

## APLIKASI PENGENALAN GAMBAR MENGGUNAKAN MICROSOFT VISUAL BASIC 6.0

Prihastuti Harsani<sup>1)</sup>, Kiki Noviandi<sup>2)</sup>

1) Staf Pengajar Jurusan Ilmu Komputer-FMIPA Universitas Pakuan

2) Mahasiswa Jurusan Ilmu Komputer - FMIPA Universitas Pakuan

### I. PENDAHULUAN

Suatu citra terkadang mengandung beberapa pesan yang sangat sulit sekali untuk diterjemahkan. Seperti halnya dalam bidang forensik dan bidang-bidang lainnya. Salah satu kesulitan yang dihadapi dalam pengolahan citra ini adalah banyaknya kombinasi warna yang mencapai  $\pm 16$  juta kombinasi. Sehingga salah satu cara yang mungkin ditempuh adalah dengan membawa kebentuk dua warna yaitu hitam dan putih dimana warna ini akan lebih mudah untuk ditransformasikan ke bentuk lain.

Untuk menyelesaikan hal tersebut kepandaian yang diberikan pada komputer seperti kecerdasan buatan atau "*Artificial Intelligent*" dapat dijadikan sebagai salah satu alternatif. Kepandaian yang diberikan pada komputer merupakan hasil representasi kecerdasan manusia ke dalam komputer yang ditranslasikan dalam bentuk instruksi dan langkah-langkah yang diperlukan untuk menyelesaikan permasalahan itu sendiri. Metode yang digunakan untuk menyelesaikan masalah dalam aplikasi pengenalan gambar adalah dengan metode pencarian *Heuristic*. Dalam prosesnya metode *Heuristic* ini melakukan pelacakan gambar, representasi ulang gambar dan pembelajaran terhadap proses yang telah dilakukan serta menarik sebuah kesimpulan dengan berbekal pengetahuan yang didapatkannya.

### II. IDENTIFIKASI DAN ANALISA MASALAH

Gambar adalah sebuah objek yang sangat kompleks untuk dikenali, hal ini dikarenakan kompleksnya kombinasi warna yang mencapai  $\pm 16$  juta warna. Hal ini sangat tidak memungkinkan untuk mengenali setiap bagian warna tersebut. Oleh karena itu diperlukan penyandian gambar kedalam kode-kode yang mudah diingat dan diimplementasikan seperti halnya kode *binary*.

Basis pengetahuan bagi penyelesaian masalah ini yaitu dengan kode *binary* 0 dan 1. Kode *binary* 0 menggambarkan bahwa elemen matriks tersebut berwarna putih dan 1 berwarna hitam. *Inference Engine* akan dirancang dengan menggunakan metode *Heuristic* dan *Learning Vector Quantization* (LVQ).

Proses pengenalan gambar akan merepresentasikan ulang suatu gambar ke dalam bentuk *binary* dan mencocokkannya dengan data *binary* suatu objek yang telah disimpan sebelumnya. Proses ini akan diproses menggunakan satu metode yang dikenal dengan metode *Learning Vector Quantization* (LVQ). Hasil pencocokan akan disimpan dalam satu *variable* bobot. Dimana bobot yang terkecil adalah kelas atau objek yang mempunyai banyak kemiripan dengan data atau gambar yang dilatih.

### III. PERANCANGAN DAN IMPLEMENTASI.

Dalam perancangan aplikasi pengenalan gambar ini akan ditambahkan fungsi deteksi tepi, representasi ulang data dan fungsi pengenalan. Dalam perancangan ini dipergunakan alat bantu berupa *pseudocode* sebagai implementasinya.

#### 3.1. Pseudocode Deteksi Tepi.

##### DESKRIPSI

x1 : Long Integer  
 x2 : Long Integer  
 y1 : Long Integer  
 y2 : Long Integer  
 isfirst : boolean  
 Width : Long Integer  
 Height : Long Integer  
 N\_step : Integer  
 x : Integer  
 y : Integer

##### DEKLARASI

*isfirst* : true  
*find* : xColor  
 x1,x2,y1,y2 : 0  
*width* : Width [Surface]  
*Height* : Height [Surface]

##### ALGORITMA

```

for x = 1 to Width
  for y = 1 to height
    if (x,y) = find then
      if not isfirst then
        if x < min(x1) then
          set x1 = min(x)
        if x > max(x2) then
          set x2 = max(x)
        if y < min(y1) then
          set y1 = min(y)
        if y > max(y2) then
          set y2 = max(y)
        else isfirst ? false
          set x1,x2 ? (min,max,x)
          set y1,y2 ? (min,max,y)
      Endif
    Next x,y
  
```

#### 3.2. Pseudocode Representasi gambar.

##### DESKRIPSI

Strdata : String  
 a : Long Integer  
 b : Long Integer  
 x : Long Integer  
 y : Long Integer

##### DEKLARASI

Strdata = ""  
 X,Y = 0  
 y = 0  
 a = width / 256  
 b = height / 256

##### ALGORITMA

```

For x ? x1 to x2 step (x2-x1)/ 256
  For y ? y1 to y2 step (y2-y1)/256
    If ( x , y ) = x then
      Set ( a , b ) ? x
      Strdata ( x , y ) ? 1
    Else
      Set ( a , b ) ? ?x
      Strdata ( x , y ) ? 0
    End if
    b ? b + ( Height / 256 )
    Doevents
  Next y
  b ? b + ( height / 256 )
  a ? a + ( width / 256 )
Next x
  
```

#### 3.3. Pseudocode Fungsi Pengenalan Gambar.

##### DESKRIPSI

Strbuffer : String  
 bobot : Long Integer  
 polabobot : Long Integer  
 polatempbobot1 : Long Integer  
 polatempbobot : Long Integer  
 polabin,i,j,h : integer  
 polabinl : integer  
 polanama : string

#### DEKLARASI

```

Strbuffer      = ""
isfirst       = true
bobot         = 0
polabobot     = 0
Polatempbobot1 = 0
polatempbobot = 0
polabin       = 0
polabin1      = 0
polanama      = ""
h             = Len(strdata)
    
```

#### ALGORITMA

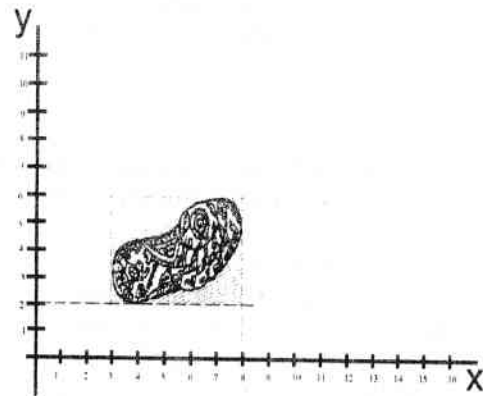
```

If rsdua.RecordCount < 1 then
    End
Else
    Rsdua(1)
    Strbuffer =? rsdua!Binary
    Do while strbuffer ? EOF
For i ? 1 to h
    Polabin = Mid(strbuffer,i,1)
    Polabin1=?
    id(strdata,i,1)Polatempbobot1
    Polatempbobot + ((Polabin
    polabin1)^2)
    If polabin polabin1 = 0 then
        Polabobot = polabobot + 1
    Else
        Polabobot = polabobot * 1
    End if
End if
Next i
Polatempbobot= sqrt(polatempbobot1)
If isfirst ? true then
    Polatempbobot = polatempbobot1
    Polanama = rsdua (Nama)
    Isfirst = false
Else
    If polatempbobot1 <= polatempbobot then
        Polatempbobot= polatempbobot1
        Polanama = rsdua (Nama)
    End if
End if
If polabobot > Bobot then
    Bobot ? polabobot
    Polanama ? rsdua (nama)
End if
Polatempbobot1 ? 0
Polabobot ? 0
Strbuffer ? 0
Loop
End if
    
```

## 4. HASIL DAN PEMBAHASAN

### 4.1. Proses Deteksi Tepi

Proses ini dimaksudkan untuk mempersempit ruang lingkup gambar dan membuang daerah yang bukan merupakan penyusun gambar. Ilustrasi dari hal ini akan ditunjukkan dalam gambar berikut.



Gambar 21. Ilustrasi Deteksi Tepi

Pada gambar diatas terlihat sebuah gambar dalam wilayah (x,y). tetapi gambar ini tidak semuanya menempati wilayah (x,y), ada wilayah yang kosong yang tidak digunakan oleh gambar itu sendiri, deteksi tepi digunakan untuk mendeteksi wilayah yang merupakan wilayah penyusun gambar yang dalam gambar diatas di tunjukan oleh wilayah yang diarsir dan membuang wilayah kosong atau wilayah yang tidak diarsir.

Proses deteksi tepi ini dilakukan sepanjang jalur atau lintasan dengan membandingkan nilai jalur atau lintasan dengan tujuan yang diharapkan. Dalam lintasan x, maka  $x_1$  adalah nilai minimum dan  $x_2$  adalah nilai maximum dari jalur x.

Begitupula dengan lintasan  $y$  dimana  $y_2$  adalah nilai maximum sedangkan  $y_1$  adalah nilai minimum dari jalur  $y$ .

Hasil dari proses deteksi tepi ini adalah koordinat-koordinat penyusun gambar yaitu titik titik terluar dari wilayah yang diarsir. Dari gambar diatas koordinat yang didapatkan adalah :

- $x_1 = 3$  titik awal pada sumbu  $x$
- $x_2 = 8$  titik akhir pada sumbu  $x$
- $y_1 = 2$  titik teratas pada sumbu  $y$
- $y_2 = 6$  titik terendah pada sumbu  $y$

#### 4.2. Proses Representasi Ulang Gambar Dalam Bentuk Matriks (256x256)

Proses representasi ulang gambar adalah proses menyusun ulang gambar ke bentuk yang dikehendaki yang dalam ruang lingkupnya adalah berukuran 256x256.

Untuk mendapatkan gambar berukuran 256x256 dari sebuah gambar dapat dilakukan dengan membuat titik atau koordinat dimana  $x_1$  adalah titik awal dan membuat titik selanjutnya dengan jarak dari hasil bagi titik  $x_2$  dgn  $x_1$ . pembuatan titik ini terus dilakukan hingga mencapai titik yang ke 256. Demikian pula untuk membuat titik  $y$ , dimana  $y_1$  adalah sebagai titik awal dan membuat titik selanjutnya dengan jarak hasil pembagian titik  $y_2$  dengan  $y_1$ . Pembuatan titik ini dilakukan pula hingga mencapai akhir yaitu 256, hasil dari proses ini akan didapatkan matriks 256x256 dalam wilayah baru  $x(n)$ ,  $y(m)$  yang bersesuaian dengan matriks awal. Adapun proses perhitungan jarak untuk proses representasi ini adalah :

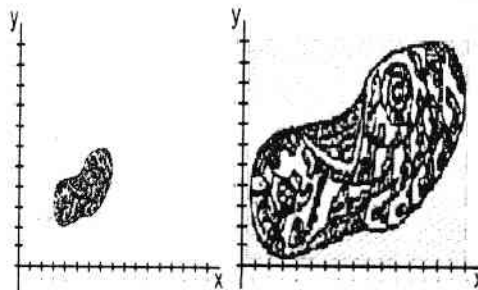
$$x_1 = 3 \quad ; \quad y_1 = 2$$

$$x_2 = 8 \quad ; \quad y_2 = 6$$

– jarak antar elemen  $x(n) = (x_2 - x_1) / 256$

– jarak antar elemen  $y(m) = (y_2 - y_1) / 256$

adapun ilustrasi dari perhitungan ini adalah :



Gambar 22. Ilustrasi representasi ulang gambar

Dari gambar diatas dapat dilihat bahwa wilayah hasil deteksi tepi ini akan dibagi kedalam kolom-kolom. Jarak antar tiap kolom adalah sama atau sebanding dengan hasil bagi titik akhir dengan titik awal suatu lintasan.

Setiap kolom yang dihasilkan dari proses representasi ini akan diberikan sebuah nilai atau kode *binary* 0 dan 1. kumpulan kode-kode yang merupakan data tentang suatu data gambar ini dapat disimpan untuk dipergunakan kembali suatu saat. Data akan disimpan dalam sebuah *DataBase* berikut keterangan yang diperlukan.

#### 4.3. Proses Pengenalan Gambar

Proses pengenalan gambar adalah proses membandingkan data yang didapat dari hasil pemrosesan gambar dengan data yang telah disimpan sebelumnya. Hal ini ditujukan untuk menguji sebuah komputer dalam mengenali sebuah pola

Objek berdasarkan karakteristik datanya.

Adapun metode yang digunakan dalam pengenalan pola gambar ini adalah metode *Learning Vector Quantization* (LVQ). Metode ini akan membandingkan setiap kode yang ada dengan mencari jarak atau lintasan terkecil dari setiap data. Adapun pembahasan perhitungan dari metode LVQ ini adalah sebagai berikut :

- Misal jika diketahui matriks yang akan dilatih adalah:  
1010011100
- sedangkan matriks bobot adalah :  
1. 0001110001  
2. 1111000010
- Jarak Pada :

Bobot 1 yaitu

$$\begin{aligned}
 &= (1-0)^2 + (0-0)^2 + (1-0)^2 + (0-1)^2 + \\
 &(0-1)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + \\
 &(0-0)^2 + (0-1)^2 \\
 &= \text{sqr}(7) \\
 &= 2.6457
 \end{aligned}$$

kesamaan pola adalah dari dua data adalah : 3 point

Bobot 2 yaitu :

$$\begin{aligned}
 &= (1-1)^2 + (0-1)^2 + (1-1)^2 + (0-1)^2 + \\
 &(0-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + \\
 &(0-1)^2 + (0-0)^2 \\
 &= \text{sqr}(6) \\
 &= 2.4494
 \end{aligned}$$

kesamaan pola dari dua data adalah : 4 point  
Jarak terkecil adalah terhadap data bobot ke 2 yaitu 2.4494

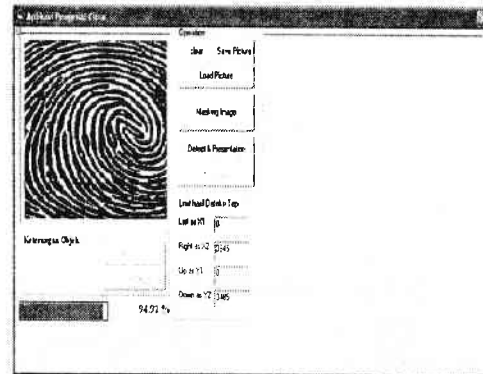
Sehingga dapat diambil kesimpulan bahwa data yang dilatih adalah termasuk bobot atau kelas ke 2, hal ini dikarenakan bahwa data yang dilatih memiliki perbedaan jarak yang paling minimum dan persamaan pola yang lebih banyak dibandingkan terhadap data yang lain.

## 5 Uji Coba Aplikasi.

### 5.1. Uji Coba Proses Deteksi Tepi

Proses ini dimaksudkan untuk mengetahui titik atau koordinat penyusun citra. Jika di-input sebuah gambar hitam putih dan memenuhi seluruh tempat penampung gambar yang berukuran mxn, maka dapat disimpulkan bahwa titik tepi dari gambar merupakan titik tepi dari penampung gambar itu sendiri. Sehingga nilai titik yang didapatkan dari proses ini adalah :

$$\begin{aligned}
 x_1 &= 0 ; & x_2 &= m ; & y_1 &= 0 ; & y_2 &= n \\
 & & & & & & & ;
 \end{aligned}$$

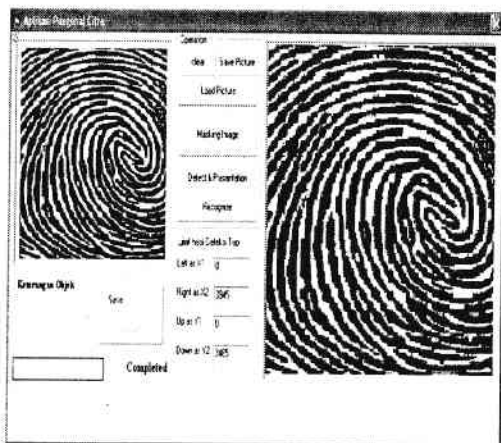


Gambar 23. Hasil Proses Deteksi Tepi

Dari proses diatas, bila nilai ataupun titik koordinat yang ditemukan selama proses deteksi tepi ini sesuai dengan titik koordinat dari penyusun citra, maka proses deteksi tepi ini telah berhasil dan berfungsi dengan baik. Hal ini dapat dilihat pada saat representasi ulang gambar ke dalam matriks 256 x 256. Jika output yang dihasilkan sama seperti gambar awal, maka secara tidak langsung proses deteksi tepi dan representasi ulang ini dapat dikatakan telah berfungsi dengan baik.

### 5.2. Uji Coba Proses Representasi Ulang Gambar ke Matrik 256 x 256.

Proses Representasi Ulang ini adalah proses menyusun ulang gambar ke dalam bentuk yang lebih terdefinisi yaitu matriks 256 x 256, dimana setiap elemen matriks ini akan memiliki nilai yang berupa kode *binary*. Proses penyusunan citra ini dimulai dari titik-titik koordinat yang dihasilkan dari deteksi tepi. Sehingga gambar ataupun citra yang dihasilkan hanya citra ataupun pola yang menyusun gambar awal. Adapun proses pengujian yang akan dilakukan adalah dengan menempatkan sebuah gambar *input* dan melihat *output* gambar yang dihasilkan. Bila kedua gambar antara gambar *input* dan gambar *output* persis atau sama maka proses ini dapat dikatakan telah berfungsi dengan baik. Adapun gambaran dari proses ini dapat dilihat pada gambar sebagai berikut.



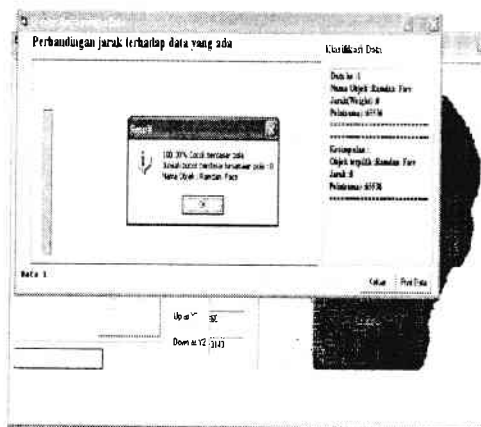
Gambar 24. Hasil Representasi Ulang Gambar Dalam Matriks 256x256

Dari gambar diatas dapat dibandingkan antara gambar *input*

dengan gambar *output*, bila kedua gambar mempunyai kemiripan atau dapat dikatakan sama maka proses ini dikatakan telah berfungsi dengan baik.

### 5.3. Uji Coba Pengenalan gambar.

Proses uji coba pengenalan gambar ini dimaksudkan untuk mengetahui sejauh mana komputer dapat menginterpretasikan gambar kedalam bentuk data yang dapat diukur serta mensimulasikan bagaimana komputer dapat menarik kesimpulan terhadap suatu permasalahan terutama dalam permasalahan pengenalan pola gambar. Proses pengenalan gambar ini dapat diuji dengan cara meng-*input* sebuah gambar yang telah diproses serta telah disimpan sebelumnya. Bila *output* yang dihasilkan oleh proses ini sesuai dengan data yang telah disimpan sebelumnya, maka proses pengenalan ini serta pengambilan keputusan ini telah berfungsi dengan baik. Adapun gambaran proses dari pengenalan gambar ini adalah :



Gambar 25. Proses Pengenalan Gambar

Uji coba ini dilakukan dengan meng-input sebuah gambar yang telah disimpan sebelumnya yaitu gambar "Ramdan Face" dan dicoba untuk mengenalinya kembali. Dari hasil *Output* yang diberikan diketahui bahwa gambar tersebut adalah Gambar dari "Ramdan Face". Hal ini membuktikan bahwa pengenalan gambar ini telah berjalan dengan baik dan sesuai dengan *input* yang diberikan.

## VI. KESIMPULAN

Dari hasil penelitian ini dapat disimpulkan bahwa sebuah gambar dalam sebuah bidang  $x,y$  dapat dikatakan sebagai suatu kesatuan matriks berukuran  $x,y$  dimana nilai untuk setiap  $x,y$  adalah nilai yang sama dalam koordinat atau titik dalam gambar pada titik  $x,y$ .

Atau dapat dikatakan bahwa dalam menguji dan mengenali sebuah sistem fisis maka pengenalan boleh ditandai oleh beberapa perwujudan fisisnya, yang kembali dinyatakan oleh beberapa kumpulan pengukuran yang membentuk "Ruang Pengukuran".

Beberapa proses yang dilakukan dalam aplikasi ini adalah proses deteksi tepi, berfungsi untuk mencari titik atau tepian gambar. Proses Representasi adalah proses membentuk ulang citra atau gambar tanpa menghilangkan identitas asal. Proses Pengenalan adalah proses pencarian data yang memiliki kesamaan paling besar dengan jarak yang seminimum mungkin.

Proses-proses ini semua saling terkait guna menghasilkan efisiensi dan kecepatan akses bagi aplikasi dalam menentukan sebuah keputusan.

Beberapa metode yang digunakan

dalam proses tersebut antara lain metode *heuristic* dan *learning vector quantization* (LVQ).

## DAFTAR PUSTAKA

- Ario Suryo Kusumo., 2002, *Buku Latihan Pemrograman Database dengan Visual Basic 6.0*, PT. Elex Media Komputindo, Jakarta.
- Djoko Pramono., 2000, *Mudah Menguasai Visual Basic 6.0*, Elex Media Komputindo, Jakarta.
- Nalwan, A., 2000, *Buku Latihan Microsoft Visual Basic 6.0*, PT Elex Media Komputindo, Jakarta.
- Sandi Setiawan., 1993, *Artificial Intelligence*, ANDI OFFSET, Yogyakarta.
- Sri Kusumadewi., 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Penerbit Graha Ilmu, Yogyakarta
- Sri Kusumadewi., 2004, *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*, Penerbit Graha Ilmu, Yogyakarta.