**L A P O R A N**

“Pengenalan Kepribadian Seseorang Berdasarkan Bentuk Kuku”

**PRAKTIKUM PENGOLAHAN CITRA**



**Oleh :**

Nadila Wirdatul Hidayah

(2110161022)

&

Efriza Luluk Savira

(2110161021)

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**TAHUN AJARAN**

**JULI 2018**

**L A P O R A N**

* Latar Belakang

Seiring dengan perkembangan teknologi di bidang komputer saat ini, baik dalam perangkat keras (hardware) maupun perangkat lunak (software), hampir sebagian besar pekerjaan manusia kini diselesaikan dengan komputer. Dengan demikian, komputer dapat dikatakan sebagai salah satu alat bantu manusia dalam menyelesaikan suatu pekerjaan. Pemakaian komputer sering digunakan untuk hal-hal yang berkenaan dengan pemrosesan data (data processing), pengolahan kata (word processing), serta **pengolahan gambar (image processing).** Salah satu alasan, mengapa komputer cenderung digunakan sebagai alat bantu dalam menyelesaikan suatu pekerjaan karena pekerjaan yang dilakukan menggunakan komputer memiliki kecepatan proses yang lebih dapat diandalkan.

Banyak software pengolah gambar yang telah digunakan dan dikembangkan saat ini, salah satu yang paling sederhana adalah software pengolah gambar Image / Photo Editor. Namun disana kami belum menemukan aplikasi photo editor yang lebih mengarah pada deteksi object tertentu khususnya mengenai topik yang kita ambil ini. Karena semua aplikasi yang ada disana rata-rata adalah aplikasi yang memang dibutuhkan untuk editing foto.

Maka dari itu, kami memutuskan untuk membuat aplikasi berbasis desktop yang dimana dapat mendeteksi suatu bentuk kuku yang telah dimasukkan oleh user sebelumnya. Disini user akan memasukkan 9 sampel berbagai macam bentuk kuku dan satu citra uji untuk menentukan kuku seseorang tersebut termasuk dalam bentuk kuku yang mana. Beserta kami akan menampilkan sifat seseorang menurut bentuk kuku yang terpilih.

* Algoritma Penyelesaian

Pada bab ini akan dibahas mengenai rancangan sistem aplikasi dan implementasi dari aplikasi deteksi bentuk kuku untuk menrntukan kepribadian seseorang. Aplikasi ini menggunakan bahasa pemograman C# dengan bantuan aplikasi Microsoft Visual Studio 2010. Untuk memudahkan pemahaman setiap program dalam sistem ini, maka dibuatlah algoritma yang merepresentasikan alur kerja setiap program. Berikut adalah alur algoritma yang kami terapkan.

Inisialisai objBitmap sebanyak 10 dan d1,d2,...d9

.

Masukkan 1 Citrra Uji yang berbeda dari 9 sampel sebelumnya

User memasukkan 9 Citra Sampel

Diambil Pixel RGB dari tiap object

Hitung nilai d1, d2, d3, d4, ..., d9

Menentukan keanggotaan citra baru

**KETERANGAN** :

Aplikasi ini diawali dengan pembuatan 10 objBitmap beserta variabel-variabel lain yang dibutuhkan untuk pembuatan program ini. Kemudian user memasukkan data citra berupa 9 sampel bentuk kuku yang berbeda-beda. Proses selanjutnya adalah user memasukkan citra baru yang berbeda dari 9 sampel yang telah dimasukkan sebelumnya . Lalu diambil pixel RGB dari tiap gambar sebelumnya yg telah dimasukkan ke dalam objBitmap tiap masing-masing gambar. Setelah itu hitung nilai d1, d2, d3, d4, d5, d6, d7, d8, d9 berdasarkan gambar yang telah dimasukkan tadi.

Lalu langkah selanjutnya adalah menentukan keanggotan dari citra baru yang kita masukkan sesudah 9 sampel sebelumnya. Menentukan keanggotaan tersebut adalah dengan menggunakan nilai d1, d, d3,...d9. Algoritmanya adalah jika d1 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-1 dan mempunyai sifat “pendiam dan romantis”. Lalu jika d2 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-2 dan mempunyai sifat “pintar dan bersemangat”.

Dan jika d3 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-3 dan mempunyai sifat “santai dan cinta damai”. Sedangkan jika d4 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-4 dan mempunyai sifat “santai dan cinta damai”. Kondisi selanjutnya, jika d5 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-5 dan mempunyai sifat “serius dan keras kepala”.

Lalu, jika d6 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-6 dan mempunyai sifat “jenius dan sensitif”. Namun, jika d7 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-7 dan mempunyai sifat “jenius dan sensitif”. Jika d8 lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-8 dan mempunyai sifat “setia dan jujur”. Kondisi terakhir yaitu jika d9 adalah lebih kecil dari semua diantara nilai d yang ada, maka citra baru tersebut termasuk dalam kategori bentuk kuku ke-9 dan mempunyai sifat “idealis dan berambisi”. Itulah berbagai macam kondisi yang ada dalam algoritma yang kami terapkan didalamnya.

* Implementasi

Disini kami menggunakan Microsoft Visual Studio 2010 untuk mengimplementasikan algoritma diatas ke dalam program yang sedemikian rupa. Nantinya program inilah yang akan digunakan oleh user untuk menentukan kepribadian seseorang berdasarkan bentuk kuku. Berikut adalah kode program yang telah kami buat sebelumnya.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Week\_13

{

public partial class Form1 : Form

{

Bitmap objBitmap1;

Bitmap objBitmap2;

Bitmap objBitmap3;

Bitmap objBitmap4;

Bitmap objBitmap5;

Bitmap objBitmap6;

Bitmap objBitmap7;

Bitmap objBitmap8;

Bitmap objBitmap9;

Bitmap objBitmap10;

int[,] Citra1 = new int[300, 300];

int[,] Citra2 = new int[300, 300];

int[,] Citra3 = new int[300, 300];

int[,] Citra4 = new int[300, 300];

int[,] Citra5 = new int[300, 300];

int[,] Citra6 = new int[300, 300];

int[,] Citra7 = new int[300, 300];

int[,] Citra8 = new int[300, 300];

int[,] Citra9 = new int[300, 300];

int[,] CitraBaru = new int[300, 300];

double d1, d2, d3, d4, d5, d6, d7, d8, d9;

public Form1()

{

InitializeComponent();

}

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

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap1 = new Bitmap(openFileDialog1.FileName);

pictureBox1.Image = objBitmap1;

}

}

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

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap2 = new Bitmap(openFileDialog1.FileName);

pictureBox2.Image = objBitmap2;

}

}

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

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap3 = new Bitmap(openFileDialog1.FileName);

pictureBox3.Image = objBitmap3;

}

}

private void button4\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap10 = new Bitmap(openFileDialog1.FileName);

pictureBox4.Image = objBitmap10;

}

}

private void button5\_Click(object sender, EventArgs e)

{

for (int x = 0; x < objBitmap1.Width; x++)

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

{

Color wc1 = objBitmap1.GetPixel(x, y);

int wc1r = wc1.R;

int wc1g = wc1.G;

int wc1b = wc1.B;

int c1 = (int)((wc1r + wc1g + wc1b) / 3);

Citra1[x, y] = (int)c1;

}

for (int x = 0; x < objBitmap2.Width; x++)

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

{

Color wc2 = objBitmap2.GetPixel(x, y);

int wc2r = wc2.R;

int wc2g = wc2.G;

int wc2b = wc2.B;

int c2 = (int)((wc2r + wc2g + wc2b) / 3);

Citra2[x, y] = (int)c2;

}

for (int x = 0; x < objBitmap3.Width; x++)

for (int y = 0; y < objBitmap3.Height; y++)

{

Color wc3 = objBitmap3.GetPixel(x, y);

int wc3r = wc3.R;

int wc3g = wc3.G;

int wc3b = wc3.B;

int c3 = (int)((wc3r + wc3g + wc3b) / 3);

Citra3[x, y] = (int)c3;

}

for (int x = 0; x < objBitmap4.Width; x++)

for (int y = 0; y < objBitmap4.Height; y++)

{

Color wc4 = objBitmap4.GetPixel(x, y);

int wc4r = wc4.R;

int wc4g = wc4.G;

int wc4b = wc4.B;

int c4 = (int)((wc4r + wc4g + wc4b) / 3);

Citra4[x, y] = (int)c4;

}

for (int x = 0; x < objBitmap5.Width; x++)

for (int y = 0; y < objBitmap5.Height; y++)

{

Color wc5 = objBitmap5.GetPixel(x, y);

int wc5r = wc5.R;

int wc5g = wc5.G;

int wc5b = wc5.B;

int c5 = (int)((wc5r + wc5g + wc5b) / 3);

Citra5[x, y] = (int)c5;

}

for (int x = 0; x < objBitmap6.Width; x++)

for (int y = 0; y < objBitmap6.Height; y++)

{

Color wc6 = objBitmap6.GetPixel(x, y);

int wc6r = wc6.R;

int wc6g = wc6.G;

int wc6b = wc6.B;

int c6 = (int)((wc6r + wc6g + wc6b) / 3);

Citra6[x, y] = (int)c6;

}

for (int x = 0; x < objBitmap7.Width; x++)

for (int y = 0; y < objBitmap7.Height; y++)

{

Color wc7 = objBitmap7.GetPixel(x, y);

int wc7r = wc7.R;

int wc7g = wc7.G;

int wc7b = wc7.B;

int c7 = (int)((wc7r + wc7g + wc7b) / 3);

Citra7[x, y] = (int)c7;

}

for (int x = 0; x < objBitmap8.Width; x++)

for (int y = 0; y < objBitmap8.Height; y++)

{

Color wc8 = objBitmap8.GetPixel(x, y);

int wc8r = wc8.R;

int wc8g = wc8.G;

int wc8b = wc8.B;

int c8 = (int)((wc8r + wc8g + wc8b) / 3);

Citra8[x, y] = (int)c8;

}

for (int x = 0; x < objBitmap9.Width; x++)

for (int y = 0; y < objBitmap9.Height; y++)

{

Color wc9 = objBitmap9.GetPixel(x, y);

int wc9r = wc9.R;

int wc9g = wc9.G;

int wc9b = wc9.B;

int c9 = (int)((wc9r + wc9g + wc9b) / 3);

Citra9[x, y] = (int)c9;

}

for (int x = 0; x < objBitmap10.Width; x++)

for (int y = 0; y < objBitmap10.Height; y++)

{

Color wcb = objBitmap10.GetPixel(x, y);

int wcbr = wcb.R;

int wcbg = wcb.G;

int wcbb = wcb.B;

int cb = (int)((wcbr + wcbg + wcbb) / 3);

CitraBaru[x, y] = (int)cb;

}

for (int x = 0; x < objBitmap4.Width; x++)

for (int y = 0; y < objBitmap4.Height; y++)

{

Color wc4 = objBitmap4.GetPixel(x, y);

int wc4r = wc4.R;

int wc4g = wc4.G;

int wc4b = wc4.B;

int c4 = (int)((wc4r + wc4g + wc4b) / 3);

Citra4[x, y] = (int)c4;

}

for (int x = 0; x < objBitmap5.Width; x++)

for (int y = 0; y < objBitmap5.Height; y++)

{

Color wc5 = objBitmap5.GetPixel(x, y);

int wc5r = wc5.R;

int wc5g = wc5.G;

int wc5b = wc5.B;

int c5 = (int)((wc5r + wc5g + wc5b) / 3);

Citra5[x, y] = (int)c5;

}

for (int x = 0; x < objBitmap6.Width; x++)

for (int y = 0; y < objBitmap6.Height; y++)

{

Color wc6 = objBitmap6.GetPixel(x, y);

int wc6r = wc6.R;

int wc6g = wc6.G;

int wc6b = wc6.B;

int c6 = (int)((wc6r + wc6g + wc6b) / 3);

Citra6[x, y] = (int)c6;

}

for (int x = 0; x < objBitmap7.Width; x++)

for (int y = 0; y < objBitmap7.Height; y++)

{

Color wc7 = objBitmap7.GetPixel(x, y);

int wc7r = wc7.R;

int wc7g = wc7.G;

int wc7b = wc7.B;

int c7 = (int)((wc7r + wc7g + wc7b) / 3);

Citra7[x, y] = (int)c7;

}

for (int x = 0; x < objBitmap8.Width; x++)

for (int y = 0; y < objBitmap8.Height; y++)

{

Color wc8 = objBitmap8.GetPixel(x, y);

int wc8r = wc8.R;

int wc8g = wc8.G;

int wc8b = wc8.B;

int c8 = (int)((wc8r + wc8g + wc8b) / 3);

Citra8[x, y] = (int)c8;

}

for (int x = 0; x < objBitmap9.Width; x++)

for (int y = 0; y < objBitmap9.Height; y++)

{

Color wc9 = objBitmap9.GetPixel(x, y);

int wc9r = wc9.R;

int wc9g = wc9.G;

int wc9b = wc9.B;

int c9 = (int)((wc9r + wc9g + wc9b) / 3);

Citra9[x, y] = (int)c9;

}

for (int x = 0; x < objBitmap10.Width; x++)

for (int y = 0; y < objBitmap10.Height; y++)

{

Color wcb = objBitmap10.GetPixel(x, y);

int wcbr = wcb.R;

int wcbg = wcb.G;

int wcbb = wcb.B;

int cb = (int)((wcbr + wcbg + wcbb) / 3);

CitraBaru[x, y] = (int)cb;

}

d1 = 0;

d2 = 0;

d3 = 0;

d4 = 0;

d5 = 0;

d6 = 0;

d7 = 0;

d8 = 0;

d9 = 0;

for (int x = 0; x < objBitmap10.Width; x++)

for (int y = 0; y < objBitmap10.Height; y++)

{

d1 = d1 + ((Citra1[x, y] - CitraBaru[x, y]) \* (Citra1[x, y] - CitraBaru[x, y]));

d2 = d2 + ((Citra2[x, y] - CitraBaru[x, y]) \* (Citra2[x, y] - CitraBaru[x, y]));

d3 = d3 + ((Citra3[x, y] - CitraBaru[x, y]) \* (Citra3[x, y] - CitraBaru[x, y]));

d4 = d4 + ((Citra4[x, y] - CitraBaru[x, y]) \* (Citra4[x, y] - CitraBaru[x, y]));

d5 = d5 + ((Citra5[x, y] - CitraBaru[x, y]) \* (Citra5[x, y] - CitraBaru[x, y]));

d6 = d6 + ((Citra6[x, y] - CitraBaru[x, y]) \* (Citra6[x, y] - CitraBaru[x, y]));

d7 = d7 + ((Citra7[x, y] - CitraBaru[x, y]) \* (Citra7[x, y] - CitraBaru[x, y]));

d8 = d8 + ((Citra8[x, y] - CitraBaru[x, y]) \* (Citra8[x, y] - CitraBaru[x, y]));

d9 = d9 + ((Citra9[x, y] - CitraBaru[x, y]) \* (Citra9[x, y] - CitraBaru[x, y]));

}

d1 = Math.Sqrt(d1);

d2 = Math.Sqrt(d2);

d3 = Math.Sqrt(d3);

d4 = Math.Sqrt(d4);

d5 = Math.Sqrt(d5);

d6 = Math.Sqrt(d6);

d7 = Math.Sqrt(d7);

d8 = Math.Sqrt(d8);

d9 = Math.Sqrt(d9);

textBox1.Text = d1.ToString();

textBox2.Text = d2.ToString();

textBox3.Text = d3.ToString();

textBox4.Text = d4.ToString();

textBox5.Text = d5.ToString();

textBox6.Text = d6.ToString();

textBox7.Text = d7.ToString();

textBox8.Text = d8.ToString();

textBox9.Text = d9.ToString();

if ((d1 < d2) && (d1 < d3) && (d1 < d4) && (d1 < d5) && (d1 < d6) && (d1 < d7) && (d1 < d8) && (d1 < d9))

{

label1.Text = System.Convert.ToString("1. Orangnya pendiam dan romantis");

}

else if ((d2 < d1) && (d2 < d3) && (d2 < d4) && (d2 < d5) && (d2 < d6) && (d2 < d7) && (d2 < d8) && (d2 < d9))

{

label1.Text = System.Convert.ToString("2. Orangnya pintar dan bersemangat");

}

else if ((d3 < d1) && (d3 < d2) && (d3 < d4) && (d3 < d5) && (d3 < d6) && (d3 < d7) && (d3 < d8) && (d3 < d9))

{

label1.Text = System.Convert.ToString("3. Orangnya santai dan cinta damai");

}

else if ((d4 < d1) && (d4 < d2) && (d4 < d3) && (d4 < d5) && (d4 < d6) && (d4 < d7) && (d4 < d8) && (d4 < d9))

{

label1.Text = System.Convert.ToString("4. Orangnya santai dan cinta damai");

}

else if ((d5 < d1) && (d5 < d2) && (d5 < d3) && (d5 < d4) && (d5 < d6) && (d5 < d7) && (d5 < d8) && (d5 < d9))

{

label1.Text = System.Convert.ToString("5. Orangnya serius dan keras kepala");

}

else if ((d6 < d1) && (d6 < d2) && (d6 < d3) && (d6 < d4) && (d6 < d5) && (d6 < d7) && (d6 < d8) && (d6 < d9))

{

label1.Text = System.Convert.ToString("6. Orangnya jenius dan sensitif");

}

else if ((d7 < d1) && (d7 < d2) && (d7 < d3) && (d7 < d4) && (d7 < d5) && (d7 < d6) && (d7 < d8) && (d7 < d9))

{

label1.Text = System.Convert.ToString("7. Orangnya jenius dan sensitif");

}

else if ((d8 < d1) && (d8 < d2) && (d8 < d3) && (d8 < d4) && (d8 < d5) && (d8 < d6) && (d8 < d7) && (d8 < d9))

{

label1.Text = System.Convert.ToString("8. Orangnya setia dan jujur");

}

else

{

label1.Text = System.Convert.ToString("9. Orangnya idealis dan berambisi");

}

}

private void button8\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap4 = new Bitmap(openFileDialog1.FileName);

pictureBox7.Image = objBitmap4;

}

}

private void button7\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap5 = new Bitmap(openFileDialog1.FileName);

pictureBox6.Image = objBitmap5;

}

}

private void button6\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap6 = new Bitmap(openFileDialog1.FileName);

pictureBox5.Image = objBitmap6;

}

}

private void button11\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap9 = new Bitmap(openFileDialog1.FileName);

pictureBox10.Image = objBitmap9;

}

}

private void button9\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap7 = new Bitmap(openFileDialog1.FileName);

pictureBox8.Image = objBitmap7;

}

}

private void button10\_Click(object sender, EventArgs e)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap8 = new Bitmap(openFileDialog1.FileName);

pictureBox9.Image = objBitmap8;

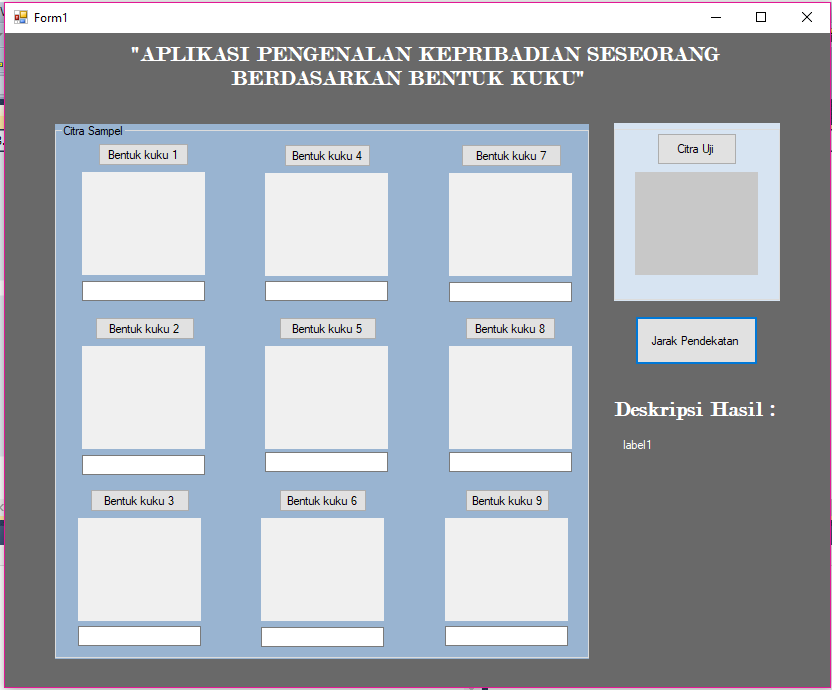
}

}

* Hasil Uji Coba

Setelah aplikasi selesai dibuat, maka alangkah baiknya untuk dilakukan pengujian beberapa sampel yang dimasukkan dalam pengujian. Agar terlihat apakah program telah sesuai dengan yang dirancang tadinya. Sebelum pengujian dilakukan, akan kami tunjukkan tampilan aplikasi yang kami buat. Beserta beberapa pengujian yang akan kami lakukan disitu.

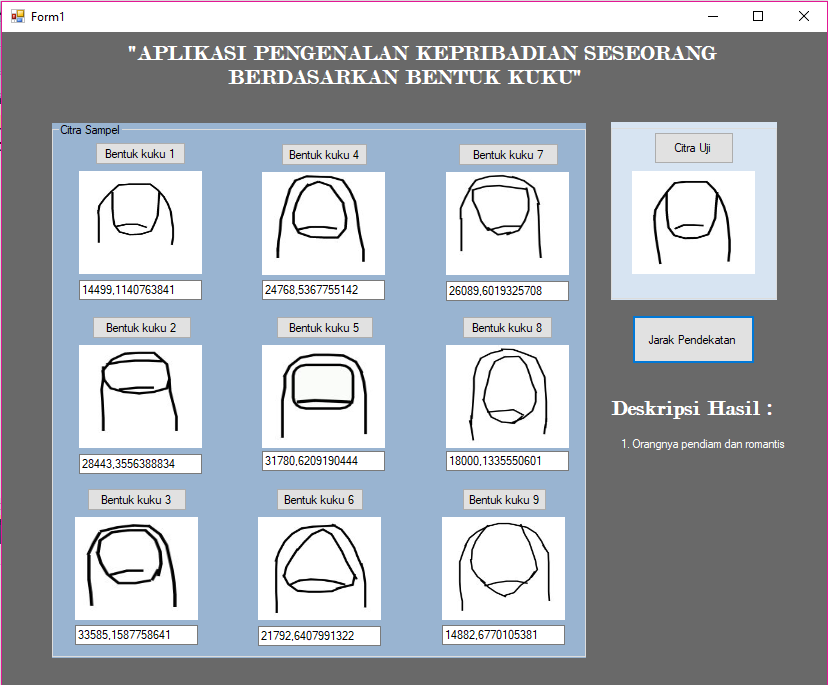
Berikut adalah tampilan utama aplikasi.



Dan berikut adalah beberapa pengujian yang kami lakukan.

1. **Uji Pertama**

Pada pengujian pertama ini, saya memasukkan 9 sampel bentuk kuku yang berbeda-beda. Lalu untuk citra baru yang saya ujikan ini termasuk kuku kategori bentuk pertama. Maka dari itu, output yang dihasilkan program adalah sedemikian rupa. Pada program ini, bentuk kuku pertama adalah keprobadian seseorang dengan sifat pendiam dan romantis. Maka ketika user memasukkan citra uji yang termasuk dalam kategori pertama, akan menampilkan sifat tersebut. Seperti pada gambar dibawah ini.



1. **Uji Kedua**

Pada pengujian pertama ini, saya memasukkan 9 sampel bentuk kuku yang berbeda-beda. Lalu untuk citra baru yang saya ujikan ini termasuk kuku kategori bentuk kedua. Maka dari itu, output yang dihasilkan program adalah sedemikian rupa. Pada program ini, bentuk kuku kedua adalah kepribadian seseorang dengan sifat pintar dan bersemangat. Maka ketika user memasukkan citra uji yang termasuk dalam kategori kedua, akan menampilkan sifat tersebut. Seperti pada gambar dibawah ini.

