

PRAKTIKUM 11

Restorasi Citra dengan metode Median Filtering

11.1. Tujuan :

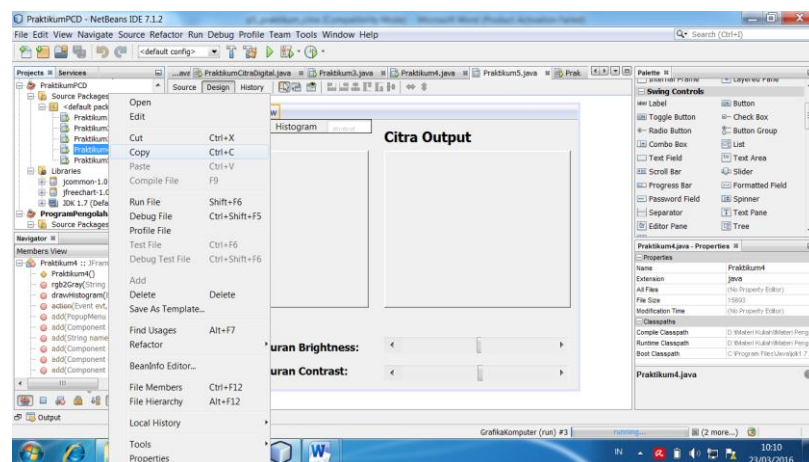
Mahasiswa mengetahui cara membuat program rsetorasi citra dengan menggunakan metode median filtering pada sebuah citra biner

11.2. Dasar Teori :

11.2.1. Algoritma Thinning

11.3. Langkah Praktikum :

- 1). Copy class JFrame Form pada praktikum sebelumnya dengan cara klik kanan pada folder Source Package Praktikum10 lalu pilih Copy seperti tampak pada gambar 11.1 berikut:

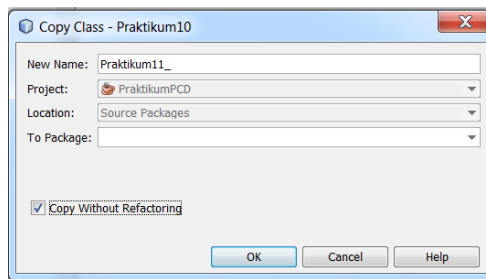


Gambar 11.1. Copy class JFrame Form

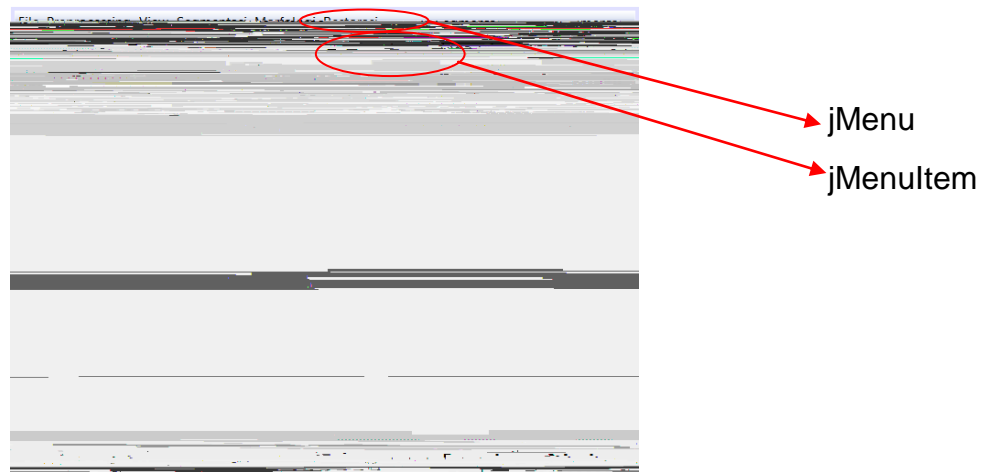
- 2). Kemudian klik kanan pada folder source packages pada <default Packages>, kemudian pilih Paste kemudian pilih Copy, seperti tampak pada gambar 11.2.

Gambar 11.2. Paste ke class JFrame Form baru

- 3). Rubah nama class menjadi Praktikum11 kemudian centang pilihan Copy Without Refactoring. Kemudian klik OK. Seperti tampak pada gambar 11.3. kemudian rubah nama class dari Praktikum10_1 menjadi Praktikum11.



Gambar 11.3



Gambar 11.3. Design Form

- 5). Tambahkan Kode metode untuk proses segmentasi pada bagian Source Praktikum11 pada event method `jMenuItem` median filetring sebagai berikut.

```
private void jMenuItem12ActionPerformed(java.awt.event.ActionEvent evt) {
    BufferedImage grayscale = medianfilter(sumber);
    int x = jLabel2.getWidth();
    int y = jLabel2.getHeight();
    ImageIcon imagelcon = new ImageIcon(resize(grayscale, x, y));
    jLabel2.setIcon(imagelcon);
}
```

- 7). Tambahkan method baru untuk proses thinning dengan kode sebagai berikut:

```
public BufferedImage medianfilter(String sumber) {
    BufferedImage prosesGambar;
    BufferedImage loading = loadImage(sumber);
    ukuranX = loading.getWidth();
    ukuranY = loading.getHeight();
    prosesGambar = new BufferedImage(ukuranX, ukuranY,
        BufferedImage.TYPE_BYTE_GRAY);
    Graphics g = prosesGambar.getGraphics();
    g.drawImage(loading, 0, 0, null);
    WritableRaster raster = prosesGambar.getRaster();
    for (int x = 1; x < (ukuranX - 1); x++) {
        for (int y = 1; y < (ukuranY - 1); y++) {
            int rgb11 = loading.getRGB((x - 1), (y - 1));
            p[0] = (rgb11 >> 8) & 0xff;
            int rgb12 = loading.getRGB(x, (y - 1));
            p[1] = (rgb12 >> 8) & 0xff;
            int rgb13 = loading.getRGB((x+1), (y - 1));
            p[2] = (rgb13 >> 8) & 0xff;
            int rgb21 = loading.getRGB((x - 1), (y));
            p[3] = (rgb21 >> 8) & 0xff;
            int rgb22 = loading.getRGB(x, y);
            p[4] = (rgb22 >> 8) & 0xff;
            int rgb23 = loading.getRGB((x+1), y);
            p[5] = (rgb23 >> 8) & 0xff;
            int rgb31 = loading.getRGB((x - 1), (y+1));
```

