

## 2

### MEMBACA DAN MENGOLAH DATA CITRA

Pada pokok bahasan ini, mahasiswa akan mempelajari tentang cara membaca, mengolah data citra dan membuat kode program dalam memanipulasi citra.

#### **Pokok Bahasan :**

1. Membaca file citra
2. Manipulasi data citra

#### **Latihan :**

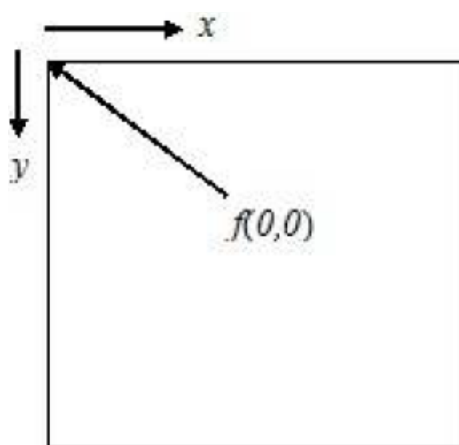
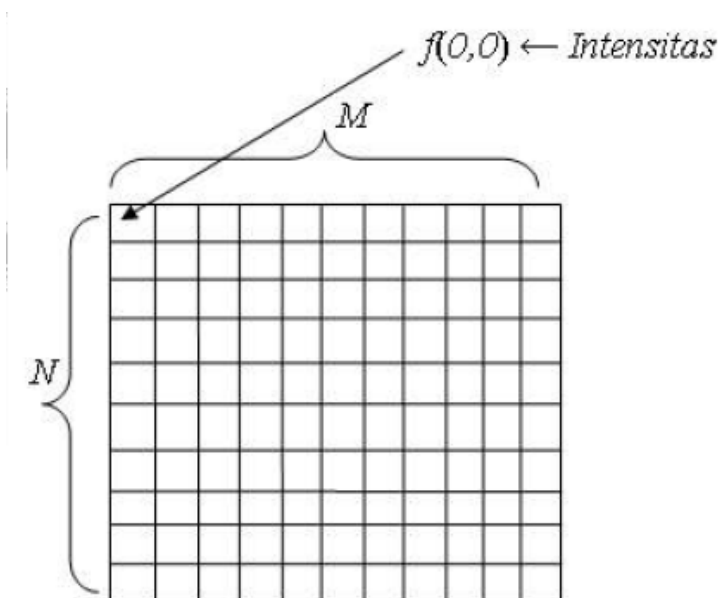
1. Membuat aplikasi *flip image* dengan menggunakan visual C#
2. Membuat aplikasi *rotate image* dengan menggunakan visual C#
3. Membuat aplikasi *image translation* dengan menggunakan visual C#

## 2.1 Membaca file Citra

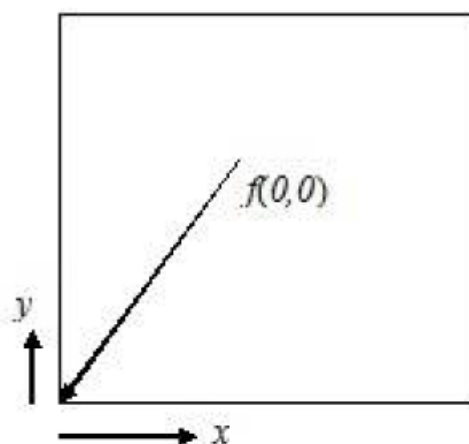
Capaian pembelajaran : mampu membaca file citra menggunakan visual C#

Citra/gambar secara konsep data dinyatakan dalam bentuk matrik 2D atau array 2D, dimana setiap pixelnya pada posisi  $(x,y)$  dinyatakan dalam  $f(x,y)$ . Setiap pixel pada data gambar mempunyai nilai sesuai dengan format gambar yang digunakan:

- Pada gambar berwarna, setiap titik mempunyai nilai 24 bit RGB, dimana masing-masing komponen warna R, G dan B mempunyai nilai 8 bit, atau dengan kata lain setiap komponen warna mempunyai nilai 0 s/d 255.
- Pada gambar grayscale atau derajat keabuan, setiap titik mempunyai nilai 8 bit, atau 0 s/d 255.
- Pada gambar hitam putih (BW), setiap titik mempunyai nilai 0 atau 1.



(a) Koordinat pada citra



(b) Koordinat kartesian

## 2.2 Manipulasi File Citra

Capaian pembelajaran : mampu memanipulasi citra sehingga menjadi bentuk lain dari citra.

Untuk bisa mengolah data citra (misalkan dengan nama `sourceImage`) maka sebelumnya harus didefinisikan sebuah obyek citra dalam type `Bitmap`, yaitu

**`Bitmap sourceImage;`**

Dalam sebuah obyek citra **`sourceImage`** terdapat beberapa parameter dan fungsi yang dapat digunakan yaitu:

- **`sourceImage.width`** : menyatakan panjang dari sebuah gambar dalam jumlah pixel horisontal.
- **`sourceImage.height`** : menyatakan tinggi dari sebuah gambar dalam jumlah pixel vertikal.
- **`sourceImage.getPixel(x,y)`** adalah sebuah fungsi untuk membaca data RGB dari pixel ( $x,y$ ). Hasil pembacaan berupa nilai warna  $w$  yang bertipe `Color` dan menghasilkan 3 data yaitu  $w.R$  untuk warna merah,  $w.G$  untuk warna hijau dan  $w.B$  untuk warna biru.
- **`sourceImage.setPixel (x, y, w)`** adalah sebuah fungsi untuk memberikan nilai warna  $w$  RGB pada *pixel* ( $x,y$ ) dalam *`sourceImage`*.

Proses pembacaan image dapat ditulis dalam sintak :

```
for (int x = 0; x < imageWidth; x++)
    for (int y = 0; y < imageHeight; y++)
    {
        //fungsi untuk mengambil warna pada setiap titik pixel gambar
        Color w = sourceImage.GetPixel(x, y);

        byte warnaMerah = w.R;
        byte warnaHijau = w.G;
        byte warnaBiru = w.B;

        Color wBaru = Color.FromArgb(warnaMerah, warnaHijau, warnaBiru);

        //fungsi untuk mengganti warna pada setiap titik pixel gambar
        //dengan warna baru
        tempImage.SetPixel(x, y, wBaru);
    }
```

### 2.2.1 Image Flipping

Flipping adalah operasi geometri yang sama dengan pencerminan. Ada dua macam flipping : horizontal dan vertical.

Flipping horizontal adalah pencerminan pada sumbu-Y dari citra A menjadi citra B, yang diberikan oleh persamaan :

$$X' = \text{imageWidth} - 1 - X;$$

$$Y' = Y;$$

Flipping vertikal adalah pencerminan pada sumbu-X dari citra A menjadi citra B, yang diberikan oleh persamaan :

$$X' = X;$$

$$Y' = \text{imageHeight} - 1 - Y;$$

Dengan X' dan Y' adalah posisi yang baru

Flip Horizontal artinya pixel di sebelah kiri akan diganti dengan pixel sebelah kanan sementara pixel sebelah kanan akan diganti dengan pixel di sebelah kiri.

Flip Vertical artinya pixel di sebelah atas akan diganti dengan pixel sebelah bawah sementara pixel sebelah bawah akan diganti dengan pixel di sebelah atas.

### 2.2.2 Image Rotation

Rotasi Citra adalah operasi geometri berupa perputaran citra terhadap titik tengah gambar. Rotasi citra bisa berupa CCW (*Counter Clock Wise*) atau CW (*Clock Wise*).

### 2.2.3 Image Translation

Image translation adalah pergeseran posisi suatu pixel ke posisi yang baru. Image translation terbagi dua : translation sepanjang sumbu X (*x translation*) dan translation sepanjang sumbu Y (*y translation*).

X translation diformulasikan dengan persamaan :

$$x' = x + xT ;$$

Y translation diformulasikan dengan persamaan :

$$y' = y + yT ;$$

Dengan X' dan Y' adalah posisi yang baru sedangkan xT dan yT adalah besarnya pergeseran pada sumbu X dan sumbu Y.

## 2.3 Latihan

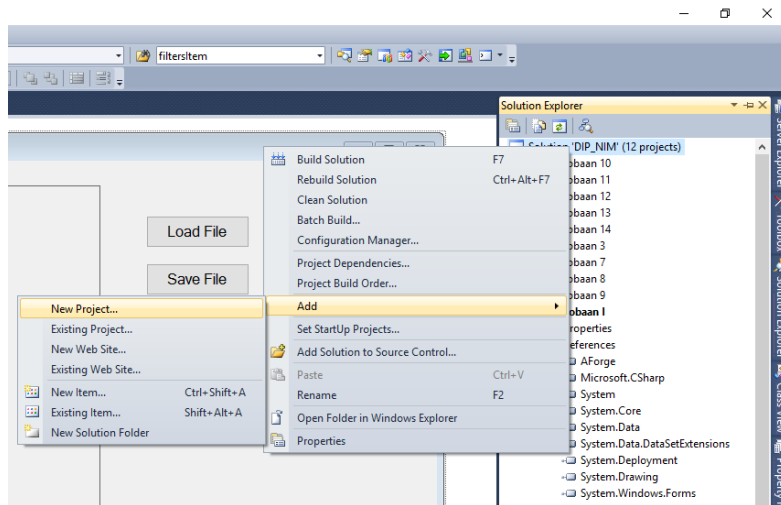
### Tujuan

Memahami cara membaca file gambar dengan microsoft visual C#

Mampu memanipulasi data citra

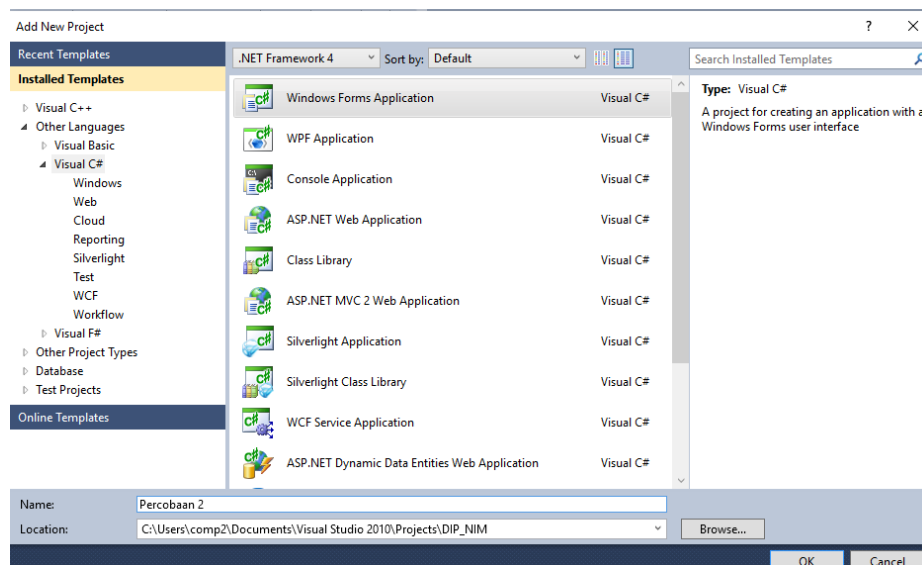
### Prosedur

1. Tambahkah proyek baru pada solution anda dengan cara klik kanan solution anda yang berada di solution explorer kemudian pilih menu **Add→New Project** seperti gambar 2.1.



Gambar 2.1 Menu penambahan proyek baru

2. Pilih **Windows Forms Application** dan beri nama **Percobaan 2**.

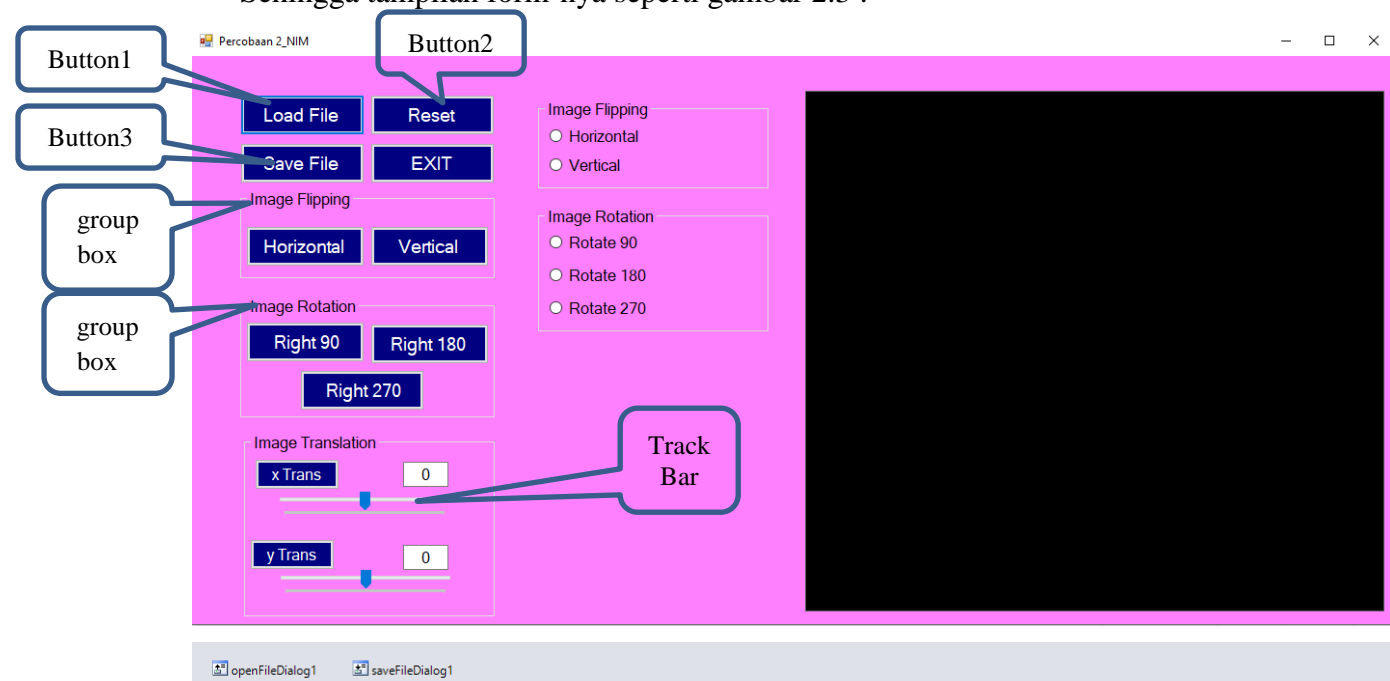


Gambar 2.2 Menu penambahan proyek baru

3. Setelah keluar form baru, tambahkan

- 1 buah **pictureBox** dengan 1 buah **openFileDialog1** dan 1 buah **saveFileDialog1**
- 5 buah **groupBox** : 2 untuk image Flipping dan 2 untuk Image Rotation dan 1 untuk image translation
- 9 buah **button**
- 2 buah **trackBar**
- 2 buah **textBox**
- 5 buah **radioButton**

Sehingga tampilan form-nya seperti gambar 2.3 :



Gambar 2.3. Tampilan VC# setelah membuat proyek baru

4. Setting properties **pictureBox** seperti percobaan 1:
5. Setting properties **openFileDialog1** dan **saveFileDialog1** seperti percobaan 1
6. Tambahkan library berikut pada program anda

```
//tambahkan ini
using System.Drawing.Imaging;
using System.IO;
```

7. Tambahkan global variable sebagai berikut :

```
//global variable
Bitmap sourceImage, tempImage;
int imageHeight, imageWidth;

//variable of image flipping
int imageFlipping;

//variable of image rotation
int imageRotation;
```

8. Double klik tombol **Load File**, kemudian tuliskan sub rutin program berikut :

```
private void button1_Click(object sender, EventArgs e)
{
    openFileDialog1.ShowDialog();
}
```

9. Double klik **OpenFileDialog1** kemudian tuliskan sub rutin program berikut :

```
private void openFileDialog1_FileOk (object sender, EventArgs e)
{
    sourceImage = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);

    tempImage = new Bitmap(sourceImage);

    pictureBox1.Image = sourceImage;

    //mencari tinggi dan lebar image
    imageHeight = sourceImage.Height;
    imageWidth = sourceImage.Width;

    //translation trackBar init
    trackBar1.Minimum = - imageWidth / 2;
    trackBar1.Maximum = imageWidth / 2;

    trackBar2.Minimum = -imageHeight / 2;
    trackBar2.Maximum = imageHeight / 2;
}
```

10. Tambahkan Fungsi inisialisasi seperti di bawah ini:

```
public Percobaan2()
{
    InitializeComponent();

    //initialization
    trakbarInitialization();
    textBoxInitialization();
}
```

11. Tambahkan pula fungsi berikut :

```
private void trakbarInitialization()
{
    //trackBar init pos
    trackBar1.Value = 0;
    trackBar2.Value = 0;
}

private void textBoxInitialization()
{
    textBox1.Text = "0";
    textBox2.Text = "0";
}
```

12. Double klik tombol **Reset** dan tambahkan rutin berikut :

```
private void button2_Click(object sender, EventArgs e)
{
    if (sourceImage == null) return;

    pictureBox1.Image = sourceImage;
    //initialization
    trakbarInitialization();
    textBoxInitialization();
}
```

13. Double klik tombol **Save File** dan tambahkan sub rutin untuk **Save File** seperti latihan pada bab 1.

14. Double klik tombol **Exit** dan tambahkan rutin berikut :

```
private void button4_Click(object sender, EventArgs e)
{
    Close();
}
```

15. Double klik tombol **Filp Horizontal** dan tambahkan rutin berikut :

```
private void button5_Click(object sender, EventArgs e)
{
    imageFlipping = 1;
    setImageFlipping(imageFlipping);
}
```

16. Tambahkan program berikut untuk image flipping

```
private void setImageFlipping(int flipping)
{
    if (tempImage == null) return;

    //image flipping
    // 1 = horizontal
    // 2 = vertical

    Bitmap flipImage = new Bitmap(tempImage);
    for (int x = 0; x < imageWidth; x++)
        for (int y = 0; y < imageHeight; y++)
        {
            Color w = flipImage.GetPixel(x, y);

            if (flipping == 1) //flip horizontal
            {
                tempImage.SetPixel(imageWidth - 1 - x, y, w);
            }
            else if (flipping == 2) //flip vertical
            {
                tempImage.SetPixel(x, imageHeight - 1 - y, w);
            }
        }
    pictureBox1.Image = tempImage;
}
```



17. Tambahkan tombol **Rotate Right 90**, kemudian tambahkan kode program berikut

```
private void button7_Click(object sender, EventArgs e)
{
    imageRotation = 90;
    setImageRotation(imageRotation);
}
```

18. Tambahkan fungsi berikut untuk image rotation

```
private void setImageRotation(int rotation)
{
    if (tempImage == null) return;

    if (rotation == 90)
    {
        tempImage.RotateFlip(RotateFlipType.Rotate90FlipNone);
    }
    else if (rotation == 180)
    {
        tempImage.RotateFlip(RotateFlipType.Rotate180FlipNone);
    }
    else if (rotation == 270)
    {
        tempImage.RotateFlip(RotateFlipType.Rotate270FlipNone);
    }

    pictureBox1.Image = tempImage;
}
```

19. Double klik button dan trackbar **xTrans** dan tambahkan rutin berikut :

```
private void button10_Click(object sender, EventArgs e)
{
    if (tempImage == null) return;

    int xTrans = int.Parse (textBox1.Text);
    int yTrans = int.Parse (textBox2.Text);

    //setting X translation
    setTranslation(xTrans, yTrans);

    //menampilkan nilai translasi pada trackBar
    trackBar1.Value = int.Parse(textBox1.Text);
    trackBar2.Value = int.Parse(textBox2.Text);
}

private void trackBar1_Scroll(object sender, EventArgs e)
{
    if (tempImage == null) return;

    int xTrans = trackBar1.Value;
    int yTrans = trackBar2.Value;

    //setting X translation
    setTranslation(xTrans, yTrans);

    //menampilkan nilai translasi pada textBox
    textBox1.Text = string.Format("{0}", trackBar1.Value);
    textBox2.Text = string.Format("{0}", trackBar2.Value);
}
```

20. Tambahkan pula fungsi berikut

```
private void setTranslation(int xTrans, int yTrans)
{
    Bitmap transImage = new Bitmap(imageWidth, imageHeight);
    for (int x = 0; x < imageWidth; x++)
        for (int y = 0; y < imageHeight; y++)
        {
            Color w = tempImage.GetPixel(x, y);
            byte warnaMerah = w.R;
            byte warnaHijau = w.G;
            int xT = x + xTrans;
            int yT = y + yTrans;
            if (yT < imageHeight && yT > 0 && xT < imageWidth && xT > 0)
                transImage.SetPixel(xT, yT, w);
        }
    pictureBox1.Image = transImage;
}
```

21. Jalankan program dan **Load** salah satu file gambar. Kemudian jalankan semua tombol yang ada dan **Save** setiap gambar hasilnya.

## 2.4 Tugas dan Pertanyaan

1. Tambahkan koding pada button **FlipVertical**
2. Tambahkan koding pada button **Rotate Right 180** dan tombol **Rotate Right 270**
3. Tambahkan koding pada button dan trackBar **yTrans**
4. Tambahkan koding untuk masing-masing radio button pada image flipping dan image rotation.
5. Laporan merupakan hasil **screenshot/save** hasil running program