

PENERAPAN JARINGAN SYARAF TIRUAN (JST) UNTUK MENDETEKSI POLA GAMBAR PADA PERMUKAAN YANG MEMILIKI CORAK TERTENTU

John Adler¹, Julerman N. Panjaitan¹, Asep Solih Awaluddin²

¹Teknik Komputer, Universitas Komputer Indonesia (UNIKOM)
Jl. Dipati Ukur 112-116 Bandung

²Matematika, Universitas Islam Negeri Sunan Gunung Jati (UIN-SGD)
Jl. AH. Nasution 105 Cibiru Bandung - 40614
email : john.adler007@gmail.com

ABSTRAK

Dalam hal pengenalan terhadap suatu pola, informasi dan rangsangan dari luar dapat diinterpretasikan secara berbeda oleh setiap manusia. Hal ini bergantung pada proses berpikir dan kemampuan analisa otak manusia. Sistem komputer dapat dijadikan alat pengenalan pola dengan meniru cara kerja otak manusia. Teknologi ini dikenal sebagai Jaringan Syaraf Tiruan (JST).

Pola yang ada dalam penelitian ini adalah pola kotak, lingkaran dan segitiga. Pola tersebut merupakan hasil foto dari kamera cell phone dengan ukuran 160x120 piksel pada permukaan bercorak yang telah ditentukan. Dengan menggunakan software Matlab, proses pelatihan dilakukan terhadap setiap jenis pola dengan menentukan ciri khususnya. Ekstrak ciri dan deteksi tepi dapat memberikan ciri khusus dalam bentuk bilangan biner pada setiap jenis pola dan dituangkan dalam suatu matriks yang merupakan input bagi JST, kemudian pengolahan input dan pengenalan pola dilakukan dengan menggunakan metode backpropagasi, nilai dan parameter optimum JST.

Dengan nilai dan parameter JST yang optimum, Jaringan Syaraf ini mampu mengenali pola kotak dan pola lingkaran masing-masing 100% serta pola segitiga sebesar 60%, jadi total prosentase pengenalan pola pada penelitian ini sebesar 86,6%.

Kata kunci : Jaringan Syaraf Tiruan, ekstraksi ciri, deteksi tepi, pengenalan pola, backpropagasi

1. Pendahuluan

Sistem komputer dan beberapa perangkat pendukungnya merupakan pilihan yang sering dipakai dalam membantu pekerjaan manusia, karena komputer memiliki kemampuan dalam melakukan berbagai macam proses seperti perhitungan, pengolahan data, pengolahan video dan audio, pengolahan citra dan sebagainya. Oleh manusia, pengenalan visual ini dapat dikenali dengan sangat mudah, karena manusia berfikir secara dinamis dan memiliki jaringan syaraf yang memiliki peran penting dalam kemampuan otak manusia untuk mengenali pola kompleks berdasarkan elemen input pancaindra seperti mata dan telinga.

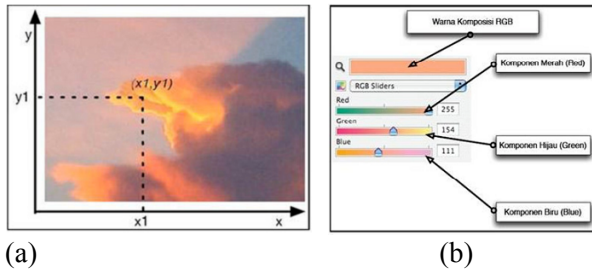
Sedangkan jaringan syaraf tiruan yang merupakan cabang ilmu yang masih baru, pada dasarnya mencoba untuk meniru cara kerja otak manusia. Salah satu yang ingin ditiru adalah bentuk dan kerja neuron (sel syaraf). Untuk dapat menyelesaikan suatu masalah dengan kompleksitas tertentu, komputer dapat meniru cara kerja otak manusia sebagai modul pemrosesan baru. Teknologi ini dikenal sebagai jaringan syaraf tiruan (*Artificial Neural Network*). Karakteristik umum dari jaringan

syaraf tiruan ini adalah kemampuan sistem untuk mengenali data bahkan yang belum pernah dipelajari sebelumnya dengan menggunakan kependaian yang didapat dari proses belajar. Pengenalan pola gambar yang beragam, baik dari segi bentuk, ukuran dan posisi sudah tentu akan membutuhkan proses pembelajaran yang lebih kompleks.

2. Tinjauan Pustaka Citra digital

Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog yang kontinu menjadi gambar diskrit melalui proses sampling. Gambar analog dibagi menjadi n baris dan m kolom sehingga menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut piksel. Citra digital dua dimensi dapat didefinisikan sebagai kumpulan koordinat x dan y, dimana x dan y adalah koordinat spasial yang memiliki intensitas citra pada setiap koordinat tersebut, hal tersebut diilustrasikan pada gambar 1a. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red,*

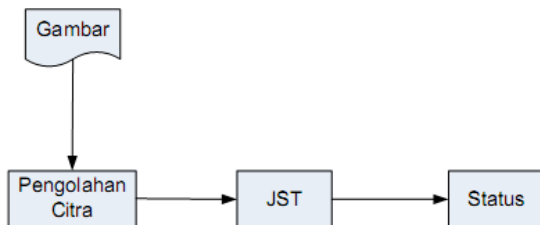
Green, Blue-*RGB*) [1]. Komposisi warna RGB tersebut dapat dijelaskan pada gambar 1b dibawah ini.



Gambar (a) Citra Digital
(b) Komposisi Warna RGB

Konsep Pengenalan dan Pengolahan Citra

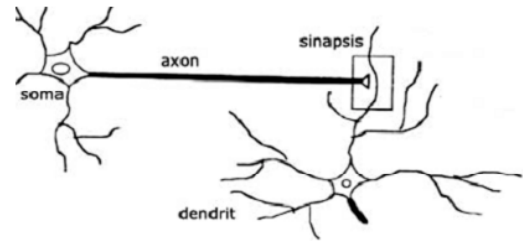
Ketika komputer memperoleh data citra masukan, komputer akan menganalisa data citra tersebut, dan mengidentifikasi citra tersebut dengan suatu pembelajaran. Dengan data-data input yang sudah ada, maka proses pembelajaran dilakukan terhadap kemungkinan-kemungkinan yang ada. Hasil atau status dari pembelajaran jaringan syaraf tiruan ini disebut pengidentifikasian citra.



Gambar 2. Data Flow Diagram Proses Pengenalan Gambar

Pengenalan Jaringan Syaraf Biologis

Otak manusia terdiri dari kurang lebih 10 miliar sel syaraf atau neuron, yang rata-rata setiap neuron saling berhubungan dengan neuron lainya melalui lebih dari 10.000 sinapsis. Otak manusia disusun oleh lebih dari 100 jenis sel neuron yang berbeda. Jumlah neuron pada otak berkisar antara 50 miliar sampai lebih dari 100 miliar. Neuron-neuron dibagi-bagi menjadi kelompok-kelompok yang berhubungan, disebut jaringan atau *network*. Jadi otak dapat dilihat sebagai kumpulan dari jaringan syaraf [2].



Gambar 3. Sel Syaraf Manusia

Sel neuron terdiri dari empat bagian, yaitu :

- Badan sel (soma), berwarna kelabu dan mengandung inti sel (nukleus dan plasma sel) dimana dalam satu neuron, terdapat beberapa dendrit dan axon yang saling berhubungan.
- Dendrit, merupakan tonjolan-tonjolan yang tidak terlalu panjang, biasanya lebih dari satu, yang berfungsi untuk meneruskan impuls syaraf (informasi) dari tepi badan sel syaraf.
- Axon, pada umumnya panjang dan hanya ada satu buah yang berfungsi untuk meneruskan impuls syaraf dari badan sel ke badan sel lainnya.
- Sinapsis akan meneruskan dan memasangkan axon dengan dendrit pada sel yang lainnya.

Konsep Umum Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah model yang mengadaptasi jaringan syaraf biologis. Sistem pemrosesan informasi dengan jaringan syaraf tiruan memiliki persamaan karakteristik dan mekanisme dengan jaringan syaraf biologis (yang dimiliki oleh manusia), tetapi tidak sekompleks jaringan syaraf manusia. Tabel 1 dibawah ini menunjukkan perbandingan antara neuron biologis dan neuron buatan.

Tabel 1. Perbandingan Neuron Biologis Dan Neuron Buatan

Neuron Biologis	Neuron Buatan
▪ Badan sel (soma)	▪ Neuron, node, elemen pemroses
▪ Dendrit	▪ Masukan
▪ Axon	▪ Keluaran
▪ Sinapsis	▪ Bobot (weight)
▪ Jumlah neuron sangat banyak	▪ Jumlah neuron sedikit

Penerapan Jaringan Syaraf Tiruan (JST) Untuk Mendeteksi Pola Gambar Pada Permukaan Yang Memiliki Corak Tertentu

Komponen Jaringan Syaraf Tiruan Bobot

Bobot merupakan suatu nilai yang dapat menghubungkan beberapa neuron dengan neuron yang lainnya pada lapisan sebelum dan sesudahnya dengan suatu pembelajaran yang diaktifkan oleh fungsi aktivasi. Inisialisasi bobot awal sangat mempengaruhi cepat atau lambatnya proses pelatihan. Penyesuaian bobot koneksi yang terletak diantara dua neuron bergantung dari turunan fungsi pengaktifan dari neuron pada satu lapisan dan juga pada neuron pada lapisan sebelumnya. Jika nilai dari bobot awal terlalu besar, maka nilai masukan awal ke neuron tersembunyi atau neuron keluaran akan mencapai daerah jenuh (*saturation region*), yaitu tempat turunan fungsi yang bernilai sangat kecil. Sebaliknya, jika nilai masukan awal ke neuron tersembunyi atau neuron keluaran terlalu kecil atau mendekati nol, maka akan berakibat proses pembelajaran akan lambat.

Layer (Lapisan)

Lapisan merupakan suatu wadah bagi neuron-neuron dalam tahapan proses jaringan syaraf. Terdiri atas tiga lapisan secara berurutan, yaitu lapisan input, lapisan tersembunyi dan lapisan output. Dalam setiap lapisan ini terdapat beberapa neuron yang bekerja dan bobot-bobot yang menghubungkan antara neuron satu dengan yang lainnya pada lapisan atau layer yang berbeda-beda.

Fungsi Aktivasi

Fungsi aktivasi berguna untuk memetakan suatu domain data masukan pada neuron yang tidak terbatas ke dalam satu jangkauan (*range*) yang sudah ditentukan terlebih dahulu. Informasi (input) akan dikirimkan ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang akan datang. Hasil penjumlahan ini kemudian dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuronnya [3,4].

Pengenalan Pola

Pengenalan pola dapat didefinisikan sebagai suatu usaha mencocokkan objek terhadap beberapa kelompok yang telah didefinisikan sebelumnya. Pola merupakan deskripsi dari objek, yaitu ciri khas dari suatu objek yang membedakannya dari objek lain. Kita mengenal objek disekitar kita, bergerak dan beraksi

sehubungan dengan objek-objek tersebut. Tahapan dan tujuan proses pengenalan pola dapat dibedakan menjadi dua, yaitu:

1. Memasukkan pola kedalam suatu pola yang belum dikenal. Prosesnya disebut *clustering* atau klasifikasi tidak terawasi.
2. Mengidentifikasi pola sebagai anggota dari kelas yang sudah dikenal sebelumnya. Prosesnya disebut klasifikasi terawasi.

Ekstraksi Ciri

Ekstraksi ciri adalah proses untuk mengubah tiap gambar yang terdapat pada file gambar menjadi susunan kode biner (angka 0 dan 1), yang selanjutnya akan dipakai sebagai input pada jaringan syaraf tiruan untuk mengambil keputusan. Ciri-ciri pada suatu objek dapat diambil berdasarkan ciri bentuk *bounding rectangle*, *bounding circle*, dan *best-ellips*. Suatu ekstraksi ciri dapat meminimumkan dimensi data dengan mempertahankan informasi penting yang terkandung didalam sebuah objek. Tujuan pemilihan ciri khas (*feature selection*) citra adalah untuk:

1. Mencari ciri yang paling optimum dari suatu objek yang dapat digunakan untuk membedakan objek tersebut dengan objek lainnya.
2. Menentukan prosedur langkah pengambilan ciri yang akan digunakan serta menentukan prosedur klasifikasinya.

Deteksi Tepi

Deteksi tepi merupakan satu langkah awal yang penting dalam teknik ekstraksi ciri dan pengenalan pola pada citra digital. Tujuannya adalah untuk mengidentifikasi suatu daerah pada citra dimana terdapat perubahan yang besar terhadap intensitas. Dengan deteksi tepi suatu sistem pemrosesan citra akan menemukan tanda-tanda atau garis bentuk yang timbul dari suatu objek. Karena pengenalan objek lebih memberi perhatian pada objeknya bukan pada pencahayaannya, maka deteksi tepi merupakan satu langkah yang rasional dalam mengenal objek. Sejumlah teknik pengolahan gambar memerlukan prioritas, baik peningkatan (*enhancement*), maupun deteksi tepi (*edge detection*). Suatu warna *gray scale* dapat digambarkan secara sederhana sebagai perubahan kasar di dalam intensitas *gray level*.

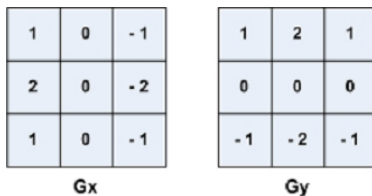
Metoda Sobel

Metoda ini akan meningkatkan dan mendeteksi tepi gambar. Operator sobel

melakukan perhitungan turunan spasial 2-D pada suatu citra dan juga menekankan daerah frekuensi tinggi pada domain spasial yang berkaitan dengan tepi. Dalam perhitungannya operator sobel memiliki kernel 3x3 pada setiap piksel, seperti yang ditunjukkan pada persamaan berikut

$$|G| = \sqrt{G_x^2 + G_y^2}$$

(1)



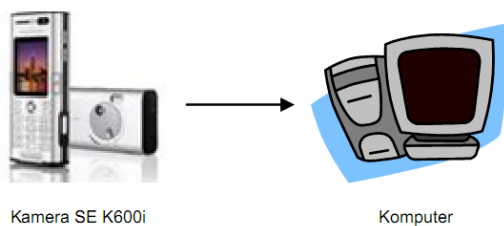
Gambar 4. Operator Sobel

3. Metode Penelitian

Deskripsi Komponen Yang Digunakan

Dalam penerapan jaringan syaraf tiruan ini diperlukan perangkat atau peralatan yang mendukung dalam desain dan pemrogramannya. Adapun perangkat yang digunakan adalah:

1. Perangkat Keras berupa PC (*Personal Computer*) menggunakan komputer dengan *motherboard* ABIT KR7A RAID, *processor* AMD Athlon 1.7 GHz, *harddisk* Seaget Barracuda 40 Gb, *Virtual Graphics Accelreation* (VGA) Gforce 4 4000 128 MHz, *memory* Visipro 256 MHz, monitor DELL 15", *Network Interface Card* D-Link DFE 853TX 10/100, dan sebuah *flash disk* AUM 512 Mb. Juga menggunakan sebuah *handphone* Sony Erickson K 600i yang memiliki fitur kamera 1,3 mega piksel dengan resolusi 160x120 piksel untuk *capture* gambar yang akan dideteksi. Untuk objek pola gambar, penulis menggunakan sebuah *styrofoam* dan kertas kado sebagai corak permukaan.

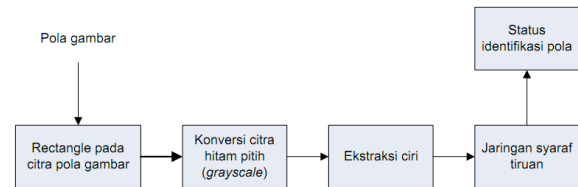


Gambar 5. Proses Pengambilan Dan Penyimpanan Gambar Pola

2. Perangkat Lunak untuk mendeteksi pola gambar pada permukaan yang memiliki corak tertentu menggunakan software MATLAB 6.0 Release 12 dan sebuah aplikasi pengolah gambar seperti ACD System 8.

Proses Pengolahan Citra Pola Gambar

Metoda *backpropagation* untuk mengetahui pola yang terdapat pada sebuah citra. Citra yang dideteksi disimpan dalam format gambar *.jpg. Skema uraian diperlihatkan seperti pada gambar dibawah ini.



Gambar 6. Proses Pengolahan Citra

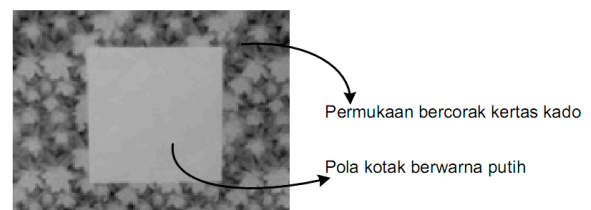
Konversi Citra Hitam Putih (*Grayscale*)

Konversi citra BW dilakukan dengan mengubah dan menyederhanakan warna dari sebuah citra. Citra akan memiliki warna hitam dan putih. Citra hitam putih (BW) direpresentasikan dengan nilai biner 0 dan 1, diperoleh dengan *threshold* ($x > \text{thres} = 1$ dan $x < \text{thres} = 0$), dengan batas *threshold* 0,5.

Pengkonversian dengan BW juga dilakukan untuk menyederhanakan perhitungan gambar konversi. Hasilnya akan didapat kumpulan piksel-piksel dengan nilai intensitas 0 (warna hitam) dan 1 (warna putih). Setelah obyek-obyek yang ada berhasil disederhanakan maka akan dilakukan perhitungan nilai rata-rata yang menghasilkan sebuah nilai yang akan dijadikan sebagai input bagi jaringan syaraf tiruan.

Ekstraksi Ciri

Tahap berikutnya adalah mengekstraksi ciri-ciri tertentu dari setiap obyek. Ciri dari sebuah citra adalah piksel-piksel yang memiliki derajat keabuan 1 dan 0. Proses ekstraksi dapat diperlihatkan pada gambar dibawah ini.



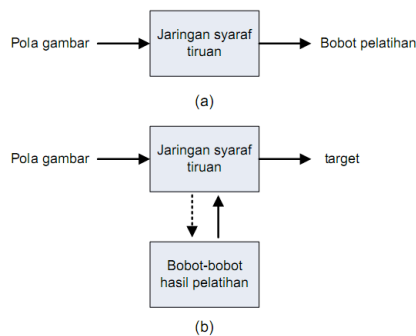
Gambar 7. Pola Kotak Hasil Ekstraksi Ciri

Penerapan Jaringan Syaraf Tiruan (JST) Untuk Mendeteksi Pola Gambar Pada Permukaan Yang Memiliki Corak Tertentu

Perancangan Jaringan Syaraf Tiruan

Arsitektur jaringan yang sering digunakan oleh *backpropagation* adalah jaringan *feedforward* dengan banyak lapisan. Lapisan masukan dilambangkan dengan x , lapisan tersembunyi dilambangkan dengan z , dan, lapisan keluaran dilambangkan dengan y . Pada jaringan *feedforward*, pelatihan untuk mengatur bobot, sehingga pada akhir pelatihan akan diperoleh bobot-bobot yang baik. Selama proses pelatihan, bobot-bobot diatur secara iteratif untuk meminimumkan fungsi kinerja jaringan.

Jaringan syaraf tiruan dilatih agar memiliki pengetahuan yang cukup dan perlu diberikan data-data yang akan dilatih berupa ciri citra dari pola yang akan dikenali. Blok diagram proses pelatihan dan pengenalan seperti dilihat pada gambar berikut ini.



Gambar 8. Proses Pembelajaran Jaringan

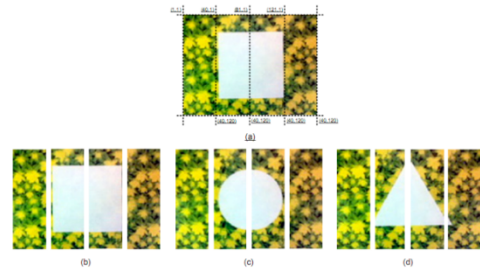
Gambar (a) memperlihatkan blok sederhana proses pembelajaran jaringan. Selanjutnya, bobot-bobot ini diperlihatkan untuk mengenali pola pada gambar (b).

Perancangan Input

Dimulai dengan menentukan potongan-potongan atau *rectangle* citra. Sebelum jaringan syaraf tiruan dilatih, terlebih dahulu representasikan input dengan cara membagi setiap bagian dalam citra tersebut menjadi empat bagian dengan koordinat-koordinat seperti diuraikan dibawah ini:

1. Potongan pertama terletak pada $rec1 = [1 \ 1 \ 40 \ 120]$
2. Potongan pertama terletak pada $rec2 = [41 \ 1 \ 40 \ 120]$
3. Potongan pertama terletak pada $rec3 = [81 \ 1 \ 40 \ 120]$
4. Potongan pertama terletak pada $rec4 = [121 \ 1 \ 40 \ 120]$

Hasil potongan pada citra ini diperlihatkan pada gambar 9 di bawah, dimana setiap potongan memiliki ukuran yang relatif sama antara satu dengan yang lainnya.



Gambar 9. a. Proses crop pola gambar
b. Hasil crop pola kotak c. Hasil crop pola lingkaran d. Hasil crop pola segitiga

Setelah piksel-piksel pada *rectangle* diperoleh, jumlah pikselnya akan dijumlahkan seluruhnya sehingga didapat jumlah total piksel. Piksel-piksel ini selanjutnya akan disimpan dalam *array* satu dimensi dengan jumlah baris adalah jumlah total baris dan kolom *rectangle* tadi. Selanjutnya nilai setiap elemen array tersebut diinisialisasikan dengan nol (*matriks zero*).

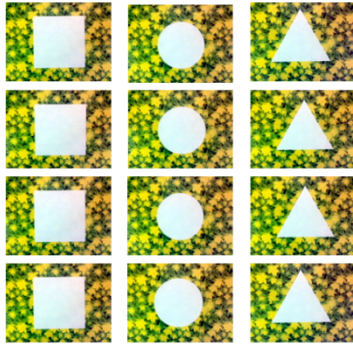
Inisialisasi Bobot

Bobot awal ditentukan bersamaan dengan penentuan parameter jaringan syaraf tiruan. Proses pelatihan dilakukan pada masing-masing jenis pola gambar. Agar mendapatkan bobot-bobot akhir yang memiliki variasi yang berbeda-beda antara setiap jenis pola gambar, dilakukan pembobotan awal yang bervariasi pula. Tujuan akhirnya adalah untuk mendapatkan bobot-bobot yang unik agar tidak terjadi kesalahan pengklasifikasian atau pengenalan pola gambar.

Setiap pelatihan jenis pola gambar yang menghubungkan neuron-neuron pada lapisan input dan lapisan tersembunyi hanya ada satu pembobot, demikian pula bobot awal yang menghubungkan neuron-neuron lapisan tersembunyi ke lapisan output. Bobot akhir diperoleh dari hasil penjumlahan antara bobot awal, *feedforward* dan *backpropagation* serta menggunakan fungsi aktivasi untuk menghitung outputnya. Ini adalah satu iterasi pembelajaran.

Pola-pola gambar yang dilatih

Pola gambar yang akan dikenali diperoleh dari tiga jenis pola, yaitu pola kotak lingkaran dan segitiga. Setiap jenis pola akan dilatih sebanyak empat kali. Masing-masing pola yang dilatih dapat dilihat pada gambar 10 di bawah ini.



Gambar 10. Pola-Pola Gambar Yang Dilatih Pelatihan Jaringan

Proses pelatihan dimulai dengan menentukan bobot awal jaringan untuk neuron-neuron input ke lapisan tersembunyi. Selanjutnya neuron-neuron lapisan tersembunyi ke lapisan keluaran. Agar memiliki pengetahuan yang cukup, JST perlu diberikan data-data yang akan dilatih berupa ciri-ciri pola gambar dari setiap gambar pola. Pelatihan jaringan syaraf tiruan diuraikan pada diagram alir data dibawah ini.



Gambar 11. Diagram Alir Pelatihan

Pada awalnya dibaca pola gambar yang dijadikan pola pelatihan. Selanjutnya citra dibagi menjadi empat bagian (rec1,rec2,rec3,rec4). Kemudian dilakukan deteksi tepi dengan menggunakan metoda sobel pada masing-masing potongan. Hasilnya adalah tepian pola dan corak permukaan yang berbentuk citra biner konversi hitam putih (BW).

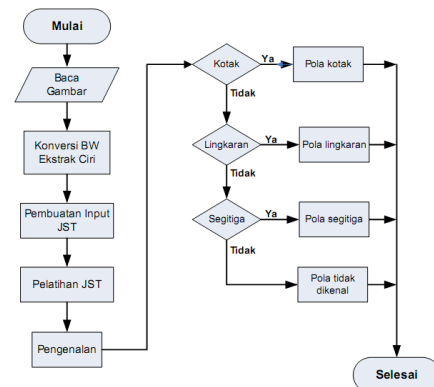
Proses selanjutnya adalah membentuk input untuk jaringan syaraf tiruan pelatihan. Cara yang dilakukan adalah memindahkan nilai piksel hasil deteksi tepi ke array kolom. Selanjutnya adalah operasi *feedforward* (antara lapisan input dan lapisan tersembunyi) dilakukan dengan mengalikan input dengan bobot-bobot interkoneksi antara kedua lapisan. Hasil proses ini diaktivasi dengan menggunakan fungsi aktivasi sigmoid biner yang ditentukan.

Diantara lapisan tersembunyi dan output, dilakukan pengalihan antara keluaran *feedforward* sebelumnya dengan bobot-bobot interkoneksi lapisan tersembunyi dan output.

Backpropagation dilakukan dengan menghitung informasi error antara lapisan tersembunyi dan lapisan output. *Backpropagation* juga dilakukan diantara lapisan tersembunyi dan lapisan input. Proses *backpropagation* dilakukan untuk memperbaiki bobot-bobot interkoneksi. Kedua proses *feedforward* dan *backpropagation* dilakukan berulang sampai kondisi error target (goal) tercapai atau telah sampai pada iterasi atau *epoch* maksimum. Jika salah satu dari kedua kondisi ini terjadi, maka pelatihan dihentikan dan didapatlah bobot-bobot optimum.

Proses Pengenalan Pola

Proses dimulai dengan membaca gambar dan mengklasifikasikan ciri dari setiap bagian pola yang ditentukan dan oleh jaringan syaraf tiruan diidentifikasi sebagai suatu pola. Blok diagram proses pengenalan pola gambar dapat dilihat pada gambar 12 berikut ini



Gambar 12. Flowchart Pengenalan Gambar

4. Hasil dan Pembahasan

Setelah melakukan beberapa percobaan dan pengujian, maka diperoleh nilai-nilai optimum dari semua parameter-parameter yang diperlukan. Tahap berikutnya akan dilakukan pengujian terhadap seluruh pola yang ada dengan menggunakan semua parameter optimum yang telah didapat pada pengujian-pengujian sebelumnya. Proses pengujian ini dilakukan oleh satu program uji (uji.m) secara bergantian dimulai dari pola kotak, lingkaran dan segitiga. Pengujian dilakukan pada pola gambar yang tidak dilatih. Jumlah pola yang dilatih sebanyak 12 buah pola (4 buah pola kotak, 4 buah lingkaran, dan 4 buah pola segitiga) dan

Penerapan Jaringan Syaraf Tiruan (JST) Untuk Mendeteksi Pola Gambar Pada Permukaan Yang Memiliki Corak Tertentu

jumlah pola yang akan diuji sebanyak 24 buah pola (8 buah pola kotak, 8 buah pola lingkaran, dan 8 buah pola segitiga). Jadi total pola yang diikuti sertakan dalam makalah ini sebanyak 36 buah pola (12 buah pola kotak, 12 buah pola lingkaran, dan 12 buah pola segitiga).

Untuk mengetahui tingkat keberhasilan pengenalan setiap pola, proses uji dilakukan secara bergantian satu persatu. Untuk mendapatkan prosentase keberhasilan yang lebih akurat dengan data yang lebih detail, pengujian terhadap satu buah pola dilakukan lebih dari satu kali pengujian, yaitu sebanyak 5 kali untuk setiap pola yang diuji. Dari 8 buah masing-masing pola diambil 4 buah pola yang akan diambil dan diuji satu persatu dan ditampilkan secara berkelompok dengan tidak memperhatikan urutannya. Sehingga banyaknya kelompok dapat dikombinasikan dengan rumus :

$$C = \binom{8}{4} = \frac{8!}{4!(8-4)!}$$

$$C = 70 \text{ kali}$$

(2)

sehingga total pengujian dilakukan sebanyak 70 kali untuk masing-masing pola. Namun karena pengujian dengan jumlah tersebut dianggap terlalu banyak, maka cukup diambil contoh kelompok pengujian sebanyak 10 kali pengujian berkelompok. Setiap pola diuji diatas sebuah permukaan yang memiliki corak tertentu. Hasil pengujian setiap pola diperlihatkan pada tabel 2, tabel 3 dan tabel 4.

Dari hasil pengujian diatas, tingkat keberhasilan jaringan syaraf tiruan ini dalam mengidentifikasi pola berbeda-beda pada masing-masing pola. Pada pengujian pola kotak dan pola lingkaran diperoleh prosentase keberhasilan mencapai 100 %. Ini menunjukkan bahwa setiap pola kotak dan pola lingkaran yang ada pada sistem ini dapat dikenali dengan baik sebagai pola kotak dan pola lingkaran. Ini juga menunjukkan bahwa proses pelatihan dengan parameter-parameter yang ada telah berhasil. Dengan demikian tidak ada pola kotak yang "tidak dikenali" atau dikenal sebagai pola lainnya ("pola lingkaran" atau "pola segitiga" untuk mengenali pola kotak serta "pola kotak" atau "pola segitiga" untuk mengenali pola lingkaran).

Tabel 2. Hasil pengujian pola kotak

Nama file	Hasil pengujian	% kerberhasilan	Nama file	Hasil pengujian	% kerberhasilan	Rata-rata % kerberhasilan
kotak5.jpg	pola KOTAK		kotak9.jpg	pola KOTAK		100.00 %
kotak6.jpg	pola KOTAK	100%	kotak10.jpg	pola KOTAK	100%	
kotak7.jpg	pola KOTAK		kotak11.jpg	pola KOTAK		
kotak8.jpg	pola KOTAK		kotak12.jpg	pola KOTAK		
kotak5.jpg	pola KOTAK		kotak6.jpg	pola KOTAK		
kotak7.jpg	pola KOTAK	100%	kotak8.jpg	pola KOTAK	100%	
kotak9.jpg	pola KOTAK		kotak10.jpg	pola KOTAK		
kotak11.jpg	pola KOTAK		kotak12.jpg	pola KOTAK		
kotak6.jpg	pola KOTAK		kotak5.jpg	pola KOTAK		
kotak7.jpg	pola KOTAK	100%	kotak8.jpg	pola KOTAK	100%	
kotak10.jpg	pola KOTAK		kotak9.jpg	pola KOTAK		
kotak11.jpg	pola KOTAK		kotak12.jpg	pola KOTAK		
kotak5.jpg	pola KOTAK		kotak7.jpg	pola KOTAK		
kotak6.jpg	pola KOTAK	100%	kotak8.jpg	pola KOTAK	100%	
kotak11.jpg	pola KOTAK		kotak9.jpg	pola KOTAK		
kotak12.jpg	pola KOTAK		kotak10.jpg	pola KOTAK		
kotak5.jpg	pola KOTAK		kotak6.jpg	pola KOTAK		
kotak8.jpg	pola KOTAK	100%	kotak7.jpg	pola KOTAK		
kotak10.jpg	pola KOTAK		kotak8.jpg	pola KOTAK	100%	
kotak11.jpg	pola KOTAK		kotak9.jpg	pola KOTAK		
kotak12.jpg	pola KOTAK		kotak10.jpg	pola KOTAK		
kotak5.jpg	pola KOTAK		kotak6.jpg	pola KOTAK		
kotak8.jpg	pola KOTAK	100%	kotak7.jpg	pola KOTAK		
kotak10.jpg	pola KOTAK		kotak8.jpg	pola KOTAK	100%	
kotak11.jpg	pola KOTAK		kotak9.jpg	pola KOTAK		
kotak12.jpg	pola KOTAK		kotak10.jpg	pola KOTAK		

Tabel 3. Pengenalan pola lingkaran

Nama file	Hasil pengujian	% kerberhasilan	Nama file	Hasil pengujian	% kerberhasilan	Rata-rata % kerberhasilan
ling5.jpg	pola LINGKARAN		ling9.jpg	pola LINGKARAN		100.00 %
ling6.jpg	pola LINGKARAN	100%	ling10.jpg	pola LINGKARAN	100%	
ling7.jpg	pola LINGKARAN		ling11.jpg	pola LINGKARAN		
ling8.jpg	pola LINGKARAN		ling12.jpg	pola LINGKARAN		
ling5.jpg	pola LINGKARAN		ling6.jpg	pola LINGKARAN		
ling7.jpg	pola LINGKARAN	100%	ling8.jpg	pola LINGKARAN	100%	
ling9.jpg	pola LINGKARAN		ling10.jpg	pola LINGKARAN		
ling11.jpg	pola LINGKARAN		ling12.jpg	pola LINGKARAN		
ling6.jpg	pola LINGKARAN		ling5.jpg	pola LINGKARAN		
ling7.jpg	pola LINGKARAN	100%	ling8.jpg	pola LINGKARAN	100%	
ling10.jpg	pola LINGKARAN		ling9.jpg	pola LINGKARAN		
ling11.jpg	pola LINGKARAN		ling12.jpg	pola LINGKARAN		
ling5.jpg	pola LINGKARAN		ling7.jpg	pola LINGKARAN		
ling6.jpg	pola LINGKARAN	100%	ling8.jpg	pola LINGKARAN	100%	
ling11.jpg	pola LINGKARAN		ling9.jpg	pola LINGKARAN		
ling12.jpg	pola LINGKARAN		ling10.jpg	pola LINGKARAN		
ling5.jpg	pola LINGKARAN		ling8.jpg	pola LINGKARAN		
ling8.jpg	pola LINGKARAN	100%	ling7.jpg	pola LINGKARAN	100%	
ling10.jpg	pola LINGKARAN		ling9.jpg	pola LINGKARAN		
ling11.jpg	pola LINGKARAN		ling12.jpg	pola LINGKARAN		

Tabel 4. Pengenalan pola segitiga

nama file	hasil uji	% kerberhasilan	nama file	hasil uji	% kerberhasilan	Rata-rata % kerberhasilan
segi5.jpg	pola SEGITIGA		segi9.jpg	pola KOTAK		60.00 %
segi6.jpg	pola SEGITIGA	100%	segi10.jpg	pola SEGITIGA	25%	
segi7.jpg	pola SEGITIGA		segi11.jpg	pola KOTAK		
segi8.jpg	pola SEGITIGA		segi12.jpg	pola KOTAK		
segi5.jpg	pola SEGITIGA		segi6.jpg	pola SEGITIGA		
segi7.jpg	pola SEGITIGA	50%	segi8.jpg	pola SEGITIGA	75%	
segi9.jpg	pola KOTAK		segi10.jpg	pola SEGITIGA		
segi11.jpg	pola KOTAK		segi12.jpg	pola KOTAK		
segi6.jpg	pola SEGITIGA		segi5.jpg	pola SEGITIGA		
segi7.jpg	pola SEGITIGA	75%	segi8.jpg	pola SEGITIGA	50%	
segi10.jpg	pola SEGITIGA		segi9.jpg	pola KOTAK		
segi11.jpg	pola KOTAK		segi12.jpg	pola KOTAK		
segi5.jpg	pola SEGITIGA		kotak7.jpg	pola SEGITIGA		
segi6.jpg	pola SEGITIGA	50%	kotak8.jpg	pola SEGITIGA		
segi11.jpg	pola KOTAK		kotak9.jpg	pola KOTAK	75%	
segi12.jpg	pola KOTAK		kotak10.jpg	pola SEGITIGA		
segi5.jpg	pola SEGITIGA		Kotak8.jpg	pola SEGITIGA		
segi6.jpg	pola SEGITIGA	75%	kotak7.jpg	pola KOTAK	25%	
segi10.jpg	pola SEGITIGA		kotak9.jpg	pola KOTAK		
segi11.jpg	pola KOTAK		kotak12.jpg	pola KOTAK		

Sedangkan pada pengujian pola segitiga diperoleh prosentase keberhasilan mencapai 60%. Ini menunjukkan bahwa tidak setiap pola segitiga yang ada pada sistem ini dapat dikenali dengan baik sebagai pola segitiga. Dapat diasumsikan bahwa proses pelatihan dengan parameter-parameter yang ada tidak berjalan

dengan baik. Pengaruh permukaan yang bercorak menyebabkan pola segitiga kurang memiliki ciri yang kuat. Karena hasil deteksi tepi yang dilakukan tidak memberi bentuk pola segitiga dengan garis sisi yang tegas.

Hal inilah yang menyebabkan pola segitiga dianggap sama atau serupa dengan pola lainnya. Hal lain yang bisa dianalisa adalah adanya kecenderungan parameter-parameter optimum pada pola kotak dan lingkaran tidak meningkatkan prosentase keberhasilan pengenalan pola segitiga. Hal ini dapat diasumsikan karena bentuk pola kotak dan lingkaran merupakan bentuk bangun simetris baik vertikal maupun horizontal yang memiliki perbedaan dengan pola segitiga, karena bentuk bangun sebuah segitiga pada penelitian ini hanya simetris secara vertikal. Hal ini merupakan perbedaan dasar pola segi tiga dengan pola kotak dan lingkaran, sehingga parameter-parameter optimum dalam proses pengenalan pola antara kotak, lingkaran dan segitiga tidak harus sama.

5. Kesimpulan

Dari perancangan dan pengujian analisis pada penelitian ini, maka ada beberapa hal yang dapat disimpulkan, antara lain :

1. Arsitektur jaringan syaraf tiruan yang optimum pada penelitian ini terdiri dari 19560 buah neuron pada lapisan input, 20 buah neuron pada lapisan tersembunyi dan 3 neuron pada lapisan output, konstanta pembelajaran yang optimum adalah 0.1 dan nilai momentum jaringan syaraf tiruan yang optimum adalah 0.3 karena iterasi yang dihasilkan lebih kecil dan keakuratan pengenalannya lebih baik.
2. Dilihat dari data hasil pengujian bahwa pola gambar yang dilatih sudah cukup optimal untuk proses pelatihan dan pengenalan pola kotak dan lingkaran (pola berhasil dikenal 100%), namun untuk pola segitiga pola gambar masih belum memberikan hasil yang optimum (pola berhasil dikenal 60%).
3. Total rata-rata prosentase keberhasilan dari pengenalan pola kotak, lingkaran dan segitiga dalam penelitian ini sebesar 86,6%.
4. Program ini hanya dapat mengidentifikasi pola kotak, lingkaran dan segitiga pada permukaan yang memiliki corak yang sama dengan yang digunakan pada penelitian ini. Dengan kata lain corak lain tidak dapat dipakai pada program ini.

6. Referensi

- [1] Tim Wikipedia, diakses pada tanggal 9 Mei 2007 dari http://id.wikipedia.org/wiki/Pengolahan_Citra
- [2] Kristanto, Andri, “*Jaringan Syaraf Tiruan (Konsep dasar, Algoritma, dan Aplikasi)*”, Gava Media, Yogyakarta, 2004.
- [3] Kusumadewi, Sri, “*Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & Excell Link*”, Graha Ilmu, Yogyakarta, 2004.
- [4] Kusumadewi, Sri, “*Artificial Intelligence (Teknik dan Aplikasinya)*”, Graha Ilmu, Yogyakarta, 2002.
- [5] Departemen Teknik Elektro, Institut Teknologi Bandung, *Image Processing Research Group Module 4*, 2003.
- [6] Green, B., “*Edge Detection Tutorial*”, 2002, diakses pada 10 Mei 2007 dari <http://www.pages.drexel.edu/%7eweg22/edgetutorial.html>
- [7] Stergiou, C. dan Siganos, D., *Neural Networks*, diakses pada tanggal 3 Maret 2007 dari www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/report.html
- [8] Krose, B. dan Smagt, P.V.d., “*An Introduction to Neural Network*”, Eight Edition, 1996.

Lampiran1. Listing program pelatihan pola kotak.

```
% PELATIHAN KOTAK
A='kotak1.jpg'; B='kotak2.jpg'; C='kotak3.jpg';
D='kotak4.jpg';

%BACA POLA GAMBAR
I=imread(A); J=imread(B); L=imread(C);
M=imread(D);

%PROSES KONVERSI DARI RGB KE GRAYSCALE
Ag = rgb2gray(I); Bg = rgb2gray(J); Cg =
rgb2gray(L); Dg = rgb2gray(M);

%RECTANGULAR PADA KORDINAT YANG DIINGINKAN
rec1=[1 1 40 120]; rec2=[41 1 40 120];
rec3=[81 1 40 120]; rec4=[121 1 40 120];

%tempatkan kordinat pada masing-masing Pola gambar
```


Penerapan Jaringan Syaraf Tiruan (JST) Untuk Mendeteksi Pola Gambar Pada Permukaan Yang Memiliki Corak Tertentu

```

I1=imcrop(I,rec1);      I2=imcrop(I,rec2);
I3=imcrop(I,rec3);      I4=imcrop(I,rec4);
J1=imcrop(J,rec1);      J2=imcrop(J,rec2);
J3=imcrop(J,rec3);      J4=imcrop(J,rec4);
L1=imcrop(L,rec1);      L2=imcrop(L,rec2);
L3=imcrop(L,rec3);      L4=imcrop(L,rec4);
M1=imcrop(M,rec1);      M2=imcrop(M,rec2);
M3=imcrop(M,rec3);      M4=imcrop(M,rec4);

%PROSES KONVERSI GAMBAR DARI RGB
KE GRAYSCALE
W1 = rgb2gray(I1); W2 = rgb2gray(I2);
W3 = rgb2gray(I3); W4 = rgb2gray(I4);
X1 = rgb2gray(J1); X2 = rgb2gray(J2);
X3 = rgb2gray(J3); X4 = rgb2gray(J4);
Y1 = rgb2gray(L1); Y2 = rgb2gray(L2);
Y3 = rgb2gray(L3); Y4 = rgb2gray(L4);
Z1 = rgb2gray(M1); Z2 = rgb2gray(M2);
Z3 = rgb2gray(M3); Z4 = rgb2gray(M4);

%PROSES DETEKSI TEPI DAN KONVERSI
BW
BW11=edge(W1,'sobel');
BW12=edge(W2,'sobel');
BW13=edge(W3,'sobel');
BW14=edge(W4,'sobel');
BW21=edge(X1,'sobel');
BW22=edge(X2,'sobel');
BW23=edge(X3,'sobel');
BW24=edge(X4,'sobel');
BW31=edge(Y1,'sobel');
BW32=edge(Y2,'sobel');
BW33=edge(Y3,'sobel');
BW34=edge(Y4,'sobel');
BW41=edge(Z1,'sobel');
BW42=edge(Z2,'sobel');
BW43=edge(Z3,'sobel');
BW44=edge(Z4,'sobel');

%PROSES JST %PEMBUATAN INPUT
[Baris11,Kolom11]=size(BW11);
[Baris12,Kolom12]=size(BW12);
[Baris13,Kolom13]=size(BW13);
[Baris14,Kolom14]=size(BW14);
[Baris21,Kolom21]=size(BW21);
[Baris22,Kolom22]=size(BW22);
[Baris23,Kolom23]=size(BW23);
[Baris24,Kolom24]=size(BW24);
[Baris31,Kolom31]=size(BW31);
[Baris32,Kolom32]=size(BW32);
[Baris33,Kolom33]=size(BW33);
[Baris34,Kolom34]=size(BW34);
[Baris41,Kolom41]=size(BW41);
[Baris42,Kolom42]=size(BW42);
[Baris43,Kolom43]=size(BW43);
[Baris44,Kolom44]=size(BW44);

Xk1=zeros(Baris11*Kolom11+Baris12*Kolom
12+Baris13*Kolom13+Baris14*Kolom14,1);

for k=1:Kolom11,
Xk1((k-
1)*Baris11+1:Baris11*k,1)=BW11(1:Baris11,k)
;
end
for k=1:Kolom12,
Xk1((k-
1)*Baris12+1+Baris11*Kolom11:Baris12*k+Ba
ris11*Kolom11,1)=BW12(1:Baris12,k);
end

for k=1:Kolom13,
Xk1((k-
1)*Baris13+1+Baris12*Kolom12+Baris11*Kol
om11:Baris13*k+Baris12*Kolom12+Baris11*K
olom11,1) =BW13(1:Baris13,k);
end

for k=1:Kolom14,
Xk1((k-
1)*Baris14+1+Baris13*Kolom13+Baris12*Kol
om12+Baris11*Kolom11:Baris14*k+Baris13*K
olom13+B
aris12*Kolom12+Baris11*Kolom11,1)=BW14(
1:Baris14,k);
end

Xk2=zeros(Baris21*Kolom21+Baris22*Kolom
22+Baris23*Kolom23+Baris24*Kolom24,1);

for k=1:Kolom21,
Xk2((k-
1)*Baris21+1:Baris21*k,1)=BW21(1:Baris21,k)
;
end

for k=1:Kolom22,
Xk2((k-
1)*Baris22+1+Baris21*Kolom21:Baris22*k+Ba
ris21*Kolom21,1)=BW22(1:Baris22,k);
end

for k=1:Kolom23,
Xk2((k-
1)*Baris23+1+Baris22*Kolom22+Baris21*Kol
om21:Baris23*k+Baris22*Kolom22+Baris21*K
olom21,1) =BW23(1:Baris23,k);
end

for k=1:Kolom24,

```

```
Xk2((k-  
1)*Baris24+1+Baris23*Kolom23+Baris22*Kol  
om22+Baris21*Kolom21:Baris24*k+Baris23*K  
olom23+B  
aris22*Kolom22+Baris21*Kolom21,1)=BW24(  
1:Baris24,k);  
end
```

```
Xk3=zeros(Baris31*Kolom31+Baris32*Kolom  
32+Baris33*Kolom33+Baris34*Kolom34,1);
```