

Jaringan Saraf Tiruan *Backpropagation* untuk Pengenalan Wajah Metode Ekstraksi Fitur Berbasis Histogram

Sigit Kusmaryanto

Abstract— One common weakness in pattern recognition for face recognition is imperative that accurate input pattern to the pattern identified. This causes the input is often not recognized or not identified so as to be in the input repeatedly. The application of face recognition using Artificial Neural Network (ANN) backpropagation with MATLAB 7.0 is expected to overcome the weakness of pattern recognition systems for face recognition. Histogram-based feature extraction methods used in this study to obtain identification characteristics of the face image and a neural network input data. Face image data using pixel size variations. Trained in facial image pixel size variations 640 x 480 pixels and 600 x 800 pixels with two of distance making on face: average(2-5m), close(<2m). The results obtained from ANN test using 18 images with frontal face: resulting number of units in the hidden layer 6, the number of input unit 255, the number of output unit 10, the maximum epoch 2500, 0001 and learning the target error rate = 0.9 with a percentage of 95% of face recognition

Keyword - Backpropagation, Face Recognition Feature Extraction, Histogram.

Abstrak— Salah satu kelemahan umum pada pengenalan pola untuk pengenalan wajah adalah keharusan pola masukan yang akurat terhadap pola teridentifikasi. Hal tersebut menyebabkan masukan sering tidak dikenali atau tidak teridentifikasi sehingga harus diinput secara berulang-ulang. Aplikasi pengenalan wajah menggunakan Jaringan Saraf Tiruan(JST) Backpropagation dengan MATLAB 7.0 diharapkan mengatasi kelemahan sistem pengenalan pola untuk pengenalan wajah. Metode ekstraksi fitur berbasis histogram digunakan dalam penelitian ini untuk mendapatkan identifikasi ciri citra wajah dan sebagai data input JST. Data citra wajah menggunakan variasi ukuran piksel. Citra wajah dilatih pada variasi ukuran piksel 640 x 480 piksel dan 600 x 800 piksel dengan dua jarak pengambilan wajah yaitu sedang(2-5m), dekat (<2m). Hasil yang didapatkan dari uji JST menggunakan 18 citra dengan wajah frontal: dihasilkan jumlah unit pada lapisan tersembunyi 6, jumlah unit input 255, jumlah unit output 10, maksimum epoch 2500, target error 0.001 dan learning rate=0.9 dengan prosentase pengenalan wajah 95%

Kata Kunci—Backpropagation, Pengenalan Wajah, Ekstraksi Fitur, Histogram.

Sigit Kusmaryanto adalah Dosen Teknik Elektro Universitas Brawijaya Malang.

I. PENDAHULUAN

APLIKASI pengenalan wajah secara umum meliputi pengenalan wajah statis atau terkontrol sampai sistem identifikasi wajah dinamis yang tak terkontrol di dalam suatu latar belakang yang berbaur. Pendekatan yang paling umum untuk pengenalan wajah didasarkan pada bentuk dan penempatan atribut wajah, seperti mata, alis mata, hidung, bibir dan dagu serta hubungan antara atribut tersebut atau analisis wajah secara keseluruhan yang menghadirkan wajah sebagai suatu kombinasi dari sejumlah wajah kanonik. Meskipun unjuk kerja dari sistem pengenalan wajah yang secara komersial tersedia cukup banyak, namun sistem ini mempunyai kelemahan dan keterbatasan atas bagaimana cara citra wajah diperoleh. Kadang diperlukan latar belakang citra wajah yang sederhana dan penerangan khusus. Sistem ini juga mempunyai kesukaran di dalam mengenali wajah dari berbagai sudut pandang (*pose*) yang berbeda dan di bawah kekuatan penerangan dengan kondisi yang berbeda-beda. Agar dalam praktek suatu sistem pengenalan wajah dapat bekerja dengan baik maka sistem harus secara otomatis dapat (i) mendeteksi kehadiran wajah pada citra yang diperoleh; (ii) menempatkan wajah jika ada; (iii) mengenali wajah dari sudut pandang umum; (dari berbagai *pose*)

Sistem pengenalan wajah digunakan untuk membandingkan satu citra wajah masukan dengan suatu database wajah dan menghasilkan wajah yang paling cocok dengan citra tersebut jika ada. Sedangkan autentikasi wajah (*face authentication*) digunakan untuk menguji keaslian/kesamaan suatu wajah dengan data wajah yang telah diinput sebelumnya. Bidang penelitian yang juga berkaitan dengan pemrosesan wajah adalah lokalisasi wajah (*face localization*) yaitu pendeteksian wajah dengan asumsi hanya ada satu wajah di dalam citra [1]. Sistem pengenalan citra wajah dibagi menjadi dua jenis, yaitu sistem *feature-based* dan sistem *image-based*. Pada sistem *feature-based* digunakan ciri yang diekstraksi dari komponen citra wajah seperti mata, hidung, mulut, dan lain-lain yang kemudian dimodelkan secara geometris hubungan antara ciri-ciri tersebut. Sedangkan pada sistem *image-based* menggunakan informasi mentah dari piksel citra yang kemudian direpresentasikan dalam metode tertentu, misalnya seperti *Principal Component Analysis* (PCA) atau *transformasi wavelet* yang digunakan untuk klasifikasi identitas citra [2]. Metode atau teknik pengenalan

wajah yang umum telah dilakukan sebelumnya, antara lain dengan menggunakan algoritma *Eigenface* [3], dengan distribusi *Gaussian* dan *Clustering* [4], dengan *Support Vector Machine* [5], dan dengan metode *Statistic* dan *Wavelet* [6]. Beberapa masalah yang mungkin terjadi yakni perubahan skala, perubahan posisi, perubahan cahaya, perubahan detil dan ekspresi wajah.

Aplikasi pengenalan wajah dengan menggunakan jaringan saraf tiruan sebagai komponen utama khususnya jaringan *backpropagation* untuk pemrosesan dan identifikasi wajah telah banyak dilakukan. Namun metode ekstraksi fitur berbasis histogram untuk identifikasi data citra wajah sebagai masukan jaringan saraf tiruan belum banyak ditemui. Dalam penelitian yang sudah ada [7] aplikasi pengenalan wajah menggunakan citra wajah yang diambil dari pose frontal dan memiliki jarak pengambilan citra yang relatif sama dan kekurangannya adalah apabila kita melakukan pemrosesan terhadap citra wajah yang sama dengan pengambilan jarak citra yang berbeda maka aplikasi tidak dapat mengekstraksi fitur wajah dengan sempurna dan hasilnya tidak akurat. Penelitian sebelumnya juga belum mengadaptasi kecerdasan buatan namun prosesing citra wajah tidak dilakukan ekstraksi fitur berbasis histogram. Berdasarkan penelitian tersebut, Peneliti bermaksud membuat sebuah aplikasi jaringan saraf tiruan *backpropagation* untuk pengenalan wajah menggunakan ekstraksi fitur berbasis histogram yang dapat menyelesaikan kelemahan umum pengenalan wajah.

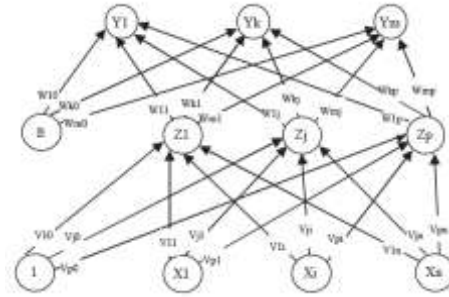
II. JARINGAN SARAF TIRUAN *BACKPROPAGATION*

Jaringan Saraf Tiruan (JST) adalah representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. Istilah buatan digunakan karena jaringan saraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. JST merupakan suatu model komputasi yang menirukan cara kerja sisitem otak manusia. Seperti halnya jaringan saraf biologis, JST juga memiliki kemampuan untuk belajar dan beradaptasi terhadap masukan-masukan. JST menyerupai otak manusia dalam dua hal, yaitu: (1) Pengetahuan diperoleh jaringan melalui proses belajar; (2) Kekuatan hubungan antar sel syaraf (*neuron*) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan.

Backpropagation adalah salah satu model JST yang mempunyai kemampuan mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

Backpropagation memiliki beberapa unit *neuron* yang ada dalam satu atau lebih layar tersembunyi. Arsitektur *Backpropagation* dengan n buah masukan (ditambah satu bias), dan sebuah layar tersembunyi

yang terdiri dari p unit (ditambah satu bias), dan juga m buah unit keluaran. Arsitektur Jaringan *Backpropagation* seperti tampak pada Gambar 1., dimana V_{ji} adalah bobot garis dari unit masukan X_i ke unit layar tersembunyi Z_j (V_{j0} adalah bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi Z_j). Sedangkan W_{kj} adalah bobot dari unit layar tersembunyi Z_j ke unit keluaran Y_k (W_{k0} adalah bobot dari bias di layar tersembunyi ke unit keluaran (Y_k))



Gambar 1. Arsitektur Jaringan *Backpropagation*

Dalam *Backpropagation* ini, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinyu, terdeferensial dengan mudah dan merupakan fungsi yang tidak turun. Fungsi yang dipakai adalah *sigmoid* biner, *tansig* dan *purelin*, dengan persamaan sebagai berikut:

Fungsi *sigmoid*:

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

$$y' = f'(x) = \theta f(x)[1 - f(x)] \tag{2}$$

Fungsi *tansig*:

$$y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \tag{3}$$

$$y' = f'(x) = [1 + f(x)][1 - f(x)] \tag{4}$$

Fungsi *tansig*:

$$y = f(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \tag{5}$$

$$y' = f'(x) = [1 + f(x)][1 - f(x)] \tag{6}$$

Algoritma Pelatihan JST *Backpropagation*

- a. Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil)
- b. Kerjakan langkah-langkah berikut selama kondisi berhenti bernilai *false* :
1. 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 tersembunyi ($Z_j, j=1,2,3,\dots,p$) menjumlahkan sinyal-sinyal input terbobot :

$$Z_in_j = V_{0k} + \sum_{i=1}^n X_i V_{ij} \tag{7}$$

- gunakan fungsi aktivasi untuk menghitung sinyal outputnya: $Z_j = f(Z_in_j)$, 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.

$$Y_{in_k} = W_{o_k} + \sum_{i=1}^p Z_i W_{jk} \quad (8)$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya: $Y_k = f(Y_{in_k})$. Kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output).

Backpropagation

- d. Tiap-tiap unit output ($Y_k, k=1,2,3,\dots,m$) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya :

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (9)$$

Hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai W_{jk}): Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai W_{o_k}):

$$\Delta W_{o_k} = \alpha \delta_k \quad (10)$$

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

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (11)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (12)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai V_{ij}):

$$\Delta V_{jk} = \alpha \delta_j X_i \quad (13)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai V_{o_j}):

$$\Delta V_{o_j} = \alpha \delta_j \quad (14)$$

- f. Tiap-tiap unit output ($Y_k=1,2,3,\dots,m$) memperbaiki bias dan bobotnya ($i=1,2,3,\dots,p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (15)$$

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

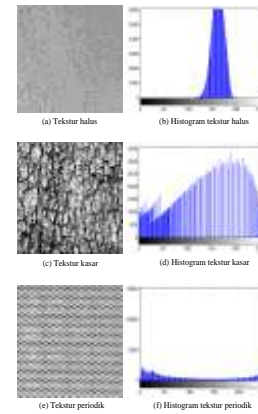
$$v_{jk}(\text{baru}) = v_{jk}(\text{lama}) + \Delta v_{jk} \quad (16)$$

2. Tes kondisi berhenti :
3. Langkah-langkah diatas adalah untuk satu kali siklus pelatihan (satu *epoch*). Proses pelatihan diulang sampai jumlah *epoch* tertentu atau telah tercapai target error yang diinginkan.
4. Hasil akhir pelatihan jaringan adalah didapatkannya bobot-bobot W_1, \dots, W_n yang kemudian disimpan untuk pengujian jaringan.

III. EKSTRAKSI FITUR BERBASIS HISTOGRAM

Metode yang sederhana untuk mendapatkan tekstur citra adalah dengan mendasarkan pada histogram. Fitur-fitur tekstur citra dapat dikenal secara statistik melalui histogram. Histogram dari tiga buah citra yang mengandung tekstur yang berbeda, yang terdapat di Gambar 2. Gambar 2(a) menunjukkan bahwa citra dengan tekstur halus memiliki daerah perubahan intensitas yang sempit. Sebaliknya, citra yang kasar memiliki kontras yang tinggi, ditandai dengan jangkauan intensitas yang lebar (Gambar 2(c) dan (d)). Menurut penglihatan, citra dalam Gambar 2(e) juga termasuk kasar dibandingkan dengan citra pada Gambar

2(a) meskipun beraturan



Gambar 2 Histogram tiga citra yang bertekstur berbeda

Fitur pertama yang dihitung secara statistik adalah rerata intensitas. Komponen fitur ini dihitung berdasar persamaan

$$m = \sum_{i=0}^{L-1} i \cdot p(i) \quad (17)$$

Dalam hal ini, i adalah aras keabuan pada citra f dan $p(i)$ menyatakan probabilitas kemunculan i dan L menyatakan nilai aras keabuan tertinggi. Rumus di atas akan menghasilkan rerata kecerahan objek.

Fitur kedua berupa deviasi standar. Perhitungannya sebagai berikut:

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (i - m)^2 p(i)} \quad (18)$$

Dalam hal ini, σ^2 dinamakan varians atau momen orde dua ternormalisasi karena $p(i)$ merupakan fungsi peluang. Fitur ini memberikan ukuran kekontrasan.

Fitur *skewness* merupakan ukuran ketidaksimetrisan terhadap rerata intensitas. Definisinya :

$$\text{skewness} = \sum_{i=0}^{L-1} (i - m)^3 p(i) \quad (19)$$

Skewness sering disebut sebagai momen orde tiga ternormalisasi. Nilai negatif menyatakan bahwa distribusi kecerahan condong ke kiri terhadap rerata dan nilai positif menyatakan bahwa distribusi kecerahan condong ke kanan terhadap rerata. Dalam praktik, nilai *skewness* dibagi dengan $(L-1)^2$ supaya ternormalisasi.

Deskriptor energi adalah ukuran yang menyatakan distribusi intensitas piksel terhadap jangkauan aras keabuan. Definisinya sebagai berikut:

$$\text{energi} = \sum_{i=0}^{L-1} [p(i)]^2 \quad (20)$$

Citra yang seragam dengan satu nilai aras keabuan akan memiliki nilai energi yang maksimum, yaitu sebesar 1. Secara umum, citra dengan sedikit aras keabuan akan memiliki energi yang lebih tinggi daripada yang memiliki banyak nilai aras keabuan. Energi sering disebut sebagai keseragaman.

Entropi mengindikasikan kompleksitas citra. Perhitungannya sebagai berikut:

$$\text{entropi} = - \sum_{i=0}^{L-1} p(i) \log_2(p(i)) \quad (21)$$

Semakin tinggi nilai entropi, semakin kompleks citra tersebut. Perlu diketahui, entropi dan energi berkecenderungan berkebalikan. Entropi juga merepresentasikan jumlah informasi yang terkandung di dalam sebaran data.

Properti kehalusan biasa disertakan untuk mengukur tingkat kehalusan/kekasaran intensitas pada citra.

Definisinya sebagai berikut:

$$R = 1 - \frac{1}{1 + \sigma^2} \tag{22}$$

Pada rumus di atas, σ adalah deviasi standar. Berdasarkan rumus di atas, Nilai R yang rendah menunjukkan bahwa citra memiliki intensitas yang kasar. Perlu diketahui, di dalam menghitung kehalusan, varians perlu dinormalisasi sehingga nilainya berada dalam jangkauan [0 1] dengan cara membaginya dengan $(L-1)^2$. Hasil perbandingan statistika tekstur untuk berbagai citra ditunjukkan pada Tabel 1.

TABEL 1
FITUR TEKSTUR CITRA BERBASIS HISTOGRAM

| Objek | Fitur |
|-------|-----------------------------|
| | Rerata intensitas : 161,408 |
| | Rerata kontras : 16,089 |
| | Skewness : -0,2545 |
| | Energi : 0,0220 |
| | Entropi : 3,9482 |
| | Smoothness : 0,0039 |
| | Rerata intensitas : 127,633 |
| | Rerata kontras : 74,137 |
| | Skewness : -0,0312 |
| | Energi : 0,0077 |
| | Entropi : 4,9436 |
| | Smoothness : 0,0779 |
| | Rerata intensitas : 162,381 |
| | Rerata kontras : 99,3366 |
| | Skewness : -6,4603 |
| | Energi : 0,1388 |
| | Entropi : 4,0303 |
| | Smoothness : 0,1318 |

Pendekatan yang serupa dengan di depan dilakukan dengan menggunakan *probability density function*. Apabila $p(i)$ adalah PDF, momen pusat PDF didefinisikan sebagai

$$m_k = \sum_{i=0}^{L-1} (i - \mu_f)^k \cdot p(i) \tag{23}$$

dengan $i=0,1,2,\dots,L-1$ adalah aras keabuan pada citra. Adapun μ_f adalah rerata aras keabuan pada citra, yang dihitung sebagai berikut:

$$\mu_f = \sum_{i=0}^{L-1} i \cdot p(i) \tag{24}$$

Momen kedua, yakni varians aras keabuan, yang berlaku sebagai ukuran ketidakhomogenan, berupa

$$\sigma_k^2 = m_2 = \sum_{i=0}^{L-1} (i - \mu_k)^2 \cdot p(i) \tag{25}$$

Momen ketiga dan keempat ternormalisasi, yang secara berturut-turut bernama *skewness* dan *kurtosis*, dinyatakan seperti berikut:

$$skewness = \frac{m_3}{m_2^{3/2}} \tag{26}$$

$$kurtosis = \frac{m_4}{m_2^2} \tag{27}$$

Skewness merupakan ukuran asimetri dan kurtosis merupakan ukuran keseragaman.

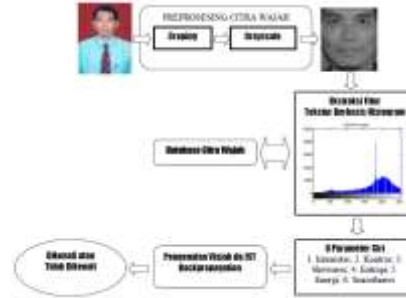
Pendekatan tekstur dengan probabilitas mempunyai kelebihan pada sifatnya yang tidak tergantung pada operasi translasi, penyekalaan, dan rotasi. Kelemahannya adalah mengabaikan hubungan antarpiksel secara lokal. Dengan kata lain, tekstur dengan susunan spasial yang berbeda tetapi memiliki distribusi aras keabuan yang sama tidak dapat dibedakan.

IV. METODE PENELITIAN

Implementasi *JST Backpropagation* untuk pengenalan wajah tampak pada Gambar 2, dibangun

melalui beberapa tahap yaitu tahap preprosesing citra wajah, tahap ekstraksi citra wajah dan tahap pengenalan wajah menggunakan *JST Backpropagation*.

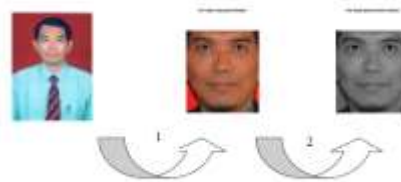
Data citra wajah diambil dengan foto digital dengan variasi jarak dan ukuran piksel. Variasi jarak yang digunakan pada penelitian ini adalah pengambilan citra digital pada jarak dekat (<2m) dan sedang (2-5m). Ukuran piksel pada citra adalah 640 x 480 piksel dan 600 x 800 piksel yang diperoleh dengan *Microsoft Office Picture Manager*.



Gambar 2. *JST Backpropagation* Untuk Pengenalan Wajah

A. Tahap preprosesing citra wajah

Seperti tampak pada Gambar 3, tahap ini dimaksudkan untuk mendapatkan aras keabuan dari citra wajah. Pada tahap ini digunakan pendekatan berbasis gambar dengan menghapus data yang berlebihan dari gambar wajah melalui kompresi gambar dengan menggunakan *Microsoft Office Picture Manager*. Untuk selanjutnya dilakukan perubahan aras keabuan pada citra dengan fungsi *default* MATLAB 7.0: `rgb2gray(citra_wajah.png)`. Pada tahap ini citra gambar yang akan diproses terdiri tiga orang dengan jenis citra: 1. citra wajah jarak dekat 2. citra wajah jarak dekat piksel 800 x 600 3. citra wajah jarak dekat piksel 640 x 480 4. citra wajah jarak sedang 2. citra wajah jarak sedang piksel 800 x 600 3. citra wajah jarak sedang piksel 640 x 480.

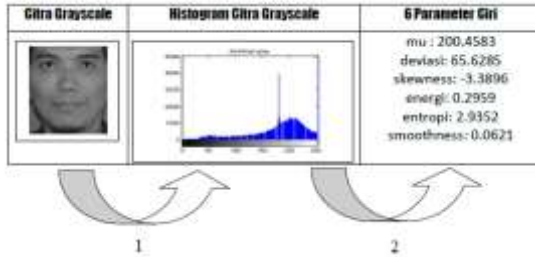


Gambar 3. Preprosesing Citra Wajah : 1. Proses *cropping* dengan *Microsoft Office Picture Manager*; 2. Proses *Grayscale* citra dengan fungsi *default* MATLAB: `rgb2gray(citra.png)`

B. Tahap ekstraksi citra wajah

Ekstraksi citra wajah pada penelitian ini menggunakan ekstraksi fitur tekstur yang berbasis histogram. Ekstraksi fitur tekstur akan menentukan ciri citra wajah berdasarkan 6 parameter yaitu: 1. rerata intensitas, 2. rerata kontras, 3. *skewness*, 4. energi, 5. entropi, 6. *smoothness*. Proses ekstraksi fitur tekstur citra wajah ditunjukkan pada Gambar 4. Seperti tampak pada Gambar 4, proses ekstraksi fitur citra wajah dilakukan pada citra *grayscale*. Selanjutnya data ekstraksi ciri

disimpan sebagai *database* citra wajah.



Gambar 4. Proses Ekstraksi Ciri Berbasis Histogram: 1. Proses Histeq menggunakan perintah: *histeq* (citra *grayscale*); 2. Ekstraksi Fitur Tekstur berbasis Histogram dengan *MATLAB*: *function* [Stat] = *stattektur*(F)

C. Tahap pengenalan dengan *JST Backpropagation*

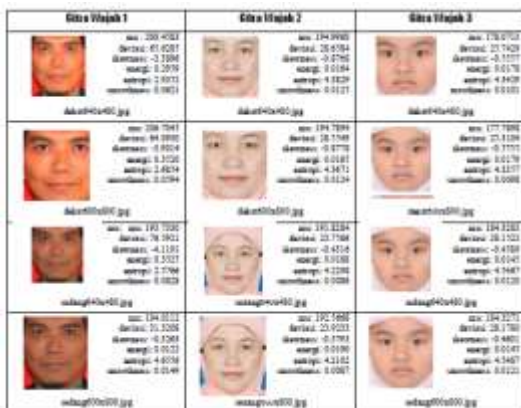
Dua fase dilakukan pada tahap pengenalan wajah dengan *JST Backpropagation* yaitu pelatihan dan pengujian. Pelatihan menggunakan sebagian data hasil ekstraksi fitur tekstur, sementara pengujian dilakukan dengan data citra wajah lainnya.



Gambar 5. Dua fase pengenalan wajah dengan *JST Backpropagation*

V. HASIL DAN PEMBAHASAN

Pada penelitian ini citra wajah terdiri atas tiga wajah berbeda yang diambil pada jarak sedang (2-5m) dan pendek (<2m) dengan variasi ukuran piksel 640 x 480 piksel dan 600 x 800 piksel Hasil ekstraksi citra wajah berbasis histogram ditunjukkan pada Gambar 6.



Gambar 6. Hasil ekstraksi citra wajah berbasis histogram

Seperti ditunjukkan pada Gambar 6., data 6 parameter hasil ekstraksi fitur citra wajah diubah menjadi data biner sebagai database masukan *JST*.

Perancangan arsitektur *JST Backpropagation* sangat tergantung pada masalah yang akan diselesaikan. Untuk

himpunan masukan berdimensi besar atau jumlah kelas keluaran yang diinginkan besar, maka diperlukan jumlah node pada lapisan tersembunyi yang lebih banyak. Atau diperlukan lebih dari satu lapisan tersembunyi, tetapi tentu saja ada batas optimumnya untuk kedua parameter tersebut.

Pengujian pertama (Tabel 2.) dilakukan terhadap jumlah lapisan tersembunyi yaitu dengan membuat variasi jumlah unit pada lapisan tersembunyi :

TABEL 2.
PENGUJIAN JUMLAH UNIT PADA LAPISAN TERSEMBUNYI

| Jumlah Lapisan Tersembunyi | Epoth | Dikenali | Tidak Dikenali | (%) |
|----------------------------------|-------|----------|-----------------------|------|
| 3 | 85 | 18 | 2 | 90 % |
| 4 | 256 | 17 | 3 | 85% |
| 5 | 98 | 17 | 3 | 85% |
| 6 | 290 | 19 | 1 | 95 % |
| Input : 225 | | | Maksimum epoch : 2500 | |
| Output : 10 | | | Target error : 0.0001 | |
| Learning rate (α) : 0.9 | | | | |

Pada pengujian selanjutnya digunakan jumlah unit pada lapisan tersembunyi 6, jumlah unit input 225, jumlah unit output 10, maksimum epoch 2500, target error 0.0001 dan dengan nilai *learning rate* (α) yang diubah-ubah. Pada pengujian ini diuji tiap citra wajah sebanyak 20 kali meng-*input*-kan pola wajah yang akan dikenali, berikut hasilnya ditunjukkan pada Tabel 3.

TABEL 3.
HASIL PENGUJIAN *JST* DENGAN VARIASI LEARNING RATE (α)

| Citra Wajah | $\alpha=0.5$ (%) | $\alpha=0.6$ (%) | $\alpha=0.7$ (%) | $\alpha=0.8$ (%) | $\alpha=0.9$ (%) |
|-------------|------------------|------------------|------------------|------------------|------------------|
| 1 | 90% | 89% | 92% | 93% | 94% |
| 2 | 87% | 91% | 92% | 92% | 94% |
| 3 | 95% | 90% | 93% | 93% | 94% |
| 4 | 91% | 93% | 94% | 93% | 95% |
| 5 | 88% | 91% | 90% | 93% | 93% |
| 6 | 89% | 90% | 90% | 92% | 95% |
| 7 | 92% | 91% | 90% | 92% | 95% |
| 8 | 91% | 92% | 91% | 94% | 95% |
| 9 | 92% | 93% | 91% | 90% | 96% |
| 10 | 89% | 94% | 92% | 90% | 96% |
| 11 | 89% | 90% | 91% | 90% | 96% |
| 12 | 91% | 88% | 90% | 90% | 96% |
| 13 | 90% | 87% | 90% | 90% | 96% |
| 14 | 92% | 90% | 90% | 92% | 97% |
| 15 | 91% | 89% | 91% | 92% | 95% |
| 16 | 94% | 90% | 91% | 92% | 95% |
| 17 | 93% | 91% | 91% | 92% | 95% |
| 18 | 90% | 90% | 90% | 93% | 93% |
| Rerata | 91% | 91% | 91% | 92% | 95% |

VI. KESIMPULAN

Dari hasil pembahasan dikaitkan dengan permasalahan dan tujuan dilakukannya penelitian ini maka dapat diambil kesimpulan:

1. Pola citra wajah dengan preprosesing menggunakan

metode ekstraksi fitur berbasis histogram dapat dikenali dengan JST menggunakan jaringan *Backpropagation*.

2. Pada pengujian JST kondisi terbaik dihasilkan jumlah unit pada lapisan tersembunyi 6, jumlah unit input 225, jumlah unit output 10, maksimum epoch 2500, target *error* 0.001 dan *learning rate*=0.9 dengan rerata dikenali sebesar 95%

REFERENCES

- [1] Syafii Nur L., *Implementasi Jaringan Saraf Tiruan Backpropagation Pada Aplikasi Pengenalan Wajah Dengan Jarak Yang Berbeda Menggunakan MATLAB 7.0*, Skripsi, Universitas Gunadarma, Depok, 2009
- [2] Al Fatta Hanif., *Rekayasa Sistem Pengenalan Wajah*, Penerbit Andi: Yogyakarta, 2009
- [3] Turk., Pentland, *Face Recognition Using Eigenfaces*, New Jersey: Prentice-Hall, Inc., 1991.
- [4] R. L. Hsu., Abdel Mottaleb., and A. K. Jain, *Face Detection in Color Images*, Proceedings of the IEEE International Conference on Image Processing, 2001.
- [5] Rein-Lien Hsu., Mohamed Abdel-Mottaleb., and Anil K. Jain, *Face Detection In Color Images*, IEEE TRANS. PAMI, Vol. 24, no. 5, pp. 696-706, May-2002.
- [6] Schneiderman., T. Kanade, *A Statistical Method For 3D Object Detection Applied To Faces And Cars*, In International Conference on Computer Vision, 2000.
- [7] Dewi Agushinta R., *Ekstraksi Fitur Dengan Segmentasi Wajah Untuk Identifikasi Pada Sistem Pengenalan Wajah*, Disertasi, Universitas Gunadarma, Depok, 2007.
- [8] Abdul Kadir., Susanto Adi, *Teori dan Aplikasi Pengolahan Citra*, Penerbit ANDI, Yogyakarta, 2012
- [9] Darma Putra, *Pengolahan Citra Digital*, Penerbit Andi, Yogyakarta, 2013.
- [10] Nugroho S., Agus H., *Penerapan Jaringan Syaraf Tiruan Untuk Mendeteksi Posisi Wajah Manusia Pada Citra Digital*, Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005) ISBN: 979-756-061-6, Yogyakarta, 2005.
- [11] Jong Jek Siang, *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, Penerbit ANDI, Yogyakarta, 2005
- [12] Darma Putra, *Sistem Biometrika: Konsep Dasar, Teknik Analisis Citra, dan Tahapan Membangun Aplikasi Sistem Biometrika*, Yogyakarta : Penerbit ANDI, 2009
- [13] Stan Z. Li., Anil K. Jain, *Handbook of Face Recognition*, New York: Springer, 2005.