

PRAKTIKUM 9

Morfologi Citra

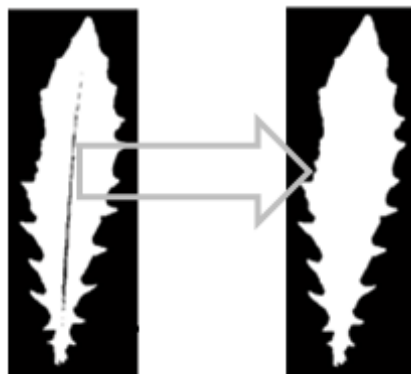
9.1. Tujuan :

Mahasiswa mengetahui cara membuat program morfologi citra dengan menggunakan operasi Dilasi dan erosi pada sebuah citra biner

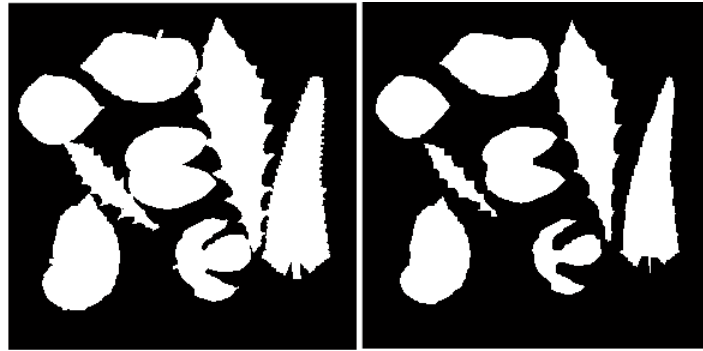
9.2. Dasar Teori :

Operasi morfologi merupakan operasi yang umum dikenakan pada citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra. Sebagai contoh, lubang pada daun dapat ditutup melalui operasi morfologi sebagaimana ditunjukkan pada Gambar 9.1. Objek-objek daun yang saling berhimpitan pun dapat dipisahkan melalui morfologi, sebagaimana ditunjukkan pada Gambar 9.2. Beberapa contoh lain aplikasi morfologi adalah sebagai berikut.

- Membentuk filter spasial.
- Memperoleh skeleton (rangka) objek.
- Menentukan letak objek di dalam citra.
- Memperoleh bentuk struktur objek.



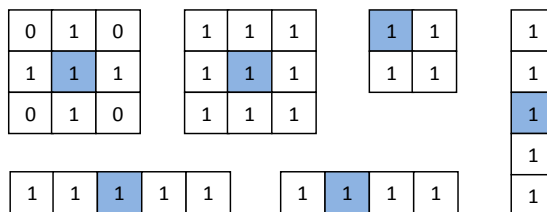
Gambar 9.1 Tulang daun dapat dianggap sebagai bagian daun melalui morfologi



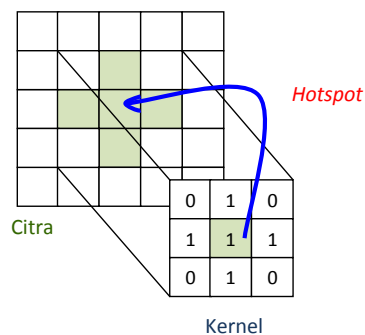
Gambar 9.2 Daun-daun yang bersinggungan dapat dipisahkan melalui morfologi, yang memperkecil ukurannya

9.2.1. Morfologi

Inti operasi morfologi melibatkan dua larik piksel. Larik pertama berupa citra yang akan dikenai operasi morfologi, sedangkan larik kedua dinamakan sebagai kernel atau *structuring element* (elemen penstruktur) (Shih, 2009). Contoh kernel ditunjukkan pada Gambar 9.3. Pada contoh tersebut, piksel pusat (biasa diberi nama *hotspot*) ditandai dengan warna abu-abu. Piksel pusat ini yang menjadi pusat dalam melakukan operasi terhadap citra, sebagaimana diilustrasikan pada Gambar 9.4.



Gambar 9.3 Contoh beberapa kernel



Gambar 9.4 Operasi kernel terhadap citra

Dua operasi yang mendasari morfologi yaitu dilasi dan erosi. Dua operasi lain yang sangat berguna dalam pemrosesan citra adalah *opening* dan *closing* dibentuk melalui dua operasi dasar itu.

9.2.2. Dilasi

Operasi dilasi biasa dipakai untuk mendapatkan efek pelebaran terhadap piksel yang bernilai 1. Operasi ini dirumuskan seperti berikut (Gonzales & Woods, 2002):

$$A \oplus B = \{z \mid [(B)_z \cap A] \subseteq A\} \quad (9.1)$$

Dalam hal ini,

- a) $\hat{B} = \{w \mid w = -b, \text{ untuk } b \in B\}$
- b) $(B)_z = \{c \mid c = a + z, \text{ untuk } a \in A\}$
- c) $z = (z_1, z_2)$

Burger & Burge (2008) mendefinisikan operasi dilasi sebagai berikut:

$$A \oplus B = \{z \mid z = a + b, \text{ dengan } a \in A \text{ dan } b \in B\} \quad (9.2)$$

Hasil dilasi berupa penjumlahan seluruh pasangan koordinat dari I dan H.

Contoh operasi dilasi dengan menggunakan Persamaan 9.1 dapat dilihat pada Gambar 9.5. Pada contoh tersebut,

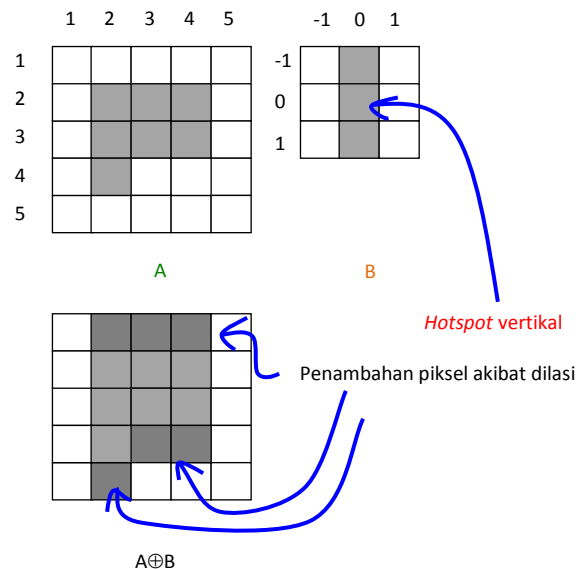
$$A = \{ (2,2), (2,3), (2,4), (3,2), (3,3), (3,4), (4,3) \}$$

$$B = \{ (-1, 0), (0,0), (1,0) \}$$

Dengan demikian,

$$\begin{aligned} A \oplus B &= \{ (2,2) + (-1, 0), (2,2) + (0, 0) + (2,2) + (1, 0), \\ &\quad (2,3) + (-1, 0), (2,3) + (0, 0) + (2,3) + (1, 0), \\ &\quad (2,4) + (-1, 0), (2,4) + (0, 0) + (2,4) + (1, 0), \\ &\quad (3,2) + (-1, 0), (3,2) + (0, 0) + (3,2) + (1, 0), \\ &\quad (3,3) + (-1, 0), (3,3) + (0, 0) + (3,3) + (1, 0), \\ &\quad (3,4) + (-1, 0), (3,4) + (0, 0) + (3,4) + (1, 0), \\ &\quad (4,3) + (-1, 0), (4,3) + (0, 0) + (4,3) + (1, 0) \} \\ &= \{ (1,2), (2,2), (3,2), (1,3), (2,3), (3,3), \\ &\quad (1,4), (2,4), (3,3), (2,2), (3,2), (4,2), \\ &\quad (2,3), (3,3), (4,3), (2,4), (3,4), (4,4), \end{aligned}$$

$$\begin{aligned}
 & (3,3), (4,3), (5,3) \} \\
 = & \{ (1,2), (1,3), (1,4), (2,2), (2,3), (2,4), \\
 & (3,2), (3,3), (3,4), (4,2), (4,3), (4,4), (5,3) \}
 \end{aligned}$$



Gambar 9.5. Efek dilasi dengan hotspot vertikal

9.2.3. Erosi

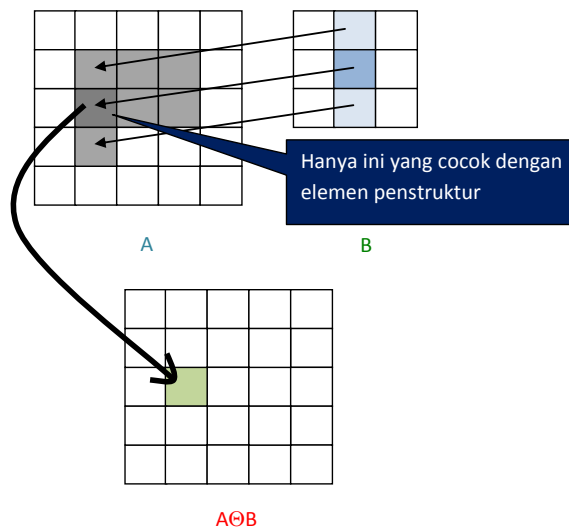
Operasi erosi mempunyai efek memperkecil struktur citra. Operasi ini dirumuskan seperti berikut (Gonzalez & Woods, 2002).

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (9.3)$$

Adapun Burger & Burge (2008) mendefinisikan erosi sebagai berikut:

$$A \ominus B = \{p \in Z^2 | (a + b) \in I, \text{ untuk setiap } b \in B\} \quad (9.4)$$

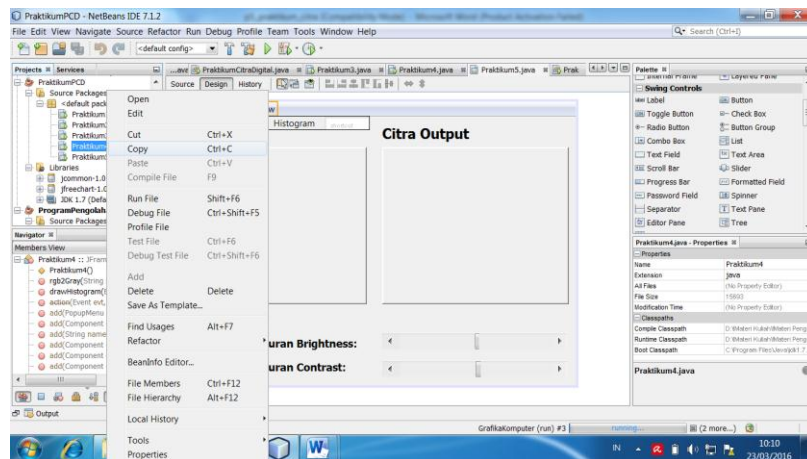
Makna yang tersirat pada Persamaan 9.3 dan 7.4 sebenarnya sama. Berdasarkan Persamaan 9.4, posisi p terdapat pada $A \ominus B$ jika seluruh nilai 1 di B terkandung di posisi p tersebut. Gambar berikut memperlihatkan bentuk visual untuk contoh di atas.



Gambar 9.6. Contoh visualisasi operasi erosi

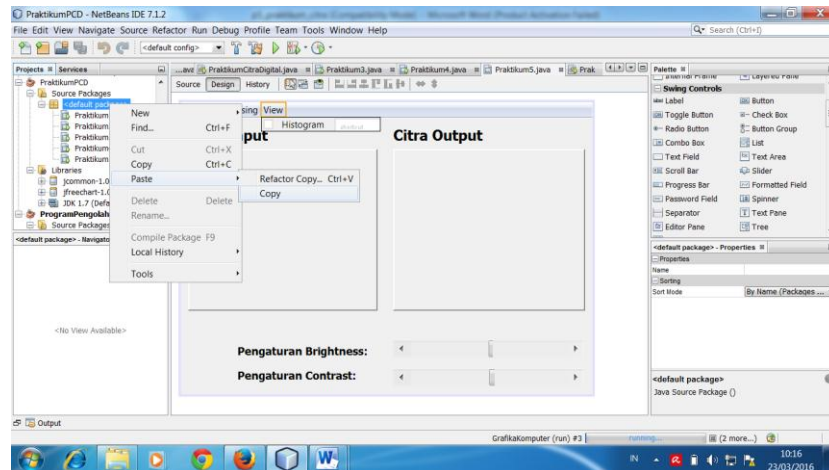
8.2. Langkah Praktikum :

- 1). Copy class JFrame Form pada praktikum sebelumnya dengan cara klik kanan pada folder Source Package Praktikum8 lalu pilih Copy seperti tampak pada gambar 9.7 berikut:



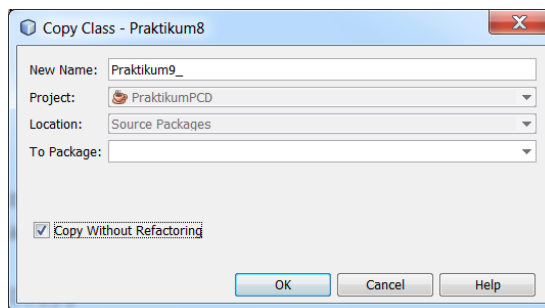
Gambar 9.7. Copy class JFrame Form

- 2). Kemudian klik kanan pada folder source packages pada <default Packages>, kemudian pilih Paste kemudian pilih Copy, seperti tampak pada gambar 9.8.



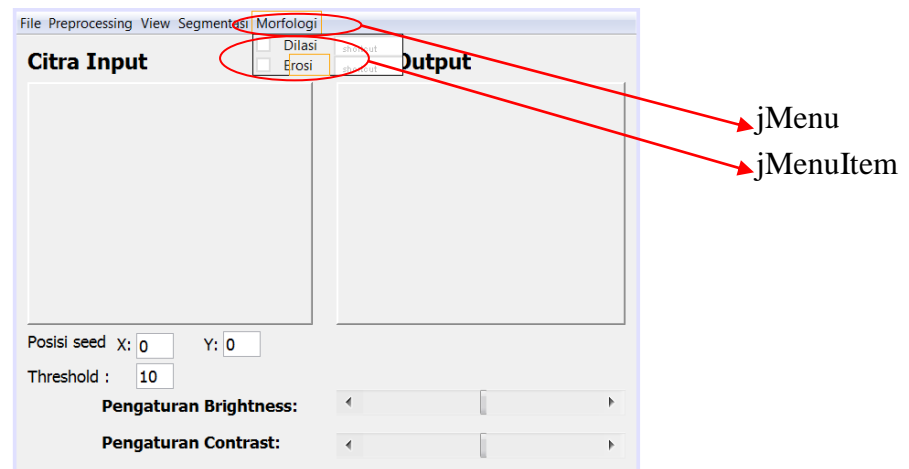
Gambar 9.8. Paste ke class JFrame Form baru

- 3). Rubah nama class menjadi `Praktikum9` kemudian centang pilihan `Copy Without Refactoring`. Kemudian klik OK. Seperti tampak pada gambar 9.9. kemudian rubah nama class dari `Praktikum8_1` menjadi `Praktikum9`.



Gambar 9.9. Jendela Dialog Copy Class

- 4). Klik Design, kemudian tambahkan 1 buah komponen `jMenu` pada `jMenuBar` dan rubah text nya menjadi `Morfologi` kemudian tambahkan 2 buah `jMenuItem` baru pada Menu `Morfologi` tersebut kemudian rubah text nya masing-masing menjadi `Dilasi` dan `Erosi`. Seperti tampak dalam gambar 9.10. berikut:



Gambar 9.10. Design Form

- 5). Tambahkan Kode metode untuk proses segmentasi pada bagian Source Praktikum9 pada event method `jMenuItem Dilasi` sebagai berikut.

```
private void jMenuItem10ActionPerformed(java.awt.event.ActionEvent evt) {
    BufferedImage grayscale = dilasi(sumber);
    int x = jLabel2.getWidth();
    int y = jLabel2.getHeight();
    ImageIcon imageIcon = new ImageIcon(resize(grayscale, x, y));
    jLabel2.setIcon(imageIcon);
}
```

- 6). Tambahkan Kode metode untuk proses segmentasi pada bagian Source Praktikum9 pada event method `jMenuItem Erosi` sebagai berikut.

```
private void jMenuItem11ActionPerformed(java.awt.event.ActionEvent evt) {
    BufferedImage grayscale = erosi(sumber);
    int x = jLabel2.getWidth();
    int y = jLabel2.getHeight();
    ImageIcon imageIcon = new ImageIcon(resize(grayscale, x, y));
    jLabel2.setIcon(imageIcon);
}
```

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

```
public BufferedImage dilasi(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 rgb22 = loadIng.getRGB(x, y);
        }
    }
}
```

```
        int p1 = (rgb22 >> 8) & 0xff;
        if (p1==0) {
            raster.setSample(x-1, y-1, 0, 0);
            raster.setSample(x, y-1, 0, 0);
            raster.setSample(x+1, y-1, 0, 0);
            raster.setSample(x-1, y, 0, 0);
            raster.setSample(x+1, y, 0, 0);
            raster.setSample(x-1, y+1, 0, 0);
            raster.setSample(x, y+1, 0, 0);
            raster.setSample(x+1, y+1, 0, 0);
        }
    }
    return prosesGambar;
}
```

8). Tambahkan metode baru untuk proses dilasi dengan kode sebagai berikut:

```
public BufferedImage erosi(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));
            int p9 = (rgb11 >> 8) & 0xff;
            int rgb12 = loadIng.getRGB(x, (y-1));
            int p2 = (rgb12 >> 8) & 0xff;
            int rgb13 = loadIng.getRGB((x+1), (y-1));
            int p3 = (rgb13 >> 8) & 0xff;
            int rgb21 = loadIng.getRGB((x-1), (y));
            int p8 = (rgb21 >> 8) & 0xff;
            int rgb22 = loadIng.getRGB(x, y);
            int p1 = (rgb22 >> 8) & 0xff;
            int rgb23 = loadIng.getRGB((x+1), (y));
            int p4 = (rgb23 >> 8) & 0xff;
            int rgb31 = loadIng.getRGB((x-1), (y+1));
            int p7 = (rgb31 >> 8) & 0xff;
            int rgb32 = loadIng.getRGB(x, (y+1));
            int p6 = (rgb32 >> 8) & 0xff;
            int rgb33 = loadIng.getRGB((x+1), (y+1));
            int p5 = (rgb33 >> 8) & 0xff;
            if ((p2 != 0) | (p3 != 0) | (p4 != 0) | (p5 != 0)
                | (p6 != 0) | (p7 != 0) | (p8 != 0) | (p9 != 0))
            {
                raster.setSample(x, y, 0, 255);
            }
        }
    }
    return prosesGambar;
}
```


9). Jalankan program sehingga tampilannya seperti gambar 9.11. berikut:



Gambar 9.11. Tampilan aplikasi hasil proses morfologi dilasi