

PRAKTIKUM 3

Kuantisasi, implementasi konversi RGB to Gray Scale

Tujuan :

Mahasiswa mengerti cara memprogram untuk melakukan teknik kuantisasi pada konversi model citra

Dasar Teori :

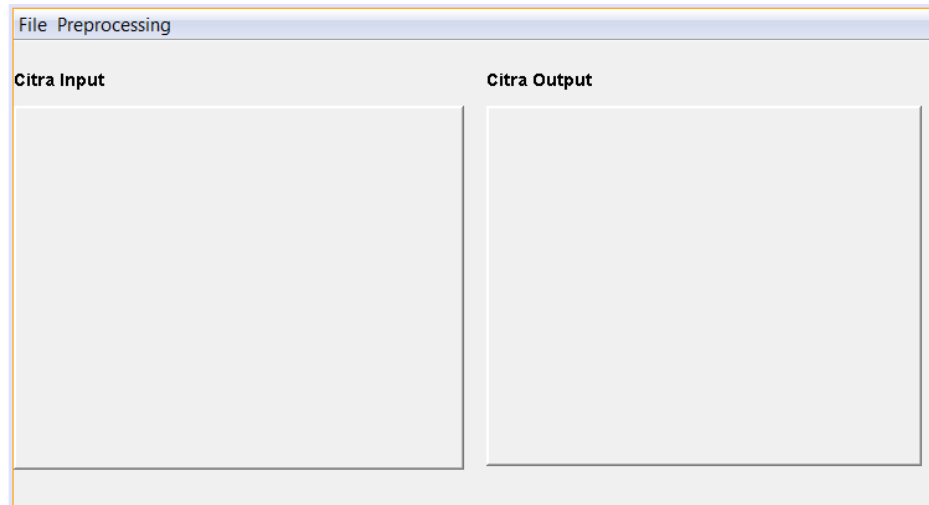
Citra gray-scale atau skala keabuan memiliki satu lapisan/layer dengan nilai 0 sampai 255, pada *image processing* proses awal yang dilakukan terhadap citra berwarna adalah mengubah menjadi citra gray-scale, hal tersebut dilakukan agar model citranya menjadi lebih sederhana. Citra berwarna memiliki tiga lapisan/layer yaitu lapisan merah (R-Layer), Lapisan Hijau (G-Layer) dan Lapisan biru (B-Layer).

Sehingga jika ingin melakukan proses pengolahan citra terhadap citra berwarna harus melibatkan tiga lapisan warna tersebut dan tentunya perhitungan yang dilakukan tiga kali lebih banyak dari pada citra gray-scale untuk itu konversi citra berwarna ke citra gray-scale perlu dilakukan dengan cara mengubah jumlah layer yang tadinya 3 layer menjadi 1 layer, Pada citra gray-scale tidak ada lagi warna yang ada hanyalah skala keabuan. Konversi dapat dilakukan dengan cara mengambil nilai rata-rata dari nilai r, g dan b yaitu :

$$s = \frac{r + g + b}{3} \quad \text{atau} \quad s = (0,5 * r) + (0,3 * g) + (0,2 * b)$$

Langkah Praktikum :

- 1). Buat class JFrame Form baru dengan nama Praktikum 3 dengan mengikuti langkah-langkah di Praktikum 2.
- 2). Desain Form dengan menggunakan komponen-komponen, JPanel, JLabel, MenuBar dan JMenuItem dengan bentuk seperti gambar berikut:



Gambar 3.1. Desain Form

3). Tambahkan Import Library dan variable Global sbb:

```
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.awt.Image;
import java.io.File;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.image.WritableRaster;

public class Praktikum3 extends javax.swing.JFrame {
    String sumber;
    BufferedImage gambar;
    Image gambarhasil;
    int ukuranX;
    int ukuranY;
```

4). Tambahkan Methode Event MenuItem File-> Open dengan Kode sbb:

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser fc = new JFileChooser();
    int open = fc.showOpenDialog(jPanel1);
    if (open == JFileChooser.APPROVE_OPTION) {
        sumber = fc.getSelectedFile().getPath();
        BufferedImage loadImg = loadImage(sumber);
        int x = jLabel1.getWidth();    int y = jLabel1.getHeight();
        int x1 = jLabel2.getWidth();    int y1 = jLabel2.getHeight();
        ImageIcon imageIcon1 = new ImageIcon(resize(loadImg, x, y));
        ImageIcon imageIcon2 = new ImageIcon(resize(loadImg, x1, y1));
        jLabel1.setIcon(imageIcon1);
        jLabel2.setIcon(imageIcon2);
    }
}
```

5). Salin Methode loadImage() dan resize() dari kode praktikum 2.

6). Tambahkan metode untuk konversi ke grayscale di atas main class dengan kode sbb:

```
// deklarasi metode untuk grayscale
public BufferedImage rgb2Gray(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 = 0; x < ukuranX; x++) {
        for (int y = 0; y < ukuranY; y++) {
            int rgb = loadIng.getRGB(x, y);
            int alpha = (rgb << 24) & 0xff;
            int merahg = (rgb >> 16) & 0xff;
            int hijau = (rgb >> 8) & 0xff;
            int biru = (rgb >> 0) & 0xff;
            float gray = (float)((0.5*merahg) + (0.3*hijau) + (0.2*biru));
            raster.setSample(x, y, 0, gray);
        }
    }
    return prosesGambar;
}
```

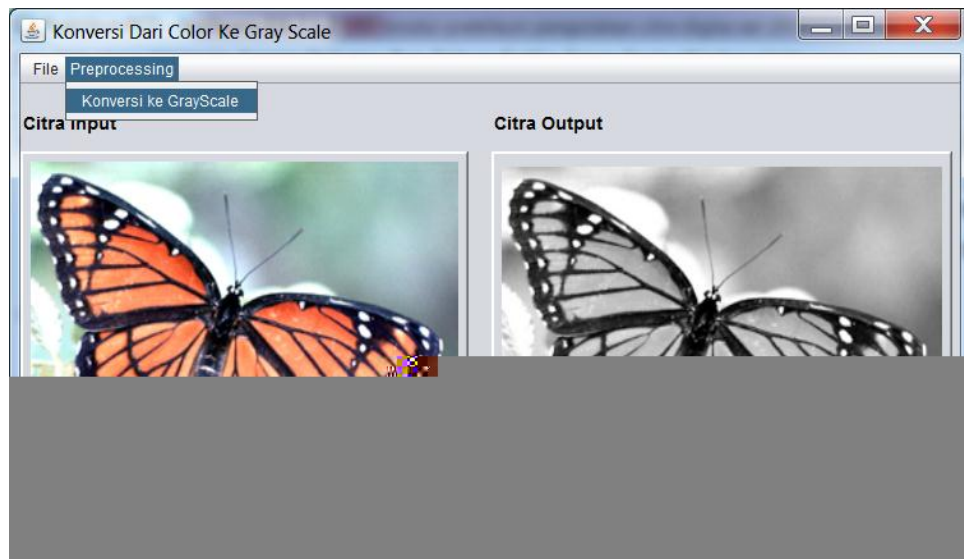
7). Tambahkan kode program event metode pada MenuBar Preprocessing -> MenuItem Konversi ke grayScale sbb:

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
    BufferedImage grayscale = rgb2Gray(sumber);
    int x = jLabel2.getWidth();
    int y = jLabel2.getHeight();
    ImageIcon imageIcon = new ImageIcon(resize(grayscale, x, y));
    jLabel2.setIcon(imageIcon);
}
```

8). Tambahkan kode program event metode pada MenuBar File -> MenuItem Exit sbb:

```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

9). jalankan program dan lakukan proses konversi setelah gambar berwarna di ambil sehingga tampilannya seperti gambar berikut:



Gambar 3.2. Tampilan Program Konversi citra berwarna ke citra Gray Scale

Tugas:

Modifikasi program di atas, agar dapat melakukan konversi dari citra Gray Scale ke citra biner dengan nilai threshold 128, jadi setiap piksel yang nilai < 128 dijadikan 0, dan yang ≥ 128 dijadikan 255.