



Achmad Basuki Nana R Fadilah Fahrul
Politeknik Elektronika Negeri Surabaya

Praktikum Pengolahan Citra - Pertemuan 1.2



Content:

1. Representasi Obyek Citra -> Matrik 2 D dari Intensitas Piksel
2. Format Citra
3. Obyek Citra -> Objek Bitmap Bitmap
4. Fungsi Setpixel() dan Getpixel()
5. Contoh Copy, dan Flip Citra

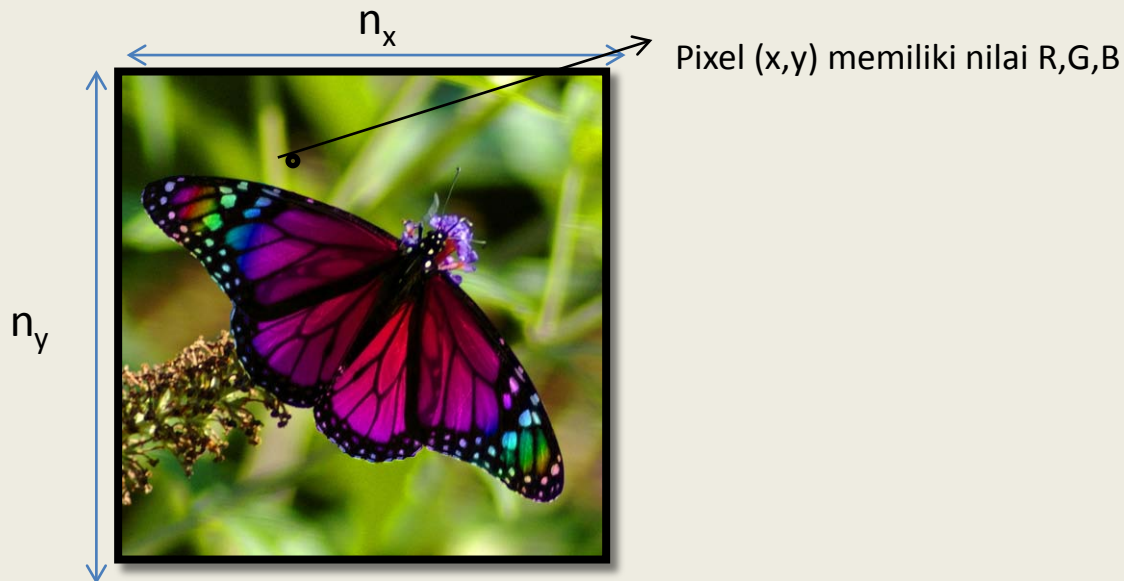


Teori Penunjang

Fungsi Setpixel dan Getpixel untuk Copy dan Flip Citra

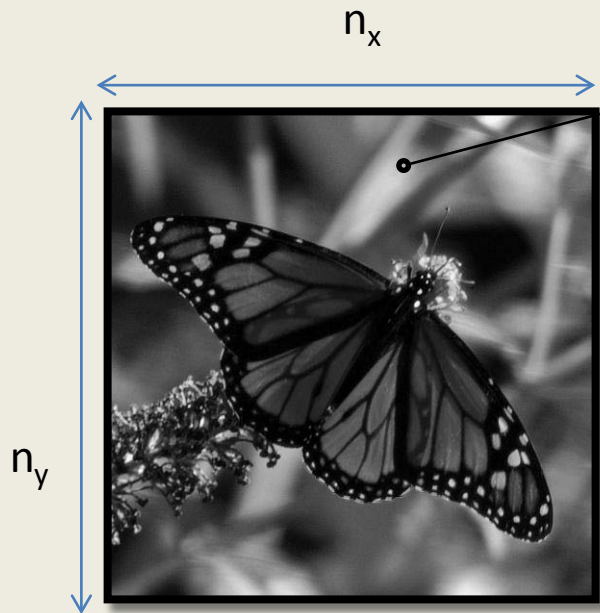
Citra sebagai Matrik 2D

- Citra/gambar secara konsep dapat dinyatakan dalam bentuk 2D atau array 2D, dimana setiap pixelnya pada posisi (x,y) dinyatakan dalam $P(x,y)$.
 - Setiap pixel pada data gambar mempunyai nilai sesuai dengan format gambar yang digunakan:
1. Pada gambar berwarna, setiap titik mempunyai nilai 24bit RGB, dimana masing-masing komponen warna R,G dan B mempunyai nilai 8bit, atau dengan kata lain setiap komponen warna memiliki nilai 0 s/d 255.
 2. Pada gambar grayscale atau derajat keabuan, setiap titik mempunyai nilai 8bit, atau 0 s/d 255.
 3. Pada gambar hitam putih (BW), setiap titik mempunyai nilai 0 atau 1.



Format Citra

Apapun format gambar yang digunakan, program di dalam Visual Studio .Net hanya mengenal format gambar berwarna RGB, sehingga untuk menampilkan gambar grayscale atau BW digunakan nilai yang sama untuk setiap komponen R,G dan B.



Pixel (x,y) tetap memiliki nilai R,G,B
Namun nilai R,G dan B yang sama

Misal:

R = 99
G = 99
B = 99

Citra sebagai objek Bitmap

- Untuk bisa mengolah data citra maka sebelumnya harus didefinisikan sebuah obyek citra dalam type Bitmap.
- Bitmap objBitmap;
- Dalam sebuah obyek Bitmap terdapat beberapa parameter dan fungsi yang dapat digunakan yaitu:
 - objBitmap.width
menyatakan panjang dari sebuah gambar (jumlah pixel horisontal)
 - objBitmap.height
menyatakan tinggi dari sebuah gambar (jumlah pixel vertical)
 - objBitmap.getPixel
fungsi untuk membaca data RGB dari pixel(x,y).
Hasil pembacaan berupa nilai warna yang bertipe Color yang memiliki tiga data yaitu color.R, color.G dan color.B
 - objBitmap.setPixel
fungsi untuk memberikan nilai warna RGB pada pixel(x,y) dalam objBitmap.

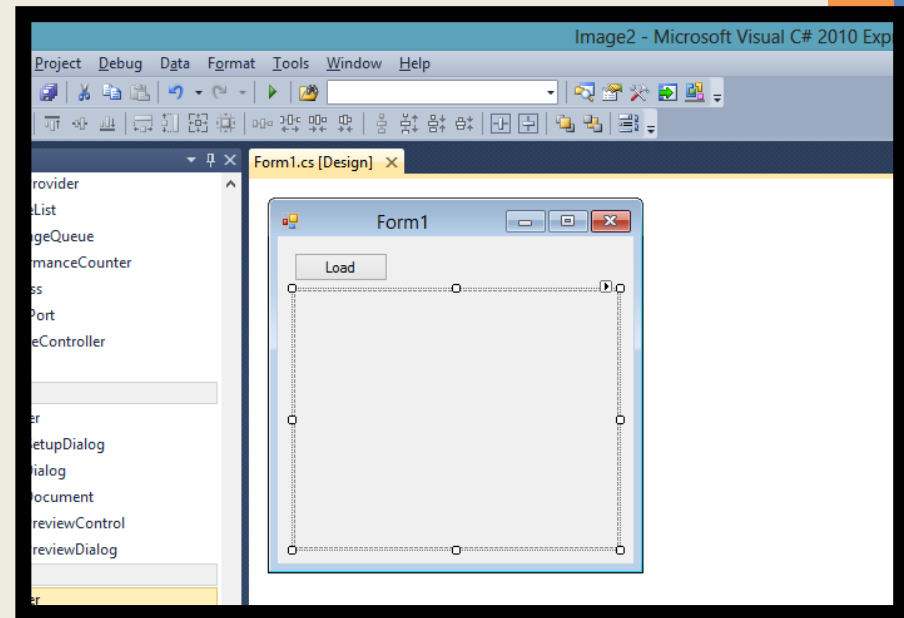


Petunjuk Praktikum

MEMBACA DATA GAMBAR DARI FILE

Praktikum ini menunjukkan langkah-langkah membuat aplikasi untuk membaca data setiap pixel pada sebuah gambar.

1. Buka Visual Studio .Net 2010
2. Pilih File → New → New Project
3. Pilih Visual C# [Windows Forms Application].
4. Isi Name dan Solution Name dengan "Image2" kemudian tekan tombol [OK].
5. Setelah muncul form new project, tambahkan 1 Button, 1 PictureBox dan 1 OpenFileDialog.
6. Ubah text Button1 menjadi "Load". Atur size-mode pada pictureBox1 dengan StretchImage.
7. Aturilah tampilan seperti gambar disamping.



8. Tambahkan sebuah obyek Bitmap dengan nama objBitmap sebelum baris kode `public Form1()`

```
namespace image1
{
    public partial class Form1:Form
    {
        Bitmap objBitmap;
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

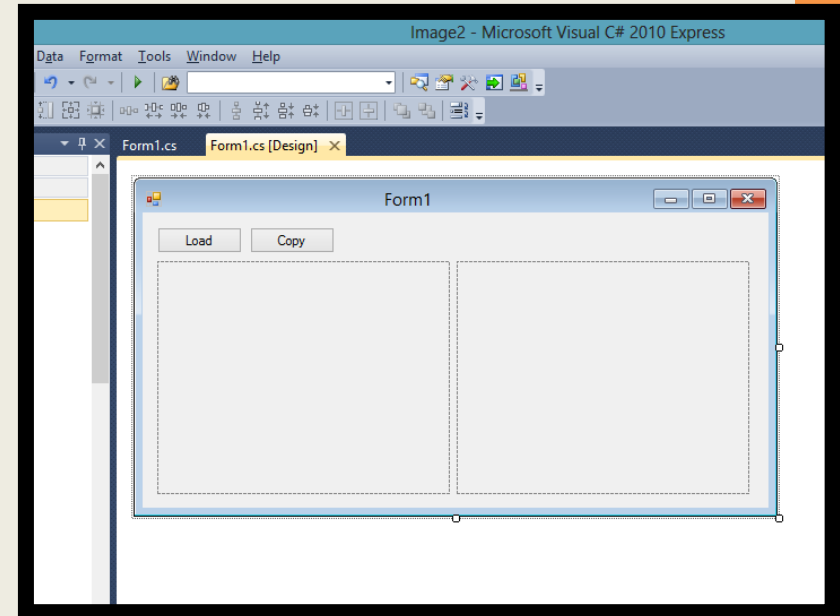
9. Double click Button1 dan tambahkan program berikut:

```
DialogResult d = openFileDialog1.ShowDialog();
if(d == DialogResult.OK)
{
    objBitmap = new Bitmap(openFileDialog1.FileName);
    pictureBox1.Image = objBitmap;
}
```

MEMBACA DAN MENGCOPY DATA GAMBAR

Berikut ini adalah langkah-langkah membuat aplikasi yang dapat mengambil data warna setiap pixel pada gambar dengan melanjutkan project Image2 yang telah dikerjakan sebelumnya.

1. Pada form di Project Image2, tambahkan 1Button dan 1PictureBox.
2. Ubah text Button2 menjadi "Copy"
3. Pada PictureBox2, atur size-mode dengan StrecthImage.
4. Atur tampilan seperti gambar disamping.



5. Tambahkan satu obyek bitmap baru di class Form1 untuk penampung data gambar. Beri nama object Bitmap baru dengan nama objBitmap1 dan letakkan setelah object Bitmap objBitmap;

```
Bitmap objBitmap1;
```

6. Double click pada Button2 dan tambahkan program berikut:

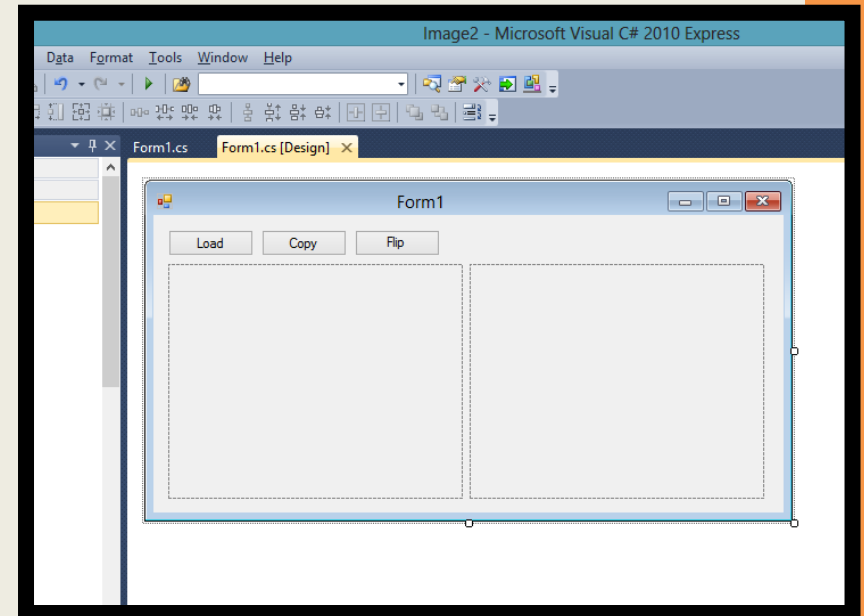
```
objBitmap1 = new Bitmap(objBitmap);  
for(int x=0; x<objBitmap.Width; x++)  
    for(int y=0; y<objBitmap.Height; y++)  
    {  
        Color w = objBitmap.GetPixel(x,y);  
        objBitmap1.SetPixel(x,y,w);  
    }  
pictureBox2.Image = objBitmap1;
```

Jalankan program kemudian tekan tombol load untuk memilih gambar dari hardisk. Kemudian tekan tombol Copy. Maka gambar akan tercopy pada PictureBox2.

FLIP HORIZONTAL

Berikut ini adalah langkah-langkah untuk mengambil data warna setiap pixel dan meletakkannya secara flip horisontal dengan melanjutkan project Image2 yang sebelumnya.

1. Pada form di Project Image2, tambahkan 1Button dan beri text "Flip Horizontal".
2. Atur tampilan seperti gambar disamping.



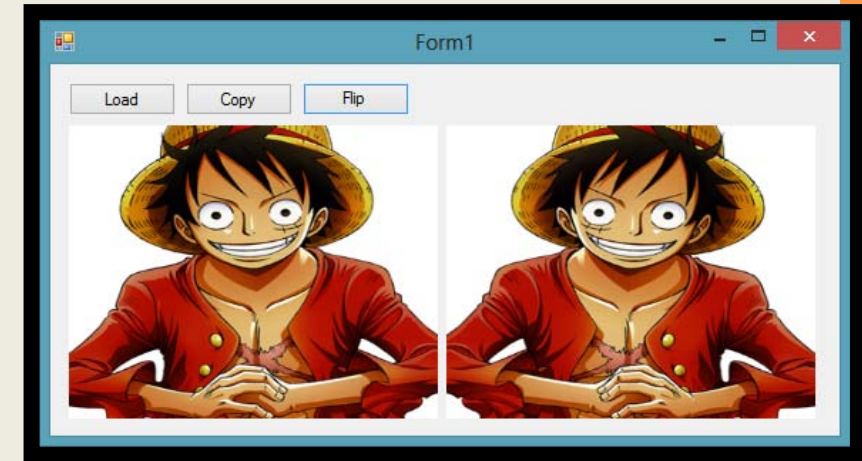
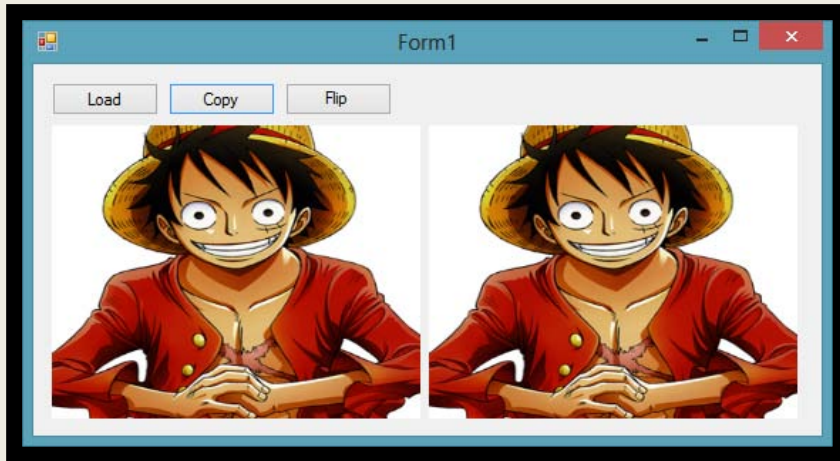
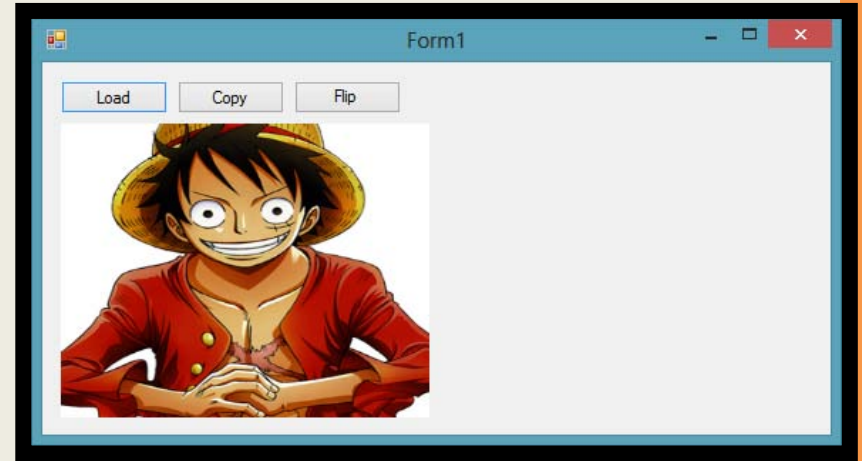
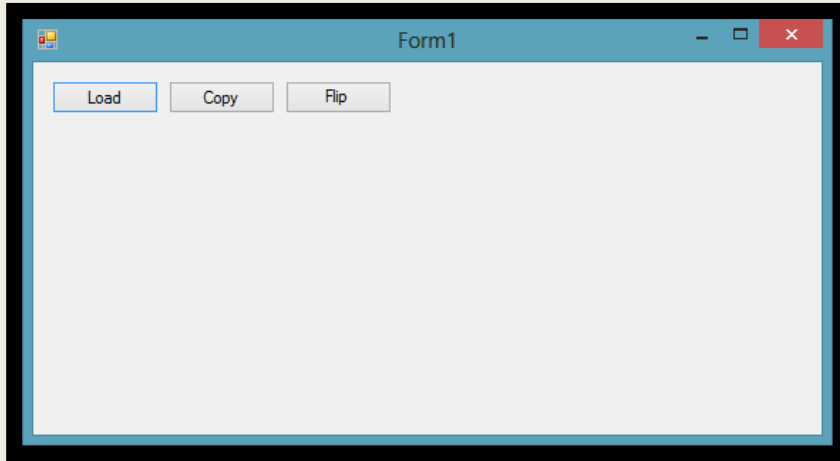
3. Double click pada Button3 (Flip Horizontal) dan tambahkan program berikut:

```
objBitmap1 = new Bitmap(objBitmap);  
for(int x=0; x<objBitmap.Width; x++)  
    for(int y=0; y<objBitmap.Height; y++)  
    {  
        Color w = objBitmap.GetPixel(x,y);  
        objBitmap1.SetPixel(objBitmap.Width-1-x,y,w);  
    }  
pictureBox2.Image = objBitmap1;
```

Jalankan program kemudian tekan tombol load untuk memilih gambar dari hardisk. Kemudian tekan tombol Copy. Maka gambar akan tercopy secara flip horizontal pada PictureBox2.

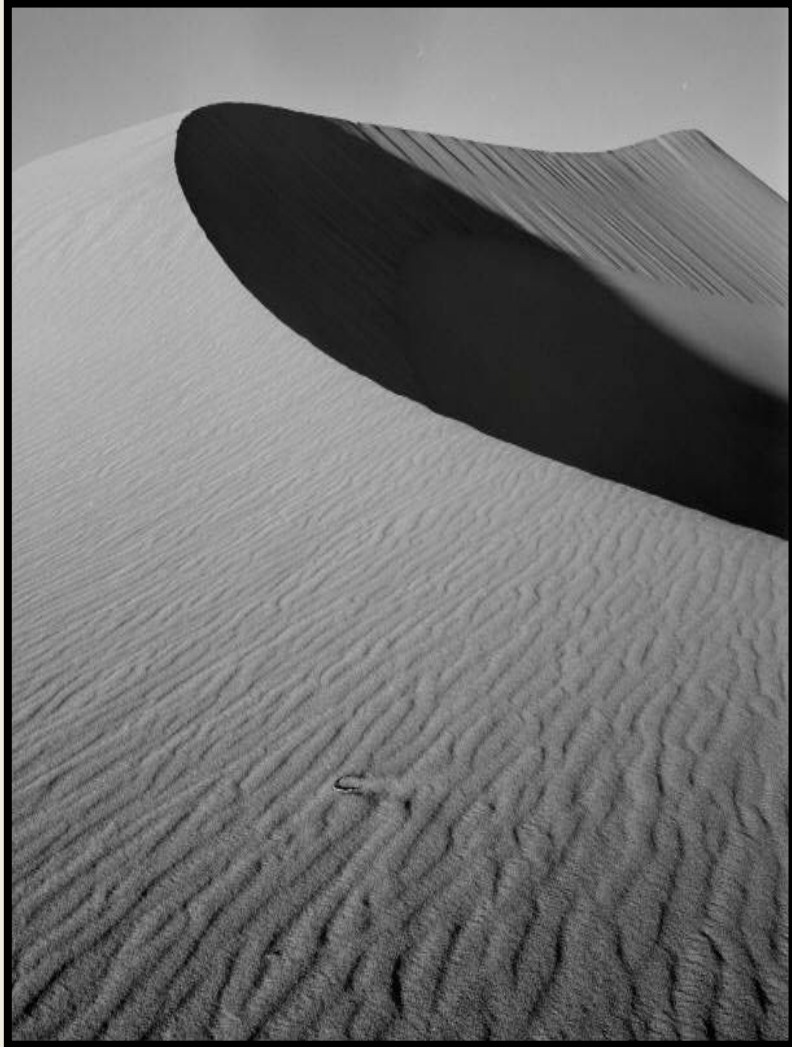
Pada dasarnya proses flip horizontal adalah meletakkan titik horizontal secara berkebalikan dimana setiap pixel x pada gambar asal (objBitmap) akan diletakkan di pixel width-(x+1) pada gambar hasil (objBitmap1).

Berikut adalah output program dari beberapa percobaan di atas:





Tugas dan Laporan praktikum



TUGAS :

1. Tambahkan menu pada project Image2 yang telah dikerjakan sebelumnya untuk Flip Vertical, Rotate 90° dan Rotate 180°

LAPORAN:

1. Tuliskan koding lengkap dari program di atas (termasuk Flip Vertical dan rotasi).
2. Berikan analisa dan keterangan pada setiap baris program yang anda anggap penting.