

Aplikasi Jaringan Syaraf Tiruan untuk Mendeteksi *White Spot*

Dedy Harto, M. Sarosa, Wijono dan Suprpto

Abstrak—Deteksi *white spot* (bintik putih) mempunyai peran yang penting untuk mengetahui adanya penyakit *white spot* pada udang. Deteksi penyakit *white spot* selama ini dilakukan di laboratorium menggunakan alat PCR (*Polymer Chain Reaction*) dengan cepat tetapi memerlukan biaya yang mahal sedangkan menggunakan cara hispatologi memerlukan waktu yang lama.

Dalam penelitian ini dilakukan pengembangan perangkat lunak untuk mendeteksi *white spot* pada udang dengan menggunakan teknologi pengolahan citra digital dan jaringan syaraf tiruan (JST). Citra digital yang dianalisis adalah citra udang sehat dan udang sakit *white spot*. Proses pengolahan citra digital dimulai dari proses cropping, grayscale, histogram yang menghasilkan nilai-nilai piksel grayscale yang digunakan sebagai masukan pada JST.

Arsitektur terbaik pada saat dilakukan pelatihan adalah pada saat laju pembelajaran 0.1, jumlah neuron pada lapis tersembunyi1 sebanyak 20 buah dan lapis tersembunyi2 sebanyak 20 buah dengan jumlah iterasi sebesar 51267 kali dan target error 0.001 menghasilkan tingkat akurasi untuk data pelatihan sebesar 100% dan untuk data pengujian sebesar 93.33%.

Kata kunci : jaringan syaraf tiruan, pengolahan citra, penyakit udang, *white spot*

I. PENDAHULUAN

PENYAKIT *White Spot Syndrome Virus* (WSSV) mewabah di Indonesia sejak tahun 1995, dan menyerang udang pada semua stadia baik benur maupun udang dewasa. WSSV dapat menyebabkan kematian pada udang hingga 100% selama 3-10 hari sejak gejala klinis muncul. [2,4,6,7].

Untuk mendeteksi penyakit *white spot* dapat dilakukan berdasarkan pada tingkah laku atau kondisi fisik udang dan uji laboratorium. Pengujian melalui laboratorium menggunakan mikroskopis dan *Polymerase Chain Reaction* (PCR). Pengujian dengan menggunakan PCR hanya dapat dilakukan di Balai Besar Pengembangan Budidaya Air Payau (BBPBAP) saja karena harganya yang mahal.

Dedy Harto adalah Mahasiswa Program Magister dan Doktor Teknik Elektro Universitas Brawijaya, Malang, Indonesia (email: defariz@gmail.com).

M. Sarosa adalah dosen Teknik Elektro Politeknik Malang, Indonesia (email: msarosa@yahoo.com).

Wijono adalah dosen Teknik Elektro Universitas Brawijaya, Malang, Indonesia (Telp.0341-665144)

Suprpto adalah dosen Program Teknologi Informatika dan Ilmu Komputer Universitas Brawijaya, Malang, Indonesia (Telp.0341-665144)

Selain pengujian laboratorium, cara lain yang dilakukan oleh petambak untuk mendeteksi penyakit bintik putih pada udang adalah dengan melihat tanda-tanda yang muncul pada fisik udang. Cara ini kurang efektif dan teliti untuk jumlah udang yang cukup banyak. Maka penggunaan teknologi informasi memungkinkan untuk mendeteksi penyakit bintik putih pada citra udang dengan bantuan komputer.

Perkembangan teknologi pengolah citra digital hingga saat ini terus diperluas dengan tujuan untuk membantu manusia dalam melakukan pekerjaannya. Pengolahan citra digital itu sendiri merupakan salah satu jenis teknologi untuk menyelesaikan masalah mengenai pemrosesan citra. Dalam pengolahan citra, citra yang ada diolah sedemikian rupa sehingga citra tersebut lebih mudah untuk diproses lebih lanjut [1].

Jaringan saraf tiruan yang berupa susunan sel-sel syaraf tiruan (*neuron*) dibangun berdasarkan prinsip kerja otak manusia. Perhatian yang besar pada jaringan saraf tiruan disebabkan adanya keunggulan yang dimilikinya seperti kemampuan untuk belajar, sifat toleransi kesalahan (*fault tolerance*). Salah satu Metoda yang digunakan dalam JST adalah *backpropagation* (propagasi balik). Metoda ini telah banyak diaplikasikan secara luas, sekitar 90% telah berhasil diaplikasikan di berbagai bidang diantaranya pengenalan pola suara, pengolah citra medika, pengenalan pola tulisan tangan dan masih banyak lagi keberhasilan propagasi balik sebagai salah satu metoda komputasi yang handal [9,10].

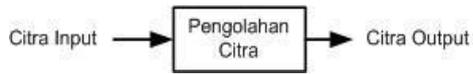
Penelitian ini bertujuan untuk merancang dan membuat perangkat lunak yang dapat mendeteksi penyakit bintik putih pada citra udang dengan menggunakan program Delphi versi 7. Dengan menggunakan program tersebut dapat divisualisasikan pengolah citra berupa pemotongan citra, grayscale dan histogram. Dari hasil histogram tersebut didapatkan nilai piksel mulai dari 0 sampai 255 yang digunakan sebagai data masukan pada JST untuk dilatih dan diuji. Hasil dari pengujian tersebut dapat menyimpulkan apakah udang tersebut sehat atau berpenyakit bintik putih.

II. LANDASAN TEORI

A. Pengolahan Citra Digital

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau komputer. Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain, artinya masukannya berupa citra dan keluarannya juga berupa

citra tetapi kualitasnya lebih baik dari citra masukan [7], seperti ditunjukkan dalam Gambar 1.



Gambar 1. Diagram Blok Pengolahan Citra [6]

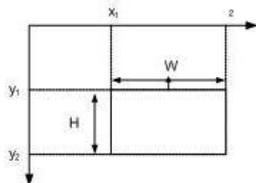
Citra digital pada umumnya berbentuk persegi panjang atau bujur sangkar (pada beberapa sistem pencitraan ada pula yang berbentuk segienam) yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau piksel sehingga ukuran citra selalu bernilai bulat. Setiap titik memiliki koordinat sesuai posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan [1,5].

B. Pemotongan (Cropping) Citra

Cropping pada pengolahan citra berarti memotong satu bagian dari citra sehingga diperoleh citra yang diharapkan [11]. Ukuran pemotongan citra dapat dirumuskan sebagai berikut:

$$W = x_2 - x_1 \text{ dan } H = y_2 - y_1 \quad (1)$$

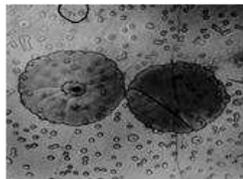
Titik (x_1, y_1) adalah koordinat titik pojok kiri atas dan titik (x_2, y_2) adalah koordinat titik pojok kanan bawah, seperti terlihat dalam Gambar 2.



Gambar 2. Proses Pemotongan Citra [11]

C. Grayscale

Grayscale (derajat keabuan) adalah tingkat warna abu-abu dari sebuah piksel. Nilai yang terkandung dalam piksel menunjukkan tingkat keabu-abuan piksel tersebut dari hitam ke putih. Biasanya nilainya ditetapkan dari nilai 0 sampai 255 (untuk 256 derajat keabuan), dengan 0 adalah warna hitam dan 255 adalah warna putih [1,11]. Citra derajat keabuan ditunjukkan dalam Gambar 3.



Gambar 3. Citra *Grayscale* Mikroskopi Penyakit Bintik Putih [6].

Proses *grayscale* adalah tahap penyederhanaan warna, untuk mendapat citra derajat keabuan dari representasi warna RGB (*Red, Green, Blue*). Warna disederhanakan dari citra 24 bit menjadi citra 8 bit atau 256 warna pokok. Nilai *grayscale* diperoleh dengan menggunakan rumus konversi yang dikeluarkan oleh *CCIR Recommendation 601-1* [8] seperti dalam Persamaan (2) sebagai berikut:

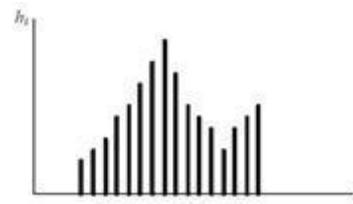
$$Y = 0,299 * R + 0,587 * G + 0,114 * B \quad (2)$$

D. Histogram Citra

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (*relative*) dari intensitas pada citra tersebut [1,5,11]. Secara matematis histogram citra dapat dihitung dengan rumus :

$$h_i = \frac{n_i}{n} \quad i = 0, 1, \dots, L-1 \quad (3)$$

h_i merupakan peluang piksel, n_i adalah jumlah piksel yang memiliki derajat keabuan i , n adalah jumlah seluruh piksel di dalam citra, dan L adalah derajat keabuan dengan nilai 0 sampai $L-1$. Secara grafis histogram ditampilkan dengan diagram batang seperti ditunjukkan dalam Gambar 4.



Gambar 4. Histogram Citra [5]

E. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) atau Artificial Neural Network adalah suatu metode komputasi yang meniru sistem jaringan syaraf biologis. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi [9,11], dengan asumsi bahwa:

- Pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*)
- Sinyal dikirimkan antara neuron-neuron melalui penghubung-penghubungnya
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
- Untuk menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi yang dikenakan pada jumlah masukan yang diterima. Besarnya keluaran ini selanjutnya dibandingkan dengan suatu batas ambang

JST dapat ditentukan dengan 3 hal, yaitu:

- Pola hubungan antar neuron (disebut arsitektur jaringan).
- Metode untuk menentukan bobot penghubung yang disebut metode training.
- Fungsi aktivasi yaitu fungsi yang digunakan untuk menentukan keluaran suatu neuron.

F. Pelatihan Backpropagation

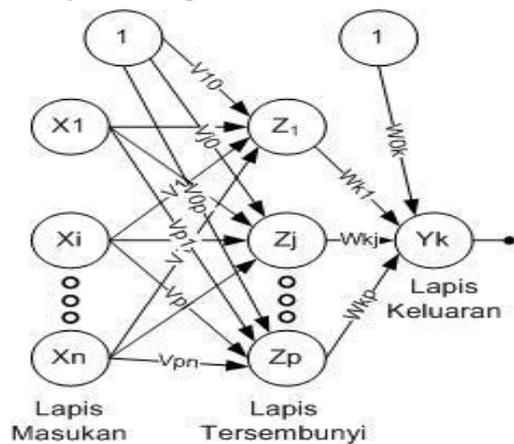
Algoritma pelatihan *backpropagation* (propagasi balik) pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart bersama McClelland untuk dipakai pada JST. Algoritma ini termasuk metoda pelatihan terbimbing (*supervised*) dan didesain untuk operasi pada JST *feed forward* lapis jamak [9].

Arsitektur JST propagasi balik terdiri dari lapis

masukan, lapis tersembunyi dan lapis keluaran seperti ditunjukkan dalam Gambar 5.

Algoritma pelatihan propagasi balik terdiri dari 3 tahapan [10], yaitu: tahap propagasi maju, tahap perambatan mundur, dan tahap perubahan bobot, sehingga dapat dituliskan dalam prosedur pelatihan propagasi balik sebagai berikut:

- Langkah 0:
Inisialisasi bobot (diberi nilai kecil secara acak)
- Langkah 1:
Ulangi langkah 2 hingga 9 sampai kondisi akhir iterasi terpenuhi
- Langkah 2:
Untuk setiap pasang data pelatihan, lakukan langkah 3 sampai 8



Gambar 5. Arsitektur Pelatihan Backpropagation [10]

Tahap propagasi maju

- Langkah 3:
Tiap unit masukan (X_i , $i = 1, \dots, n$) menerima sinyal masukan X_i dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit tersembunyi).
- Langkah 4:
Masing-masing unit dilapis tersembunyi dikalikan dengan bobot dan dijumlahkan serta ditambah dengan biasnya.

$$Z_in_j = V_{j0} + \sum_{i=1}^n X_i V_{ji} \quad (4)$$

V_{j0} = bias pada unit tersembunyi j aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya, $Z_j = f(Z_in_j)$, dan kirimkan sinyal ini keseluruhan unit pada lapis di atasnya (unit keluaran).

- Langkah 5:
Tiap unit keluaran (y_k , $k = 1, \dots, m$) jumlahkan bobot sinyal masukannya.

$$Y_in_k = V_{ko} + \sum_{j=1}^p Z_j W_{kj} \quad (5)$$

W_{ko} = bias pada unit keluaran k dan aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya, $Y_k = f(Y_in_k)$.

Tahap Perambatan Mundur:

- Langkah 6:
Tiap unit keluaran (y_k , $k = 1, \dots, m$) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan informasinya,

$$\delta_k = (t_k - Y_k) f'(Y_in_k) \quad (6)$$

hitung koreksi bobotnya (digunakan untuk memperbaharui W_{jk} nantinya),

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (7)$$

hitung koreksi biasnya (digunakan untuk memperbaharui w_{ko} nantinya), dan kirimkan δ_k ke unit-unit pada lapisan dibawahnya,

- Langkah 7:
Setiap unit lapisan tersembunyi (Z_j , $j = 1, \dots, p$) jumlahkan hasil perubahan masukannya (dari unit-unit lapisan di atasnya),

$$\delta_in_j = W_{ok} + \sum_{k=1}^m \delta_k W_{jk} \quad (8)$$

kalikan dengan turunan fungsi aktivasinya untuk menghitung informasi kesalahannya,

$$\delta_j = \delta_in_j f'(z_in_j) \quad (9)$$

Hitung koreksi bobot

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (10)$$

Hitung koreksi bias

$$\Delta v_{oj} = \alpha \delta_j \quad (11)$$

Tahap memperbaiki bobot dan bias

- Langkah 8:
Hitung semua perubahan bobot
Perubahan bobot garis yang menuju ke unit keluaran ($k=1,2,\dots,m$ dan $j = 0,1,\dots,p$):

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (12)$$

Perubahan bobot garis yang menuju ke unit tersembunyi ($j = 1, 2, \dots, p$ dan $i = 0, 1, \dots, n$):

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (13)$$

- Setelah pelatihan selesai dilakukan, jaringan dapat digunakan untuk pengenalan pola dengan menggunakan langkah 4 dan 5 (tahap propagasi maju).

G. Penyakit Bercak Putih Viral (White Spot Syndrome Virus, WSSV)

Penyakit yang paling sering ditemukan terkait dengan kematian adalah penyakit bintik putih viral. Udang yang terserang penyakit ini menunjukkan tanda adanya bintik putih di seluruh tubuhnya, dari karapas hingga pangkal ekor.

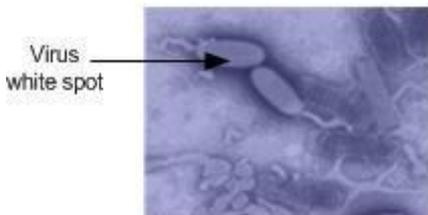
Penyebab penyakit bercak putih ini adalah virus yang disebut "Bacilliform". Dalam istilah asing penyakit ini dinamai SEMBV (*Systemic Ectodermal and Mesodermal*

Basulovirus), karena virus ini menimbulkan kerusakan dan kematian sel (*necrosis multivocal*) pada jaringan *ektodermal* dan *mesodermal* seperti sel *epitel subkutikuler* (kulit bagian dalam). Nekrosis pada sel *epitel subkutikuler* inilah yang menghasilkan bintik putih pada kulit udang [7], seperti ditunjukkan dalam Gambar 6 dan Gambar 7.

Pada umumnya gejala-gejala penyakit bintik putih pada udang dapat diidentifikasi melalui dua pendekatan yaitu: (i) tingkah laku/kondisi udang yang terinfeksi penyakit ini, dan (ii) identifikasi penampakan jenis penyakit pada anggota tubuh udang [7].

Selanjutnya gejala-gejala yang sering muncul jika udang terinfeksi jenis penyakit ini antara lain :

- Penurunan nafsu makan udang secara mendadak bahkan dalam waktu relatif singkat udang tidak makan sekali.
- Pencernaan udang kosong.
- *Hepatopanchreas* menyusut, dengan warna biru/putih.
- Udang terlihat pasif.
- Udang melayang di permukaan air
- Udang banyak menempel di dinding tambak.
- Pada kondisi parah terjadi kematian massal di dasar tambak.



Gambar 6. Morfologis Virus Pemotretan Dengan Scanning Mikroskopi [7]



Gambar 7. Udang Terserang Bercak Putih Viral, (a) Terlihat Bintik Keputihan pada Seluruh Tubuh, dan Karapas Udang, (b) Gambar Mikroskopi Bintik Putih [6]

Gejala-gejala penyakit bintik putih seperti tersebut di atas, merupakan gejala yang biasa diperoleh melalui pengamatan secara visual di lapangan. Sedangkan untuk mengetahui gejala-gejala secara klinis dari penyakit ini sebaiknya dilakukan melalui pengamatan laboratorium (jika ada) terhadap beberapa sampel udang dari tambak.

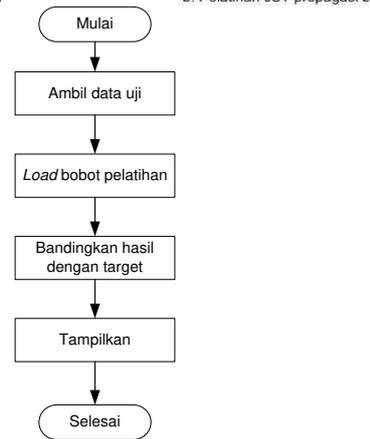
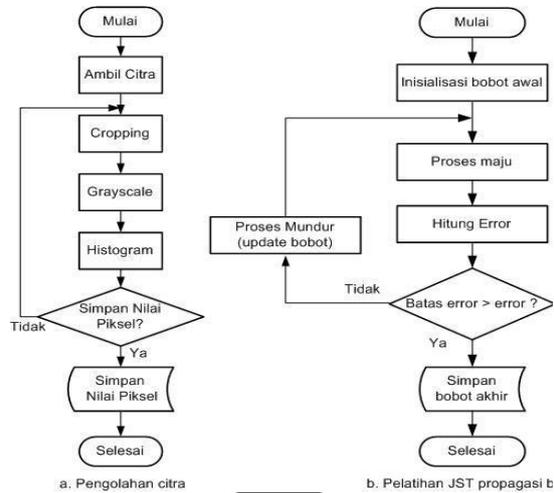
III. PERANCANGAN DAN IMPLEMENTASI

Perancangan Perangkat Lunak

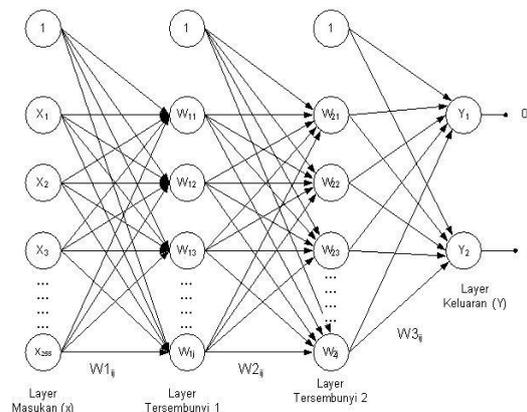
Perancangan sistem deteksi penyakit bintik putih pada citra udang menggunakan program Delphi 7, dimulai dari perancangan pengolahan citra dan dilanjutkan dengan perancangan JST propagasi balik seperti ditunjukkan dalam Gambar 8.

Arsitektur jaringan syaraf tiruan (Gambar 9) yang digunakan dalam penelitian ini terdiri dari 3 lapisan yaitu lapisan masukan, lapisan tersembunyi dan lapisan keluaran dengan rincian sebagai berikut:

- Jumlah *neuron* lapisan masukan sebanyak 256 *neuron*. Penentuan 256 *neuron* pada lapisan masukan berdasarkan pada data dari hasil proses grayscale citra udang yang merupakan nilai piksel



Gambar 8. Diagram Alir Perancangan Perangkat Lunak



Gambar 9. Arsitektur Pelatihan Backpropagation

- Jumlah lapisan tersembunyi sebanyak 2 lapisan dengan jumlah *neuron* tiap-tiap lapis tersembunyi dilakukan secara variasi

- Jumlah *neuron* lapis keluaran terdiri dari 2 *neuron*. Setiap kondisi udang direpresentasikan dengan bilangan biner 01 untuk kondisi udang sehat dan 10 untuk kondisi udang sakit.

Pada metode *backpropagation* terdapat 3 tahap, yaitu: proses maju, proses perhitungan error, dan proses mundur, dengan langkah-langkah sebagai berikut:

- Langkah awal
Inisialisasi weight (pemberian nilai semua bobot jaringan secara acak) $W1_{ij}$, $W2_{ij}$, dan $W3_{ij}$
- Langkah-langkah proses pelatihan

1. Proses propagasi maju

Layer 1:

- Kalikan semua masukan dengan bobot1

$$temp1_j = W1_{0j} + \sum_{i=1}^n X_{ik} W1_{ij}$$

- Kemudian dihitung sesuai dengan fungsi pengaktif yang digunakan:

$$V1_j = \frac{1}{1 + \exp(-temp1_j)}$$

Layer 2:

- Kalikan semua output layer 1 dengan bobot2

$$temp2_j = W2_{0j} + \sum_{i=1}^m V1_j W2_{ij}$$

- Kemudian dihitung sesuai dengan fungsi pengaktif yang digunakan:

$$V2_j = \frac{1}{1 + \exp(-temp2_j)}$$

Layer 3:

- Kalikan semua output layer 2 dengan bobot3

$$temp3_j = W3_{0j} + \sum_{i=1}^o V2_j W3_{ij}$$

- Kemudian dihitung sesuai dengan fungsi pengaktif yang digunakan:

$$V3_j = \frac{1}{1 + \exp(-temp3_j)}$$

2. Proses perhitungan error

- Hitung rata-rata error setiap *neuron* keluaran

$$Error_n = |Target_{jk} -$$

3. Proses perbaikan bobot

Layer 3:

- Hitung $\delta 3$ (delta bobot3)

$$\delta 3_j = (target_j - V3_j) V3_j'$$

karena

$$V3_j' = f'(temp3_j) = \frac{1}{1 + \exp(-temp3_j)}$$

menggunakan fungsi sigmoid biner, maka:

$$f'(temp3_j) = f(temp3_j)(1 - f(temp3_j)) = V3_j(1 - V3_j)$$

Perbaiki bobot3

$$W3_{ij} = W3_{ij}(lama) + \alpha \delta 3_j V2_j$$

Layer 2:

- Hitung $\delta 2$ (delta bobot2)

$$\delta 2_j = \left(\sum_{i=1}^o \delta 3_i W3_{ji} \right) V2_j'$$

Karena

$$V2_j' = f'(temp2_j) = \frac{1}{1 + \exp(-temp2_j)}$$

menggunakan fungsi sigmoid biner, maka :

$$f'(temp2_j) = f(temp2_j)(1 - f(temp2_j)) = V2_j(1 - V2_j)$$

Perbaiki bobot2

$$W2_{ij} = W2_{ij}(lama) + \alpha \delta 2_j V1_j$$

Layer 1:

$$\delta 1_j = \left(\sum_{i=1}^m \delta 2_i W2_{ji} \right) V1_j'$$

karena

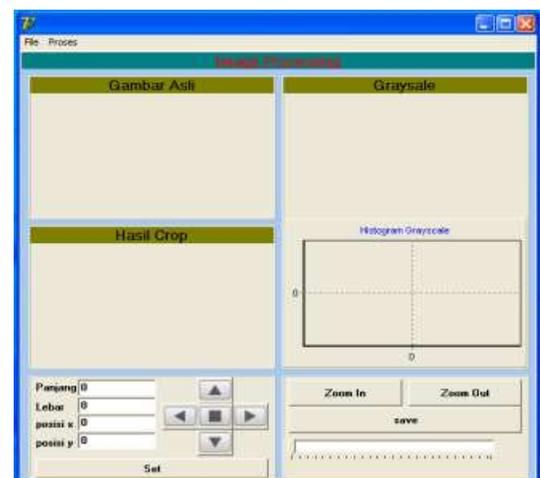
$$V1_j' = f'(temp1_j) = \frac{1}{1 + \exp(-temp1_j)}$$

menggunakan fungsi sigmoid biner, maka:

$$f'(temp1_j) = f(temp1_j)(1 - f(temp1_j)) = V1_j(1 - V1_j)$$

Perbaiki bobot1

$$W1_{ij} = W1_{ij}(lama) + \alpha \delta 1_j X_{jk}$$



Gambar 10. Menu Pengolahan Citra

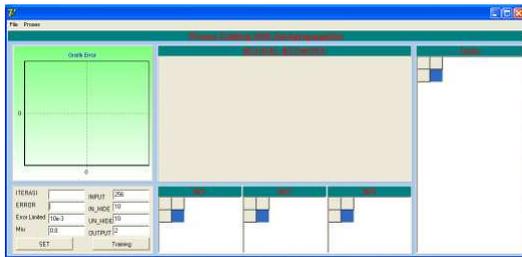
- Memeriksa *Stopping Condition*

Jika *Stopping Condition* telah terpenuhi, maka pelatihan jaringan saraf dapat dihentikan. *Stopping Condition* yang digunakan adalah dengan membatasi error dengan metode *Mean Square Error* (MSE) untuk menghitung rata-rata error antara keluaran

Implementasi Sistem Dalam Program

Tahap implementasi sistem ini dibagi menjadi 3 menu, yaitu:

- Proses *image*, digunakan untuk mencari nilai piksel citra dari hasil proses *cropping* citra dan menyimpan hasilnya dalam format teks, seperti dalam Gambar 10.
- Proses neural network digunakan untuk melatih data nilai piksel dari citra udang yang ditunjukkan dalam Gambar 11.
- Proses test, digunakan untuk menguji keakuratan dalam pengenalan penyakit bintik putih pada citra udang, ditunjukkan dalam Gambar 12.

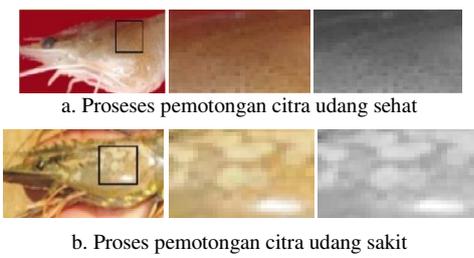


Gambar 11. Tampilan Menu Pelatihan

IV. PENGUJIAN DAN PEMBAHASAN

A. Proses Pengolahan Citra

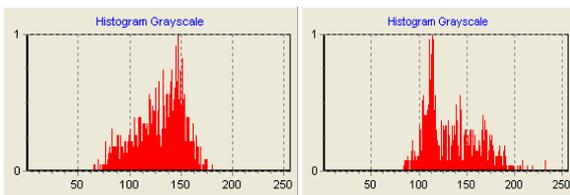
Proses pengolahan citra dimulai dari proses *cropping* citra dilakukan dengan menentukan nilai panjang dan lebar yang dibuat seragam pada semua citra, yaitu 30 x 25 piksel, yang diubah-ubah hanya pada posisi x dan y saja untuk menentukan lokasi dari proses pemotongan citra seperti ditunjukkan dalam Gambar 13.



a. Proseses pemotongan citra udang sehat

b. Proses pemotongan citra udang sakit

Gambar 13. Proses Pemotongan Citra



a. Udang sehat

b. Udang sakit

Gambar 14. Hasil Histogram Cirta

Dari hasil pemotongan citra, kemudian dilakukan proses *grayscale* dan histogram untuk mendapatkan nilai piksel pada udang sehat dan udang sakit yang ditunjukkan dalam Tabel 1 dan Tabel 2. Hasil histogram citra ditunjukkan dalam Gambar 14.

B. Pelatihan JST Propagasi balik

Pelatihan jaringan digunakan untuk melatih set data yang telah dibuat, yaitu data masukan berupa nilai piksel udang meliputi udang sehat dan udang sakit bintik putih. Sedangkan data targetnya adalah kondisi udang.

TABEL 1
NILAI PIKSEL HASIL PEMOTONGAN CITRA UDANG SEHAT

| No. | Nama Gambar | Posisi X | Y | Nilai I Piksel |
|-----|-------------|----------|----|----------------|
| 1 | H1a | 100 | 8 | 50 - 139 |
| 2 | H2a | 125 | 6 | 38 - 142 |
| 3 | H3a | 110 | 9 | 46 - 117 |
| 4 | H4a | 100 | 10 | 42 - 140 |
| 5 | H5a | 110 | 10 | 51 - 145 |
| 6 | H6a | 110 | 2 | 46 - 138 |
| 7 | H7a | 99 | 12 | 41 - 140 |
| 8 | H8a | 100 | 10 | 38 - 126 |
| 9 | H9a | 100 | 10 | 51 - 142 |
| 10 | H10a | 120 | 5 | 37 - 133 |
| 11 | H11a | 106 | 19 | 37 - 115 |
| 12 | H12a | 116 | 7 | 52 - 173 |
| 13 | H13a | 99 | 9 | 54 - 147 |
| 14 | H14a | 110 | 10 | 38 - 109 |
| 15 | H15a | 102 | 9 | 36 - 137 |
| 16 | H16a | 100 | 10 | 37 - 121 |
| 17 | H17a | 100 | 9 | 46 - 129 |
| 18 | H18a | 100 | 10 | 35 - 130 |

TABEL 2
NILAI PIKSEL HASIL PEMOTONGAN CITRA UDANG SAKIT

| No. | Nama Gambar | Posisi X | Y | Nilai Piksel |
|-----|-------------|----------|----|--------------|
| 1 | K1 | 196 | 41 | 83 - 231 |
| 2 | K2 | 159 | 54 | 107 - 240 |
| 3 | K3 | 146 | 49 | 97 - 218 |
| 4 | K4 | 146 | 49 | 87 - 199 |
| 5 | K5 | 130 | 49 | 84 - 211 |
| 6 | K6 | 115 | 29 | 84 - 210 |
| 7 | K7 | 54 | 11 | 124 - 241 |
| 8 | K8 | 249 | 39 | 81 - 185 |
| 9 | K9 | 83 | 20 | 47 - 253 |
| 10 | K10 | 76 | 11 | 115 - 256 |
| 11 | K11 | 121 | 10 | 98 - 162 |
| 12 | K12 | 117 | 37 | 98 - 174 |
| 13 | K13 | 208 | 42 | 117 - 212 |
| 14 | K14 | 83 | 39 | 145 - 239 |
| 15 | K15 | 47 | 13 | 5 - 225 |
| 16 | K16 | 34 | 10 | 134 - 246 |
| 17 | K17 | 141 | 27 | 92 - 216 |
| 18 | K18 | 200 | 25 | 93 - 221 |

Untuk menentukan jumlah *neuron* pada lapisan tersembunyi yang terbaik, maka dilakukan pengamatan terhadap lapis tersembunyi1 dan 2, dengan variasi jumlah *neuron* adalah 5 dan 5, 10 dan 10, 20 dan 20, 25 dan 25, 30 dan 30 *neuron* dengan variasi laju pembelajaran 0.1, 0.3, dan 0.5. Pelatihan akan berhenti jika nilai error ≤ nilai target error. Pengamatan dilakukan dengan target error = 0.001, maka hasilnya dapat dilihat dalam Tabel 3. Dan untuk menentukan besarnya nilai akurasi dapat dihitung dengan cara :

$$\text{Akurasi \%} = \frac{\text{Jumlah data sesuai target}}{\text{Jumlah data}} \times 100\%$$

Setelah dilakukan pelatihan dengan berbagai macam variasi nilai laju pembelajaran dan jumlah *neuron* pada lapis tersembunyi, maka hasil terbaik didapatkan pada

laju pembelajaran 0.1 dengan jumlah neuron pada lapis tersembunyi1 dan 2 adalah 20 dan 20. Hal ini dapat dilihat berdasarkan jumlah iterasi sebesar 51267 dengan nilai error 0,00999 yang ditunjukkan dalam Tabel 3. Selanjutnya nilai bobot dari hasil pelatihan tersebut disimpan dan digunakan pada saat proses pengujian.

TABEL 3
HASIL PELATIHAN DENGAN VARIASI LAJU PEMBELAJARAN DAN JUMLAH NEURON PADA LAPIS TERSEMBUNYI 1 DAN 2

| Laju Pembelajaran | Jml Pola Masukan | Jlm neuron | Iterasi | Error |
|-------------------|------------------|------------|---------|---------|
| 0.1 | 6 | 5:5 | 130990 | 0.00999 |
| | | 10:10 | 161685 | 0.00999 |
| | | 15:15 | 59037 | 0.00999 |
| | | 20:20 | 51267 | 0.00999 |
| | | 25:25 | 64952 | 0.00999 |
| 0,3 | 6 | 30:30 | 83537 | 0.00999 |
| | | 5:5 | 614389 | 0.00999 |
| | | 10:10 | 214388 | 0.00999 |
| | | 15:15 | 321789 | 0.00999 |
| | | 20:20 | 81451 | 0.00999 |
| 0,5 | 6 | 25:25 | 160201 | 0.00999 |
| | | 30:30 | 172640 | 0.00999 |
| | | 5:5 | 685411 | 0.00999 |
| | | 10:10 | 919267 | 0.00999 |
| | | 15:15 | 1267748 | 0.00999 |
| | | 20:20 | 1896005 | 0.00899 |
| | | 25:25 | 900248 | 0.00961 |
| | | 30:30 | 477651 | 0.00876 |

C. Pengujian Jaringan Syaraf Tiruan

Jaringan yang telah dilatih dan mencapai hasil yang dikehendaki perlu diuji untuk mengetahui kemampuannya pada saat mempelajari data latih yang diberikan. Selain itu juga pengujian dapat dilakukan menggunakan data yang belum pernah dilatihkan sebelumnya untuk melihat tingkat akurasi sistem yang telah dibuat, yaitu menggunakan data uji sebanyak 30 pola data.

TABEL 4
HASIL PENGUJIAN TERHADAP DATA LATIH

| No | Pola Masukan | Hasil Pengujian | Target pengenalar | Keterangan |
|----|--------------|-----------------|-------------------|------------|
| 1 | H1 | Sehat | Sehat | Benar |
| 2 | H2a | Sehat | Sehat | Benar |
| 3 | H4a | Sehat | Sehat | Benar |
| 4 | K2 | Sakit | Sakit | Benar |
| 5 | K8 | Sakit | Sakit | Benar |
| 6 | K11 | Sakit | Sakit | Benar |

Berdasarkan Tabel 4 dan Tabel 5, hasil pengujian terhadap data latih dan data uji dapat dianalisis dengan mengamati ketepatan atau akurasi antara target dengan keluaran jaringan, yaitu:

- Pengujian terhadap 6 data yang telah dilatih menghasilkan keakurasian sebesar 100% (semua data latih dapat dikenali).
- Data uji sebanyak 30 data yang terdiri dari udang sehat sebanyak 15 data dan udang sakit sebanyak 15 data.
- Pengujian terhadap 15 data udang sehat menghasilkan tingkat akurasi 100%, dimana JST dapat mengambil keputusan dengan benar terhadap data yang diuji. Sedangkan pengujian terhadap 15 data udang sakit menghasilkan 13 data dapat dikenali sebagai udang sakit dan 2 data dianggap sebagai udang sehat, yaitu data k9 dan

k15. Sehingga tingkat akurasi terhadap 30 data uji sebesar 93.33%. Dari hasil tersebut dapat dikatakan bahwa JST dapat mengenali nilai piksel udang sehat dan nilai piksel udang sakit.

TABEL 5
HASIL PENGUJIAN TERHADAP DATA UJI

| No. | Pola Masukkan | Hasil Pengujian | Target Pengenalan | Keterangan |
|-----|---------------|-----------------|-------------------|------------|
| 1. | H1a | Sehat | Sehat | Benar |
| 2 | H3a | Sehat | Sehat | Benar |
| 3 | H5a | Sehat | Sehat | Benar |
| 4 | H6a | Senat | Senat | Benar |
| 5 | H7a | Sehat | Sehat | Benar |
| 6 | H8a | sehat | sehat | Benar |
| 7 | H9a | Sakit | Sakit | Benar |
| 8 | H10a | Sehat | Sehat | Benar |
| 9 | H11a | Sehat | Sehat | Benar |
| 10 | H12a | Sehat | Sehat | Benar |
| 11 | H13a | Sehat | Sehat | Benar |
| 12 | H14a | Sehat | Sehat | Benar |
| 13 | H15a | Sehat | Sehat | Benar |
| 14 | H16a | Sehat | Sehat | Benar |
| 15 | H17a | Sehat | Sehat | Benar |
| 16 | H18a | Sehat | Sehat | Benar |
| 17 | K2 | Sehat | Sehat | Benar |
| 18 | K3 | Sakit | Sakit | Benar |
| 19 | K4 | Sakit | Sakit | Benar |
| 20 | K5 | Sakit | Sakit | Benar |
| 21 | K6 | Sakit | Sakit | Benar |
| 22 | K7 | Sakit | Sakit | Benar |
| 23 | K9 | Sehat | Sakit | Salah |
| 24 | K10 | Sakit | Sakit | Benar |
| 25 | K12 | Sakit | Sakit | Benar |
| 26 | K13 | Sakit | Sakit | Benar |
| 27 | K14 | Sakit | Sakit | Benar |
| 28 | K15 | Sehat | Sakit | Salah |
| 29 | K17 | Sakit | Sakit | Benar |
| 30 | K18 | Sakit | Sakit | Benar |

V. KESIMPULAN DAN SARAN

Berdasarkan aplikasi yang telah di buat beserta uji coba yang telah dilakukan maka dapat ditarik kesimpulan bahwa nilai intensitas piksel yang ditampilkan oleh histogram dipengaruhi oleh penentuan posisi dalam proses pemotongan citra.

Untuk menghasilkan kinerja yang baik dalam jaringan pelatihan maka dilakukan uji coba terhadap laju pembelajaran dan jumlah neuron dengan nilai yang bervariasi, sehingga didapatkan nilai terbaik berdasarkan iterasi terendah yaitu: untuk laju pembelajaran didapatkan 0.1, *neuron* pada lapis tersembunyi1 = 20 dan lapis tersembunyi2 = 20 dengan jumlah iterasi 51267 kali dan target error yang ditentukan 0.001. Dari hasil tersebut dilakukan pengujian terhadap data latih yang menghasilkan akurasi sebesar 100% dan untuk data uji sebanyak 30 data menghasilkan kesalahan dalam mengambil keputusan sebanyak 2 data dengan tingkat akurasi sebesar 93.33%.

Untuk pengembangan penelitian deteksi penyakit *white spot* pada udang dapat dilakukan secara *real-time* dalam pengambilan data melalui perangkat kamera, selanjutnya dilakukan proses pengolahan citra dan hasilnya dijadikan masukkan JST. Dalam pengambilan citra perlu dilakukan penentuan dan penetapan intensitas cahaya, warna latar belakang yang sama, dan jarak antara kamera dengan objek, agar citra menjadi seragam.

DAFTAR PUSTAKA

- [1]. Ahmad Usman. 2005. *Pengolahan Citra Digital dan Teknik Pemrograman*. Graha Ilmu. Yogyakarta.
- [2]. Amri K. 2003. *Budi Daya Udang Windu Secara Intensif*. PT. Agro Media Pustaka, Tangerang
- [3]. Alifuddin M, Dana D,dkk. 2003. *Patogenesis Infeksi Virus White Spot (WSSV) Pada Udang Windu (Penaeus Monodon Fab.)*. *Jurnal Akuakultur Indonesia*, 2(2). Bogor
- [4]. Aryani D, Susanto NG, dkk. 2008. *Pengaruh Perubahan Salinitas Terhadap Virulensi Terhadap White Spot Syndrome Virus Pada Udang Putih*. Prosiding Seminar Nasional Sain dan Teknologi II. Universitas Lampung. Lampung.
- [5]. Munir Rinaldi. 2004. *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Informatika. Bandung.
- [6]. Murdjani M, dkk. 2007, *Budidaya Udang Windu (Penaeus Monodon Fabricius) Intensif*, BBPBAP Jepara
- [7]. Nasrulamin. 2008. *Waspada Penyakit Bintik Putih Di Tambak Anda*. <http://nasrulamin.wordpress.com/>. Diakses pada tanggal 25 Maret 2011
- [8]. Nugroho HF. 2005. *Pengenalan Wajah dengan JST Backpropagation*. SNATI. Yogyakarta.
- [9]. Purnomo HM, Kurniawan A. 2006, *Supervised Neural Network dan Aplikasinya*, Graha Ilmu, Yogyakarta.
- [10]. Siang JJ. 2004. *Jaringan Saraf Tiruan & Pemrograman Menggunakan Matlab*, Andi, Yogyakarta.
- [11]. Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, D.O., dan Wijanarto, 2009, *Teori Pengolahan Citra Digital*, Andi, Yogyakarta.

Dedy Harto, lahir di Air Batu, Kab. Asahan Sumatera Utara pada tanggal 23 November 1972 adalah anak keempat dari ayah (alm) Bambang Ermanto dan ibu Ati Suparty, menempuh pendidikan SD sampai SMP di Kab. Asahan dan SMK di Kota Medan lulus tahun 1992, dan menyelesaikan pendidikan sarjana Teknik Elektro di Fakultas Teknik Sekolah Tinggi Teknologi Nasional Yogyakarta tahun 2001. Saat ini bekerja sebagai dosen di Jurusan Teknik Elektro Fakultas Teknik Universitas Borneo Tarakan mulai tahun 2002