //
Histogram Grayscale
    }
```

- public Form1(): Ini adalah **konstruktor** kelas. Metode ini dijalankan *satu kali* saat formulir pertama kali dibuat.
- InitializeComponent();: Fungsi standar WinForms yang memuat semua komponen visual (tombol, *picture box*, menu) yang Anda desain di *Form Designer*.
- button5.Click += ...: Ini menghubungkan peristiwa Click dari button5 ke sebuah *lambda expression*. Saat button5 diklik, ia akan memanggil metode ShowOnlyColorChannel dengan parameter 0 (untuk Merah).
- button6.Click & button7.Click: Sama seperti di atas, tetapi untuk button6 (memanggil channel 1, Hijau) dan button7 (memanggil channel 2, Biru).
- openToolStripMenuItem.Click += ...: Ini menghubungkan item menu openToolStripMenuItem ke fungsi TampilkanHistogramChannel(0) (Histogram Merah).
- saveToolStripMenuItem.Click += ...: Menghubungkan item menu saveToolStripMenuItem ke TampilkanHistogramChannel(1) (Histogram Hijau).
- histogramBToolStripMenuItem.Click += ...: Menghubungkan item menu histogramBToolStripMenuItem ke TampilkanHistogramChannel(2) (Histogram Biru).
- histogramGrToolStripMenuItem.Click += ...: Menghubungkan item menu histogramGrToolStripMenuItem ke TampilkanHistogramChannel(3) (Histogram Grayscale).

Event Handler Tombol

button1_Click (Load Gambar)

C#

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

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            originalImage = new Bitmap(openFileDialog.FileName);

            // Pindahkan data gambar ke array 3D
            LoadImageDataTo3DArray(originalImage);

            // Buat Bitmap baru dari array 3D dan tampilkan di pictureBox1
            Bitmap bmpFromArray = CreateBitmapFrom3DArray();
            if (bmpFromArray != null)
            {
                pictureBox1.Image = bmpFromArray;
                pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;
            }
        }
    }
```

- Metode ini dijalankan saat button1 (Tombol Load) diklik.
- OpenFileDialog: Membuat kotak dialog "Buka File".
- openFileDialog.Filter: Membatasi file yang bisa dipilih hanya file gambar.
- if (openFileDialog.ShowDialog() == DialogResult.OK): Membuka dialog. Jika pengguna memilih file dan mengklik "OK"...
- originalImage = new Bitmap(openFileDialog.FileName);: Memuat file gambar yang dipilih ke dalam variabel originalImage sebagai *backup*.
- LoadImageDataTo3DArray(originalImage);: Memanggil metode *custom* Anda untuk memproses gambar dan menyimpannya ke imageData3D.
- Bitmap bmpFromArray = CreateBitmapFrom3DArray();: Memanggil metode *custom* Anda untuk membuat gambar *baru* berdasarkan data di imageData3D.
- pictureBox1.Image = bmpFromArray;: Menampilkan gambar baru tersebut di pictureBox1.
- pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;: Mengatur agar gambar pas di dalam *picture box*.

button2_Click (Simpan Nilai Piksel ke Teks)

C#

```
private void button2_Click(object sender, EventArgs e)
{
    // ... (Pemeriksaan null) ...
    Bitmap bmp = new Bitmap(pictureBox1.Image);
    // ... (Setup SaveFileDialog) ...
    if (sfd.ShowDialog() == DialogResult.OK)
    {
        // ... (Try-catch block) ...
        var sb = new System.Text.StringBuilder();
        for (int y = 0; y < bmp.Height; y++)
        {
            for (int x = 0; x < bmp.Width; x++)
            {
                Color pixelColor = bmp.GetPixel(x, y);
                sb.Append($"{x},{y}: R={pixelColor.R}, G={pixelColor.G}, B={pixelColor.B}\n");
            }
        }
        System.IO.File.WriteAllText(sfd.FileName, sb.ToString());
        // ... (MessageBox sukses/error) ...
    }
}
```

- Metode ini dijalankan saat button2 (Tombol Save) diklik.
- Ia mengambil gambar yang *saat ini* ada di pictureBox1.
- Ia membuka SaveFileDialog yang disaring untuk file .txt.
- var sb = new System.Text.StringBuilder();: Menggunakan StringBuilder yang jauh lebih efisien untuk membangun string panjang daripada konkatenasi string biasa.
- *Loop* for (y ...) dan for (x ...): Melakukan iterasi melalui setiap piksel gambar.
- Color pixelColor = bmp.GetPixel(x, y);: Mendapatkan nilai warna (R, G, B) dari piksel di koordinat (x, y).
- sb.Append(...): Menambahkan baris teks yang berisi koordinat (x, y) dan nilai R, G, B-nya ke StringBuilder.
- System.IO.File.WriteAllText(...): Menulis seluruh isi StringBuilder ke file teks yang dipilih pengguna.

buttonRestore_Click

C#

```
    private void buttonRestore_Click(Object sender, EventArgs e)
    {
        if(originalImage != null)
        {
            pictureBox1.Image = (Bitmap)originalImage.Clone();
        }
    }
```

- Memeriksa apakah originalImage sudah diisi (yaitu, pengguna sudah memuat gambar).
- pictureBox1.Image = (Bitmap)originalImage.Clone();: Mengatur gambar di pictureBox1 kembali ke gambar asli.
- .Clone(): Ini penting. Ini membuat *salinan* dari gambar asli. Jika Anda tidak menggunakan .Clone(), perubahan apa pun pada pictureBox1.Image di masa depan juga dapat memengaruhi originalImage.

buttonGrayscale_Click

C#

```
    private void buttonGrayscale_Click(Object sender, EventArgs e)
    {
        if (imageData3D == null)
        {
            return;
        }
        ShowOnlyColorChannel(3);
    }
```

- Saat tombol Grayscale diklik, metode ini hanya memanggil ShowOnlyColorChannel dengan parameter 3. Berdasarkan logika Anda, channel 3 adalah channel Grayscale.

Metode Logika Inti

LoadImageDataTo3DArray

C#

```
    private void LoadImageDataTo3DArray(Bitmap bmp)
    {
        // ...
        imageData3D = new int[width, height, 4]; // 4 channel: R, G, B, Grayscale
        for (int y = 0; y < height; y++)
        {
            for (int x = 0; x < width; x++)
            {
                Color pixelColor = bmp.GetPixel(x, y);
                int r = pixelColor.R;
                int g = pixelColor.G;
                int b = pixelColor.B;
                int gray = (int)(0.3 * r + 0.59 * g + 0.11 * b);

                imageData3D[x, y, 0] = r;
                imageData3D[x, y, 1] = g;
                imageData3D[x, y, 2] = b;
                imageData3D[x, y, 3] = gray; // grayscale disimpan di channel ke-3
            }
        }
    }
```

- Metode ini adalah "jantung" dari penyimpanan data Anda.
- Ia mengambil Bitmap sebagai input.
- imageData3D = new int[width, height, 4];: Menginisialisasi array 3D dengan ukuran yang tepat.
- *Loop* for (y ...) dan for (x ...): Iterasi per piksel.
- Color pixelColor = bmp.GetPixel(x, y);: Membaca data piksel.
- int gray = (int)(0.3 * r + 0.59 * g + 0.11 * b);: Menghitung nilai *grayscale* (luminance) menggunakan formula standar.
- imageData3D[x, y, 0] = r;: Menyimpan nilai R, G, B, dan Gray ke dalam "slot" yang sesuai di array 3D untuk piksel (x, y) tersebut.

CreateBitmapFrom3DArray

C#

```
    private Bitmap CreateBitmapFrom3DArray(int channel = -1)
    {
        // ... (Pemeriksaan null) ...
        // ... (Loop y dan x) ...
                int r = imageData3D[x, y, 0];
                int g = imageData3D[x, y, 1];
                int b = imageData3D[x, y, 2];
                int gray = imageData3D[x, y, 3];

                if (channel == 0)     { bmp.SetPixel(x, y, Color.FromArgb(r, 0, 0)); } // hanya red
                else if (channel == 1) { bmp.SetPixel(x, y, Color.FromArgb(0, g, 0)); } // hanya green
                else if (channel == 2) { bmp.SetPixel(x, y, Color.FromArgb(0, 0, b)); } // hanya blue
                else if (channel == 3) { bmp.SetPixel(x, y, Color.FromArgb(gray, gray, gray)); } //
grayscale
                else                   { bmp.SetPixel(x, y, Color.FromArgb(r, g, b)); } // tampilkan full color
        // ...
        return bmp;
    }
```

- Metode ini kebalikan dari LoadImageDataTo3DArray. Ia *membangun* Bitmap *dari* data di imageData3D.
- int channel = -1: Ini adalah **parameter opsional**. Jika Anda memanggil CreateBitmapFrom3DArray() tanpa parameter (seperti di button1_Click), channel akan bernilai -1.
- Ia membaca nilai r, g, b, gray dari array 3D.
- Logika if-else if:
  - Jika channel adalah 0, 1, atau 2, ia membuat piksel *berwarna* (merah murni, hijau murni, atau biru murni) menggunakan nilai dari channel tersebut.
  - Jika channel adalah 3, ia membuat piksel *abu-abu* menggunakan nilai gray.
  - Jika channel adalah -1 (default), ia membuat piksel *berwarna penuh* menggunakan r, g, b.

ShowOnlyColorChannel

C#

```
    private void ShowOnlyColorChannel(int channel)
    {
        // ... (Pemeriksaan null dan loop) ...
                int r = 0, g = 0, b = 0;

                if (channel == 0)     { r = imageData3D[x, y, 0]; }
                else if (channel == 1) { g = imageData3D[x, y, 1]; }
                else if (channel == 2) { b = imageData3D[x, y, 2]; }
                else if (channel == 3) { r = g = b = imageData3D[x, y, 3]; } // grayscale

                bmp.SetPixel(x, y, Color.FromArgb(r, g, b));
        // ...
        pictureBox1.Image = bmp;
    }
```

- Metode ini mirip dengan CreateBitmapFrom3DArray tetapi dengan logika yang sedikit berbeda untuk visualisasi. Ini digunakan oleh tombol R, G, B, dan Grayscale.
- Ia menginisialisasi r, g, b ke 0.
- if-else if:
  - Jika channel == 0, ia *hanya* mengisi r. g dan b tetap 0. Hasilnya adalah gambar nuansa merah.
  - Jika channel == 1, ia *hanya* mengisi g. r dan b tetap 0. Hasilnya adalah gambar nuansa hijau.
  - Jika channel == 2, ia *hanya* mengisi b. r dan g tetap 0. Hasilnya adalah gambar nuansa biru.
  - Jika channel == 3, ia mengisi r, g, *dan* b dengan nilai gray yang sama. Ini adalah cara yang benar untuk membuat piksel abu-abu.
- pictureBox1.Image = bmp;: Setelah selesai membangun *bitmap* baru, ia langsung menampilkannya di pictureBox1.

Metode Histogram (TampilkanHistogramChannel)

Ini adalah metode yang paling kompleks.

C#

```
    private void TampilkanHistogramChannel(int channel)
    {
        // ... (Pemeriksaan null) ...
        Bitmap bmp = new Bitmap(pictureBox1.Image);
        int[] hist = new int[256];
```

- Ia mengambil gambar yang *saat ini* ada di pictureBox1.
- int[] hist = new int[256];: Membuat array untuk "menghitung". hist[0] akan menyimpan jumlah piksel dengan intensitas 0, hist[1] jumlah piksel dengan intensitas 1, dst.

C#

```
        // Hitung histogram dari bitmap di pictureBox1
        for (int y = 0; y < bmp.Height; y++)
        {
            for (int x = 0; x < bmp.Width; x++)
            {
                Color pixelColor = bmp.GetPixel(x, y);
                int val = 0;
                switch (channel) { /* ... */ }
                hist[val]++;
            }
        }
```

- **Perhitungan Histogram**:
  - Loop melalui setiap piksel dari gambar di pictureBox1.
  - switch (channel): Menentukan nilai mana yang akan dihitung.
    - case 0: val = pixelColor.R; (ambil nilai Merah)
    - case 1: val = pixelColor.G; (ambil nilai Hijau)
    - case 2: val = pixelColor.B; (ambil nilai Biru)
    - case 3: val = (int)(0.3 * ...); (hitung nilai Grayscale dari piksel saat ini)
  - hist[val]++;: Menambahkan 1 ke "keranjang" yang sesuai. Jika val adalah 150, hist[150] akan bertambah satu.

C#

```
        Bitmap histBmp = new Bitmap(width, height);
        using (Graphics g = Graphics.FromImage(histBmp))
        {
            g.Clear(Color.Black);
            g.SmoothingMode = SmoothingMode.AntiAlias;
            // ... (Definisi Pen, Brush, Font) ...
            int max = hist.Max();
            // ...
```

- **Persiapan Menggambar**:
  - Membuat Bitmap baru (histBmp) seukuran pictureBoxHistogram.
  - using (Graphics g = ...): Cara standar untuk mendapatkan "kanvas" (Graphics) untuk menggambar di atas histBmp.
  - g.Clear(Color.Black);: Mengisi latar belakang histogram dengan warna hitam.
  - g.SmoothingMode = SmoothingMode.AntiAlias;: Membuat garis terlihat lebih halus.
  - int max = hist.Max();: Menemukan nilai *tertinggi* dalam array hist. Ini adalah *puncak* histogram dan digunakan untuk menskalakan grafik agar pas secara vertikal.

C#

```
            // Gambar sumbu X dan Y dengan margin baru
            // ... (g.DrawLine untuk sumbu) ...

            // Label sumbu X (0 sampai 255)
            // ... (Loop untuk label X) ...

            // Label sumbu Y (jumlah pixel)
            // ... (Loop untuk label Y) ...
```

- **Menggambar Sumbu & Label**:
  - Kode ini menggambar garis untuk sumbu X dan Y dengan margin.
  - Kemudian ia masuk ke *loop* (misalnya, i dari 0 sampai 5) untuk menggambar label.
  - Untuk sumbu X, ia menghitung posisi untuk label "0", "51", "102", ..., "255".
  - Untuk sumbu Y, ia menghitung posisi untuk label "0" sampai nilai max.
  - g.DrawString(...): Menulis teks label di kanvas.

C#

```
            // Pilih warna garis histogram sesuai channel
            Pen pen;
            switch (channel) { /* ... (Pilih Pens.Red, Pens.Lime, etc.) ... */ }

            // Gambar histogram dengan offset
            PointF[] points = new PointF[256];
            for (int i = 0; i < 256; i++)
            {
                float xPos = marginLeft + (i / 255f) * xAxisLength;
                float yPos = height - marginBottom - ((hist[i] / (float)max) * (height - 40));
                points[i] = new PointF(xPos, yPos);
            }
            g.DrawLines(pen, points);
        }
        pictureBoxHistogram.Image = histBmp;
    }
```

- **Menggambar Grafik Histogram**:
  - Memilih warna Pen (Merah, Hijau/Lime, Biru, Putih) berdasarkan channel.
  - Membuat array PointF[256] untuk menampung 256 titik dari grafik.
  - *Loop* for (int i = 0; i < 256; i++):
    - float xPos: Menghitung posisi X. Ini memetakan i (0-255) ke lebar area gambar (dari marginLeft ke width - 10).
    - float yPos: Menghitung posisi Y. Ini adalah bagian terpenting.
      - (hist[i] / (float)max): Menghitung rasio/persentase ketinggian (misal 0.5 jika hist[i] setengah dari max).
      - ... * (height - 40): Menskalakan rasio itu ke ketinggian area gambar.
      - height - marginBottom - ...: Membalikkan nilai Y (karena di komputer, Y=0 ada di *atas*) dan memposisikannya relatif terhadap sumbu X (di height - marginBottom).
  - g.DrawLines(pen, points);: Menggambar satu garis yang menghubungkan semua 256 titik.
- pictureBoxHistogram.Image = histBmp;: Akhirnya, menampilkan Bitmap histogram yang sudah jadi di pictureBoxHistogram.