PAPER:

SISTEM PENGENALAN OBYEK *REAL-TIME* DENGAN JARINGAN SYARAF TIRUAN *BACKPROPAGATION*

Oleh:
LUKMAN TALIBO[1], YEFFRY HANDOKO PUTRA[2], NANA JUHANA[3]
[1)] JURUSAN TEKNIK INFORMATIKA (10100090)
[2)] DOSEN JURUSAN TEKNIK KOMPUTER
[3)] DOSEN JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
BANDUNG

Abstract

This minithesis (final project) explains about object recognition system that using artificial neural network as classifier of object types. Artificial neural network (ANN) receives four object features, that is, bounding rectangle, bounding circle, first and second momen invarian of Hu. That features extracted using image processing program and good accurate level to representating object characteristics.This system developed in real-time environment. QuickCam camera used as visual sensor to capture light from system environment. System have capability to recognizing objects as continue by good accurate level is 97.14% for normal camera position. That's mean, camera be same position when taking sample data for ANN training. System also have capability to recognizing smaller objects (a half normal size), bigger (two × normal size), and shape-changed objects. For each condition above, success procentage to recognizing objects are 94.28%, 95.71, and 87.14%. The optimum structure for this ANN: four neurons for input layer, 10 neurons for hidden layer, and seven neurons for output layer with learning rate is 0.9.

Key words: *Object recognition, object features, artificial neural network.*

1. PENDAHULUAN

Sisten kendali visual yang sering kita lihat adalah robot, baik itu robot statis (seperti manipulator *pick and place*) maupun robot yang dapat berjalan (seperti mobil robot). Kedua sistem tersebut menggunakan informasi dari lingkungan sekitarnya untuk melakukan aksi atau tindakan. Informasi ini didapat melalui sensor visual. Sensor visual ini dapat berupa sensor konvensional maupun kamera *Charge Couple Devise* (CCD). Pada manipulator, sensor ini menangkap informasi lingkungan dan obyek yang akan dimanipulasi, sedangkan pada mobil robot, sensor ini selain digunakan untuk informasi obyek, juga digunakan untuk menentukan arah geraknya.

Penelitian ini mencoba untuk menerapkan kamera QuickCam, yang relatif lebih murah dibanding kamera CCD yang sering digunakan saat ini. Dengan kamera ini, informasi citra obyek an lingkungannya dapat dicuplik (*capture*).

Pada sistem kendali visual, telah sering digunakan informasi luas (jumlah piksel yang membentuk obyek). Hal ini berakibat ketergantungan pengklasifikasi terhadap ukuran obyek, sehingga apabila ukuran obyek diperbesar atau sebaliknya, maka dapat menyebabkan kegagalan dalam pengambilan keputusan mengenai obyek. Alternatif yang akan dikembangkan dalam penelitian ini adalah menggunakan ciri-ciri geometri dan momen obyek, yaitu *bounding rectangle*, *bounding circle*, invarian momen Hu 1 dan 2. Ciri ini dapat diandalkan dan tetap dapat digunakan untuk obyek yang mengalami transformasi (translasi, rotasi, dan penskalaan).

Jaringan Syaraf Tiruan yang kita kenal sering digunakan untuk pengenalan karakter (huruf) Penelitian ini mencoba untuk menerapkan jaringan syaraf tiruan (JST) sebagai pengklasifikasi obyek berdasarkan ciri geometri dan momen yang disebutkan diatas. Dengan ciri ini sistem kendali visual dapat lebih cerdas dalam mengambil keputusan terhadap obyek yang akan dimanipulasi

2. DESKRIPSI SISTEM

Citra input yang digunakan dalam penelitian ini adalah citra RGB 24 bit dengan ukuran 320x240 piksel (hasil akuisisi). Citra ini dikonversi menjadi citra *gray-scale* 256 level dan disimpan dalam array dua dimensi. Selanjutnya citra *gray-scale* ini dithresholding sehingga menghasilkan citra biner dengan dua derajat keabua (0 dan 255). Selanjutnya pada citra tersebut dideteksi apakan terdapat obyek atau tidak dengan operasi *region growing* (RG).

Setiap obyek yang terdeteksi pada citra hasil RG akan ditentukan luas, pusat massa, dan momen. Selanjutnya ditentukan tepian obyek untuk menentukan keliling obyek. Dari sini dapat diekstrak ciri-ciri obyek, yakni *bounding rectangle*, *bounding circle*, invarian Hu 1 dan 2.

Ciri-ciri ini (sebanyak 20 data) dijadikan data input untuk training JST. JST belajar dengan iterasi maksimum yang ditentukan adalah 50000. Bobot-bobot interkoneksi akhir dari JST ini akan digunakan oleh JST klasifikasi untuk menentukan kelas dari masing-masing obyek .

3. PENGOLAHAN CITRA DIGITAL

3.1. Proses Akuisisi Citra

Citra direpresentasikan dalam bentuk array 2 dimensi (2D) atau matriks. Pada penelitian ini, akan digunakan citra dengan ukuran 240x320. Representasi citra digital diilustrasikan pada Gambar di bawah ini.

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & f(0,319) \\ f(1,0) & f(1,1) & f(1,319) \\ & & \\ f(239,0) & f(239,1) & f(239,319) \end{bmatrix}$$

Gambar 1 Fungsi f(x,y) citra

Dalam bentuk matriks citra digital dengan berbagai derajat keabuan, citra direpresentasikan sebagai.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	230	235	240	240	245	0
0	0	210	220	232	235	245	0
0	0	200	240	235	235	245	0
0	0	185	215	220	235	243	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Gambar 2. Citra digital dengan berbagai derajat keabuan

3.2. Proses Thresholding

Thresholding digunakan untuk mengubah citra yang memiliki banyak derajat keabuan (*gray scale*) menjadi citra biner atau citra yang hanya memiliki dua derajat keabuan. Secara sederhana, *thresholding* digunakan untuk menghilangkan informasi citra yang memiliki derajat keabuan di bawah *threshold* yang ditentukan.

$$g(x,y)=\begin{Bmatrix} Ga & jika & f(x,y)\le T, \\ Gb & jika & f(x,y)>T \end{Bmatrix}$$

Proses *thresholding* yang dideskripsikan oleh persamaan di atas mereduksi berbagai derajat keabuan citra sehingga hanya ada dua derajat keabuan, yaitu *Ga* dan *Gb*, sedangkan $g(x,y)$ merupakan citra yang menampung dua derajat keabuan tersebut. Manfaat dari *thresholding* adalah dapat mereduksi jumlah komputasi (*reducing computation costs*) pada pengolahan citra.

3.3. Proses Region Growing

Region growing adalah proses untuk mendeteksi keberadaan obyek pada citra yang terbentuk[1]**.** Algoritma *region growing* (RG) akan mengidentifikasi daerah-daerah yang mengandung unsur-unsur obyek dan derau. Setelah ditentukan apakah daerah itu obyek atau bukan. Jika derau, maka daerah itu akan dihilangkan dengan mengganti nilai pixelnya dengan nilai pixel latar belakang. Jika tidak, maka daerah itu merupakan obyek. Bila di suatu bidang citra terdapat dua atau lebih obyek, maka setiap obyek akan memiliki nilai pixel yang berbeda [4].

3.4. Pendeteksian Tepi

Proses pendeteksian tepi dilakukan dalam dua tahap, yaitu dilasi dan dilanjutkan dengan subtraksi. Algoritma kedua proses tersebut dijelaskan sebagai berikut[7]:

3.4.1 Dilasi

Algoritma:
UNTUK setiap pixel pada citra
  JIKA pixel pada citra adalah pixel obyek
  MAKA ganti nilai derajat keabuan kedelapan pixel tetangga (0) dengan 255
Hasil:
obyek mengalami perbesaran satu pixel

3.4.2 Subtraksi

Algoritma:
UNTUK setiap pixel pada citra
  JIKA pixel pada citra OBYEK = pixel DILASI
  MAKA ganti nilai derajat keabuan DILASI dengan 0
Hasil:
citra tepi obyek

3.5. Ekstraksi Ciri Obyek

Ada empat ciri yang akan diekstrak dari setiap obyek, yaitu *bounding rectangle*, *bounding circle*, invarian momen Hu orde 1 dan invarian momen Hu orde 2. Namun, untuk mendapatkan keempat ciri

tersebut, perlu ditentukan telebih dahulu pusat massa setiap obyek.

3.5.1 Pusat Massa

Pusat massa suatu objek dalam arah horisontal x (m_x) di dapat dengan cara menjumlahkan semua koordinat x dari pixel-pixel objek, kemudian dibagi dengan luas objek, dalam hal ini adalah jumlah pixel yang membangun objek tersebut (n). Demikian pula dengan pusat massa objek dalam arah vertikal y (m_y). Persamaan untuk mendapatkan pusat massa ini adalah:

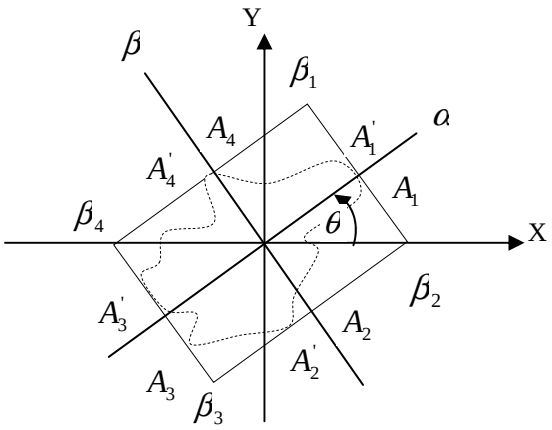
$$m_x = \frac{\sum_{i=1}^{n} x_i}{n} \qquad m_y = \frac{\sum_{j=1}^{n} y_j}{n}$$

Sebagaimana disebutkan di atas, informasi ini akan digunakan untuk menentukan ciri-ciri obyek.

3.5.2 Bounding Rectangle

Bounding Rectangle adalah perbandingan antara luas obyek dengan luas persegi/persegi panjang terkecil yang dapat melingkupi obyek. Awal dari proses penentuan nilia ciri *bounding rectangle* adalah dengan menentukan orientasi obyek.

$$\alpha = x\cos\theta + y\sin\theta$$
$$\beta = -x\sin\theta + y\cos\theta$$



Gambar 3 Bounding Rectangle

Sisi-sisi persegi panjang adalah $\alpha_{max} - \alpha_{min}$ dan $\beta_{max} - \beta_{min}$.

3.5.3 Bounding Circle

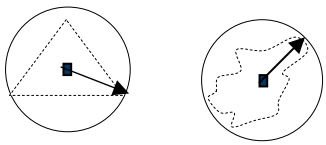
Salah satu obyek yang memiliki bentuk yang sangat sederhana adalah lingkaran. Di samping sederhana, lingkaran memiliki bentuk yang sangat simetris. Seperti diketahui, luas dan keliling sebuah lingkaran dapat diketahui dengan:

$$A = \pi R^2 \quad (2.18)$$
$$P = 2\pi R = \pi D \quad (2.19)$$

dimana R adalah jari-jari dan D merupakan garis tengah lingkaran. Komputasi untuk kedua properti lingkaran ini dapat dilakukan pada region-region citra, dan ini merupakan bentuk dasar dari ukuran kebundaran (*circularity*) [8]. Rasio P^2/A untuk sebuah lingkaran adalah 4π, dan ini merupakan nilai minimum untuk setiap region. Dari sini, dapat dirumuskan ukuran kebundaran obyek adalah:

$$C = \frac{P^2}{4\pi A} \quad (2.20)$$

C memiliki nilai minimum 1 untuk lingkaran dan terus menaik untuk region yang lebih kompleks.



Gambar 4 *Bonding Circle*

Pendekatan yang dilakukan pada penelitian ini adalah memberikan nilai maksimum 1 dimiliki untuk lingkaran dan nilai yang terus menurun untuk obyek yang lebih kompleks. Nilai ini dapat mewakili *bounding circle*. Perbandingan antara luas suatu lingkaran dengan luas lingkaran terkecil yang dapat menutupi lingkaran tersebut adalah 1.

$$C = \frac{1}{\left(\frac{P^2}{4\pi A}\right)} = \frac{4\pi A}{P^2}$$

3.5.4 Momen Invarian Hu Orde 1

Invarian momen Hu adalah perangkat popular untuk mendeskripsikan ciri geometri pada sistem identifikasi obyek dan pengenalan karakter[12]. Formulasi invarian ini independen terhadap translasi, rotasi, dan penskalaan. Terdapat tujuh jenis invarian Hu ini, namun hanya dua saja yang digunakan pada penelitian ini, yaitu invarian Hu orde 1 dan orde 2.

Momen yang mentransformasikan fungsi citra f(i,j) pada sistem diskrit dinyatakan dengan persamaan (?).

$$m_{pq} = \sum_{i=0}^{H-1}\sum_{j=0}^{W-1} i^p j^q f(i,j)$$

dimana H dan W masing-masing merupakan tinggi lebar citra. Selanjutnya, momen pusat untuk suatu citra dinyatakan dengan:

$$\mu_{pq} = \sum_{i,j \in R} \sum \left(i - \bar{i}\right)^p \left(j - \bar{j}\right)^q$$

dimana μ_{20} adalah pusat arah horisontal, μ_{02} adalah pusat arah vertikal, dan μ_{11} adalah pusat arah diagonal. Nilai $\hat{i}$ dan $\hat{j}$ masing-masing didapatkan dengan.persamaan:

$$\bar{i} = (m_{10} / m_{00}),\ \ \bar{j} = (m_{01} / m_{00})$$

m_{00} adalah jumlah total pixel yang membentuk obyek. sedangkan m_{10} dan m_{01} adalah pusat massa obyek.

Momen pusat yang terbentuk sensitif terhadap transformasi rotasi dan penskalaan. Maka dari itu dilakukan normalisasi terhadap momen pusat (μ_{pq}) melalui persamaan:

$$\eta_{pq} = \mu_{pq} / \mu_{00}^{\gamma} \quad \text{(z)}$$

dimana $\gamma = (p + q)/2 + 1$

Dari persamaan terakhir maka ciri ketiga dari obyek dapat ditentukan, yaitu invarian momen Hu orde 1 yang didefinisikan sebagai:

$$\phi_1 = \eta_{20} + \eta_{02}$$

3.5.5 Momen Invarian Hu Orde 2

Dari persamaan (z) dapat ditentukan ciri keempat obyek melalui persamaan:

$$\phi_1 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

3.6 Normalisasi Ciri

Agar dapat dijadikan input JST, keempat ciri obyek ini perlu dinormalisasi, yaitu menetapkan nilai-nilai numerik keempat ciri tersebut berada pada *range* 0 dan 1. Ciri obyek yang pertama dan kedua telah mengalami proses normalisasi 'alami', hal ini dikarenakan nilai perbandingan antara obyek dengan persegi atau lingkaran pasti kurang dari satu dan lebih dari nol.

Nilai-nilai numerik untuk ciri invarian momen Hu orde 1 terdapat pada *range* 0.10 sampai dengan 0.20. Sedangkan nilai invarian momen Hu orde 2 berada di luar range normal (0 sampai 1). Normaliasai pada ciri invarian momen Hu orde 1 dapat dilakukan dengan sederhana , yaitu dengan mengalikan ciri ini untuk setiap obyek dengan 3 sehingga nilainya tersebar dari 0 sampai mendekati 1.

$$nh_1 = 3 * \phi_1$$

Sedangkan untu orde 2, normalisasi dapat dilakukan dengan persamaan.

$$nh_2 = 0.05 * (3\log\left(\frac{\phi_2 + 0.0001}{0.0101}\right)$$

Dengan demikian, keempat ciri diatas telah memenuhi syarat sebagai input JST.

4. ALGORITMA JARINGAN SYARAF TIRUAN BACK-PROPAGATION

Arsitektur jaringan syaraf tiruan yang dibangun terdiri dari empat node pada lapisan input, 10 node pada lapisan *hidden* dan tujuh node pada lapisan output. Jumlah node lapisan input disesuaikan dengan jumlah ciri obyek yang telah dijelaskan sebelumnya, yaitu *bounding rectangle*, *bounding circle*, invarian momen Hu orde 1 dan invarian momen Hu orde 2.

4.1 Format Pebacaan Ciri Obyek

Algoritma:

```
FOR k=1 to k=jum_obyek do
    FOR l=1 to l=jum_data do
        FOR m=1 to m=jum_input do
            Baca data dari file
```

Data dari file ini kemudian disimpan pada array tiga dimensi dengan variabel **xxx**[n][k][l].

4.1 Format Target Output

Format target output yang diinginkan diperlihatkan pada tabel 1 di bawah.

Tabel 1 Target Output

Output	Kelas Obyek						
	1	**2**	**3**	**4**	**5**	**6**	**7**
O_1	1	0	0	0	0	0	0
O_2	0	1	0	0	0	0	0
O_3	0	0	1	0	0	0	0
O_4	0	0	0	1	0	0	0
O_5	0	0	0	0	1	0	0
O_6	0	0	0	0	0	1	0
O_7	0	0	0	0	0	0	1

4.2 Propagasi Maju

Tahap propagasi maju dimulai dengan memberikan suatu pola masukan paa jaringan. Kemudian tiap unit input akan mendistribusikan pola masukan ke neuron (node) pada lapisan berikutnya (*hidden*).

1. Set bobot-bobot interkoneksi sel
   Proses ini memberikan nilai awal bobot-bobot interkoneksi sel-sel JST. Nilai yang diberikan berupa bilangn acak berkisar antara -0.9 hingga 0.9
2. Baca pola belajar
   Proses ini melakukan pembacaan pola belajar, berupa pasangan empat buah ciri bentuk, momen, dan data keluaran yang sesuai, dalam representasi nilai-nilai numeriknya.

3. Set masukan dan target keluaran JST
   Proses ini menyiapkan nilai-nilai masukan sel pada lapisan input dan nilai-nilai target sel pada lapisan output berdasarkan pola belajar yang dibaca pada langkah 2. Pengaturan masukan dan target keluaran pola belajar ini dilakukan bergantian, yaitu untuk obyek pertama sampai keempat untuk sampel pengambilan pertama, lalu obyek pertama sampai ketujuh untuk sampel pengambilan kedua, demikian seterusnya sampai sampel pengambilan kesepuluh, dan kemudian dilakukan proses ulang untuk sampel pengambilan pertama, kedua dan seterusnya.

4. Proses menghitung respon unit pada lapisan input dan *hidden*.
   Berdasarkan parameter-parameter yang dimilikinya, proses ini melakukan perhitungan nilai keluaran fungsi aktivasi tiap sel pada lapisan *input* dan *hidden*.

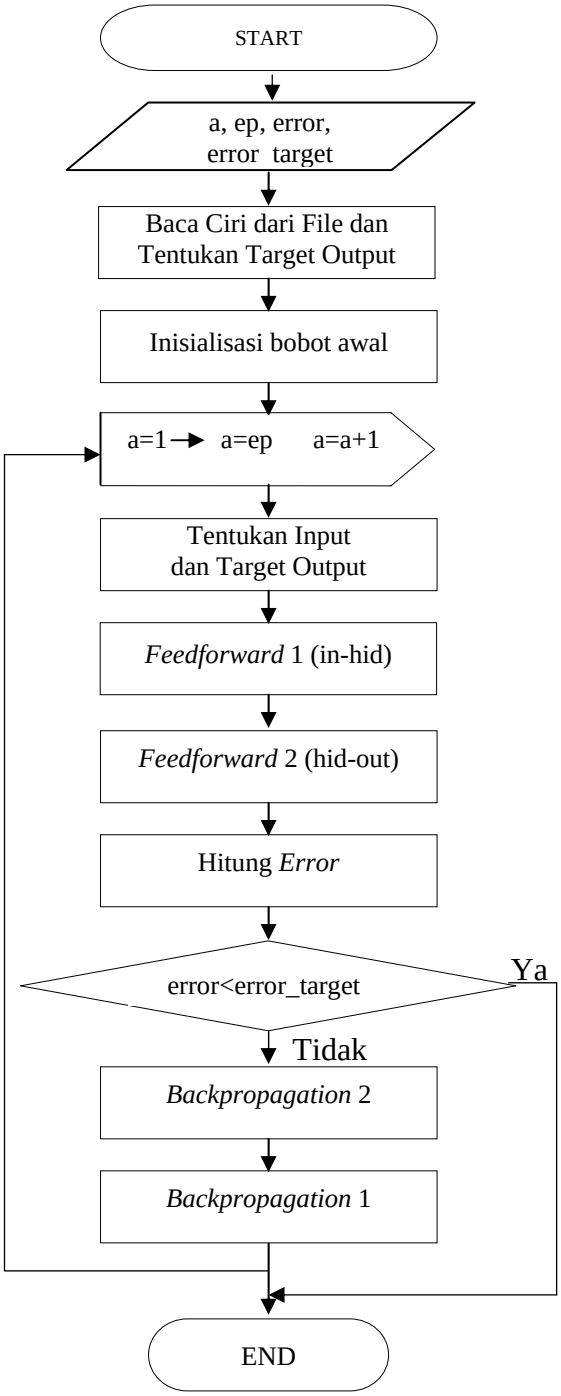
5. Proses menghitung fungsi keluaran sel-sel pada lapisan output
   Berdasarkan nilai-nilai keluaran tiap sel lapisan *hidden* yang menjadi masukan bagi sel-sel lapisan output, dilakukan proses penghitungan nilai keluaran fungsi aktivasi sel lapisan output.

4.3 Propagasi Balik

6. Proses penghitungan *error* dan penyesuaian bobot-bobot interkoneksi sel untuk lapisan output.
   Proses ini melakukan perhitungan *error* keluaran sel-sel pada lapisan output, berdasarkan nilai keluaran sel yang diperoleh pada proses 6 dan nilai nilai target keluaran sel-sel output yang telah diberikan pada proses 1. Nilai *error* yang diperoleh dipergunakan untuk memperbaiki bobot-bobot interkoneksi ke sel-sel lapisan output.

7. Proses penghitungan *error* dan penyesuaian bobot-bobot interkoneksi sel untuk lapisan *hidden*
   Error tiap sel pada lapisan output juga dirambatkan ke sel-sel lapisan dalam sehingga setiap sel lapisan dalam memiliki nilai *error*-nya masing-masing untuk memperbaiki bobot-bobot interkoneksi sel-sel lapisan dalam. Apaila lapisan dalam lebih dari satu lapis, proses merambatkan itu berlanjut, lapis demi lapis, sehingga setiap sel lapisan dalam memiliki nilai *error*-nya masing-masing memperbaiki bobot-bobot interkoneksi dari sel-sel lapisan sebelumnya.

8. Ulangi langkah 2 sampai 6 hingga diperoleh nilai error keluaran yang diperkenankan



Gambar 5 Flowchart Pelatihan JST

Nilai *error* keluaran dalam proses ini menjadi kendali untuk menghentikan proses belajar JST. Apabila *error* keluaran yang diberikan telah terpenuhi maka proses belajar akan dihentikan. Proses juga akan berhenti jika jumlah pelatiha telah mencapai jumlah iterasi maksimal.

9. Penulisan bobot interkoneksi sel
   Proses ini melakukan penyimpanan pengetahuan JST (berupa nilai-nilai bobot interkoneksi sel JST) selama melakukan

proses belajar ke media penyimpan, agar dapat dipergunakan pada proses klasifikasi. Dalam tugas akhir ini, penulisan bobot interkoneksi dituliskan ke dalam bentuk file teks (.txt) agar mudah untuk dibaca kembali dalam aplikasinya.

5. IMPLEMENTASI PERANGKAT LUNAK

Seluruh modul program ini diimplementasikan ke dalam ke dalam bentuk perangkat lunak yang dibuat dengan Microsoft Visual C++ 6.0. Perangkat lunak yang dibuat terdiri dari tiga modul, yaitu: modul pengolahan citra, modul pelatihan JST, dan modul klasifikasi obyek

5.1 Modul Pengolahan Citra

Prosedur-prosedur:

1. GrayScaling
2. ThresholdingRegionGrowing
3. DilasiObject
4. Subtraksi
5. VisualisasiCitra
6. LuasObyek
7. PusatMassa
8. BoundingRect
9. BoundingCirc
10. MomenPusat
11. MomenHu

5.2 Modul Pelatihan JST

Prosedur-prosedur:

1. ReadDataFromFile
2. RandomizeInitWeights
3. TargetOutput
4. *Feedforward*
5. *Backpropagation*
6. WriteWeightsFile

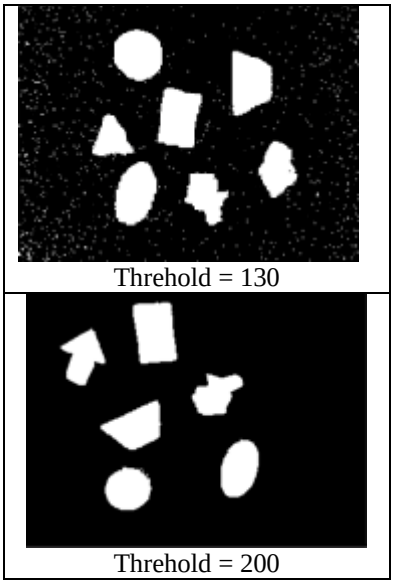
5.3 Modul Klasifikasi Obyek

Prosedur-prosedur:

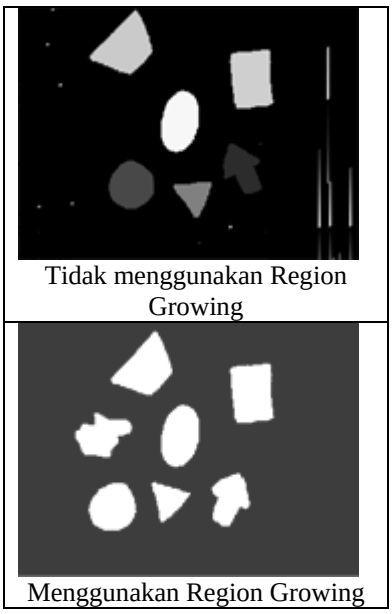
1. ReadWeightsFromFile
2. SetObjectFeatures
3. ThresholdingOutput
4. Classification
5. WriteResultsToFile

6. HASIL PENGUJIAN SISTEM

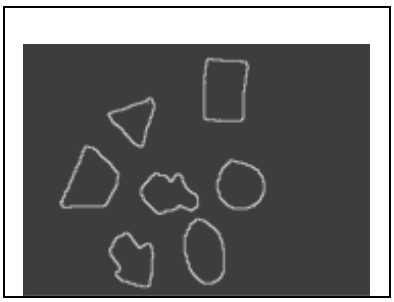
6.1 Uji Pengolahan Citra



Gambar 6 Citra Hasil *Thresholding*



Gambar 7 Citra Hasil *Region Growing*



Gambar 8 Tepi Obyek

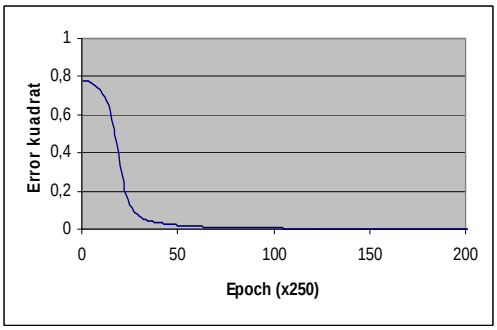
6.2 Uji Pelatihan JST

Tabel 2 Uji Variasi Arsitektur JST Backpropagation

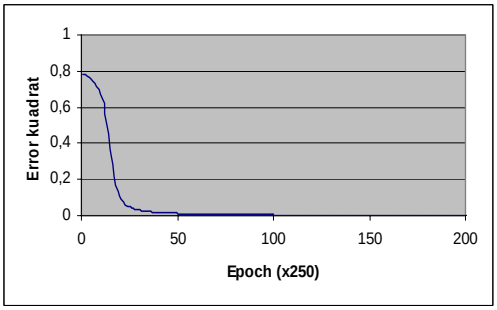
Jumlah obyek	= 7			
Neuron input	= 4			
Neuron output	= 7			
Learning rate	= 0.9			
Error target	= 0.0001			
Hidden	Epoch	Benar	Salah	Tidak dikenal
4	Gagal	5	1	1
5	29090	4	1	2
6	449100	5	1	1
7	Gagal	7	0	0
8	49220	7	0	0
9	39480	7	0	0
10	19490	7	0	0
11	9930	7	0	0

Tabel 3 Uji Variasi Alpha/Learning Rate

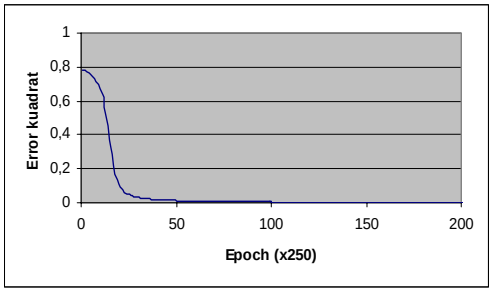
Jumlah obyek	= 7			
Neuron input	= 4			
Neuron hidden	= 10			
Neuron output	= 7			
Error	= 0.0001			
Alpha	Epoch	Benar	Salah	Tidak dikenal
0.1	Gagal	6	0	1
0.2	Gagal	6	0	1
0.3	43680	6	0	1
0.4	29220	6	1	0
0.5	21610	6	1	0
0.6	17110	5	1	1
0.7	14230	6	1	0
0.8	19150	6	1	0
0.9	19490	7	0	0



(a)



(b)



(c)

Gambar 9. Grafik Error Kuadrat dengan Alpha = (a) 0.3, (b) 0.7, dan (c) 0.9

7.3 Uji Klasifikasi Obyek untuk Posisi Acak

Tabel 4 Uji klasifikasi obyek posisi acak

Klasifikasi ke	OBYEK			
	1	2	3	4
1	1	1	1	1
2	1	1	1	1
3	1	0	1	1
4	1	1	1	1
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1
8	1	1	1	1
9	1	0	1	1
10	1	1	1	1

Klasifikasi ke	OBYEK			Keberhasilan (%)
	5	6	7	
1	1	1	1	100
2	1	1	1	100
3	1	1	1	85.71
4	1	1	1	100
5	1	1	1	100
6	1	1	1	100
7	1	1	1	100
8	1	1	1	100
9	1	1	1	85.71
10	1	1	1	100
Rata-rata				**97.14**

Tabel 4 Uji klasifikasi obyek yang diperkecil dan diperbesar

Pengenalan ke	Keberhasilan (%)
1	100
2	85.71
3	100
4	100
5	100
6	85.71
7	100
8	100
9	85.71
10	85.71
Rata-rata	**94.28**

Pengenalan ke	Keberhasilan (%)
1	85.71
2	85.71
3	100
4	100
5	100
6	100
7	100
8	85.71
9	100
10	100
Rata-rata	**95.71**

Tabel 4 Uji klasifikasi obyek yang mengalami deformasi bentuk

Pengenalan ke	Keberhasilan (%)
1	100
2	100
3	85.71
4	85.71
5	85.71
6	71.43
7	100
8	85.7
9	71.43
10	85.7
Rata-rata (%)	**87.12**

7. KESIMPULAN

1. Kamera QuickCam mampu menghasilkan citra dengan kualitas cukup baik. Dengan *frame-grabber* yang terintegrasi langsung pada *driver*-nya, kamera ini praktis untuk diimplementasikan pada sistem pengolahan citra *real-time*.
2. Ciri-ciri obyek yakni *bounding rectangle*, *bounding circle*, invarian Hu orde ke-satu dan orde ke-dua cukup akurat digunakan untuk mengklasifikasi obyek dan handal terhadap transformasi obyek (yaitu translasi, rotasi, dan penskalaan).
3. Sistem mampu mengenali obyek yang diperkecil dan diperbesar dengan prosentase keberhasilan masing-masing 94,78 % dan 95.71%. Sistem juga diuji untuk mengenali obyek yang mengalami deformasi bentuk akibat bayangan yang terbentuk karena kemiringan kamera dengan prosentase keberhasilan 87.14 %.
4. Struktur jaringan syaraf tiruan yang optimal adalah empat neuron pada lapisan input, 10 neuron pada lapisan *hidden*, tujuh neuron pada lapisan output, konstanta kecepatan belajar 0.9, dan *error* target 0.0001. Jumlah neuron 10 ditentukan melalui parameter ketepatan dalam mengklasifikasi dan waktu pelatihan minimum yang diperlukan.

Ucapant terima kasih:
- *Bapak Yeffry Handoko Putra M.T*
- *Bapak Nana Juhana ST,MT*
- *Teman-teman seangkatan (2000) di Jurusan Teknik Informatika UNIKOM yang telah memberikan kritisisasi pada hasil penelitian ini.*
- *Semua pihak yang telah membantu dalam pembuatan aplikasi pengolahan citra realtime ini.*

DAFTAR PUSTAKA

[1] Budiawan, R., *Sistem Pengendalian Robot Penangkap Objek Bergerak Berbasis Visual*, Tugas Akhir Teknik Elektro ITB, Bandung, 2000

[2] Davies, E.R., *Machine Vision: Theory, Algorithms, and Practicalities*, Academic Press, 1990

[3] Dewi, S., *Artificial Intelligence* (*Teknik dan Aplikasinya*), Yogyakarta, 2002

[4] Jahne, B., Haufsecker, H., *Computer Vision and Apllications*, Academic Press, 2000

[5] Lau, K.L, Siu W.C., Law N.F., *Best-fit Ellipse Concept from the Second Order Moments for Object Retrieval*, Massachusetts Institute of Technology (MIT), 2002, pp.
URL: http://www.mit.edu

[6] Masters, Timothy, *Signal and Image Processing with Neural Networks – A C++ Sourcebook*, Toronto, John Wiley and Sons, Inc., 1994

[7] Oh, P. *Template for Real-Time Image Processing Development*, Drexel University, 2001. URL: http://www.cs.columbia.edu

[8] Parker, J.R., *Practical Computer Vision Using C*, John Wiley and Sons, 1994

[9] Putra R., *Sistem Kendali Robot Manipulator Berbasis Visual Dengan Pegendali ANFIS Untuk Penjejakan Objek Bergerak*, Tugas Akhir Teknik Elektro ITB, Bandung, 2002

[10] Schalkoff, R.J., *Digital Image Processing and Computer Vision*, Wiley, 1989

[11] Schildt, H., *C: The Complete Reference*, Third Edition, Osborne McGraw-Hill, California, 1995

[12] Siong, A.W., Resmana, *Pengenalan Citra Obyek Sederhana Dengan Menggunakan Metode Jaringan Syaraf Tiruan SOM*, UK Petra, Surabaya, 1999, pp.
URL: http://www.petra.ac.id

[13] Weeks, A.R., *Fundamentals of Electronic Image Processing*, IEEE Press, 1996