

## PENGENALAN OBJEK SEDERHANA BERDASARKAN CIRI WARNA DENGAN METODE *BACK PROPAGATION*

Uning Lestari<sup>1</sup>

### ABSTRACT

*Human being has extraordinary ability for recognizing objects based on these feature knowledge that has been observed. Computer is a good, fast, accurate and efficient data processing device. Computer able to solve the unstructured problem, uncomplete information and qualitative data Artificial Intelligence (AI) problem.*

*Pattern recognition is one of AI applications for classifying and identifying pattern or complex object by characteristics measurement or object features (color, shape, texture). A system to identify two object that have almost the same shape but different colors is made in this research. Pattern recognition method is made by one of the methods of Artificial Neural Network that is supervised (Supervised Learning) that is Back Propagation. Example object in this research using apple image and orange image. Calculation result in Backpropagation shows with 14 images colour, 10x10 matrixes pixel and random selected weight, so sometimes to get large error. The researcher define if error more than 0.5 the system is not able to recognize tested images. To get high accurately simulation data is needed. The result of this research shows that learning process by Backpropagation, systems recognize different objects based on the color feature.*

**Key words:** *Pattern Recognition, Backpropagation, image processing*

### INTISARI

Manusia memiliki kemampuan yang luar biasa untuk mengenali objek-objek berdasarkan ciri-ciri atau pengetahuan yang pernah diamatinya. Komputer merupakan alat pengolah data yang baik, cepat, akurat, dan efisien selain itu juga dapat untuk menyelesaikan masalah-masalah yang tidak terstruktur, informasi tidak lengkap, dan data yang bersifat kualitatif (masalah *Artificial Intelligence*).

Pengenalan pola yang merupakan salah satu aplikasi dalam AI (*Artificial Intelligence*) bertujuan untuk mengklasifikasikan dan mendeteksi pola atau objek kompleks melalui pengukuran sifat-sifat atau ciri-ciri objek (warna, bentuk, tekstur). Pada penelitian ini dibuat suatu sistem yang dapat mengenali pola dua buah objek yang hampir mirip bentuknya tapi mempunyai perbedaan warna yang mencolok, oleh karena itu diambil *feature* (ciri) warna pada pengenalan pola objeknya. Sistem pengenalan pola tersebut dibuat dengan salah satu metode Jaringan Syaraf Tiruan yang terawasi (*Supervised Learning*) yaitu *Back Propagation*.

Hasil penelitian ini menunjukkan dengan pembelajaran yang dilakukan beberapa kali sistem dapat mengenali dua buah citra objek sederhana yaitu citra Jeruk dan Apel berdasarkan ciri warnanya. Jika  $error < 0.5$  sistem dapat mengenali kedua citra tersebut.

**Kata Kunci:** *Back Propagation, Pengenalan Pola, colour feature*

### PENDAHULUAN

Manusia memiliki kemampuan luar biasa untuk mengenali objek-objek berdasarkan ciri-ciri atau pengetahuan yang pernah diamatinya. Perkembangan teknologi komputer sekarang ini sangat cepat, selain komputer merupakan alat pengolah data yang sangat baik, cepat, akurat, dan efisien, komputer juga dapat berfungsi yang lebih jauh lagi yaitu kemampuan komputer untuk menyelesaikan-

kan masalah-masalah yang tidak terstruktur, informasi tidak lengkap, dan data yang bersifat kualitatif. Masalah-masalah tersebut harus diselesaikan berdasarkan pengalaman dan membutuhkan pengetahuan (*knowledge*) atau fakta dari permasalahan yang ada. Dengan kata lain komputer harus mempunyai intelegensia atau kecerdasan. Dengan kecerdasan tersebut komputer dapat mengam-

<sup>1</sup> Staf pengajar Jurusan Teknik Informatika, ISTA, Yogyakarta

bil kesimpulan atau solusi berdasarkan masalah yang diinputkan.

*Artificial Intelligence* merupakan ilmu komputer yang mengembangkan kemampuan komputer agar dapat membantu memecahkan masalah yang membutuhkan kecerdasan dan fungsi-fungsi khusus yang mempunyai analogi terhadap tingkah laku manusia. Banyak aplikasi yang termasuk dalam *Artificial Intelligence* (AI) misalnya di robotika, games, expert system, penginderaan, pemahaman bahasa, *problem solving*, pengenalan pola (*pattern recognition*) dan lain-lain.

Jaringan Syaraf Tiruan (JST) adalah suatu sistem pemroses informasi yang berdasarkan model jaringan Syaraf biologi, bermacam-macam aplikasi berbasis jaringan syaraf tiruan telah dikembangkan diberbagai bidang salah satunya dalam hal pengenalan pola (*Pattern and Recognition*) (Schalkoff, 2000). Dengan adanya teknologi ini komputer dapat menganalisa serta mengetahui objek-objek yang ada baik berbentuk 2 dimensi maupun 3 dimensi sesuai dengan tujuan yang akan diimplementasikan kedalam perangkat lunak dan perangkat keras.

Contoh implementasi pengenalan pola yaitu sebagai aplikasi untuk pengenalan pola citra buah berdasarkan ciri warna dimana fungsi utamanya menganalisa berdasarkan data yang telah ada melalui proses belajar dan proses pengenalan karakter, hal ini memanfaatkan algoritma Propagasi Balik (*Backpropagation*) karena algoritma ini termasuk ke dalam proses belajar terawasi (*Supervised Learning*) dimana proses atau tahap pelatihan dilakukan dengan perbaikan kesalahan serta pencapaian suatu target tertentu. Model arsitektur jaringan yang digunakan adalah jaringan layar jamak (*multi layer network*) karena *Backpropagation* mempunyai beberapa lapisan seperti lapisan masukan, lapisan keluaran dan lapisan tersembunyi, sedang fungsi Aktivasi yang dipakai adalah fungsi biner sigmoid karena sangat sesuai dengan algoritma Propagasi Balik.

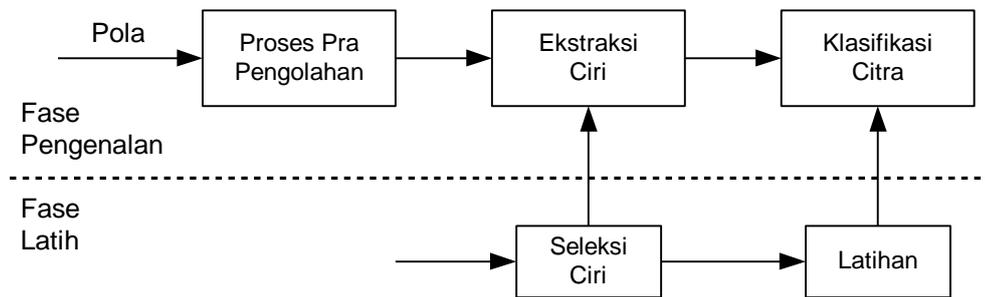
Penelitian di bidang pengenalan pola ini telah banyak digunakan untuk aplikasi-aplikasi misalnya untuk pemetaan tanah melalui foto udara, pengenalan huruf, mendeteksi kelainan tubuh berdasar-

kan foto sinar X dan lain sebagainya. Penelitian tentang pengenalan pola ini telah dilakukan juga oleh beberapa peneliti diantaranya Ang Wie Siong dkk (2000) yang membuat sistem pengenalan pola objek sederhana dengan menggunakan metode Jaringan Syaraf Tiruan *Self Organizing Map* (SOM). Pada penelitian ini jaringan dilatih untuk mengenali pola sebanyak lima belas macam meliputi tiga bentuk (kubus, tabung, dan kerucut). Jaringan dilatih terus menerus sampai diperoleh error tertentu dan diharapkan sistem dapat mengenali citra dengan eror yang minimum.

Pada penelitian ditujukan untuk menelaah lebih jauh pembelajaran (*learning*) pada kasus *feature based image processing* untuk kasus pengenalan objek berdasarkan pola warna (*colour feature*) dengan menggunakan salah satu metode Jaringan Syaraf Tiruan yaitu metode *Backpropagation*.

Untuk pengenalan objek dalam citra, dilakukan dengan memasukkan citra ke sistem, disini diberikan contoh untuk citra Apel dan Jeruk dimana bentuk kedua buah tersebut hampir mirip hanya warna yang membedakan. Dari ciri (*feature*) tersebut diambil ciri warna untuk pembedanya dan diharapkan sistem dapat mengenali apakah citra tersebut buah Apel atau Jeruk berdasarkan ciri warnanya.

Sistem pengenalan pola ini memiliki batasan-batasan yaitu: citra hanya terdiri dari satu objek, perbedaan warna objek satu citra dengan objek citra yang lain sangat mencolok, warna latar belakang (*background*) pada setiap citra dibuat sama, bentuk objek pada setiap citra hampir sama dengan warna yang berbeda, di sini digunakan citra dengan objek jeruk dan apel, dan citra sudah terbebas dari noise. Pola adalah intensitas yang terdefinisi dan dapat diidentifikasi melalui ciri-cirinya (*features*). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi.



Gambar 1 Model proses pengenalan pola

Secara umum pengenalan pola bertujuan untuk mengklasifikasikan dan mendeteksi pola atau objek kompleks melalui pengukuran sifat-sifat atau ciri-ciri objek. Tahapan dan tujuan proses pengenalan pola dibedakan menjadi dua yaitu:

- a. Memasukkan pola kedalam suatu kelas pola yang belum dikenal prosesnya disebut clustering atau klasifikasi tidak terawasi
- b. Mengidentifikasi pola sebagai anggota dari kelas yang sudah dikenal, prosesnya disebut klasifikasi terawasi.

Fase proses pengenalan pola beroperasi dalam dua fase (lihat gambar 1) yaitu:

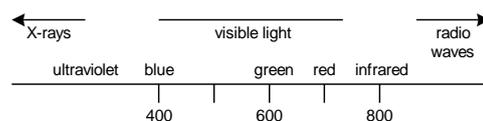
1. Fase latihan  
Pada fase ini pengekstraksi ciri menentukan ruang ciri yang sesuai untuk merepresentasikan pola,

2. Fase Pengenalan  
Pengklasifikasi menentukan suatu pola masukan sebagai salah satu kategori objek yang telah dispesifikasikan menurut ciri-ciri objek.

Citra didefinisikan sebagai fungsi intensitas cahaya dua dimensi  $f(x,y)$ , dimana  $x$  dan  $y$  merupakan koordinat spasial dan nilai  $f$  pada suatu titik  $(x,y)$  adalah intensitas cahaya (brightness/gray level) dari citra tersebut di titik  $(x,y)$ . Citra digital adalah citra dengan  $f(x,y)$  yang nilainya didigitalisasi (dibuat diskrit) baik dalam kordinat spasialnya maupun gray levelnya. Digitalisasi dari koordinat spasial citra disebut dengan *image sampling*, sedangkan digitalisasi dari gray level citra disebut dengan *gray*

*level quantization*. Citra digital dapat dideskripsikan sebagai suatu matriks dimana baris dan kolomnya menunjukkan suatu titik dalam citra (Munir, 2002), dan nilai elemen matriks tersebut menunjukkan gray level di titik tersebut. Elemen-elemen dari ci-tra tersebut disebut pixel (*picture ele-ments*).

Persepsi visual citra berwarna (*color image*) umumnya lebih kaya dibandingkan dengan citra hitam putih (*grayscale*). Citra berwarna menampilkan citra objek seperti warna aslinya. Warna yang diterima oleh mata dari sebuah objek ditentukan oleh warna sinar yang dipantulkan oleh objek tersebut. Warna sinar yang direspon oleh mata adalah sinar tampak (*visible light*) dengan panjang gelombang 400 (biru) sampai 700 nm (merah) (gambar 2).



Gambar 2 : Spektrum Warna

Kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red (R)*, *green (G)*, dan *blue (B)*. Ketiga warna tersebut dinamakan warna pokok (*primaris*), dan sering disingkat sebagai warna dasar RGB. Warna-warna lain dapat dihasilkan dengan mengkombinasikan ketiga warna pokok tersebut dengan persentase tertentu.

$$W = a R + b G + c B$$

- $W$  : warna kombinasi  
 $a, b, c$  : persentase dari masing-masing warna pokok  
 $R, G, B$  : *red, green, blue*

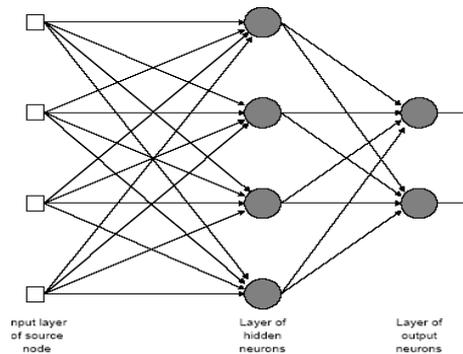
Jaringan syaraf tiruan adalah suatu sistem pemrosesan informasi yang cara kerjanya memiliki kesamaan tertentu dengan jaringan syaraf biologis (Fausett, 1994). Jaringan syaraf tiruan dikembangkan sebagai model matematis dari syaraf biologis dengan berdasar asumsi bahwa :

1. Pemrosesan terjadi pada elemen-elemen sederhana yang disebut neuron
2. Sinyal dilewatkan neuron melalui penghubung
3. Setiap penghubung mempunyai bobot yang akan mengalikan sinyal yang lewat
4. Setiap neuron memiliki fungsi aktivasi yang akan menentukan nilai sinyal output

Jaringan syaraf dapat digolongkan menjadi berbagai jenis berdasarkan arsitekturnya, yaitu pola hubungan antara neuron-neuron, dan algoritma trainingnya, yaitu cara penentuan nilai bobot pada penghubung. Untuk menyelesaikan masalah ini, digunakan metode *algoritma backpropagation*.

Metode yang digunakan untuk menyelesaikan masalah di atas adalah dengan menggunakan jaringan syaraf tiruan dengan metode *backpropagation / feedforward* dengan algoritma pelatihan yang sederhana. Pelatihan dilakukan untuk pengaturan bobot-bobot yang baik. Selama proses pelatihan, bobot-bobot diatur secara iteratif untuk meminimumkan fungsi kinerja jaringan. Fungsi kinerja jaringan yang sering digunakan untuk *backpropagation* adalah *mean square error (MSE)*, fungsi ini akan mengambil rata-rata kuadrat error yang terjadi antara output jaringan dan target.

Sebagian besar algoritma pelatihan untuk jaringan *feedforward* menggunakan gradien dari fungsi kinerja untuk menentukan bagaimana mengatur bobot-bobot untuk meminimumkan kinerja. Gradien ini ditentukan dengan suatu teknik yang disebut *backpropagation*. Pada dasarnya algoritma pelatihan *backpropagation* akan menggerakkan bobot dengan arah gradien negatif.



Gambar 3. JST Propagasi Balik dengan empat node pada input layer, satu hidden layer dengan empat node, dan dua node pada output layer

Prinsip dasar algoritma *backpropagation* sederhana adalah memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi kinerja turun dengan cepat (lihat gambar 3).

*Backpropagation* merupakan algoritma pembelajaran yang biasanya digunakan oleh perseptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dengan arah mundur (*back-ward*). Untuk mendapatkan *error* tersebut tahap perambatan maju harus dikerjakan lebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan memakai aktivasi yang dapat didiferensialkan, seperti sigmoid:

$$y = f(x) = \frac{1}{1 + e^{-\alpha x}}$$

dengan :

$$f'(x) = \alpha f(x)[1 - f(x)]$$

$$\text{atau tansig : } y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{atau } y = f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

dengan :

$$f'(x) = [1 + f(x)][1 - f(x)]$$

$$\text{atau purelin: } y = f(x) = x \text{ dengan } f'(x) = 1$$

Algoritma *Backpropagation* yaitu:

- Inisialisasi bobot (ambil bobot awal dengan nilai random yang kecil)
- Tetapkan : Maksimum Epoch, Target Error, dan Learning Rate ( $\alpha$ ).
- Kerjakan langkah-langkah berikut selama (Epoch < Maksimum Epoch) dan (MSE > Target Error):

1. Epoch = Epoch+1
2. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

- *feedforward*:

- a. Tiap-tiap unit input ( $X_i$ ,  $i = 1, 2, 3, \dots, n$ ) menerima sinyal  $X_i$  dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- b. Tiap-tiap unit pada suatu lapisan tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) menjumlahkan sinyal-sinyal input terbobot:

$$z\_in_j = b1_j + \sum_{i=1}^n x_i v_{ij},$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:  $z_j = f(z\_in_j)$ , dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

- c. Tiap-tiap unit output ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) menjumlahkan sinyal-sinyal input terbobot:

$$y\_in_k = b2_k + \sum_{i=1}^p z_i w_{jk},$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:  $y_k = f(y\_in_k)$ , dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit output).

Catatan : langkah (b) dilakukan sebanyak jumlah lapisan tersembunyi.

- *Backpropagation*

- a. Tiap-tiap unit output ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) menerima target pola yang berhubung-

an dengan pola input pembelajaran hitung informasi errornya:

$$\delta 2_k = (t_k - t_k) f'(y\_in_k)$$

$$\varphi 2_{jk} = \delta_k z_j$$

$$\beta = \delta_k$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $w_{jk}$ ):  $\Delta w_{jk} = \alpha \varphi 2_{jk}$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b2_k$  :

$$\Delta b2_k = \delta \beta 2_k$$

langkah (a) ini juga dilakukan sebanyak lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

- b. Tiap-tiap unit tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di atasnya:

$$\delta\_in_j = \sum_{k=1}^m \delta 2_k w_{jk}$$

kalikan nilai ini dengan turunan dari fungsi aktivasi-nya untuk menghitung informasi error:

$$\delta 1_j = \delta\_in_j f'(z\_in_j)$$

$$\varphi 1_{ij} = \delta 1_j x_j$$

$$\beta 1_j = \delta 1_j$$

kemudian hitung koreksi bobot-bobot (yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$ ):  $\Delta v_{ij} = \alpha \varphi 1_{ij}$  hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai  $b1_j$  ) :

$$\Delta b1_j = \alpha \beta 1_j$$

- c. Tiap-tiap unit output ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) memperbaiki bias dan bobotnya ( $j = 0, 1, 2, \dots, p$ ):  
 $w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$   
 $b2_k(\text{baru}) = b2_k(\text{lama}) + \Delta b2_k$

Tiap-tiap unit tersembunyi ( $Z_j$ ,  $j = 1,2,3,\dots,p$ ) memperbaiki bias dan bobotnya ( $i = 0,1,2,\dots,n$ ):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

$$b1_j(\text{baru}) = b1_j(\text{lama}) + \Delta b1_j$$

d. Hitung MSE.

Inisialisasi bobot awal secara random. Biasanya bobot awal diinisialisasi secara random dengan nilai antara -0.5 sampai 0.5 (atau -1 sampai 1, atau interval lainnya).

Masalah utama yang dihadapi dalam *backpropagation* adalah lamanya iterasi yang harus dilakukan. *Backpropagation* tidak dapat memberikan kepastian tentang berapa epoch yang harus dilalui untuk mencapai kondisi yang diinginkan. Oleh karena itu orang berusaha meneliti bagaimana parameter-parameter jaringan dibuat sehingga menghasilkan jumlah iterasi yang relatif lebih sedikit.

a. Pemilihan Bobot dan Bias Awal

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Oleh karena itu dalam standar *Backpropagation*, bobot dan bias diisi dengan bilangan acak kecil. Nguyen dan Widrow (1990) mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat. Misalnya  $n$  = jumlah unit masukan,  $p$  = jumlah unit tersembunyi,  $\beta$  = faktor skala =  $0.7 \sqrt[n]{P}$

Algoritma inisialisasi Nguyen Widrow adalah sebagai berikut :

a. Inisialisasi semua bobot ( $v_{ji}$  (lama)) dengan bilangan acak dalam interval [-0.5, 0.5]

b. Hitung

$$\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jn}^2}$$

c. Bobot yang dipakai sebagai inisialisasi =  $v_{ji} = \frac{\beta v_{ji}(\text{lama})}{\|v_j\|}$

d. Bias yang dipakai sebagai inisialisasi =  $v_{j0}$  = bilangan acak antara -  $\beta$  dan  $\beta$

b. Jumlah Unit Tersembunyi

Hasil teoritis yang didapat menunjukkan bahwa jaringan dengan sebuah layer tersembunyi sudah cukup bagi *backpropagation* untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah layer tersembunyi kadangkala membuat pelatihan lebih mudah.

Jika jaringan memiliki lebih dari satu layer tersembunyi, maka algoritma pelatihan yang dijabarkan sebelumnya perlu direvisi. Dalam propagasi maju, keluaran harus dihitung untuk tiap layer, dimulai dari layer tersembunyi paling bawah (terdekat dengan masukan). Sebaliknya, dalam propagasi mundur, faktor  $\delta$  perlu dihitung untuk tiap layer tersembunyi, dimulai dari layer keluaran.

c. Jumlah Pola Pelatihan

Tidak ada kepastian tentang berapa banyak pola yang diperlukan agar jaringan dapat dilatih dengan sempurna. Jumlah pola yang dibutuhkan dipengaruhi oleh banyaknya bobot dalam jaringan serta tingkat akurasi yang diharapkan. Aturan kasarnya dapat ditentukan berdasarkan rumusan, yaitu:

$$\text{Jumlah pola} = \frac{\text{Jumlah bobot}}{\text{tingkat akurasi}}$$

d. Lama Iterasi

Tujuan utama penggunaan *backpropagation* adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut data pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan

mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga semua kesalahan pola pelatihan = 0.

Umumnya data dibagi menjadi 2 bagian saling asing, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai untuk pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan, kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahannya sudah meningkat, pelatihan tidak ada gunanya untuk diteruskan lagi. Jaringan sudah mulai mengambil sifat yang hanya dimiliki secara spesifik oleh data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan kemampuan melakukan generalisasi.

e. Momentum

Pada standar *backpropagation*, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya (disebut momentum) yang dimasukkan. Jadi tidak hanya pola masukan terakhir saja yang diperhitungkan.

Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain (*outlier*). Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa (berarti arah gradien sudah benar), maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.

Dengan penambahan momentum, bobot baru pada waktu ke  $(t + 1)$  didasarkan atas bobot pada waktu  $t$  dan  $(t - 1)$ . Disini harus ditambahkan

2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir. Jika  $\mu$  adalah konstanta ( $0 \leq \mu \leq 1$ ) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan

$$w_{kj}(t+1) = w_{kj}(t) + a \delta_k z_j + \mu (w_{kj}(t) - w_{kj}(t-1)) \quad \text{dan}$$
$$v_{ji}(t+1) = v_{ji}(t) + a \delta_j x_i + \mu (v_{ji}(t) - v_{ji}(t-1))$$

## PEMBAHASAN

Sistem pengenalan objek sederhana ini diimplementasikan dengan program Matlab 7. File yang digunakan berupa file citra dengan format JPEG dan TIF dengan 16 gray scale. Secara garis besar program ini akan dilakukan langkah-langkah sebagai berikut (Kusumadewi, 2002):

- Membuka file citra digital dengan format TIF.
- Membaca file citra digital dalam bentuk matriks, dengan nilai matriks tersebut adalah merupakan warna dan citra telah diresolusi menjadi  $10 \times 10$  pixel matriks untuk memudahkan dan menyederhanakan perhitungan.
- Membuat pelatihan pola warna dengan data sebanyak 14 file citra warna dengan format TIF dan ukuran matriks pixel  $10 \times 10$ .
- Dibuat matrix  $10 \times 10$  dari data pelatihan mejadi vektor kolom untuk semua data pelatihan
- Menentukan data pengujian dengan data sebanyak 3 file citra warna yang diinputkan dengan format TIF dengan ukuran matriks pixel  $10 \times 10$ .
- Membuat nilai-nilai warna citra yang berupa matriks pixel menjadi nilai riil dengan type *double*.
- Mencari nilai maksimal dari nilai maksimal vektor matiks data pelatihan maupun data pengujian
- Menormalisasi data pelatihan dan pengujian
- Menentukan data target latih dan data target uji
- Membangun jaringan syaraf tiruan dengan metode *feedforward*
- Menentukan maksimum epoch, *learning rate*, *show step*

- Melakukan pembelajaran
- Menampilkan bobot-bobot awal input, lapisan, dan bias
- Melakukan simulasi
- Melakukan pengujian data
- Menentukan error
- Menarik kesimpulan

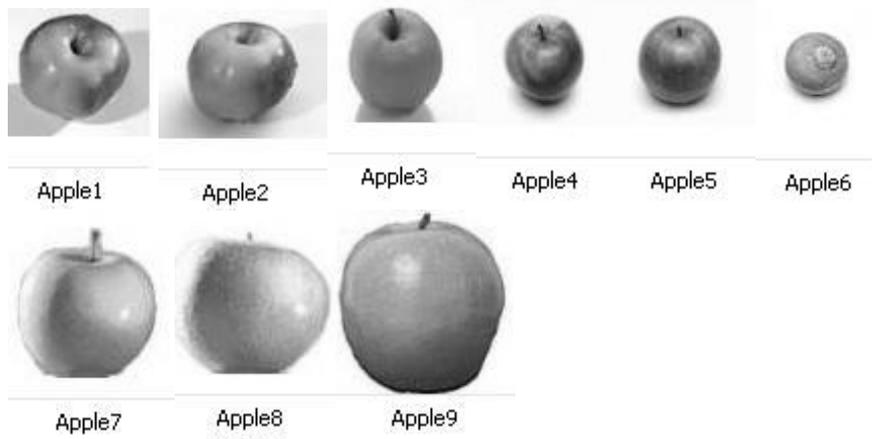
Pada pembuatan sistem menggunakan fungsi-fungsi yang ada di Matlab diantaranya fungsi:

- imread  
Fungsi ini berguna untuk membaca citra dari suatu file. Jika citra yang dibaca memiliki format warna grayscale, fungsi ini akan menghasilkan array dua dimensi yang berisi informasi intensitas grayscale dari citra tersebut. Fungsi ini mendukung format BMP, JPEG, TIF, PNG, HDF, PCX, dan XWD.
- reshape  
Menampilkan matrix  $m \times n$ , setiap elemen diletakkan dalam bentuk kolom (vektor kolom)
- double  
Befungsi membuat nilai suatu variabel  $X$  menjadi double array
- net=newff  
Befungsi untuk membangun jaringan syaraf *feedforward*
- traingdx  
Memperbaiki bobot  $dW = lr * gW$ , dimana  $dW$  = perubahan bobot,  $lr$  = learning rate, dan  $gW$  = gradien kinerja
- logsig  
Fungsi sigmoid biner, fungsi aktivasi identitas.
- net.trainParam.epochs=MaxEpoch  
Menentukan maksimum epoch, yaitu jumlah epoch maksimum yang boleh dilakukan selama pelatihan. Iterasi akan dihentikan apabila nilai epoch melebihi nilai maksimum epoch.
- net.trainParam.goal = TargetError  
Menentukan kinerja tujuan, yaitu target nilai fungsi kinerja. Iterasi akan dihentikan apabila fungsi kinerja kurang dari atau sama dengan kinerja tujuan.

- net.TrainParam.max\_perf\_inc = MaxPerfinc  
Menentukan maksimum kenaikan kinerja, yaitu nilai maksimum kenaikan error yang diizinkan, antara error saat ini dan error sebelumnya.
- net.trainParam.lr=LearningRate  
Menentukan learning rate, yaitu laju pembelajaran.
- net.trainParam.lr\_inc = IncLearningRate  
Menentukan rasio yang berguna sebagai faktor pengali untuk kenaikan learning rate apabila learning rate yang ada terlalu rendah atau mencapai kekonvergenan.
- net.trainParam.lr\_dec = DecLearningRate  
Menentukan rasio untuk menurunkan learning rate. Rasio ini berguna sebagai faktor pengali untuk menurunkan learning rate apabila learning rate yang ada terlalu tinggi dan menuju ke ketidakstabilan.

File citra yang disimpan untuk Implementasi Program adalah sebagai berikut:

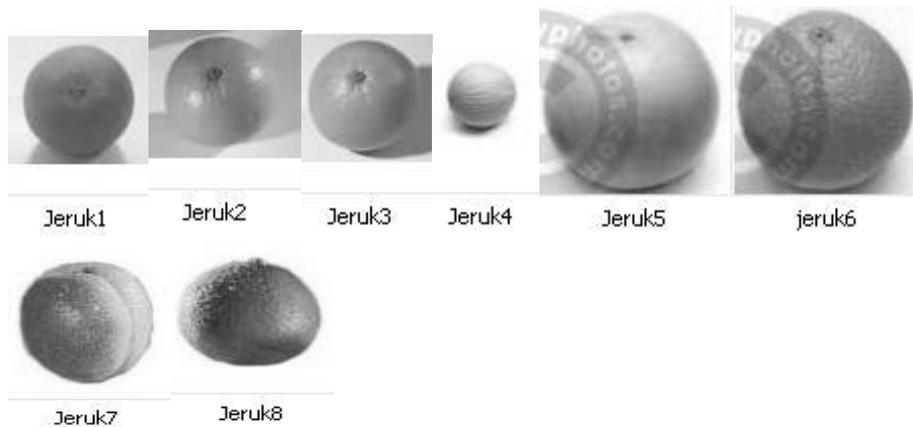
- Untuk citra apel dengan format JPEG dapat dilihat pada gambar 4.
- kemudian untuk proses implementasi program file-file citra apel tersebut diresolusi menjadi matriks pixel 10 x 10 dengan format TIF (gambar 5)
- Untuk File citra jeruk dengan format JPEG dapat dilihat pada gambar 6.
- kemudian untuk proses implementasi program file-file citra jeruk tersebut diresolusi menjadi matriks pixel 10 x 10 dengan format TIF dapat dilihat pada gambar 7.
- File citra yang digunakan untuk data pelatihan/pebelajaran dapat dilihat pada gambar 8:
- File citra yang digunakan untuk data uji dapat dilihat pada gambar 9.



Gambar 4 Citra apel format JPEG



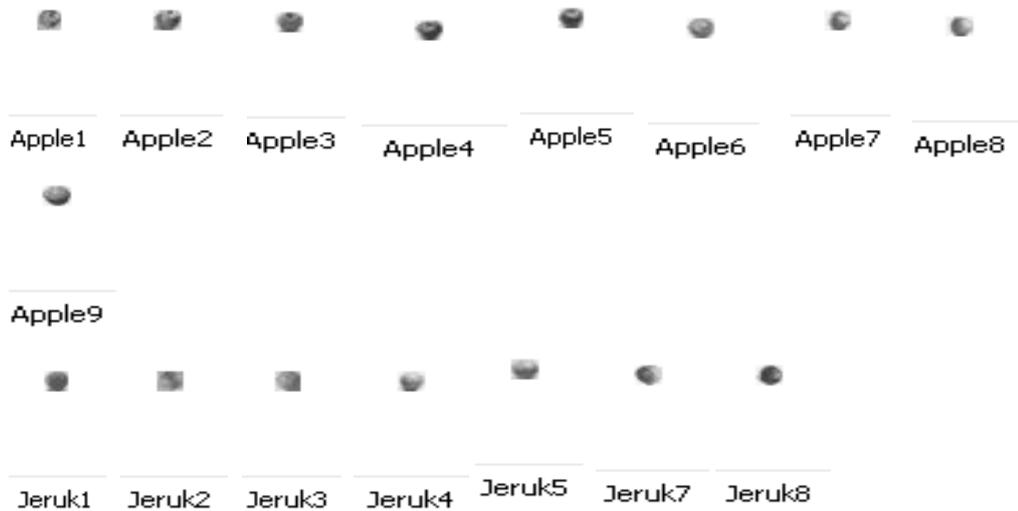
Gambar 5 Citra Apel dengan Format TIF



Gambar 6 Citra jeruk dengan format JPEG



Gambar 7 Citra jeruk dengan format TIF



Gambar 8 Citra apel dan jeruk untuk proses pelatihan



Gambar 9 Citra yang akan diuji

Hasil Running Program untuk langkah-langkah pada implementasi program di atas adalah sebagai berikut:

1. Kesimpulan Running program yang pertama:

=====  
 Error Pengujian Dengan Data Uji

EU = 0.8858 0.0630 0.0862

Kesimpulan

Tidak Dikenal

Apple

Apple

(lihat grafik pada gambar 10)

2. Kesimpulan untuk running program kedua:

=====  
 Error Pengujian Dengan Data Uji

EU = 0.1388 0.7378 0.0254

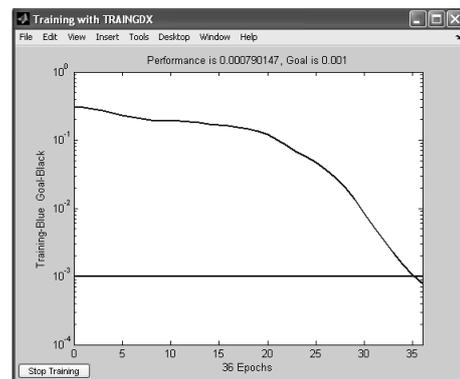
Kesimpulan

=====  
 Jeruk

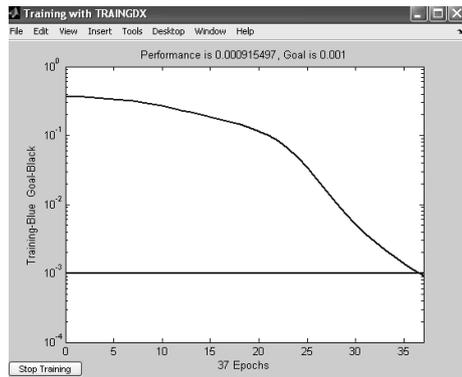
Tidak Dikenal

Apple

(lihat grafik pada gambar 11)



Gambar 10. Hasil running program pertama



Gambar 11. Hasil running program kedua

3. Kesimpulan untuk running program ketiga:

=====  
Error Pengujian Dengan Data Uji  
=====

EU = 0.4634 0.3147 0.1933  
=====

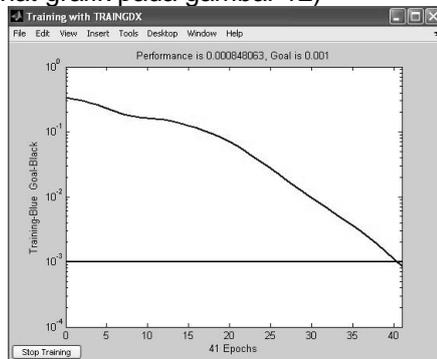
Kesimpulan  
=====

Jeruk

Apple

Apple

(lihat grafik pada gambar 12)



Gambar 12. Hasil running program ketiga

4. Kesimpulan untuk running program keempat:

=====  
Error Pengujian Dengan Data Uji  
=====

EU = 0.2498 0.0416 0.2091  
=====

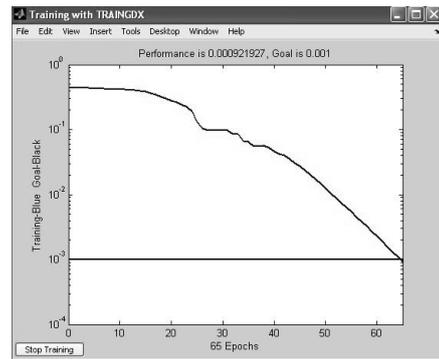
Kesimpulan  
=====

Jeruk

Apple

Apple

(lihat grafik pada gambar 13)



Gambar 13. Hasil running program keempat

5. Kesimpulan Untuk running program kelima adalah :

=====  
Error Pengujian Dengan Data Uji  
=====

EU = 0.1601 0.4764 0.0767  
=====

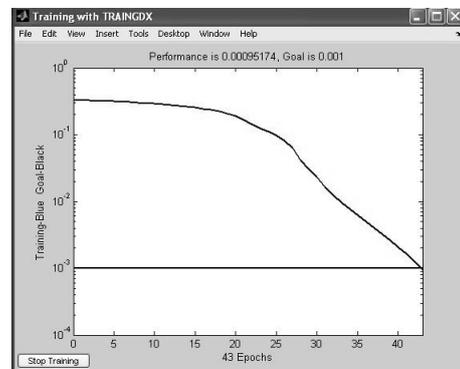
Kesimpulan  
=====

Jeruk

Apple

Apple

(lihat grafik pada gambar 14)



Gambar 14. Hasil running program kelima

6. Untuk running program keenam adalah:

=====  
Error Pengujian Dengan Data Uji  
=====

EU = 0.3089 0.0226 0.2665  
=====

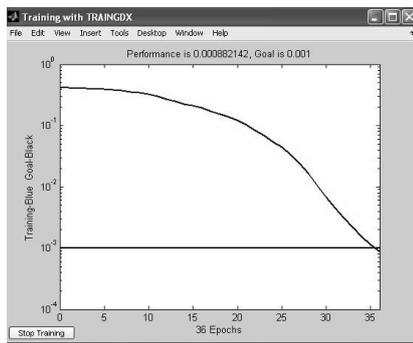
Kesimpulan  
=====

Jeruk

Apple

Apple

(lihat grafik pada gambar 15)



Gambar 15. Hasil running program keenam

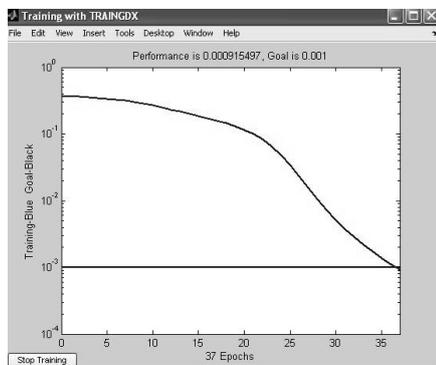
7. Kesimpulan untuk running program ketujuh:

=====  
Error Pengujian Dengan Data Uji  
=====

EU = 0.0192 0.1975 0.0194

=====  
Kesimpulan  
=====

Jeruk  
Apple  
Apple  
(lihat grafik pada gambar 16)



Gambar 16. Hasil running program ketujuh

**KESIMPULAN**

Pada data pengujian pertama, diinputkan data jeruk untuk input yang pertama, untuk jeruk diperoleh target untuk jeruk adalah 1, maka apabila diperoleh  $R1 = 1$  dan nilai error kurang dari 0.5 maka data input pertama yang berupa jeruk dapat dikenali sebagai jeruk, namun selain keadaan itu maka input tidak dapat dikenali (nilai error lebih dari 0.5).

Pada data pengujian kedua dan ketiga, diinputkan data apel, diperoleh

target adalah 0 maka apabila diperoleh  $R2$  dan  $R3 = 0$  dan nilai error kurang dari 0.5 maka data input kedua dan ketiga yang berupa apel dapat dikenali sebagai apel, selain keadaan itu maka input tidak dapat dikenali (nilai error lebih dari 0.5).

Dari delapan kali running program, dapat ditarik kesimpulan bahwa sistem pengenalan objek melalui pola warna dengan algoritma *backpropagation* disini, dapat mengenali input data uji yang dimasukkan, yaitu jeruk, apel, apel.

Masih terdapat error yang cukup besar, karena pelatihan data hanya sedikit, yaitu 14 citra warna dengan ukuran matriks pixel  $10 \times 10$ , serta pemilihan bobot dilakukan secara random, sehingga kadang-kadang error yang diperoleh cukup besar, sehingga perlu diberi batasan jika error melebihi 0.5 maka sistem ini tidak dapat mengenali data input yang diujikan.

Untuk mendapatkan ketelitian yang cukup tinggi, diperlukan data pelatihan yang lebih banyak. Perlu dilakukan penelitian lanjutan untuk citra yang terdiri dari satu jenis objek dan warna yang lebih kompleks serta citra warna yang masih terdapat noise dan belum dilakukan perbaikan (*enhancement*) citra.

**DAFTAR PUSTAKA**

- Fausett L, R, 1994, "Fundamental of neural Network, Arcitectures, Algorithms, and Applications", USA, Praticce-Hall, INC
- Kusumadewi, S., 2002, *Membangun Jaringan Syaraf Tiruan menggunakan MATLAB & EXCEL LINK*, Penerbit Graha Ilmu, Yogyakarta
- Munir, R., 2002, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Penerbit Informatika Bandung
- Schalkoff, R., 2000, *Pattern Recognition: Statistical, Structural and Neural approaches*, John Wiley and Sons Inc, New York
- Siong, A.W., Resmana, *Pengenalan Citra Objek Sederhana dengan Menggunakan Metode Jaringan Syaraf Tiruan SOM*, Prosiding Seminar Nasional Kecerdasan Komputasional, Universitas Indonesia, Juli 1999