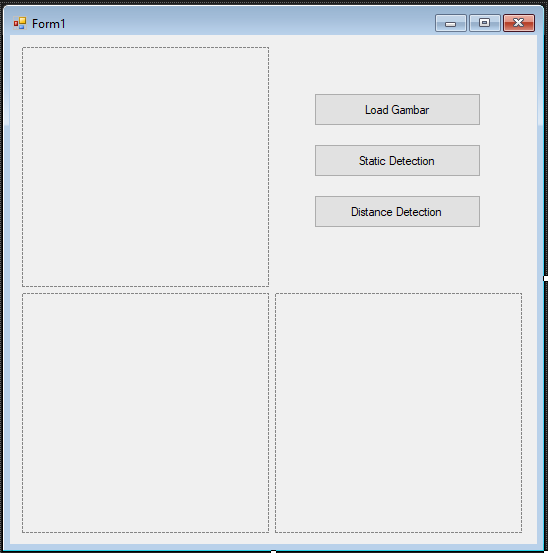
Laporan Resmi Praktikum Pengolahan Citra

Deteksi Warna

Dimas Rizky H.P. – 2110141011 – 3 D4 IT A

1. Kode Program

* Desain form aplikasi deteksi warna



*Gambar 1.0 Desain form aplikasi deteksi warna*

Pada aplikasi pendeteksi warna ini, terdapat 3 buah picture box, yaitu picturebox untuk menampilkan gambar input, menampilkan gambar yang telah di deteksi warnanya dan di filter warnanya menggunakan static detection dan distance detection.

Terdapat tiga button yang merupakan fitur dari aplikasi ini, yaitu button untuk load gambar, button untuk static color detection dan button distance color detection.

* Button Load Gambar

Load gambar digunakan untuk membuka file citra/gambar yang akan dideteksi warnanya. Gambar yang akan digunakan adalah gambar 2.0, yang akan dideteksi warna kulitnya dengan threshold detection bernilai (RED(162-194), GREEN(125-176), BLUE(117-164)).



*Gambar 2.0 Citra yang digunakan*

Kode program dapat dilihat pada box code 1, pada button load gambar sebagai berikut :

private void button1\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if(d == DialogResult.OK)

{

objBitmap = new Bitmap(openFileDialog1.FileName);

pictureBox1.Image = objBitmap;

}

}

*Box Code 1, koding button load gambar*

* Button Static Detection

Static detection digunakan untuk memfilter warna citra yang berada di luar batas/threshold yang ditentukan. Pada citra ini, threshold yang ditentukan adalah (RED(162-194), GREEN(125-176), BLUE(117-164)). Artinya, pada pixel yang berada di luar range warna tersebut akan diubah menjadi warna hitam.

Coding dapat dilihat pada box code 2, yang merupakan code untuk button static detection.

private void button2\_Click(object sender, EventArgs e)

{

objBitmap1 = new Bitmap(objBitmap);

for(int x=0; x<objBitmap.Width; x++)

{

for(int y=0; y < objBitmap1.Height; y++)

{

Color w = objBitmap.GetPixel(x, y);

int r = w.R;

int g = w.G;

int b = w.B;

//if (((r > 188) && (r < 239)) && ((g > 139) && (g < 211)) && ((b > 106) && (b < 200)))

//if (((r > 213) && (r < 239)) && ((g > 45) && (g < 118)) && ((b > 28) && (b < 87)))

if (((r >= 162) && (r <= 194)) && ((g >= 125) && (g <= 176)) && ((b >= 117) && (b <= 164)))

{

continue;

}

else

{

r = 0; g = 0; b = 0;

}

Color wb = Color.FromArgb(r, g, b);

objBitmap1.SetPixel(x, y, wb);

}

pictureBox2.Image = objBitmap1;

}

}

*Box Code 2, koding untuk button static detection*

* Button Distance Detection

Berbeda dengan static detection, distance color detection melakukan filtering warna piksel dengan menggunakan jarak bersifat dinamis yang thresholdnya juga ditentukan sebesar (RED=194, GREEN=176, BLUE=117). Dengan distance detection ini akan meningkatkan range deteksi warna secara dinamis sehingga akan menghasilkan persebaran range warna yang lebih baik.

Coding untuk button distance color detection dapat dilihat pada box code 3 yang diterapkan pada button3

private void button3\_Click(object sender, EventArgs e)

{

//int r = 162, g = 125, b = 117;

int r = 194, g = 176, b = 164;

double d;

objBitmap2 = new Bitmap(objBitmap);

for(int x=0; x < objBitmap.Width; x++)

{

for(int y=0; y< objBitmap2.Height; y++)

{

Color w = objBitmap.GetPixel(x, y);

int red = w.R;

int green = w.G;

int blue = w.B;

d = Math.Abs(red - r) + Math.Abs(green - g) + Math.Abs(blue - b);

if (d < 85)

{

continue;

} else

{

red = 0; green = 0; blue = 0;

}

Color wb = Color.FromArgb(red, green, blue);

objBitmap2.SetPixel(x, y, wb);

}

pictureBox3.Image = objBitmap2;

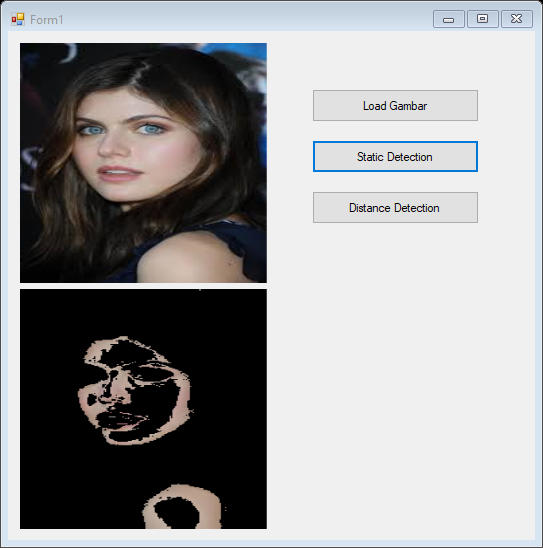
}

}

*Box Code 3, koding untuk button distance color detection*

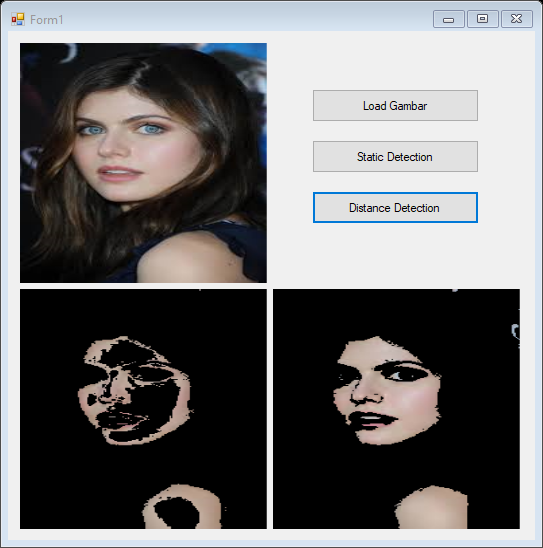
1. Output

* Static Color Detection



*Gambar 3.0, static color detection*

* Distance Color Detection



*Gambar 4.0, Distance color detection*

1. Kesimpulan

* *Static colo detection akan melakukan filter terhadap warna citra, pixel dengan kombinasi RGB yang berada di dalam range threshold akan tetap memiliki nilai warna awal, sedangkan yang di luar range threshold akan diubah menjadi hitam*
* *Distance color detection menggunakan acuan penghitungan selisih antara range warna threshold dengan warna pixel yang sedang diproses. Hal ini akan menyebabkan lebih lebarnya dan lebih mendekatinya akurasi terhadap warna yang ingin dideteksi*