

SISTEM DETEKSI PENYAKIT Pengeroposan Tulang Dengan Metode Jaringan Syaraf Tiruan *Backpropagation* Dan Representasi Ciri Dalam Ruang *Eigen*

Is Mardianto¹; Dian Pratiwi²

^{1,2}Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Trisakti,
Jln. Kyai Tapa No.1, Grogol, Jakarta Barat 11440
georadar@yahoo.com

ABSTRACT

There are various ways to detect osteoporosis disease (bone loss). One of them is by observing the osteoporosis image through rontgen picture or X-ray. Then, it is analyzed manually by Rheumatology experts. Article present the creation of a system which could detect osteoporosis disease on human, by implementing the Rheumatology principles. The main areas identified were between wrist and hand fingers. The working system in this software included 3 important processing, which were process of basic image processing, pixel reduction process, pixel reduction, and artificial neural networks. Initially, the color of digital X-ray image (30 x 30 pixels) was converted from RGB to grayscale. Then, it was threshold and its gray level value was taken. These values then were normalized to an interval [0.1, 0.9], then reduced using a PCA (Principal Component Analysis) method. The results were used as input on the process of Backpropagation artificial neural networks to detect the disease analysis of X-ray being inputted. It can be concluded that from the testing result, with a learning rate of 0.7 and momentum of 0.4, this system had a success rate of 73 to 100 percent for the non-learning data testing, and 100 percent for learning data.

Keywords: osteoporosis, image processing, PCA, artificial neural networks

ABSTRAK

Ada berbagai cara untuk mendeteksi penyakit osteoporosis (pengeroposan tulang), salah satunya adalah dengan melihat gambaran osteoporosis melalui foto rontgen atau X-ray. Kemudian, dianalisis secara manual oleh pakar Rheumatologi. Artikel menunjukkan pengembangan suatu sistem yang dapat mendeteksi penyakit osteoporosis pada manusia dengan menerapkan prinsip Rheumatologi. Daerah utama yang diidentifikasi adalah antara pergelangan tangan hingga jari tangan. Sistem kerja di dalam perangkat lunak ini meliputi 3 pemrosesan penting, yaitu proses pengolahan citra dasar, proses reduksi piksel, dan proses jaringan syaraf tiruan. Awalnya, citra X-ray digital (30 x 30 piksel) dilakukan pengkonversian warna dari RGB ke grayscale. Kemudian, di-thresholding dan diambil nilai gray level-nya. Nilai-nilai tersebut selanjutnya dinormalisasi ke interval [0.1, 0.9], lalu direduksi menggunakan metode PCA (Principal Component Analysis). Hasilnya dijadikan input pada proses jaringan syaraf tiruan Backpropagation untuk mengetahui analisis penyakit dari X-ray yang dimasukkan. Disimpulkan dari hasil pengujian dengan learning rate sebesar 0.7 dan momentum sebesar 0.4, sistem ini memiliki tingkat keberhasilan 73 hingga 100 persen untuk uji data non-learning, dan 100 persen untuk data learning.

Kata kunci: osteoporosis, pengolahan citra, PCA, jaringan syaraf tiruan

PENDAHULUAN

Latar Belakang

Dewasa ini, kebutuhan akan piranti pengenalan/pendeteksi penyakit yang handal sangat dibutuhkan. Pengembangan teknologi pengenalan berupa sistem kecerdasan buatan menjadi sangat penting dan membantu karena selain mempermudah, juga mempercepat pekerjaan manusia. Salah satu ilmu yang mendukung teknologi tersebut adalah *artificial neural network* (jaringan syaraf tiruan), di mana dalam 10 tahun terakhir pengaplikasiannya telah banyak dikembangkan di berbagai bidang. Misalnya, dalam bidang kedokteran di Indonesia, yang sudah banyak menggunakan teknologi jaringan syaraf tiruan untuk memudahkan pekerjaan paramedis maupun dokter dalam mendeteksi setiap penyakit pasien. Beberapa alat berbasis teknologi jaringan syaraf tiruan yang telah digunakan di rumah-rumah sakit diantaranya

adalah CT. SCAN, MRI, Ultrasonografi (USG), Mesin Sinar-X, Mamografi, dan lain-lain.

Sejalan dengan perkembangan perangkat lunak dan perangkat keras, teknik analisis jaringan syaraf tiruan membutuhkan terobosan teoritis untuk mempercepat kinerja jaringan. Sebagian besar aplikasi yang telah dikembangkan menggunakan jaringan syaraf tiruan umpan maju, dengan pembobot dikoreksi yang dinamakan metode *Backpropagation*. Hal ini disebabkan metode *Backpropagation* termasuk ke dalam pelatihan *supervised* (terbimbing) dan cocok untuk berbagai aplikasi, serta hasil yang dikerjakan lebih optimal dibandingkan dengan metode jaringan syaraf tiruan lainnya. Oleh karena itu, penulis membuat sistem pendeteksi penyakit *osteoporosis* (pengeroposan tulang) pada tangan melalui media citra (*X-Ray*) digital, dengan menggunakan metode *Backpropagation*, di mana representasi cirinya dilakukan dalam ruang *eigen*.

Pada umumnya, untuk mendeteksi *osteoporosis* pada tubuh dapat dilihat dari gejala-gejala yang ditimbulkan, seperti

nyeri pada bagian tertentu, bengkak-bengkak, dan kram atau otot sulit digerakkan. Namun, hal ini tidaklah cukup untuk membuktikannya. Oleh karena itu, diperlukan pemeriksaan lebih lanjut lagi seperti pemeriksaan radiologi untuk menilai densitas massa tulang yang sensitif, dan hasilnya berupa gambar *X-ray* yang lalu dianalisis oleh pakar Rheumatologi.

Beberapa analisis yang digunakan pakar Rheumatologi maupun spesialis penyakit, sebagai acuan untuk menentukan *osteoporosis* dari suatu gambar *X-ray* yaitu dengan melihat ada atau tidaknya pembengkakan, pengapuran (penipisan massa tulang), perubahan bentuk tulang yang signifikan (membengkok/memendek), atau bahkan terjadi *fracture* (patah tulang). Sistem analisis inilah yang kemudian penulis terapkan pada perangkat lunak deteksi *osteoporosis*.

Permasalahan

Permasalahan yang ada pada perancangan sistem berupa pertanyaan seperti: bagaimana mengumpulkan sejumlah citra *X-ray* yang hanya meliputi daerah tangan; bagaimana meng-*crop* citra tanpa harus kehilangan informasi penting di dalamnya dan tidak mengubah bentuk; bagaimana mendapatkan karakteristik *osteoporosis* pada tangan dengan metode PCA dan prinsip *eigen*; serta bagaimana menentukan nilai *threshold*, jumlah unit *hidden layer*, nilai momentum, dan nilai laju pembelajaran yang tepat, sehingga didapat hasil pengenalan yang optimal dengan waktu iterasi seminimal mungkin.

Tujuan dan Manfaat

Program aplikasi yang dibuat ini memiliki tujuan, yaitu: mempermudah orang-orang untuk melakukan pendeteksian gangguan *osteoporosis* dengan cara yang mudah dan murah karena hanya memerlukan gambar *X-ray* tangan dari radiologi; memperoleh hasil prediksi yang cepat dan optimal dalam mengenali penyakit *osteoporosis*; serta memperoleh persentase keberhasilan jaringan dalam mengenali penyakit, sehingga dapat diketahui tingkat kesalahannya.

Ruang Lingkup

Ruang lingkup di dalam penelitian ini dibagi menjadi beberapa bagian, yaitu: sistem, citra, dan proses.

Sistem

Secara garis besar, aplikasi deteksi *osteoporosis* ini tersusun atas 3 sistem, yaitu sistem pengolahan citra, representasi ciri, dan jaringan syaraf tiruan. Sistem pengolahan citra digunakan untuk mengubah kekontrasan dan *format* citra asli, sehingga menjadi data yang dapat di-*input* untuk pemrosesan berikutnya. Sistem representasi ciri, dalam hal ini dimaksudkan untuk mengambil sejumlah data (mereduksi pixel) yang dianggap mewakili karakteristik atau ciri-ciri *osteoporosis* yang sebenarnya dengan menggunakan teknik PCA (*Principal Component Analysis*), serta prinsip ruang vektor dan nilai *eigen*. Sistem jaringan syaraf tiruan digunakan untuk menganalisis dan mendeteksi adanya *osteoporosis*, sebagaimana cara kerja otak para pakar Rheumatologi dalam menganalisis.

Citra

Citra yang digunakan dalam program aplikasi ini mempunyai batasan-batasan yaitu: citra masukan merupakan citra *X-Ray* yang telah di-*scanning*; ukuran citra telah di-*crop* menjadi 30 x 30 pixel; *format* citra merupakan *format* yang telah diubah menjadi *.PCX; daerah citra yang diambil dan dideteksi hanya berkisar pergelangan tangan hingga jari-jari tangan; serta prediksi pada citra hanya terdiri dari 2 kategori,

yaitu citra normal (bernilai '0') atau citra *osteoporosis* (bernilai '1').

Proses

Ruang lingkup proses keseluruhan pada penelitian ini, yaitu: membuka file citra tangan yang berformat *.PCX; mengubah warna citra ke bentuk *grayscale*; memisahkan daerah objek dengan *background* menggunakan *thresholding*; mengambil nilai intensitas keabuan (*greylevel*) untuk setiap pixel citra; normalisasi nilai *greylevel*; mereduksi jumlah pixel di dalam ruang *eigen*; menjalankan proses jaringan syaraf tiruan *Backpropagation*; *training* jaringan hingga *error*, yang didapat lebih kecil dari batas toleransi atau telah melebihi jumlah iterasi yang dimasukkan; menyimpan bobot-bobot akhir yang bersesuaian; serta *testing* atau *mapping* jaringan, dengan memasukkan pola baru ataupun pola yang telah dilatih.

Landasan Teori

Pengolahan Citra

Pengolahan citra merupakan suatu sistem di mana proses dilakukan dengan masukan berupa citra, dan menghasilkan citra pula dengan kualitas yang lebih baik. Pada penelitian ini, citra yang digunakan adalah citra *X-ray* bergambar rangka tulang tangan, di mana setelah melalui proses *scan* akan menghasilkan citra digital (*.PCX). Citra digital adalah citra yang didefinisikan sebagai fungsi $f(x,y)$ di mana x menyatakan nomor baris, y menyatakan nomor kolom, dan f menyatakan nilai derajat keabuan (*greylevel*) pada citra.

RGB

Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra, warna direpresentasikan dengan nilai hexadesimal 0x00000000 sampai dengan 0x00ffffff.

Grayscale

Dalam pengolahan citra, perubahan warna citra menjadi citra *grayscale* digunakan untuk menyederhanakan model citra. Citra berwarna memiliki 3 komposisi warna, yaitu *red* (R), *green* (G), dan *blue* (B). Untuk mendapatkan citra *grayscale*, maka 3 komponen tersebut dirata-rata. Dalam citra ini, tidak ada lagi warna; yang ada hanya derajat keabuan.

Thresholding

Selain digunakan untuk mengatur derajat keabuan yang ada pada citra, *thresholding* juga untuk memisahkan bagian citra yang sesuai dengan objek (*foreground*) dan latar belakangnya (*background*), serta mengkonversi data citra menjadi data biner (binerisasi), dengan tujuan agar proses selanjutnya menjadi mudah.

Setiap citra meskipun mengandung objek yang sama, tentunya memiliki karakteristik yang berbeda dalam sifat-sifat pencahayaannya. Hal ini menyebabkan sulitnya menentukan nilai *threshold* yang cocok untuk diterapkan ke semua kondisi citra. Oleh karena setiap citra mempunyai nilai *threshold* masing-masing, maka dapat dilakukan pembelajaran berupa pengetahuan sifat-sifat dari masing-masing citra yang akan diproses sebelum menentukan nilai *threshold* yang cocok. Dengan demikian, setelah melalui *trial and error* akan ditemukan nilai *threshold*, yang kurang lebih cocok diterapkan untuk semua citra.

Normalisasi

Normalisasi digunakan untuk mengecilkan *range* data gambar, yang dalam hal ini berupa *range* piksel. Normalisasi data dapat dilakukan dengan berbagai cara, diantaranya dengan cara mentransformasi ke dalam interval tertentu. Jika fungsi aktivasi yang akan digunakan berupa fungsi aktivasi *sigmoid biner*, data harus ditransformasi atau normalisasi ke interval $[0, 1]$. Tapi, akan lebih baik jika ditransformasikan ke interval yang lebih kecil, yaitu interval $[0.1, 0.9]$. Hal ini dikarenakan fungsi *sigmoid biner* merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1.

$$x' = \frac{0.8(x - a)}{b - a} + 0.1 \dots\dots\dots (2.1)$$

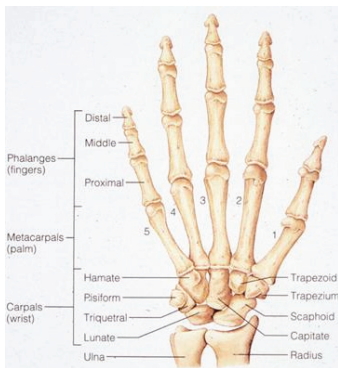
Di mana a adalah data minimum dan b adalah data maksimum. Jika fungsi aktivasi yang digunakan adalah fungsi *sigmoid bipolar*, data dinormalisasi ke interval $[-1, 1]$ terlebih dulu.

Rheumatologi

Tulang merupakan bagian terpenting dari tubuh manusia dan makhluk hidup lainnya. Tulang terbentuk dari matrik dan garam-garam non organik berupa kalsium, fosfat, dan dilapisi membran atau ligamen berserat di sekitarnya untuk melicinkan pergerakan sendi. Tubuh manusia, normalnya tersusun atas 206 tulang yang memberikan struktur dan bentuk. Tulang-tulang ini berfungsi untuk melindungi organ-organ dalam yang vital dan rawan, sebagai tempat penyimpanan mineral-mineral, dan tempat pembentukan sel-sel darah.

Rheumatologi dan Landasan Ilmiah Anatomi Tangan

Rheumatologi adalah salah satu cabang ilmu penyakit dalam yang berkonsentrasi pada diagnosis dan terapi penyakit rematik. Salah satu metode diagnosis yang digunakan para ahli Rheumatologi adalah melalui sinar-X atau radiograf untuk menghasilkan gambar medikal (*X-ray* atau *rontgen*). Tulang pada bagian sendi pergelangan hingga jari-jari tangan merupakan salah satu daerah yang dapat dijadikan gambaran kondisi ada atau tidaknya tanda-tanda penyakit *osteoporosis*, karena seringkali rasa nyeri dan pembengkakan yang terjadi pada tangan (Gambar 1) terletak pada sekitar daerah tersebut.

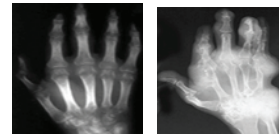


Gambar 1 Struktur Rangka Tangan Manusia

Osteoporosis

Osteoporosis adalah penyakit metabolisme tulang, di mana tulang-tulang menjadi rapuh dan mudah patah akibat pengurangan massa dan kemunduran mikroarsitektur tulang. Pemeriksaan radiologi digunakan untuk menilai densitas

massa tulang yang sudah mengalami pengurangan sebesar 50 persen. Gambaran radiologi yang khas pada *osteoporosis* adalah penipisan korteks dan daerah trabekular (tulang bagian dalam dan berongga). Pada tulang tangan (Gambar 2), daerah *osteoporosis* banyak ditemukan di sekitar *falang medial* jari II dan III, dan juga pada tulang *metacarpal*.



Gambar 2 X-ray Tangan yang Positif Menderita Osteoporosis

Jaringan Syaraf Tiruan

Jaringan Saraf Tiruan (JST) didefinisikan sebagai suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan saraf manusia (Gambar 3). Jaringan saraf tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia.

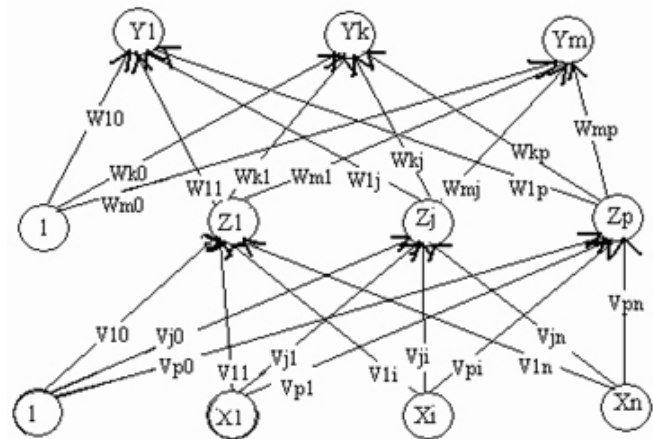


Gambar 3 Struktur Neuron Jaringan Syaraf Tiruan

Secara umum, proses pembelajaran JST dapat dikategorikan menjadi 2 jenis proses, yaitu: *supervised training* (pelatihan terbimbing), yaitu tiap pola *input* memiliki pola target; dan *unsupervised training* (pelatihan tidak terbimbing). Pada pelatihan ini, vektor target tidak dibutuhkan untuk keluarannya, sehingga tidak ada perbandingan untuk menentukan respon yang ideal.

Jaringan Syaraf Tiruan Backpropagation

Metode *Backpropagation* atau disebut juga propagasi balik, pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart bersama McClelland, di mana algoritma ini termasuk metode pelatihan terbimbing (*supervised*) dan didesain untuk operasi pada jaringan syaraf tiruan *feed forward multi-layer*. Arsitektur dari JST *Backpropagation* terdiri dari 3 bagian, yang meliputi masukan, lapis tersembunyi, dan keluaran (Gambar 4).



Gambar 4 Struktur JST Backpropagation [7]

Pemilihan Bobot dan Bias Awal

Pada tahun 1990, Nguyen dan Widrow mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi, sehingga menghasilkan iterasi lebih cepat. Algoritma inisialisasi *Nguyen Widrow* adalah sebagai berikut. Pertama, inisialisasi semua bobot (v_{ji} (lama)) dengan bilangan acak dalam interval $[-0.5, 0.5]$. Kedua, hitung $\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jn}^2}$ (2.2). Ketiga, bobot yang dipakai sebagai inisialisasi $v_{ji} = [\beta \cdot v_{ji} \text{ (lama)}] / \|v_j\|$ (2.3). Keempat, bias yang dipakai sebagai inisialisasi $v_{j0} = \text{bilangan acak antara } -\beta \text{ dan } \beta$; di mana n adalah jumlah unit masukan, p adalah jumlah unit tersembunyi, dan β adalah faktor skala sebesar $0.7^{-\sqrt{p}}$ (2.4)

Fungsi Aktivasi

Dalam *Backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah, dan merupakan fungsi yang tidak turun. Salah satunya adalah fungsi *sigmoid biner* dengan *range* $[0, 1]$:

$$f(x) = \frac{1}{1 + e^{-x}} \text{(2.5)}$$

Pelatihan Standar *Backpropagation*

Algoritma pelatihan *Backpropagation* terdiri dari 2 tahapan, *feed forward* dan *Backpropagation* dari galatnya, yaitu: langkah 0, pemberian inisialisasi penimbang (diberi nilai kecil secara acak); langkah 1, ulangi langkah 2 hingga 9 sampai kondisi akhir iterasi dipenuhi; dan langkah 2, untuk masing-masing pasangan data pelatihan lakukan langkah 3 hingga 8.

Umpan maju (*feed forward*), terdiri dari: langkah 3, masing-masing unit masukan ($X_i, i = 1, \dots, n$) menerima sinyal masukan X_i dan sinyal tersebut disebarkan ke unit-unit bagian berikutnya (unit-unit lapis tersembunyi); langkah 4, masing-masing unit dilapis tersembunyi, dikalikan dengan penimbang dan dijumlahkan, serta ditambah dengan biasnya; dirumuskan dengan $Z_{in_j} = V_{j0} + \sum X_i V_{ji}$ (2.6). Kemudian dihitung sesuai dengan fungsi pengaktif yang digunakan adalah: $Z_j = f(Z_{in_j})$ (2.7). Bila yang digunakan adalah fungsi sigmoid maka bentuk fungsi tersebut adalah:

$$Z_j = \frac{1}{1 + \exp(-z_{in_j})} \text{(2.8). Sinyal keluaran dari fungsi}$$

pengaktif tersebut dikirim ke semua unit di lapis keluaran; berikutnya adalah langkah 5, masing-masing unit keluaran ($Y_k, k = 1, 2, 3 \dots m$) dikalikan dengan penimbang dan dijumlahkan serta ditambah dengan biasnya. Rumusnya adalah $Y_{in_k} = W_{k0} + \sum Z_j W_{kj}$ (2.9). Kemudian dihitung kembali sesuai dengan fungsi pengaktif sebagai $y_k = f(y_{in_k})$ (2.10)

Backpropagasi (Backpropagation) dan Galatnya, terdiri dari: langkah 6, masing-masing unit keluaran ($Y_k, k = 1, \dots, m$) menerima pola target sesuai dengan pola masukan saat pelatihan/training dan dihitung galatnya dengan rumus $\delta_k = (t_k - y_k) f'(y_{in_k})$ (2.11). Karena $f'(y_{in_k}) = y_k$ menggunakan fungsi sigmoid, maka $f'(y_{in_k}) = f(y_{in_k})(1 - f(y_{in_k})) = y_k(1 - y_k)$ (2.12). Menghitung perbaikan penimbang (kemudian untuk memperbaiki W_{kj}) dengan rumus $\Delta W_{kj} = \alpha \cdot \delta_k \cdot Z_j$ (2.13). Sedangkan untuk menghitung perbaikan korelasi dengan rumus $\Delta W_{k0} = \alpha \cdot \delta_k$ (2.14). Dan menggunakan nilai delta (δ_k) pada semua unit lapis sebelumnya; berikutnya adalah langkah 7, masing-masing penimbang yang menghubungkan unit-unit lapis keluaran dengan unit-unit pada lapis tersembunyi ($Z_j, j = 1, \dots, p$) dikalikan dengan delta (δ_k) dan dijumlahkan sebagai masukan ke unit-unit lapis berikutnya dengan rumus $\delta_{in_j} = \sum \delta_k W_{kj}$ (2.15). Selanjutnya dikalikan dengan turunan dari fungsi pengaktifnya untuk menghitung galat dengan rumus $\delta_j = \delta_{in_j} f'(y_{in_j})$ (2.16). Langkah selanjutnya

untuk menghitung perbaikan penimbang (digunakan untuk memperbaiki V_{ji}) dengan rumus $\Delta V_{ji} = \alpha \cdot \delta_j \cdot X_i$ (2.17). Kemudian menghitung perbaikan bias (untuk memperbaiki V_{j0}) dengan rumus $\Delta V_{j0} = \alpha \cdot \delta_j$ (2.18).

Memperbaiki penimbang dan bias, terdiri dari: langkah 8, masing-masing keluaran unit ($Y_k, k = 1, \dots, m$) diperbaiki bias dan penimbangnya ($j = 0, \dots, p$) dengan rumus $W_{kj}(\text{baru}) = W_{kj}(\text{lama}) + \alpha \cdot \delta_k \cdot Z_j$ (2.19). Atau apabila parameter momentum (μ) digunakan menjadi: $W_{kj}(\text{baru}) = W_{kj}(\text{lama}) + \alpha \cdot \delta_k \cdot Z_j + (\mu (W_{kj}(\text{lama}) - W_{kj}(\text{lama} - 1)))$ (2.20). Masing-masing unit tersembunyi ($Z_j, j = 1, \dots, p$) diperbaiki bias dan penimbangnya ($j = 1, \dots, n$) dengan rumus $V_{ji}(\text{baru}) = V_{ji}(\text{lama}) + \alpha \cdot \delta_j \cdot X_i$ (2.21). Atau apabila parameter momentum digunakan menjadi: $V_{ji}(\text{baru}) = V_{ji}(\text{lama}) + \alpha \cdot \delta_j \cdot X_i + (\mu (V_{ji}(\text{lama}) - V_{ji}(\text{lama} - 1)))$ (2.22); berikutnya adalah langkah 9, yakni dengan uji kondisi pemberhentian (akhir iterasi).

Parameter Pelatihan

Beberapa parameter JST yang digunakan yaitu: inisialisasi penimbang, jenis adaptasi penimbang, *learning rate*, momentum, dan penentuan jumlah lapis tersembunyi.

Proses Pelatihan

Untuk menentukan jumlah pelatihan, digunakan 2 macam *stopping* kriteria yaitu: berdasarkan jumlah pelatihan yang dilakukan (*epochs*), misalnya pelatihan akan dihentikan setelah dilakukan sejumlah pelatihan; serta berdasarkan Galat *Mean Square Error* (MSE), maka proses pelatihan akan terus dilakukan sampai *error*-nya menjadi lebih kecil dari batas toleransi. Perhitungan MSE yakni:

$$MSE = \frac{\sum (\text{target} - \text{output})^2}{\text{jumlah pola}} \text{(2.23)}$$

Proses Mapping

Proses *mapping* dilakukan dengan menjalankan prosedur *feed forward* dengan penimbang yang telah disimpan sebelumnya (dilakukan *load* data penimbang dari file). Hasil keluaran dari JST merupakan hasil dari proses *mapping*. Sehingga dengan proses *mapping* ini, dapat dilakukan proses identifikasi, sekaligus didapatkan diagnosa dari hasil *X-ray*.

Nilai Eigen dan Vektor Eigen

Untuk mencari nilai *eigen* matriks A yang berukuran $n \times n$, maka kita menuliskan kembali $Ax = \lambda x$ sebagai: $Ax = \lambda \cdot I \cdot x$ (2.24).

Secara ekuivalen, vektor *eigen* yang bersesuaian dengan λ adalah vektor tak nol dalam ruang pemecahan dari $(\lambda I - A)x = 0$. Dan ruang pemecahan ini dinamakan sebagai ruang *eigen* (*eigen space*) dari A yang bersesuaian dengan λ .

Teknik *Principal Component Analysis* (PCA)

Metode PCA atau *Principal Component Analysis* pertama kali diajukan pada 1933 oleh Hottelling dengan tujuan untuk mengurangi dimensi sebuah ruang yang direpresentasikan oleh variabel statistik x_1, x_2, \dots, x_n .

Langkah-langkah untuk mereduksi dimensi pada metode PCA yaitu sebagai berikut. Pertama, jika terdapat himpunan p buah, citra pelatihan yang dapat direpresentasikan ke dalam $[x_1, x_2, \dots, x_p]$, dengan dimensi tiap citra adalah q (baris \times kolom), maka dapat dibentuk matriks data pelatihan Y dengan orientasi vektor kolom sebagai berikut:

$$Y = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \dots & \dots & \dots & \dots \\ x_{q1} & x_{q2} & \dots & x_{qp} \end{bmatrix} \text{ (2.25)}$$

Dengan x_{ij} menunjukkan nilai intensitas dari piksel citra pelatihan ke- j variabel (dimensi) ke- i .

Kedua, mencari rata-rata vektor citra dengan rumus

$$\mu_i = \frac{1}{p} \sum Z_{ip} \dots (2.26)$$
, di mana Z_{ip} adalah total piksel citra ke- i dari citra pelatihan ke- p . Sehingga dapat dibentuk sebuah citra matriks rata-rata $\mu = [\mu_1 \mu_2 \dots \mu_q]$. Ketiga, kemudian hitung selisih vektor citra dengan rata-rata vektor citra dengan rumus $A = Y - \mu \dots (2.27)$. Keempat, dari matriks A diatas dapat dihitung matriks total scatter berupa matriks kovarian (real, simetrik): $S = A.A^T$ (berordo $q \times q$)..... (2.28). Kelima, cari *eigen* vektor V dan nilai *eigen* D dari matriks S dan urutkan *eigen* vektor berdasarkan nilai *eigen* yang terbesar. Keenam, tentukan dimensi ruang ciri dengan rumus

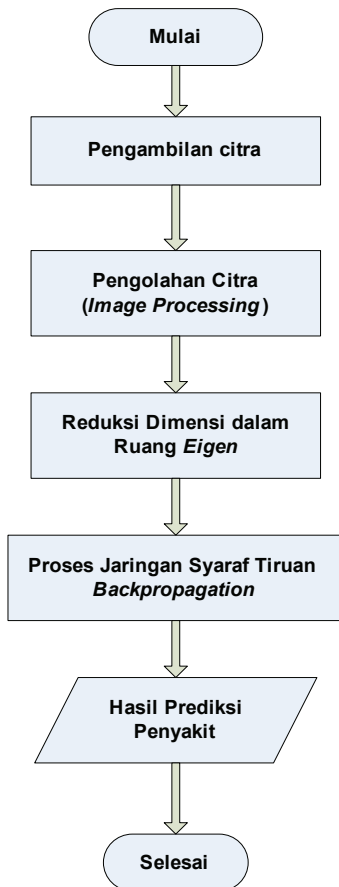
$$m = \min q \left[\begin{array}{l} \sum_{i=1}^m D_i > \theta \\ \sum_{i=m+1}^q D_i > \theta \end{array} \right] \dots (2.29).$$

eigen menjadi vektor *eigen* V' (*eigenspace*) sesuai dengan nilai m . Kedelapan, transformasi vektor citra menjadi vektor ciri dengan rumus $Y' = V^T.A$ (2.30), di mana kolom untuk citra, dan baris untuk piksel-piksel.

Analisis dan Perancangan Program

Gambar 5 merupakan gambar diagram *Use-Case* sistem deteksi penyakit pengeroposan tulang. Sedangkan pada Gambar 6, dapat dilihat gambar diagram alir perangkat lunak.

Diagram Alir Perangkat Lunak



Gambar 6 Diagram Alir Perangkat Lunak

Konversi ke Grayscale

Algoritma dari pengkonversian RGB ke *grayscale*



Gambar 5 Diagram *Use-Case* Sistem Deteksi Penyakit Pengeroposan Tulang

adalah sebagai berikut:

```

Picture.Height = tinggi citra
Picture.Width = lebar citra
For j = 1 to picture.Height
  For i = 1 to picture.Width
    r = Picture.Pixel[i,j]
    g = Picture.Pixel[i,j]
    b = Picture.Pixel[i,j]
    //mencari nilai grayscale
    gray = (r + g + b) / 3
    Picture.Pixel[i,j] = gray
  Next i
Next j
  
```

Thresholding

Algoritma dari *thresholding* yaitu sebagai berikut:

```

picture.Height = tinggi citra
picture.Width = lebar citra
threshold = 80 //misalnya nilai = 80
For j = 1 to picture.Height
  For i = 1 to picture.Width
    gray = Picture.Pixel[i,j]
    //seleksi
    If gray > threshold Then
      gray = gray //grayscale tetap
    Else
      gray = 0 //grayscale bernilai nol
    End If
    Picture.Pixel[i,j] = gray
  Next i
Next j
  
```

Normalisasi Data

Normalisasi atau transformasi data pada program ini dilakukan ke interval $[0.1, 0.9]$ yang dalam perhitungan mendekati interval sebenarnya yaitu $[0, 1]$.

Algoritmanya sebagai berikut :

```

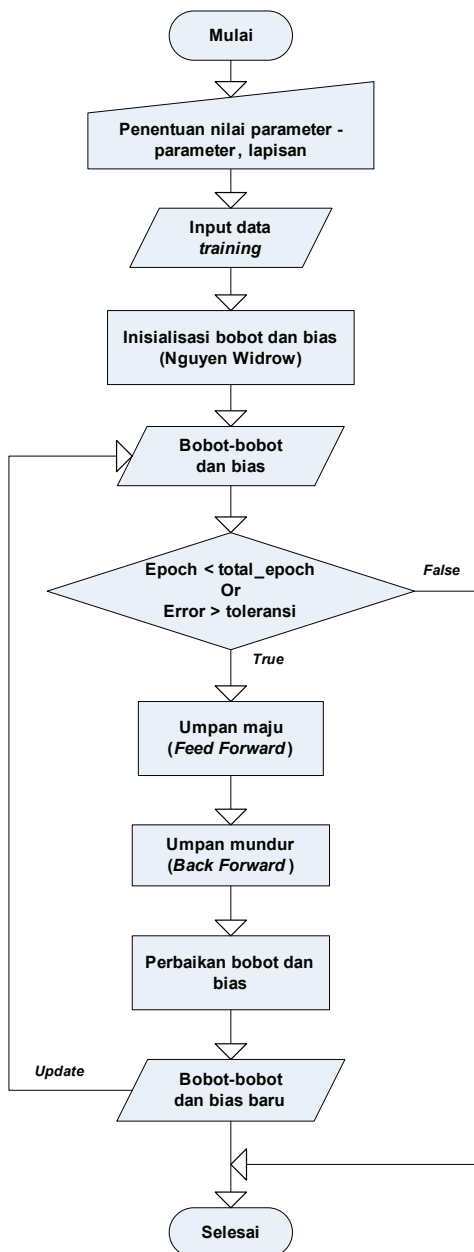
picture.Height = tinggi citra
picture.Width = lebar citra
For j = 1 to picture.Height
For i = 1 to picture.Width
    grayscale = Picture.Pixel[i,j]
    //Normalisasi
    Trans = ((0.8 * (grayscale - 0)) / (255 - 0)) + 0.1
    //menyimpan hasil ke array
    index[i,j] = Trans
Next i
Next j

```

Mereduksi Jumlah Piksel

Pada tahap ini, piksel-piksel setiap citra yang dikumpulkan dalam sebuah matriks ($N_{total_piksel} \times M_{data}$) akan mengalami pengurangan sejumlah baris atau sejumlah kolom tertentu, sesuai dengan nilai *eigen* yang paling rendah.

Di dalam program ini, pengaplikasian metode PCA digunakan bersamaan dengan *module* KLT yang telah ada untuk mengambil karakteristik dari *input* data-data yang diberikan dalam mencari nilai dan vektor *eigen*.



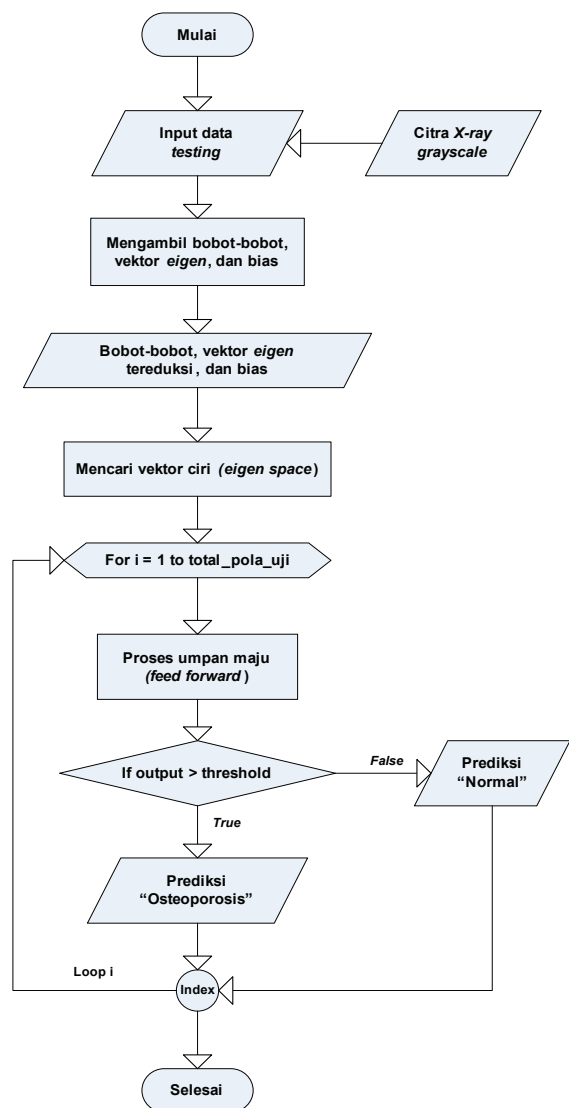
Gambar 7 Diagram Alir Proses *Training* JST *Backpropagation*

Proses *Training* Jaringan Syaraf Tiruan

Pada perancangan program sistem deteksi penyakit pengeroposan tulang ini, konfigurasi jaringan syaraf yang digunakan adalah: metodenya adalah *Backpropagation*; jumlah lapis sebanyak 3 lapis, terdiri atas satu lapis masukan, satu lapis tersembunyi, dan satu lapis keluaran; jumlah unit *output* sebanyak satu unit; fungsi aktivasinya adalah fungsi *sigmoid biner*; jenis *training* adalah pelatihan terbimbing (*supervised training*); *threshold* sebesar 0,55; momentum dari 0.1 sampai 0.9; learning rate dari 0.1 sampai 0.9; unit target sebanyak satu unit; *tolerance* sebesar 0.001 hingga 0.002; iterasi (*epoch*) sebesar 10000; metode inisialisasi bobot dan bias awalnya adalah dengan metode *Nguyen Widrow*; serta jenis adaptasi bobotnya adalah dengan adaptasi biasa (*incremental updating*). Diagram alir proses *training* JST *Backpropagation* dapat dilihat pada Gambar 7.

Proses *Mapping*

Pada proses *mapping* atau pengujian ini, algoritma *Backpropagation* yang dijalankan hanyalah pada bagian proses *feedforward* (umpan maju) saja, yaitu dengan memasukkan bobot-bobot yang sesuai dan telah disimpan di harddisk dari hasil proses *training* ke unit-unit *input* hingga menghasilkan keluaran. Untuk flowchart proses *testing* JST *Backpropagation* dapat dilihat pada Gambar 8.



Gambar 8 Flowchart Proses *Testing* JST *Backpropagation*

Implementasi dan Hasil Uji Coba Program

Pengujian citra pada proses *mapping*, dalam hal ini dilakukan dengan jumlah perbandingan 25 : 75; yaitu dari keempat puluh citra sampel, 10 sampel digunakan untuk proses *training* dan 30 sampel lainnya untuk proses *mapping*.

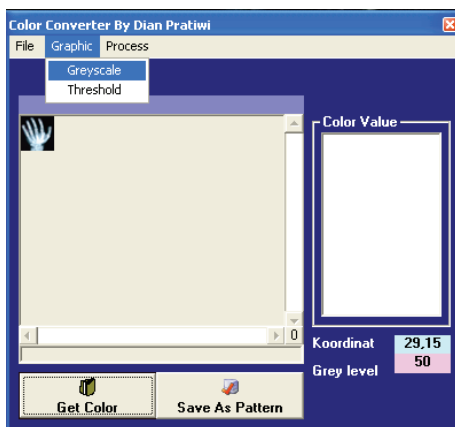
Pengambilan Citra Pengujian

Citra-citra *X-ray* yang akan diujicobakan pada proses *training* dan *testing* jaringan syaraf tiruan merupakan citra yang sebelumnya telah melalui proses *scanning* dan *cropping*. Hasilnya yaitu akan berupa citra *X-ray* digital berformat PCX berukuran 30x30 piksel.

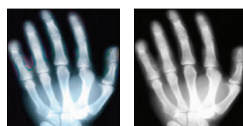
Pengujian Perangkat Lunak

Langkah-langkah yang dilakukan adalah: dengan mengkonversi warna ke *grayscale* (Gambar 9 dan 10); berikutnya adalah dengan *thresholding* (Tabel 1). Berdasarkan rata-rata uji coba (*trial and error*) yang telah dilakukan, nilai *threshold* yang paling sesuai untuk digunakan bernilai antara 60 hingga 80. Hal ini dikarenakan nilai *threshold* 60 hingga 80 tidak berlebihan dalam menghilangkan daerah berintensitas rendah dan sudah tidak meninggalkan daerah *background* pada citra. Dengan demikian, dari interval nilai *threshold* 60 hingga 80 dipilihlah nilai *threshold* tetap secara subjektif, yaitu 80 untuk keseluruhan *thresholding*. Jadi, jika citra dengan derajat keabuan kurang dari 80, maka derajat keabuan diubah menjadi 0 (*background*). Jika sebaliknya, derajat keabuan akan bernilai tetap (*foreground*).

Citra	I	II	III	IV
Normal				
50				
60				
70				
80				
90				
100				



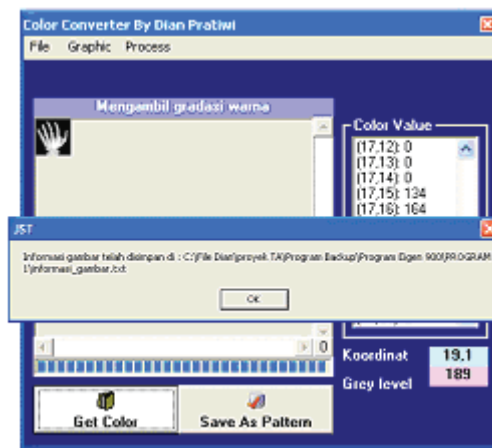
Gambar 9 Form *Color Converter*



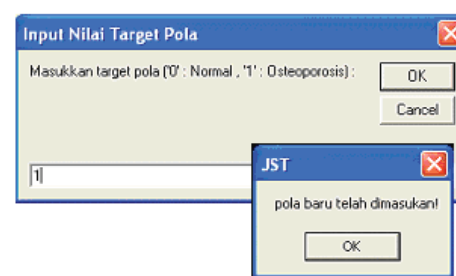
Gambar 10 Citra Hasil Proses *Grayscale*; Citra *X-ray* Sebelum Diproses (Kiri), Citra *X-ray* Setelah Diproses (Kanan).

Mengambil Piksel Citra

Langkah-langkah dari tahap ini yaitu dengan mengambil dan menampilkan *grayscale* tiap piksel (Gambar 11); serta menyimpan *grayscale* sebagai pola (Gambar 12). Pada tahap ini, *graylevel* yang telah ditampilkan akan dijadikan sebagai pola (satu pola satu citra) setelah melalui proses normalisasi, di mana setiap polanya akan memiliki sebuah target ('1' atau '0') yang dimasukkan.



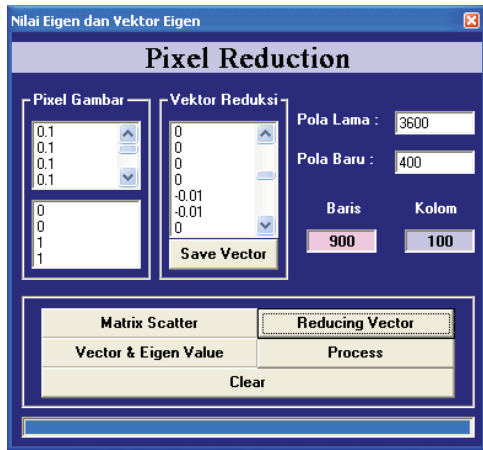
Gambar 11 Form *Color Converter*; *Grayscale* Ditampilkan dan Disimpan.



Gambar 12 Tampilan *Input Box* Nilai Target dan Konfirmasinya

Mereduksi Dimensi Citra

Pada proses ini, jumlah piksel yang terkandung pada citra direduksi (dikurangi) dengan metode PCA dan prinsip ruang *eigen*, di mana hasil dari reduksinya berupa vektor ciri yang akan dipergunakan sebagai *input* data proses *training* jaringan syaraf tiruan, dan vektor *eigen* tereduksi yang dapat disimpan sebagai file teks (*.txt) untuk digunakan dalam proses *mapping* (Gambar 13).



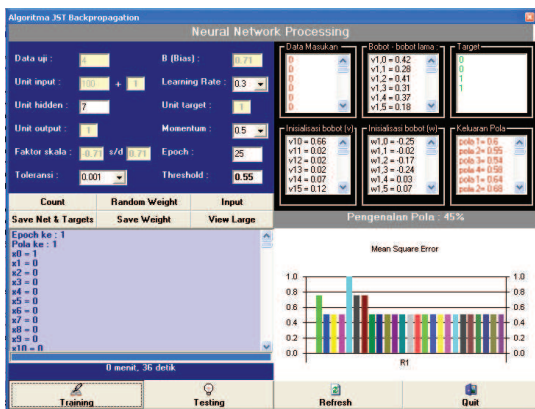
Gambar 13 Form Pixel Reduction

Memulai Training JST Backpropagation

Pada proses *training* JST di dalam program ini, nilai-nilai masukan diambil dari hasil proses reduksi yang berupa vektor ciri. Vektor ciri ini selanjutnya akan mengalami pelatihan jaringan berulang-ulang hingga jumlah iterasi (*epoch*) selesai dilakukan atau nilai galat (*error*) telah mencapai toleransi yang diberikan. Apabila hasil pengenalan pola pada proses ini telah mencapai 100 persen; yaitu saat galat lebih kecil dari toleransi, maka bobot-bobot dan bias dapat disimpan untuk kemudian digunakan dalam tahap *mapping* (Gambar 14).

Dari Tabel 2 dapat dilihat bahwa dengan memasukkan jumlah *input* yang semakin banyak, waktu iterasi akan semakin lama pula untuk mencapai galat minimum. Namun tidak pada percobaan 2 dan 3, yang ternyata memberikan hasil yang baik dengan lama iterasi cukup sedikit dibandingkan dengan nomor percobaan lainnya. Hal ini dapat dikarenakan kesesuaiannya nilai-nilai parameter yang diberikan terhadap banyaknya unit *input*, sehingga menghasilkan pengenalan pola yang lebih optimal. Oleh karena itu, penyusun menetapkan mengambil jumlah unit *input* training hanya sebanyak 500 dan 600 unit dari 10 citra (pola) untuk dibandingkan kembali dengan memasukkan nilai-nilai parameter yang berbeda, dan kemudian dilatih untuk mengetahui hasil pengenalan yang paling baik.

Berdasarkan Tabel 3 sampai 5 yang merupakan hasil pengujian, proses *training* akan menghasilkan pengenalan pola yang baik hingga 100 persen dan lama iterasi yang cukup singkat apabila digunakan nilai parameter sebesar 0.7 dan 0.4 (Tabel 5 percobaan 4) serta 0.7 dan 0.5 (Tabel 4



Gambar 14 Form Training JST Backpropagation

percobaan 6) pada nilai *learning rate* dan *momentum*. Bobot-bobot dari hasil kedua pengujian ini, kemudian bisa disimpan dan dipergunakan pada tahap *mapping*. Namun, jika diamati lagi dari hasil kedua pengujian ini, hasil yang paling baik adalah pada pengujian dengan menggunakan nilai *momentum* dan *learning rate* sebesar 0.4 dan 0.7, yaitu di mana lama *training* hanya berkisar 11 menit. Sehingga dengan demikian, bobot-bobot yang akan disimpan adalah bobot-bobot hasil dari pengujian ini.

Menjalankan Testing JST (Mapping)

Pada tahap ini (Gambar 15), bobot-bobot, vektor *eigen* tereduksi, maupun pola-pola citra *X-ray* yang telah disimpan akan diproses bersamaan, di mana prosesnya hanya berupa tahap *feed-forward* (umpan maju). Sedangkan Gambar 16 merupakan tampilan hasil prediksi dari dua puluh pola citra *X-ray*, di mana 25 persen dari total pola dideteksi normal atau *tidak osteoporosis* dan 75 persen terdeteksi *osteoporosis*.

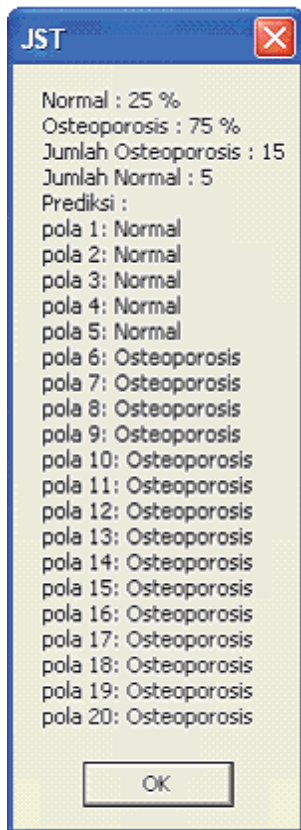
Hasil Uji Coba Testing Perangkat Lunak

Pada Tabel 6 ditunjukkan bahwa hasil dari pengujian atau *mapping* pola-pola yang sebelumnya telah dilatih akan memberikan persentase kebenaran 100 persen dalam mengenali pola. Hal ini dikarenakan bobot-bobot yang disimpan dan digunakan merupakan hasil pelatihan yang sama dengan pola-pola tersebut. Untuk data gambar selengkapnya dapat dilihat pada lampiran 2.

Pada Tabel 8 dapat juga dilihat hasil pengujian pola-pola citra *X-ray* yang belum dilatih. Dari tabel ini dapat diketahui juga bahwa pengenalan pola masih belum optimal, di mana masih terdapat kesalahan pendeteksian sejumlah 8 pola *X-ray* (27 persen) dari 30 pola *non-thresholding* dan 10 pola *X-ray* (33 persen) dari 30 pola *thresholding* yang diujikan. Namun, apabila diamati lagi, kesalahan tersebut tidaklah terlalu besar jika dibandingkan keberhasilan yang didapat. Dan dari pengamatan kedua jenis citra (citra *thresholding* dan citra *non-thresholding*), dapat diambil kesimpulan bahwa penggunaan citra *non-thresholding* akan menghasilkan pengenalan lebih baik dari citra *thresholding*. Sehingga, pada saat *mapping* selanjutnya, citra-citra yang ingin diuji tidak perlu sebelumnya dilakukan proses *thresholding*. Pada Tabel 7 dapat dilihat perbedaan gambar *X-ray Training* dan *Non-Training*.



Gambar 15 Tampilan Form Testing JST Backpropagation



Gambar 16 Tampilan Detil Output Prediksi Penyakit

PENUTUP

Berdasarkan hasil-hasil yang telah dicapai selama perancangan, pembuatan, dan pengujian perangkat lunak tugas akhir ini, maka dapat disimpulkan bahwa: metode *Backpropagation* pada jaringan syaraf tiruan merupakan salah satu metode yang dapat digunakan untuk membuat aplikasi pengenalan atau pendeteksian dengan tingkat keberhasilan yang cukup baik, karena lebih dari 50 persen jumlah data citra yang di-input dan diuji dapat dikenali dengan benar; penggunaan nilai ambang *thresholding* sebesar 80, parameter *learning rate* sebesar 0.7, *momentum* sebesar 0.4, iterasi sebesar 100 *epoch*, unit-unit lapis tersembunyi sebanyak 10 buah, dan unit *input* sebanyak 600 unit menunjukkan sistem telah berhasil mengenali pola-pola X-ray learning dan kurang berhasil mengenali pola-pola X-ray *non-learning*; untuk mendapatkan bobot-bobot yang sesuai, diperlukan uji coba pelatihan yang berulang-ulang karena bobot-bobot yang dihasilkan dari pelatihan akan menentukan tingkat pengenalan pada proses *mapping*; serta pengujian dengan sejumlah 30 citra dari 10 citra yang dilatihkan dengan jumlah perbandingan uji masukan *osteoporosis* dan normal adalah 50:50 memiliki hasil persentase keberhasilan sebesar 73 persen dan kegagalan sebesar 27 persen.

DAFTAR PUSTAKA

Ahmad, U. (2005). *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.
 Anton, H. (1998). *Aljabar Linier Elementer*, Edisi kelima. Jakarta: Erlangga.
 A Tutorial on *Principal Component Analysis*. (2002). <http://www.vision.auc.dk/sig/Teaching/Flerdim/Current/hotelling/hotelling.htm>, diakses Mei 2007.
 Fowler, M. (2005). *UML Distilled Panduan Singkat Tentang*

Bahasa Pemodelan Objek Standar, Edisi ketiga. Yogyakarta: Andi.
Glosarium Data dan Informasi Kesehatan. (2005). <http://www.depkes.go.id>, diakses Maret 2007.
 Grant, J.V. Basmajian, dan Charles E. Slonecker. (1995). *Metode Anatomi Berorientasi pada Klinik*, Edisi kesebelas. Jakarta: Binarupa Aksara.
 Hermawan, A. (2006). *Jaringan Syaraf Tiruan Teori dan Aplikasi*. Yogyakarta: Andi.
 Jolliffe, I.T., (2002). *Principal Component Analysis*, 3rd Edition. New York: Springer Verlag.
Manifestasi Klinis dan Penatalaksanaan Osteoporosis. (2004). http://www.dexa-medica.com/test/htdocs/dexamedica/article_files/_osteoporosis.pdf, diakses Mei 2007.
 Mardianto, I. (2007). *Bahan Kuliah Sistem Jaringan Syaraf – 12 Principal Component Analysis*. Jakarta: Jurusan FTI-Informatika, Universitas Bina Nusantara.
Osteoporosis, Low Back Pain, and Other Bone Disorders. (2004). http://www.healthsquare.com/condition_and_treatment/condition_topic_index/osteoporosis-lowbackpain-otherbonedisorders.htm, diakses Juni 2007.
Perbandingan Morfometri Vertebra Lumbal dengan Metacarpal Untuk Mendeteksi Dini Osteoporosis. (2007). <http://med.unhas.ac.id/en/-Medical Faculty of Hasanuddin University>, diakses Mei 2007.
 Purnomo, H., Mauridhi, dan Agus K. (2006). *Supervised Neural Network dan Aplikasinya*. Yogyakarta: Graha Ilmu.
 Putra, R. (2006). *The Best Source Code Visual Basic*. Jakarta: PT Gramedia.
 Sadikin, R (2006). *Bahan Kuliah Analisis Desain Sistem – Pemodelan Use-case*. Jurusan FTI-Informatika.
 Siang, J.J. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.
Suplemen Kalsium Cegah Osteoporosis. (2007). <http://www.MediaSehat.com>, diakses Mei 2007.
 ----- *Format File*. <http://slametriyanto.web.id/FormatFile.pdf>, diakses Mei 2007.
 ----- *Image*. <http://www.google.com>.
 ----- *Kelebihan PCX*. <http://tri1405.blogspot.com/2007/03/09/format-file-gambar-2-file-pcx>, diakses Juli 2007.
 ----- *Matrix*. <http://www.chemeng.ui.ac.id/s1/metnum/matrix.htm>, diakses Mei 2007.
 ----- *Matriks dan Komputasi*. <http://www.fisika.ui.ac.id/matriks.htm>, diakses Mei 2007.
 ----- *Modul KLT*. <http://www.planetsourcecode.com>, diakses Juni 2007.
 ----- *Osteoporosis*. <http://www.mitrakeluarga.com/informasi/osteoporosis.htm>, diakses Mei 2007.
 ----- *Principle Component Analysis*. http://dewey.petra.ac.id/spektra/module/catalog/docs/digital_thesis, diakses Mei 2007.
 ----- *Reumatologi*. <http://id.wikipedia.org/wiki/Reumatologi>, diakses Juni 2007.
 ----- *Vektor dan Nilai Eigen*. <http://www.bluebit.gr/matrix/eigenvalue-eigenvector.htm>, diakses Juni 2007.
 ----- <http://www.cs.cmu.edu/~cil/v-images.html>, diakses April 2007.

APPENDIX

Tabel 2 Data Pengujian Berdasarkan Jumlah Unit *Input* Proses Training

Proses Training	1	2	3	4	5
Data Uji	10	10	10	10	10
Unit Input	400	500	600	700	800
Unit Hidden	10	10	10	10	10
Unit Output	1	1	1	1	1
Toleransi	0,001	0,001	0,001	0,001	0,001
Learning Rate	0,8	0,8	0,8	0,8	0,8
Momentum	0,5	0,5	0,5	0,5	0,5
Epoch	10000	10000	10000	10000	10000
Treshold	0,55	0,55	0,55	0,55	0,55
Pengenalan Pola	100% (epoch ke 82)	100% (epoch ke 38)	100% (epoch ke 35)	100% (epoch ke 49)	100% (epoch ke 52)
Lama Training	174 menit 59 detik	87 menit 13 detik	70 menit 49 detik	102 menit 55 detik	153 menit 30 detik

Tabel 3 Data Pengujian Berdasarkan Jumlah Unit *Hidden* Proses Training

Proses Training	1	2	3	4	5	6
Data Uji	10	10	10	10	10	10
Unit Input	500	500	500	600	600	600
Unit Hidden	5	15	20	5	15	20
Unit Output	1	1	1	1	1	1
Toleransi	0,001	0,001	0,001	0,001	0,001	0,001
Learning Rate	0,8	0,8	0,8	0,8	0,8	0,8
Momentum	0,5	0,5	0,5	0,5	0,5	0,5
Epoch	10000	10000	10000	10000	10000	10000
Treshold	0,55	0,55	0,55	0,55	0,55	0,55
Pengenalan Pola	100% (epoch ke 65)	100% (epoch ke 48)	100% (epoch ke 52)	100% (epoch ke 52)	100% (epoch ke 39)	100% (epoch ke 37)
Lama raining	139 menit 40 detik	120 menit 40 detik	133 menit 34 detik	104 menit 3 detik	93 menit 19 detik	98 menit 12 detik

Tabel 4 Data Pengujian Berdasarkan *Learning Rate* Proses Training

Proses Training	1	2	3	4	5	6
Data Uji	10	10	10	10	10	10
Unit Input	500	500	500	600	600	600
Unit Hidden	10	10	10	10	10	10
Unit Output	1	1	1	1	1	1
Toleransi	0,001	0,001	0,001	0,001	0,001	0,001
Learning Rate	0,5	0,6	0,7	0,5	0,6	0,7
Momentum	0,5	0,5	0,5	0,5	0,5	0,5
Epoch	10000	10000	10000	10000	10000	10000
Treshold	0,55	0,55	0,55	0,55	0,55	0,55
Pengenalan Pola	100% (epoch ke 49)	100% (epoch ke 13)	100% (epoch ke 17)	100% (epoch ke 103)	100% (epoch ke 26)	100% (epoch ke 5)
Lama Training	102 menit 51 detik	25 menit 20 detik	33 menit 34 detik	218 menit 22 detik	55 menit 39 detik	13 menit 13 detik

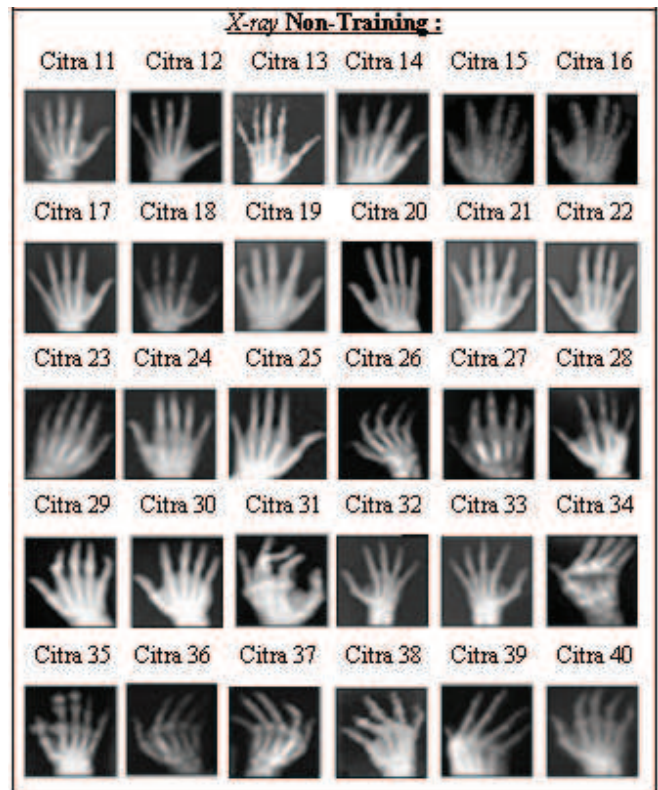
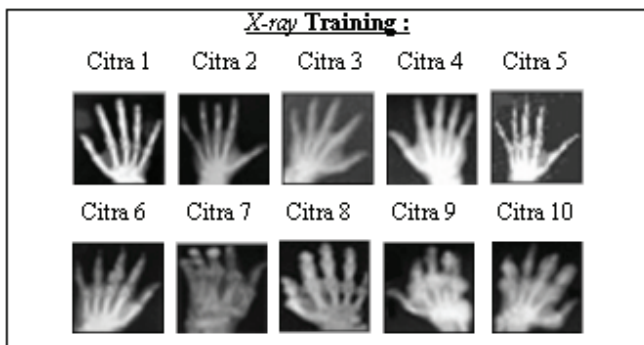
Tabel 5 Data Pengujian Berdasarkan Momentum Proses Training

Proses Training	1	2	3	4	5	6
Data Uji	10	10	10	10	10	10
Unit Input	500	500	500	600	600	600
Unit Hidden	10	10	10	10	10	10
Unit Output	1	1	1	1	1	1
Toleransi	0,001	0,001	0,001	0,001	0,001	0,001
Learning Rate	0,7	0,7	0,7	0,7	0,7	0,7
Momentum	0,4	0,7	0,8	0,4	0,7	0,8
Epoch	10000	10000	10000	10000	10000	10000
Treshold	0,55	0,55	0,55	0,55	0,55	0,55
Pengenalan Pola	100% (epoch ke 13)	100% (epoch ke 98)	100% (epoch ke 154)	100% (epoch ke 5)	100% (epoch ke 80)	100% (epoch ke 101)
Lama Training	22 menit 11 detik	168 menit 30 detik	304 menit 1 detik	11 menit 43 detik	170 menit 3 detik	207 menit 14 detik

Tabel 6 Hasil Pengujian Mapping Pola X-ray Training

X-Ray Tangan	Target	Output
Citra 1	0	0
Citra 2	0	0
Citra 3	0	0
Citra 4	0	0
Citra 5	0	0
Citra 6	1	1
Citra 7	1	1
Citra 8	1	1
Citra 9	1	1
Citra 10	1	1

Tabel 7 X-ray Training dan Non-Training



Tabel 8 Hasil Pengujian *Mapping Pola X-ray Non-Training*

X-Ray (Non-Thresholding)	X-Ray (Thresholding)	Target	Output (Non-Thresholding)	Output (Thresholding)
Citra 11	Citra 11	0	0	0
Citra 12	Citra 12	0	0	0
Citra 13	Citra 13	0	0	0
Citra 14	Citra 14	0	0	0
Citra 15	Citra 15	0	0	0
Citra 16	Citra 16	0	0	1
Citra 17	Citra 17	0	0	1
Citra 18	Citra 18	0	1	1
Citra 19	Citra 19	0	1	1
Citra 20	Citra 20	0	1	1
Citra 21	Citra 21	0	1	1
Citra 22	Citra 22	0	1	1
Citra 23	Citra 23	0	1	1
Citra 24	Citra 24	0	1	1
Citra 25	Citra 25	0	1	1
Citra 26	Citra 26	1	1	1
Citra 27	Citra 27	1	1	1
Citra 28	Citra 28	1	1	1
Citra 29	Citra 29	1	1	1
Citra 30	Citra 30	1	1	1
Citra 31	Citra 31	1	1	1
Citra 32	Citra 32	1	1	1
Citra 33	Citra 33	1	1	1
Citra 34	Citra 34	1	1	1
Citra 35	Citra 35	1	1	1
Citra 36	Citra 36	1	1	1
Citra 37	Citra 37	1	1	1
Citra 38	Citra 38	1	1	1
Citra 39	Citra 39	1	1	1
Citra 40	Citra 40	1	1	1