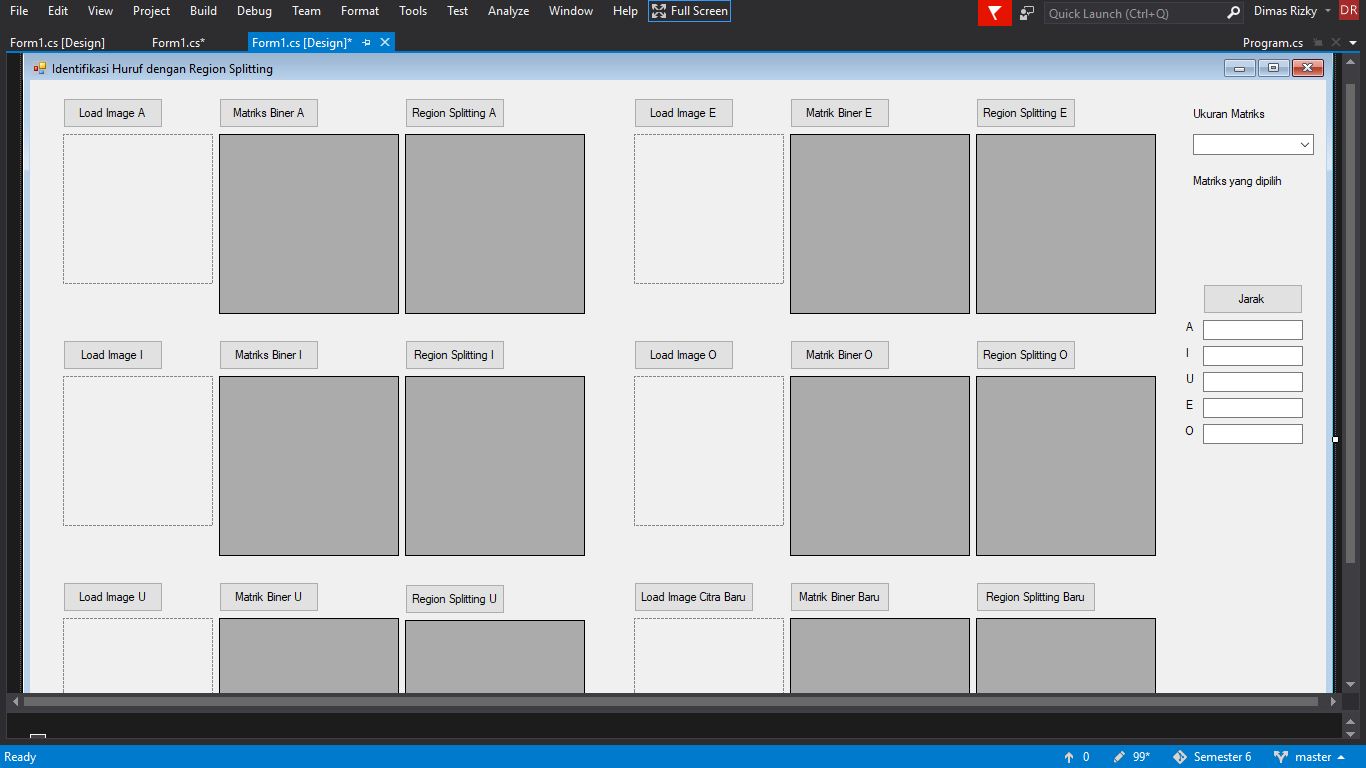
Laporan Resmi – Praktikum Pengolahan Citra

Region Splitting, Template Matching dan Segmentasi Citra

Dimas Rizky H.P. – 2110141011

1. Design program & source code

* Region Splitting



*Gambar 1.0, desain program untuk program region splitting*

Program region splitting menerima input 5 buah input untuk data awal yaitu huruf vokal A,I,U,E,O dan input citra lainnya yang ingin dibandingkan dengan citra data awal sebelumnya. Dalam melakukan identifikasi, terdapat ukuran matriks yang dapat digunakan mulai dari matriks 2x2, sampai dengan 5x5.

Jika program telah dijalankan, dan semua citra database di-*load*, serta citra baru di masukan, program akan menghitung jarak antara citra baru dan citra database satu per satu sehingga akan diketahui jarak antara citra baru dan citra yang ada pada database. Jarak dengan nilai paling kecil merupakan citra yang memiliki kemiripan paling besar diantara citra database lainnya.

* Source Code

Setiap komponen load citra pada program memiliki masing masing, 3 button, 1 picturebox dan 2 gridview. Button yang ada adalah, load image, matriks biner dan region splitting

1. Global variable

Bitmap objBitmapA, objBitmapI, objBitmapU,

objBitmapE, objBitmapO, objBitmapBaru;

int[,] MatriksA = new int[100, 100];

int[,] MatriksI = new int[100, 100];

int[,] MatriksU = new int[100, 100];

int[,] MatriksE = new int[100, 100];

int[,] MatriksO = new int[100, 100];

int[,] MatriksBaru = new int[100, 100];

int selectedMatriks, val;

int[] reg, regA, regI, regU, regE, regO, regBaru;

double dA, dI, dU, dE, dO;

1. Button load image

private Bitmap LoadImage(Bitmap objBitmap, PictureBox pictureBox)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap = new Bitmap(openFileDialog1.FileName);

pictureBox.Image = objBitmap;

}

return objBitmap;

}

1. Button matriks biner

private int[,] MatriksBiner(Bitmap objBitmap, int[,] Matriks, DataGridView dataGridView)

{

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

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

{

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

int wr = w.R;

int wg = w.G;

int wb = w.B;

int c = (int)((wr + wg + wb) / 3);

if (c > 127)

c = 0;

else

c = 255;

c = (int)c / 255;

Matriks[x, y] = c;

}

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

{

DataGridViewColumn newCol = new DataGridViewColumn();

DataGridViewCell cell = new DataGridViewTextBoxCell();

newCol.CellTemplate = cell;

newCol.HeaderText = (i + 1).ToString();

newCol.Name = "Oke";

newCol.Visible = true;

newCol.Width = 100;

dataGridView.Columns.Add(newCol);

dataGridView.Rows.Add(cell);

}

dataGridView.Rows.Add(objBitmap.Height);

for (int i = 0; i < objBitmap.Height; i++)

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

dataGridView.Rows[i].Cells[j].Value = Matriks[j, i].ToString();

return Matriks;

}

1. Buttin region splitting

private int[] RegionSplit(int[,] Matriks, int row, DataGridView dataGridView)

{

if (row == 2)

reg = new int[4];

else if (row == 5)

reg = new int[25];

int count = 0; int bb = 0; int ba = val;

for (int i = 0; i < row; i++)

{

int kb = 0; int ka = val; int sum = 0;

for (int j = 0; j < row; j++)

{

DataGridViewColumn newCol = new DataGridViewColumn();

DataGridViewCell cell = new DataGridViewTextBoxCell();

newCol.CellTemplate = cell;

newCol.Name = "Column";

newCol.Visible = true;

dataGridView.Columns.Add(newCol);

dataGridView.Rows.Add();

for (int x = bb; x < ba; x++)

for (int y = kb; y < ka; y++)

sum = sum + Matriks[x, y];

reg[count] = sum;

dataGridView[i, j].Value = reg[count].ToString();

kb = ka + 1;

ka = ka + val;

count = count + 1;

sum = 0;

}

bb = ba + 1;

ba = ba + val;

}

return reg;

}

1. Button jarak

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

{

dA = 0; dI = 0; dU = 0; dE = 0; dO = 0;

for (int i = 0; i < reg.Length; i++)

{

dA = dA + ((regA[i] - regBaru[i]) \* (regA[i] - regBaru[i]));

dI = dI + ((regI[i] - regBaru[i]) \* (regI[i] - regBaru[i]));

dU = dU + ((regU[i] - regBaru[i]) \* (regU[i] - regBaru[i]));

dE = dE + ((regE[i] - regBaru[i]) \* (regE[i] - regBaru[i]));

dO = dO + ((regO[i] - regBaru[i]) \* (regO[i] - regBaru[i]));

}

dA = Math.Sqrt(dA); dI = Math.Sqrt(dI); dU = Math.Sqrt(dU); dE = Math.Sqrt(dE); dO = Math.Sqrt(dO);

textBox1.Text = dA.ToString();

textBox2.Text = dI.ToString();

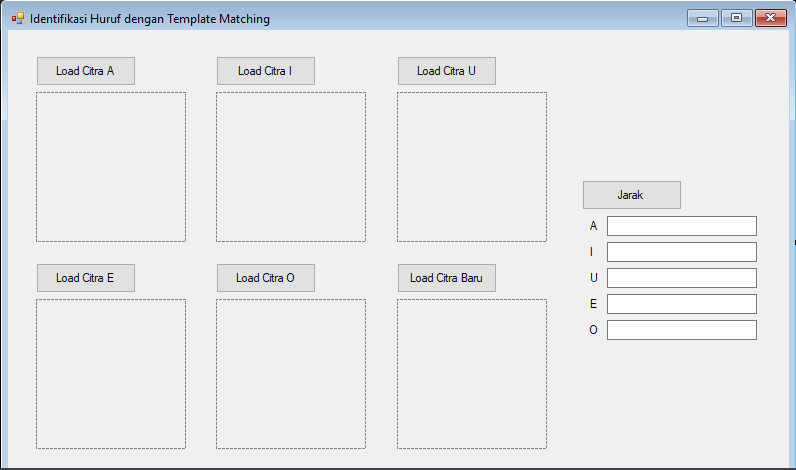
textBox3.Text = dU.ToString();

textBox4.Text = dE.ToString();

textBox5.Text = dO.ToString();

}

* Template Matching



*Gambar 1.1, desain program untuk program template matching*

Pada template matching hampir sama dengan program region splitting, namun template matching tidak perlu menggunakan matriks tertentu untuk menghitung jarak antara citra database dan citra baru yang akan di identifikasi.

Semakin kecil jarak yang dihasilkan (mendekati 0) berarti citra tersebut memiliki tingkat kemiripan yang lebih tinggi

* Source code

Program ini memiliki 2 komponen utama untuk masing masing segmen citra nya, yaitu button load dan picturebox. Juga terdapat button jarak untuk menghitung jarak antara citra baru dan citra database

1. Global variable

Bitmap objBitmapA, objBitmapI, objBitmapU,

objBitmapE, objBitmapO, objBitmapNew;

int[,] CitraA = new int[100, 100];

int[,] CitraI = new int[100, 100];

int[,] CitraU = new int[100, 100];

int[,] CitraE = new int[100, 100];

int[,] CitraO = new int[100, 100];

int[,] CitraNew = new int[100, 100];

double dA, dI, dU, dE, dO;

1. Button load

private Bitmap LoadImage(Bitmap objBitmap, PictureBox pictureBox)

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap = new Bitmap(openFileDialog1.FileName);

pictureBox.Image = objBitmap;

}

return objBitmap;

}

1. Button jarak

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

{

EkstraksiCitra(objBitmapA, CitraA);

EkstraksiCitra(objBitmapI, CitraI);

EkstraksiCitra(objBitmapU, CitraU);

EkstraksiCitra(objBitmapE, CitraE);

EkstraksiCitra(objBitmapO, CitraO);

EkstraksiCitra(objBitmapNew, CitraNew);

dA = 0; dI = 0; dU = 0; dE = 0; dO = 0;

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

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

{

dA = dA + ((CitraA[x, y] - CitraNew[x, y]) \* (CitraA[x, y] - CitraNew[x, y]));

dI = dI + ((CitraI[x, y] - CitraNew[x, y]) \* (CitraI[x, y] - CitraNew[x, y]));

dU = dU + ((CitraU[x, y] - CitraNew[x, y]) \* (CitraU[x, y] - CitraNew[x, y]));

dE = dE + ((CitraE[x, y] - CitraNew[x, y]) \* (CitraE[x, y] - CitraNew[x, y]));

dO = dO + ((CitraO[x, y] - CitraNew[x, y]) \* (CitraO[x, y] - CitraNew[x, y]));

}

dA = Math.Sqrt(dA); dI = Math.Sqrt(dI); dU = Math.Sqrt(dU); dE = Math.Sqrt(dE); dO = Math.Sqrt(dO);

textBoxA.Text = dA.ToString();

textBoxI.Text = dI.ToString();

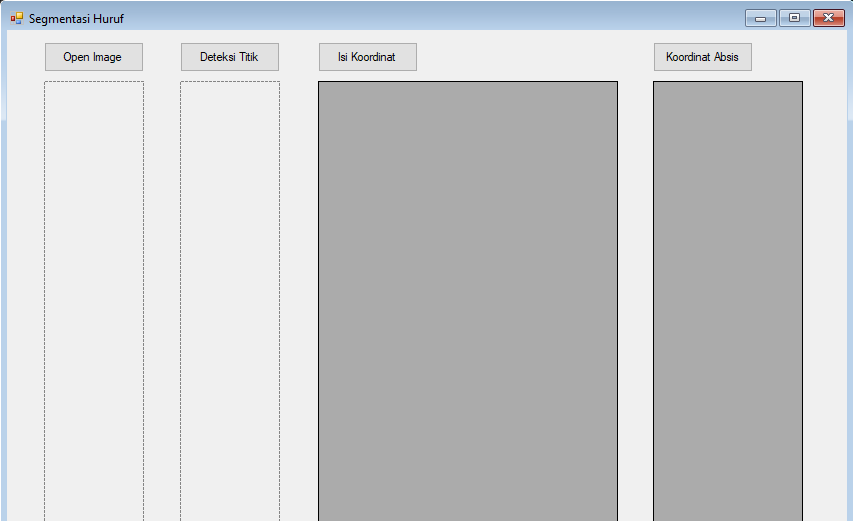
textBoxU.Text = dU.ToString();

textBoxE.Text = dE.ToString();

textBoxO.Text = dO.ToString();

}

* Segmentasi Huruf



*Gambar 1.2, desain program pada program segmentasi huruf*

Segmentasi huruf akan bekerja dengan cara mendeteksi tepi tepi citra input, lalu mengkonversikannya dalam bentuk biner di dalam tabel koordinat. Koordinat yang terbentuk akan memiliki nilai angka 1 atau 0. 1 merepresentasikan adanya titik atau adanya tepi dari citra, sedangkan 0 merupakan *fill* citra. Sehingga jika dilihat dengan seksama, koordinat yang dihasilkan akan membentuk formasi huruf/citra seperti yang ada pada citra input

* Source code

Program segmentasi huruf memiliki 2 picture box berukuran 100x500 dan 2 gridview untuk menampilkan koordinat dan tabel absisnya, serta memiliki 4 button untuk masing masing interaksi, yaitu open image, deteksi titik, isi koordinat dan koordinat absis

1. Global variable

Bitmap objBitmap1, objBitmap2;

int[,] matriks = new int[100,500];

int[] koor\_absis = new int[500];

int[,] koordinat = new int[2, 500];

String[] koor = {"X", "Y"};

int x = 0, y = 0;

int absis = 0;

1. Btn open gambar

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

{

DialogResult d = openFileDialog1.ShowDialog();

if (d == DialogResult.OK)

{

objBitmap1 = new Bitmap(openFileDialog1.FileName);

pictureBox1.Image = objBitmap1;

}

}

1. Button deteksi titik

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

{

objBitmap2 = new Bitmap(objBitmap1);

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

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

{

int sumwr = 0; int sumwg = 0; int sumwb = 0;

int[] k = { -1, -1, -1, -1, 8, -1, -1, -1, -1 };

if ((x > 0 && y > 0) && (x < objBitmap1.Width - 1 && y < objBitmap1.Height - 1))

{

int hitung = 0;

for (int i = -1; i <= 1; i++)

for (int j = -1; j <= 1; j++)

{

Color w = objBitmap1.GetPixel(x + i, y + j);

int wr = k[hitung] \* w.R;

int wg = k[hitung] \* w.G;

int wb = k[hitung] \* w.B;

sumwr += wr; sumwg += wg; sumwb += wb;

hitung++;

}

}

int rr = sumwr; int gg = sumwg; int bb = sumwb;

if (rr > 255) rr = 255; if (rr < 0) rr = 0;

if (gg > 255) gg = 255; if (gg < 0) gg = 0;

if (bb > 255) bb = 255; if (bb < 0) bb = 0;

int xg = (int)(rr + gg + bb / 3);

if (xg > 127) xg = 255;

else xg = 0;

matriks[x, y] = (int)xg / 255;

Color new\_w = Color.FromArgb(rr, gg, bb);

objBitmap2.SetPixel(x, y, new\_w);

}

pictureBox2.Image = objBitmap2;

}

1. Button isi koordinat

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

{

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

{

DataGridViewColumn newCol = new DataGridViewColumn();

DataGridViewCell cell = new DataGridViewTextBoxCell();

newCol.CellTemplate = cell;

newCol.HeaderText = (i + 1).ToString();

newCol.Visible = true;

newCol.Width = 100;

dataGridView1.Columns.Add(newCol);

dataGridView1.Rows.Add(cell);

}

dataGridView1.Rows.Add(objBitmap2.Height);

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

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

dataGridView1.Rows[i].Cells[j].Value = matriks[j, i].ToString();

}

1. Button koordinat absis

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

{

Boolean cek = false;

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

{

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

{

if (matriks[j, i] == 1)

absis = i;

else if (matriks[j, i] == 0)

cek = true;

}

if (cek && absis != 0)

koor\_absis[i] = absis;

absis = 0;

}

DataGridViewColumn newCol = new DataGridViewColumn();

DataGridViewCell cell = new DataGridViewTextBoxCell();

newCol.CellTemplate = cell;

newCol.HeaderText = "Koordinat X";

newCol.Visible = true;

dataGridView2.Columns.Add(newCol);

dataGridView2.Rows.Add(cell);

dataGridView2.Rows.Add(objBitmap2.Height);

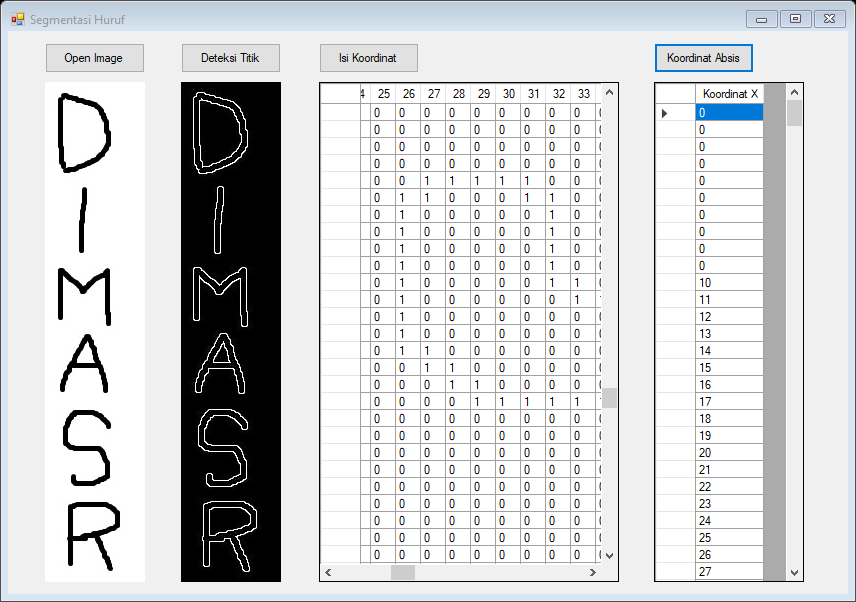
for (int i = 0; i < objBitmap2.Height; i++)

dataGridView2.Rows[i].Cells[0].Value = koor\_absis[i].ToString();

}

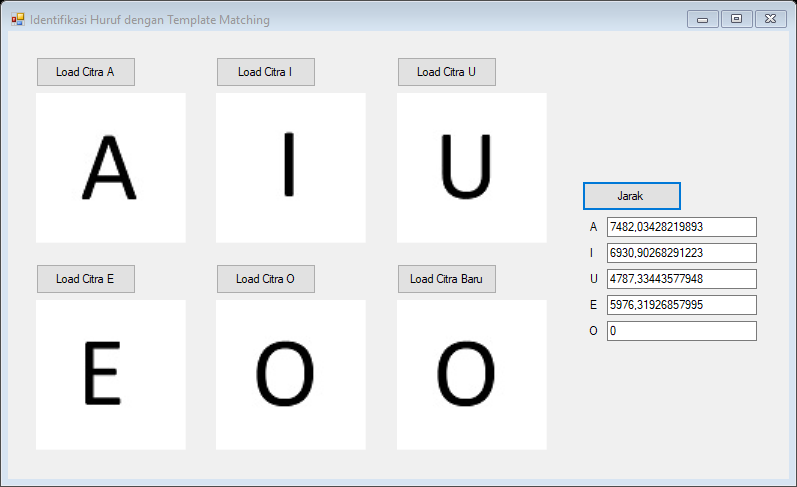
1. Output Program

* Segmentasi Huruf

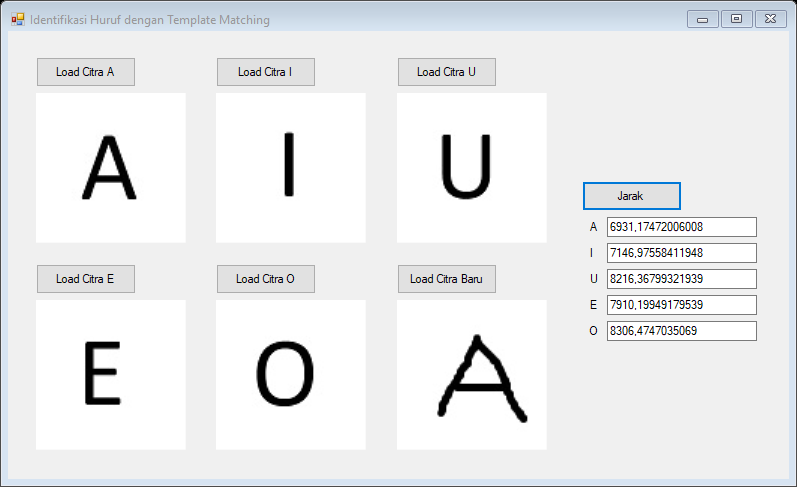


*Gambar 2.0, Segmentasi Huruf*

* Template Matching

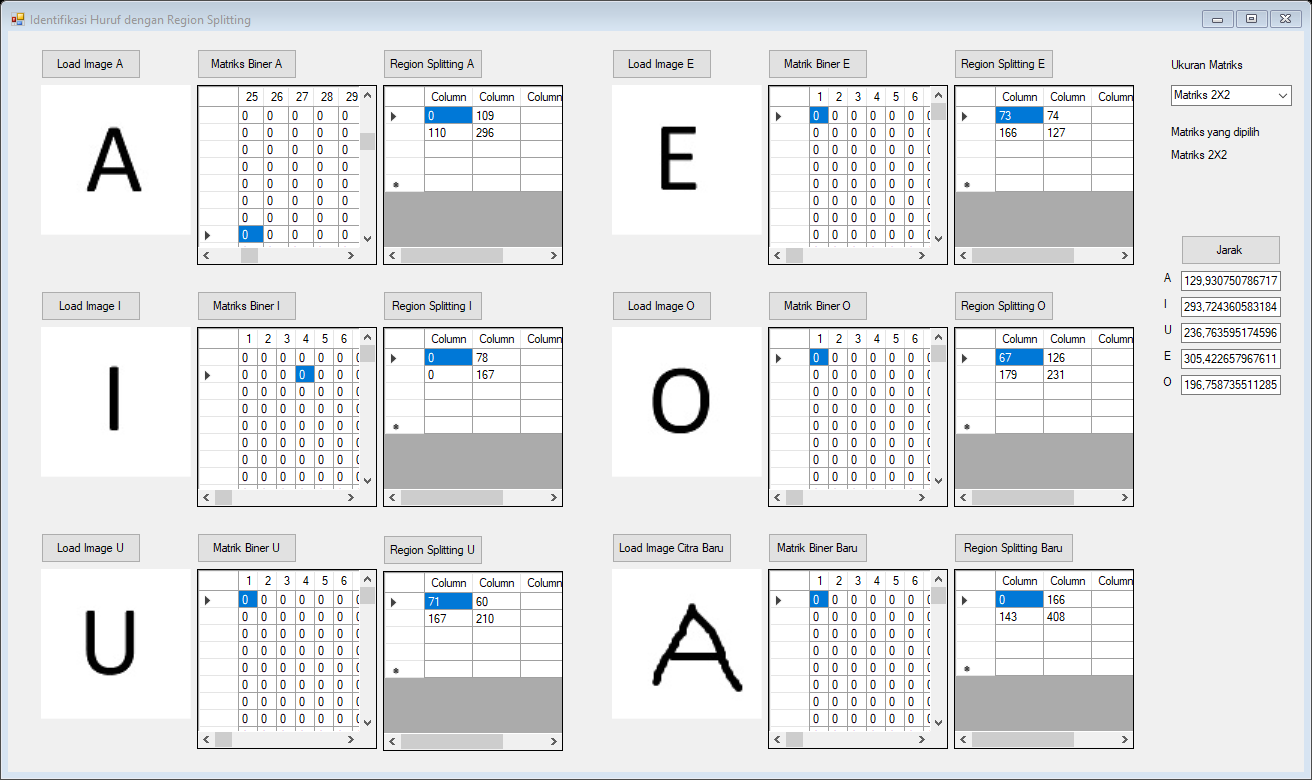


*Gambar 2.1, Template matching terlihat o memiliki jarak 0 (sama)*

**

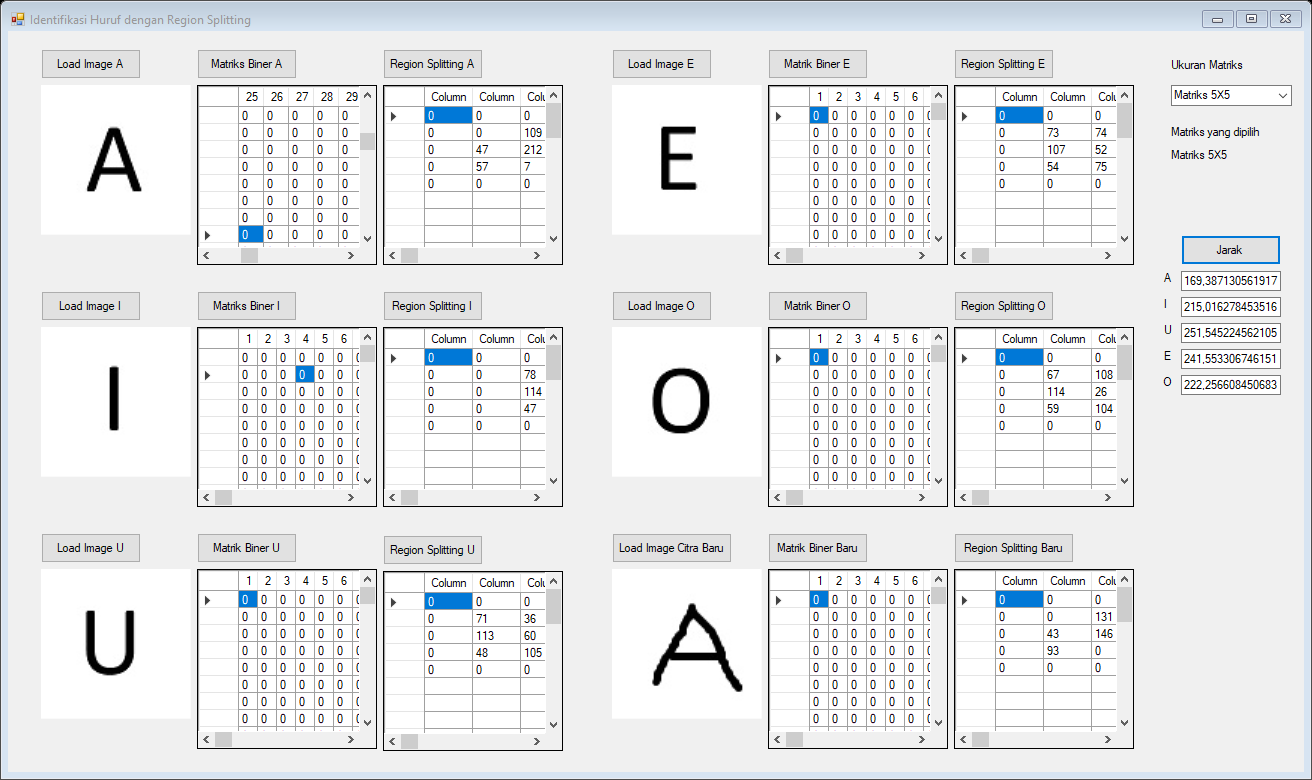
*Gambar 2.2, terlihat jarak yang paling dekat adalah dengan huruf A*

* Region Splitting 2x2



*Gambar 2.3, terlihat jarak yang terdekat adalah dengan huruf A*

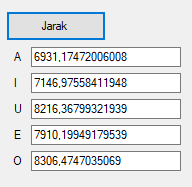
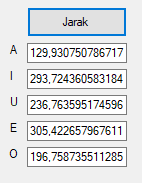
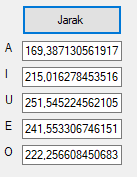
* Region Splitting 5x5



*Gambar 2.4, Region splitting dengan matriks 5x5 juga menunjukan jarak terdekat adalah dengan citra A*

1. Analisa

* Bandingkan hasil identifikasi jarak antara template matching, region splitting 2x2 dan region splitting 5x5!

*Gambar 2.5, mulai dari kanan ke kiri, hasil dengan template matching, region splitting 2x2, region splitting 5x5*

Pada identifikasi menggunakan template matching, jarak yang dihasilkan antara citra database dan citra yang akan dianalisa sangat besar mencapai angka 6900, sedangkan untuk identifikasi menggunakan region splitting 2x2 dan 5x5 hanya berkisar angka ratusan.

Pada region splitting 2x2, dan 5x5 ditunjukan hasil yang lebih mendekati adalah dengan menggunakan matriks 2x2 dengan jarak yang didapat adalah 130 pada 2x2 dan 169 pada 5x5

Namun, pada dasarnya, ketiganya berhasil mengidentifikasi bahwa citra input termasuk ke golongan huruf A.