

Baik, berikut penjelasan **baris demi baris** dari program C# tersebut 🙋

✿ Bagian 1: Import Namespace

```
using System.Drawing.Drawing2D;
using System.Windows.Forms;
```

- `using System.Drawing.Drawing2D;` → Menyediakan kelas dan fungsi untuk manipulasi gambar 2D seperti transformasi koordinat, penskalaan, dan rotasi.
 - `using System.Windows.Forms;` → Memungkinkan penggunaan komponen antarmuka Windows Forms seperti `PictureBox`, `Button`, dll.
-

🧱 Bagian 2: Namespace & Kelas

```
namespace MiniPhotoshop.Logic.Helpers
{
    public static class CoordinateHelper
```

- `namespace MiniPhotoshop.Logic.Helpers` → Mengelompokkan kelas dalam proyek *MiniPhotoshop* agar terorganisir (bagian logika bantu atau helper).
 - `public static class CoordinateHelper` → Kelas bersifat *static*, artinya tidak perlu membuat objek untuk menggunakannya (langsung dipanggil lewat nama kelas).
-

🎯 Bagian 3: Fungsi Utama

```
public static Point TranslateMouseClickedToImagePoint(PictureBox pb, Point
mouseLocation)
```

- Fungsi `TranslateMouseClickedToImagePoint` digunakan untuk **mengubah koordinat klik mouse pada PictureBox menjadi koordinat sebenarnya pada gambar** di dalamnya.
 - Parameter:
 - `pb` → objek `PictureBox` tempat gambar ditampilkan.
 - `mouseLocation` → posisi klik mouse di dalam `PictureBox`.
 - Mengembalikan nilai `Point (x, y)` yang mewakili titik klik di dalam gambar asli.
-

🚫 Pemeriksaan Awal

```
if (pb.Image == null)
{
    return mouseLocation;
}
```

- Jika PictureBox belum memiliki gambar (`pb.Image == null`), maka posisi klik dikembalikan apa adanya.
 - Tujuannya untuk mencegah error saat mencoba mengakses ukuran gambar yang belum ada.
-

Mengambil Ukuran PictureBox dan Gambar

```
float pWidth = pb.ClientSize.Width;
float pHeight = pb.ClientSize.Height;

float iWidth = pb.Image.Width;
float iHeight = pb.Image.Height;
```

- Menyimpan **lebar dan tinggi** PictureBox (`pWidth, pHeight`) dan gambar (`iWidth, iHeight`) ke dalam variabel bertipe `float`.
 - Tipe `float` digunakan agar mudah melakukan operasi perbandingan rasio dan skala nanti.
-

Menghitung Rasio Aspek (Aspect Ratio)

```
float pRatio = pWidth / pHeight;
float iRatio = iWidth / iHeight;
```

- `pRatio` → rasio lebar terhadap tinggi dari PictureBox.
 - `iRatio` → rasio lebar terhadap tinggi dari gambar.
 - Rasio ini menentukan apakah gambar di-*stretch*, *fit*, atau diberi padding saat ditampilkan.
-

Inisialisasi Variabel Ukuran dan Offset

```
float newWidth = iWidth;
float newHeight = iHeight;
float offsetX = 0;
float offsetY = 0;
```

- Menyimpan ukuran gambar yang sudah disesuaikan (nanti dihitung ulang).
- `offsetX` dan `offsetY` digunakan untuk menghitung **jarak tepi gambar dari batas PictureBox** (karena gambar bisa ditengah dan tidak memenuhi seluruh area).

Penyesuaian Skala Gambar

```
if (pRatio > iRatio)
{
    newWidth = pHeight * iRatio;
    newHeight = pHeight;
    offsetX = (pWidth - newWidth) / 2;
}
else
{
    newWidth = pWidth;
    newHeight = pWidth / iRatio;
    offsetY = (pHeight - newHeight) / 2;
}
```

- Kondisi ini menentukan apakah gambar lebih “lebar” atau “tinggi” dibanding PictureBox.
- Jika **PictureBox lebih lebar** ($pRatio > iRatio$):
 - Gambar disesuaikan berdasarkan tinggi PictureBox ($newHeight = pHeight$).
 - Lebar disesuaikan proporsional ($newWidth = pHeight * iRatio$).
 - Dihitung $offsetX$ agar gambar berada di tengah horizontal.
- Jika **PictureBox lebih tinggi atau sama**, kebalikannya:
 - Lebar gambar menyesuaikan ($newWidth = pWidth$).
 - Tinggi dihitung proporsional ($newHeight = pWidth / iRatio$).
 - $offsetY$ diatur agar gambar di tengah vertikal.

Konversi Titik Klik ke Koordinat Gambar

```
int imgX = (int)((mouseLocation.X - offsetX) * (iWidth / newWidth));
int imgY = (int)((mouseLocation.Y - offsetY) * (iHeight / newHeight));
```

- Mengubah posisi klik ($mouseLocation$) menjadi koordinat sebenarnya pada gambar.
- Dikurangi $offset$ karena area kosong ($padding$) tidak termasuk bagian gambar.
- Dikalikan skala ($iWidth / newWidth$) agar titik klik disesuaikan dengan ukuran gambar asli.
- Dikonversi ke `int` karena koordinat gambar berbentuk piksel bulat.

Validasi Batas Koordinat

```
if (imgX < 0) imgX = 0;
if (imgY < 0) imgY = 0;
if (imgX >= iWidth) imgX = (int)iWidth - 1;
if (imgY >= iHeight) imgY = (int)iHeight - 1;
```

- Menjaga agar koordinat hasil tidak keluar dari batas gambar.
 - Jika klik terjadi di luar area gambar (misal padding), koordinat “dijepit” ke batas terdekat dalam gambar.
-

Mengembalikan Hasil

```
return new Point(imgX, imgY);
```

- Mengembalikan titik koordinat hasil konversi dalam bentuk `Point (x, y)`.
 - Titik ini menunjukkan lokasi piksel yang sebenarnya di gambar (bukan di `PictureBox`).
-

Kesimpulan

Fungsi ini berguna saat kamu ingin mendeteksi posisi klik mouse **di dalam gambar yang telah di-resize** di dalam `PictureBox` — misalnya untuk mengedit, menggambar, atau memilih area tertentu pada gambar di aplikasi seperti *MiniPhotoshop*.

Apakah kamu mau saya buat **diagram visual** sederhana untuk menggambarkan hubungan antara `PictureBox`, gambar, offset, dan titik klik-nya? Itu akan membantu banget untuk memahami fungsi ini secara visual.