IMPLEMENTASI SEGMENTASI CITRA DAN ALGORITMA *LEARNING VECTOR QUANTIZATION* (LVQ) DALAM PENGENALAN BENTUK BOTOL

Andri

STMIK Mikroskil
Jl. Thamrin No. 122, 124, 140 Medan 20212

andri@mikroskil.ac.id

Abstrak

Dalam kehidupan sehari-hari, pengolahan citra digital memegang peranan yang cukup penting. Salah satu kegunaannya adalah melakukan pengenalan pola. Pengenalan pola bisa dikembangkan dalam kegiatan industri minuman untuk pengenalan bentuk botol sehingga dapat mempercepat waktu pemisahan jenis-jenis botol. Tujuan dari penelitian ini adalah merancang suatu aplikasi untuk melakukan pengenalan terhadap bentuk botol dengan segmentasi citra dan algoritma *Learning Vector Quantization* (LVQ). Aplikasi dimulai dengan input gambar botol yang dideteksi tepi dengan operator *sobel*. Deteksi tepi ini menghasilkan citra hitam putih. Selanjutnya, hasil deteksi tepi sobel diekstraksi ciri ke pola dengan ukuran 20x20 kotak, dengan masing-masing kotak terdiri dari warna hitam atau putih. Warna hitam mewakili bit 1 dan warna putih akan mewakili bit 0. Dengan demikian, terdapat 400 bit hasil ekstraksi ciri sebagai identitas dari setiap gambar tepi botol. Proses akhir adalah melatih bit-bit ini dan nilai bobot hasil pelatihan disimpan sebagai basis pengetahuan untuk proses pengenalan. Aplikasi hasil rancangan dapat digunakan untuk mengenali bentuk botol dan juga menampilkan langkah-langkah perhitungan proses pelatihan dan pengenalan. Hasil pengujian menunjukkan aplikasi ini dapat mengenal bentuk botol dengan tingkat keberhasilan hingga 88,88%. Faktor kegagalan pengenalan terletak pada gambar latar yang berbeda dan posisi botol yang tidak sama dengan posisi botol pada saat proses pelatihan.

***Kata kunci** : deteksi tepi, operator sobel, LVQ, botol*

1. Pendahuluan

1.1. Latar Belakang

Dalam kehidupan sehari-hari, pengolahan citra digital memegang peranan yang cukup penting. Salah satu kegunaannya adalah dapat melakukan pengenalan pola. Pengenalan pola adalah tahapan selanjutnya atau analisis dari pengolahan citra. Pengenalan pola bisa dikembangkan dalam kegiatan industri seperti pengenalan bentuk botol yang akan digunakan pada industri minuman. Hal ini dilakukan untuk mengenali dan memisahkan jenis-jenis bentuk botol.

Pengenalan pola dapat dilakukan dengan menggunakan metode Jaringan Syaraf Tiruan (JST). Terdapat beberapa algoritma JST yaitu *perceptron*, propagasi balik, *kohonen*, *Learning Vector Quantization* (LVQ) dan algoritma JST lainnya. Algoritma yang dipilih dan digunakan dalam penelitian ini adalah algoritma LVQ karena algoritma LVQ dapat belajar secara otomatis untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor input. Jika 2 (dua) vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama [3]. Proses utama meliputi 2 (dua) fase yaitu fase pelatihan (*training*) dan fase pengenalan (*recognition*). Awalnya semua bentuk sampel botol yang akan dikenali oleh aplikasi harus melewati fase pelatihan terlebih dahulu untuk disimpan

sebagai basis pengetahuan. Sebelum memasuki fase pelatihan atau fase pengenalan, input gambar akan melalui *pre-processing*, yaitu segmentasi citra dengan metode deteksi tepi. Tujuan deteksi tepi adalah untuk menyederhanakan citra dengan mendeteksi tepi dari bentuk botol dan mengubah gambar menjadi menjadi gambar hitam putih (dengan piksel hitam merupakan batas tepi bentuk botol, dan piksel putih merupakan warna latar). Setelah itu, gambar biner dipecah menjadi n x n kotak dan dilakukan ekstraksi ciri. Hasil ekstraksi ciri kemudian dilatih dengan menggunakan metode LVQ. Hasil pelatihan akan menghasilkan bobot-bobot pelatihan yang disimpan sebagai basis pengetahuan. Pada fase pengenalan, bobot-bobot pelatihan ini akan digunakan untuk mengenali pola yang sudah pernah dilatih sebelumnya.

1.2. Rumusan Masalah

Pengenalan dan pemisahan jenis-jenis botol dalam kuantitas yang banyak pada industri minuman dengan menggunakan tenaga manusia kurang efisien sehingga diperlukan suatu teknologi komputer untuk mempercepat proses pengenalan dan pemisahan tersebut.

1.3. Tujuan dan Manfaat

Adapun tujuan penelitian ini adalah merancang suatu aplikasi yang dapat melakukan pengenalan terhadap bentuk botol dengan menggunakan segmentasi citra dan algoritma LVQ. Manfaatnya adalah:

a. Membantu industri minuman dalam melakukan pengenalan dan pemisahan jenis-jenis botol.
b. Sebagai referensi untuk pengembangan lebih lanjut seperti sistem pengenalan dan pemisahan jenis botol otomatis.

1.4. Batasan Masalah

Ruang lingkup masalah pada penelitian ini sebagai berikut:

a. Gambar yang dimasukkan adalah gambar satu buah botol kosong, baik botol plastik ataupun botol kaca, dalam bentuk file jpg, bmp atau gif.
b. Proses deteksi tepi, dengan menggunakan operator *sobel*.
c. Proses ekstraksi ciri, untuk menghasilkan pola umum setiap bentuk botol, yang merupakan bit-bit masukan pada proses pelatihan dan proses pengenalan, dibatasi sebesar 20 x 20.
d. Setiap bentuk botol akan disertai dengan nama botol, dengan anggapan bahwa setiap nama botol bersifat unik untuk setiap bentuk botol.
e. Aplikasi akan menghasilkan langkah-langkah perhitungan algoritma LVQ pada proses pelatihan maupun proses pengenalan.
f. Aplikasi akan menampilkan waktu yang dibutuhkan untuk proses deteksi tepi, proses pelatihan dan proses pengenalan.

1.5 Metodologi Penelitian

Langkah-langkah yang dilakukan untuk menyelesaikan penelitian ini antara lain:

a. Analisis, menganalisa cara kerja dari metode yang digunakan. Analisis mencakup proses deteksi tepi *sobel*, proses ekstraksi ciri, proses pelatihan dan proses pengenalan dengan algoritma LVQ.
b. Perancangan Sistem, merancang algoritma sesuai metode yang digunakan dan merancang antar muka perangkat lunak.
c. Konstruksi Sistem, membangun perangkat lunak dengan menggunakan bahasa pemograman *Microsoft Visual Basic* .Net 2008 dan menggunakan basis data *Microsoft Access* 2007.

d. Pengujian, mengeksekusi program dengan tujuan menemukan kesalahan serta menguji hasil akurasi proses pengenalan.

2. Kajian Pustaka

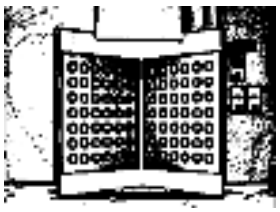
2.1 Deteksi Tepi (*Edge Detection*)

Dasar dari teknik deteksi tepi ialah dengan melakukan penelusuran citra secara vertikal dan horizontal sambil melihat apakah terjadi perubahan warna yang mendadak yang melebihi nilai sensitivitas antara dua titik yang berdempetan. Jika terjadi perubahan warna yang mendadak, maka di tempat antara kedua titik tersebut dianggap tepi sebuah objek [4].

Proses segmentasi umumnya dilakukan pada citra *graylevel* (selanjutnya disebut hanya citra). Dengan melakukan segmentasi citra menggunakan metode deteksi tepi, citra akan disederhanakan menjadi dua warna, hitam dan putih. Warna putih sebagai pinggiran kedua objek, sedangkan warna latar (*background*) dan tengah objek berwarna hitam ataupun sebaliknya warna hitam sebagai batas/pinggiran kedua objek, sedangkan warna latar dan tengah objek berwarna putih.

Hasil dari deteksi tepi tidak selalu sempurna karena proses deteksi tepi dilakukan dengan cara mendeteksi perubahan warna yang melebihi suatu nilai sensitivitas. Contohnya adalah citra asli yang diperlihatkan pada Gambar 1. Apabila dilakukan deteksi tepi terhadap citra tersebut dengan sensitivitas rendah, maka akan diperoleh citra seperti pada Gambar 2 di mana muncul titik-titik hitam yang bertebaran di dalam dan di luar objek. Titik-titik tersebut tidak seharusnya muncul sebagai tepi, tetapi deteksi tepi menganggapnya tepi dari suatu objek karena proses ini dilakukan dengan cara mendeteksi perubahan warna yang melebihi suatu nilai sensitivitas.

Gambar 1. Citra Asli

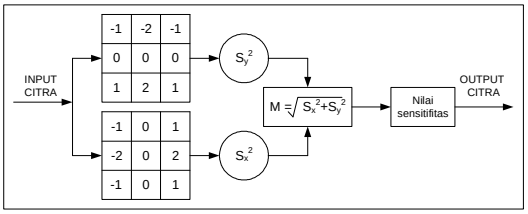


Gambar 2. Citra dengan *Threshold* Rendah

2.2 Operasi *Sobel*

Operator *sobel* diterapkan dalam dua buah *mask*. *Mask* yang pertama (G_x) digunakan untuk melakukan penelusuran terhadap suatu citra secara horizontal dan *mask* yang kedua (G_y) digunakan untuk melakukan penelusuran terhadap suatu citra secara vertikal.

Langkah-langkah dalam melakukan *edge detection* dengan menggunakan Operator *sobel* diperlihatkan pada Gambar 3.



Gambar 3. Langkah-langkah Deteksi Tepi dengan Operator *Sobel* [1]

Keterangan Gambar 3:

1. Lakukan penelusuran secara vertikal

   Dengan mengalikan piksel citra terhadap mask G_y, sehingga diperoleh piksel hasil penelusuran vertikal (S_y), di mana N_y diperoleh dengan cara :

$$S_y = (X_6 + 2.X_7 + X_8) - (X_1 + 2.X_2 + X_3)$$

2. Lakukan penelusuran secara horizontal
   Dengan mengalikan piksel citra terhadap mask G_x, sehingga diperoleh piksel hasil penelusuran horizontal (S_x), di mana N_x diperoleh dengan cara :

$$S_x = (X_3 + 2.X_5 + X_8) - (X_1 + 2.X_4 + X_6)$$

3. Lakukan penggabungan terhadap hasil penelusuran secara vertikal dan horizontal dengan cara :

$$M = \%S_x^2 + S_y^2$$

   Di mana M = Piksel N setelah dilakukan penggabungan hasil penelusuran secara horizontal dan vertikal

4. Berikan suatu nilai *threshold* (sensitivitas), dan kemudian bandingkan nilai *threshold* tersebut dengan nilai M. Jika :
   - M > nilai *threshold*, maka piksel tersebut menjadi pinggiran sebuah objek
   - M < nilai *threshold*, maka piksel tersebut sebagai warna latar atau tengah dari objek.

2.3 Metode *Learning Vector Quantization*

Metode *Learning Vector Quantization* (LVQ) adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor *input*. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor *input*. Jika 2 (dua) vektor *input* mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor *input* tersebut ke dalam kelas yang sama. [3]

Algoritmanya adalah sebagai berikut:

1. Tentukan maksimum epoh (banyaknya proses pelatihan yang akan diulangi), eps (*error* minimum yang diharapkan) dan nilai *alpha*.
2. Hasil ekstraksi ciri pertama dari masing-masing pola digunakan sebagai data awal (inisialisasi). Data inisialisasi ini akan diisi sebagai nilai bobot awal (w).
3. Epoh = 0
4. Selama (Epoh < MaxEpoh) atau (alpha > eps), maka lakukan hal berikut:
   a. Epoh = Epoh + 1
   b. Untuk setiap data hasil ekstraksi ciri, lakukan hal berikut:
      1) Set x = hasil ekstraksi ciri dari pola.
      2) Set T = nomor urut dari setiap kelas.
      3) Hitung jarak hasil ekstraksi ciri pola saat ini dengan masing-masing bobot. Misalkan dihitung jarak hasil ekstraksi ciri pola pertama dengan setiap bobot, maka rumus yang digunakan adalah sebagai berikut:

$$\text{Jarak} = \sqrt{(x_{11} - w_{11})^2 + (x_{12} - w_{12})^2 + \ldots + (x_{1m} - w_{1m})^2}$$

         dengan:
         x_{1m} = bit ekstraksi ciri dari pola-1 yang ke-m.
         w_{1m} = bobot W(1,m)
         m = banyak bit ekstraksi ciri
      4) Bila nomor kelas pada bobot yang memiliki jarak terkecil sama dengan nilai nomor urut (T) pola, maka hitung:

$$w_j \text{ (baru)} = w_j \text{ (lama)} + \alpha \text{ (x} - w_j \text{ (lama))}$$

      5) Bila tidak, maka hitung:

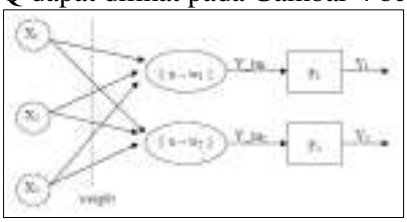
$$w_j \text{ (baru)} = w_j \text{ (lama)} - \alpha \text{ (x} - w_j \text{ (lama))}$$

   c. Kurangi nilai *Alpha*:

$$\alpha = \alpha - (0.1 * \alpha)$$

5. Simpan bobot hasil pelatihan (w).

Arsitektur jaringan LVQ dapat dilihat pada Gambar 4 berikut.



Gambar 4. Arsitektur Jaringan LVQ [3]

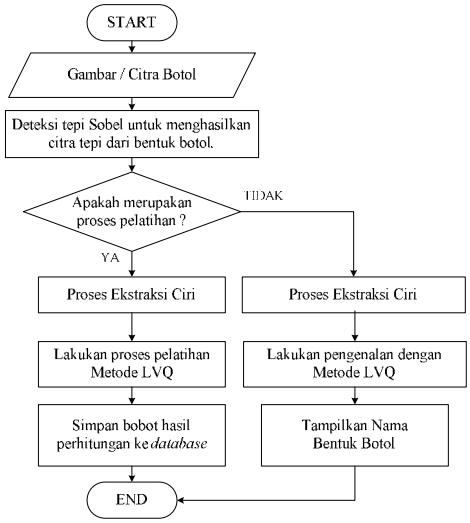
Pada Gambar 4, terlihat bahwa LVQ mempunyai beberapa vektor bobot yang menghubungkan setiap neuron masukan dengan neuron keluaran. Sehingga dapat dikatakan bahwa setiap neuron keluaran pada LVQ berhubungan dengan sebuah vektor bobot. Untuk melakukan proses pengenalan dan pembelajaran, LVQ menggunakan operasi-operasi vektor. Pola yang akan dilatih dan dikenali disajikan dalam bentuk vektor. LVQ mengenali pola berdasarkan pada kedekatan jarak antara dua vektor. Jika |v| mendekati sama, maka v tersebut dikelompokkan ke dalam kelas yang sama [3].

3. Metode Penelitian

Secara umum, cara kerja aplikasi ini dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada Gambar 5 berikut. Dari Gambar 5, dapat terlihat bahwa proses utama yang terdapat di dalam aplikasi, adalah sebagai berikut:

A. Deteksi tepi *sobel*

Deteksi tepi *sobel* merupakan teknik *pre-processing* yang digunakan untuk menyederhanakan citra berwarna menjadi citra hitam putih. Tujuannya adalah agar dapat diperoleh gambar tepi botol (piksel berwarna hitam), sehingga dapat dilakukan proses ekstraksi ciri untuk mendapatkan pola bentuk botol.



Gambar 5. Flowchart Proses Kerja Aplikasi

B. Proses Ekstraksi Ciri.

Proses ekstraksi ciri akan mendigitalisasi gambar tepi botol. Caranya gambar hasil deteksi tepi *sobel* diekstraksi ciri ke pola dengan ukuran 20 x 20 kotak, dengan masing-masing kotak terdiri dari warna hitam atau putih. Selanjutnya, warna hitam mewakili bit 1 dan warna putih akan mewakili bit 0. Dengan demikian, akan terdapat 400 bit hasil ekstraksi ciri dari masing-masing gambar tepi botol.

C. Proses Pelatihan.
Proses pelatihan dengan menggunakan metode LVQ akan memroses semua bit hasil ekstraksi ciri tepi botol yang terdapat di dalam basis data, dan menghasilkan nilai bobot. Bobot ini akan disimpan ke basis data untuk dipakai pada proses pengenalan.

D. Proses Pengenalan.
Proses pengenalan memroses bit ekstraksi ciri dari tepi botol yang akan dikenali dan bobot hasil pelatihan, untuk mendapatkan nama bentuk botol yang paling mirip dengan tepi botol yang dikenali.

Basis data yang digunakan untuk menyimpan bobot hasil pelatihan metode LVQ adalah *Microsoft Access* 2007 dengan 2 (dua) buah tabel seperti yang terlihat pada Tabel 1 dan 2, yaitu:

1. Tabel PolaBotol.
Tabel ini berfungsi untuk menyimpan nama botol dan hasil ekstraksi ciri dari setiap tepi botol. Struktur tabelnya adalah sebagai berikut.

Tabel 1. Struktur Tabel PolaBotol

No.	*Field Name*	*Data Type*	Keterangan
1.	NamaBotol	Text (50)	Berisi nama botol.
2.	EkstraksiCiri	Text (400)	Berisi 400 bit biner hasil ekstraksi ciri dari pola 20 x 20.(berfungsi sebagai *primary key* dan bersifat *unique*)
3.	NomorUrut	Integer	Berisi nomor urut pola, diurutkan berdasarkan nama botol secara *ascending*.
4.	Inisialisasi	Yes / No (1 / 0)	Bila hasil ekstraksi ciri merupakan pola pertama dari nama botol, maka Inisialisasi = 1. Sebaliknya, Inisialisasi = 0.

2. Tabel BobotLVQ
Fungsi dari tabel ini adalah untuk menyimpan bobot hasil pelatihan dari metode LVQ. Struktur tabelnya adalah sebagai berikut.

Tabel 2. Struktur Tabel BobotLVQ

No.	*Field Name*	*Data Type*	Keterangan
1.	Indeks1	Integer	Indeks *array*-1.
2.	Indeks2	Integer	Indeks *array*-2.
3.	Nilai	Double	Nilai bobot.

Sebagai contoh, bila yang ingin disimpan adalah bobot pelatihan w(1,2) = 0.9575, maka *field* Indeks1 berisi nilai 1, *field* Indeks2 berisi nilai 2 dan *field* Nilai berisi 0.9575.

4. Hasil dan Pembahasan

4.1. Hasil

a. Form Utama
Saat aplikasi dijalankan, *form* utama akan muncul seperti Gambar 6 berikut.

Gambar 6. Form Utama

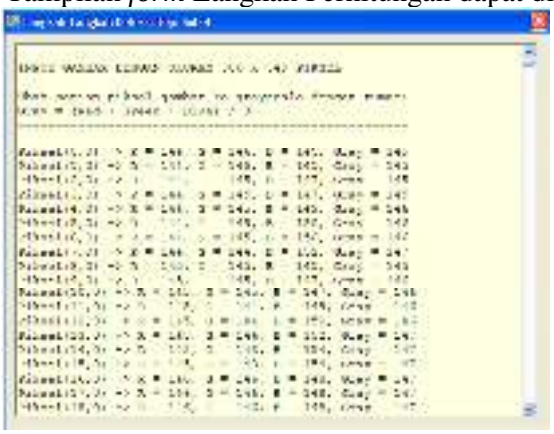
b. Form Deteksi Tepi

Setelah gambar ditampilkan, klik tombol Deteksi Tepi pada *form* Utama untuk memulai proses deteksi tepi terhadap gambar botol. Namun sebelumnya pengguna dapat mengatur nilai *threshold* yang akan digunakan untuk proses deteksi tepi. Tampilan *form* Deteksi Tepi dapat dilihat pada Gambar 7 berikut.

Gambar 7. Tampilan Form Deteksi Tepi

c. Form Langkah Perhitungan

Untuk melihat proses perhitungan dalam proses deteksi tepi *sobel*, klik tombol Langkah^2 pada *toolbar form* Utama. Tampilan *form* Langkah Perhitungan dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Form Langkah Perhitungan

d. Form Pelatihan

Untuk melatih bentuk botol yang belum pernah dilatih sebelumnya, klik tombol Pelatihan pada *toolbar form* Utama, dan *form* Pelatihan akan muncul seperti Gambar 9.

e. Form Pengenalan

Untuk mengenali bentuk botol yang sudah pernah dilatih sebelumnya, klik tombol Pengenalan pada *toolbar form* Utama, dan *form* Pengenalan akan muncul seperti Gambar 10.

Gambar 9. Tampilan Form Pelatihan

Gambar 10. Tampilan Hasil Proses Pengenalan

4.2 Pembahasan

a. Hasil Pengujian Proses Pengenalan

Pengujian hasil pengenalan dilakukan untuk 3 (tiga) jenis botol dengan masing-masing terdiri atas 6 (enam) gambar, dengan hanya setiap gambar pertama yang dilatih dari masing-masing jenis botol. Sisa lima gambar lainnya tidak dilatih dan diuji untuk proses pengenalan. Hasil pengujian disertakan pada Tabel 3.

Tabel 3. Hasil Pengujian Akurasi Pengenalan

No.	Jenis Botol	Hasil Pengenalan	Gambar
1.	Aqua-1	Benar	 Aqua-1
2.	Aqua-2	Benar	 Aqua-2
3.	Aqua-3	Benar	 Aqua-3
4.	Aqua-4	Benar	 Aqua-4
5.	Aqua-5	Salah	 Aqua-5
6.	Aqua-6	Salah	 Aqua-6

No.	Jenis Botol	Hasil Pengenalan	Gambar
7.	Dove-1	Benar	 Dove-1
8.	Dove-2	Benar	 Dove-2
9.	Dove-3	Benar	 Dove-3
10.	Dove-4	Benar	 Dove-4
11.	Dove-5	Benar	 Dove-5
12.	Dove-6	Benar	 Dove-6
13.	Sosro-1	Benar	 Sosro-1
14.	Sosro-2	Benar	 Sosro-2
15.	Sosro-3	Benar	 Sosro-3
16.	Sosro-4	Benar	 Sosro-4
17.	Sosro-5	Benar	 Sosro-5
18.	Sosro-6	Benar	 Sosro-6

Pada Tabel 3, pengujian terhadap proses pengenalan mendapatkan hanya 2 buah gambar yang salah dikenali. Penyebab kesalahan pengenalan adalah adanya latar yang tidak polos (sedangkan pelatihan menggunakan latar yang polos) dan posisi botol yang tidak sesuai dengan posisi botol pada saat proses pelatihan. Dengan demikian, akurasi persentase proses pengenalan adalah:

% Akurasi = (banyak pengenalan benar / banyaknya gambar) x 100 %

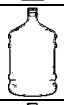
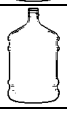
% Akurasi = (16 / 18) x 100 % = 88.88%

b. Pengaruh Nilai *Threshold* Deteksi Tepi Terhadap Hasil Pengenalan

Nilai *threshold* berpengaruh terhadap hasil deteksi tepi botol. Semakin rendah *threshold*, maka semakin banyak piksel yang akan dianggap sebagai pinggiran / tepi, karena perubahan warna yang sedikit saja sudah cukup untuk membuat suatu piksel terdeteksi menjadi pinggiran / tepi. Hasilnya akan terlihat banyak piksel hitam atau *noise* pada gambar. Sebaliknya, semakin tinggi *threshold*, maka akan semakin sedikit piksel yang dianggap sebagai pinggiran / tepi. Hasilnya akan terlihat sedikit piksel hitam pada gambar. Pada pengujian ini, gambar Aqua-1 dilatih dengan nilai *threshold* sebesar 125. Sedangkan pada proses pengenalan, digunakan beberapa nilai *threshold* deteksi tepi. Hasil pengujian dapat dilihat pada Tabel 4 berikut.

Tabel 4. Hasil Pengujian Terhadap Nilai *Threshold* Deteksi Tepi

No.	Nilai Threshold	Gambar Hasil Deteksi Tepi	Hasil Pengenalan
1.	10		Salah
2.	50		Benar

No.	Nilai Threshold	Gambar Hasil Deteksi Tepi	Hasil Pengenalan
3.	100		Benar
4.	200		Benar
5.	250		Benar

Dari Tabel 4, dapat terlihat bahwa semakin tinggi *threshold*, maka semakin sedikit piksel yang dianggap sebagai piksel tepi. Pengujian proses pengenalan pada nilai *threshold* rendah sebesar 10 menyisakan banyak piksel hitam yang dianggap sebagai tepi / pinggiran botol, sehingga terlihat banyak *noise* pada gambar dan mengakibatkan hasil pengenalan menjadi salah.

5 Kesimpulan

Kesimpulan yang bisa diambil adalah:

1. Hasil pengujian proses pengenalan berhasil mendapatkan akurasi proses pengenalan hingga 88.88%. Untuk mendapatkan akurasi pengenalan yang lebih tinggi, maka diusahakan posisi botol dan latar gambar botol, sama dengan posisi botol dan latar pada saat proses pelatihan.
2. Pada proses deteksi tepi, penggunaan nilai *threshold* yang rendah akan mengakibatkan banyak piksel dianggap sebagai tepi botol. Hasilnya adalah gambar deteksi tepi penuh dengan *noise*, dan mengakibatkan kesalahan dalam proses pengenalan.

Referensi

[1] Basuki, A., 2005, *Pengolahan Citra Menggunakan Visual Basic*, Penerbit Graha Ilmu, Yogyakarta.

[2] Desiani, A. dan Arhami, M., 2006, *Konsep Kecerdasan Buatan*, Penerbit Andi, Yogyakarta.

[3] Kusumadewi, S., 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Penerbit Graha Ilmu, Yogyakarta.

[4] Munir, R., 2004, *Pengolahan Citra Digital*, Penerbit Informatika, Bandung.

[5] Puspitaningrum, D., 2006, *Pengantar Jaringan Syaraf Tiruan*, Penerbit Andi, Yogyakarta.

[6] Bullinaria, A.J. 2007, *Learning Vector Quantization (LVQ): Introduction to Neural Computation*, *Available*: http://www.cs.bham.ac.uk/~pxt/NC/lvq_jb.pdf, [18 Februari 2012].

[7] Jaringan Syaraf Tiruan (*Artificial Neural Network*), *Available*: http://ivan.siregar.biz/courseware/CG2_NeuralNetwork_Algorithm.pdf, [18 Februari 2012].

[8] LVQ *Neural Nets*, *Available*: http://www.neural-forecasting.com/lvq_neural_nets.htm, [18 Februari 2012].

[9] Pengenalan Jenis dan Bentuk Botol Berbasis Jaringan Syaraf Tiruan, *Available*: http://lib.atmajaya.ac.id/default.aspx?tabID=61&src=k&id=39564, [18 Februari 2012].

[10] *Simple Competitive Learning*, *Available*: http://www.willamette.edu/~gorr/classes/cs449/Unsupervised/competitive.html, [18 Februari 2012].