

# IMPLEMENTASI JARINGAN SARAF TIRUAN *BACKPROPAGATION* UNTUK DETEKSI WAJAH DALAM CITRA DIGITAL

Stanly Hence Dolfi Loppies  
stanly@unmus.ac.id

## ABSTRAK

Sebuah citra adalah kumpulan piksel-piksel yang disusun dalam larik dua-dimensi. Sebuah piksel pada citra digital adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Pendeteksian wajah adalah salah satu bidang kecerdasan buatan yang berguna pada pengolahan data.

Tujuan dari penelitian ini adalah untuk mengimplementasikan jaringan saraf tiruan pada pendeteksian wajah dalam citra digital dengan menggunakan metode *backpropagation*. Deteksi wajah dalam citra digital akan dilatih dan diuji menggunakan tahap-tahap pada *backpropagation*. Vektor *input* pelatihan dan pendeteksian wajah diperoleh dari hasil pengolahan citra.

Metodologi yang digunakan dalam pengembangan perangkat lunak ini adalah *Rational Unified Process* (RUP) dengan kaskas yang digunakan adalah *Unified Modelling Language* (UML). Aplikasi dibangun dengan menggunakan bahasa pemrograman Matlab 7.0.

Berdasarkan hasil pengujian menunjukkan bahwa semua citra wajah yang dilatih berhasil dikenali. Citra wajah yang telah dimodifikasi akan memiliki nilai akurasi pendeteksian yang berbeda dengan citra wajah sebelum dimodifikasi. Hal ini disebabkan karena perbedaan vektor *input* dari citra-citra tersebut.

Kata kunci: Wajah Citra Digital, Jaringan Saraf Tiruan, *Backpropagation*.

## 1. PENDAHULUAN

Pengolahan citra secara digital dimulai pada awal tahun 1920-an, dimana sebuah foto berhasil ditransmisikan secara digital melalui kabel laut dari kota New York ke London. Pada tahun 1960-an terjadi perkembangan yang pesat seiring dengan munculnya teknologi komputer, dimana muncul berbagai algoritma pengolahan citra. Sebuah citra adalah kumpulan piksel-piksel yang disusun dalam larik dua-dimensi. Sebuah piksel pada citra digital adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Pendeteksian wajah adalah salah satu bidang kecerdasan buatan yang berguna pada pengolahan data. (Scach, 2008)

Pendeteksian wajah (*face detection*) adalah salah satu tahap awal yang sangat penting sebelum dilakukannya pengenalan wajah (*face recognition*). Pada kasus seperti pemotretan untuk pembuatan KTP (Kartu Tanda Penduduk), SIM (Surat Izin Mengemudi), dan kartu kredit, citra yang didapatkan umumnya hanya berisi satu wajah dan memiliki latar belakang seragam dan kondisi pencahayaan yang telah diatur sebelumnya sehingga deteksi wajah dapat dilakukan dengan lebih mudah.

Pengenalan wajah manusia dalam gambaran visual dapat diimplementasikan ke dalam banyak aplikasi yang mencakup pengenalan wajah, yang mengisyaratkan pemahaman, pengawasan yang dimonitoring oleh sebuah *database* manajemen video atau sistem keamanan seperti izin akses masuk ruangan, pengawasan lokasi, maupun pencarian identitas individu pada *database* kepolisian dan sebagainya (Kiki et al, 2009).

Kemampuan yang dimiliki JST dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau *input* yang dimasukkan dan membuat prediksi tentang kemungkinan *output* yang akan muncul atau menyimpan karakteristik dari *input* yang disimpan kepadanya (Puspitaningrum, 2006).

Berbagai macam algoritma telah dipakai dan dikembangkan untuk membandingkan antara algoritma satu dengan algoritma lainnya untuk mendapatkan hasil yang

lebih baik. Perambatan galat mundur (*backpropagation*) adalah sebuah metode sistematis untuk pelatihan *multilayer* jaringan saraf tiruan. Jaringan saraf tiruan merupakan salah satu jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan saraf manusia yang diimplementasikan sebagai pengklasifikasi wajah.

## 2. TINJAUAN PUSTAKA

### 2.1 Citra Digital

Sebuah piksel pada citra digital adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Sebuah citra adalah kumpulan piksel-piksel yang disusun dalam larik dua-dimensi. Indeks baris dan kolom (x,y) dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel (0,0) terletak pada sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak ke bawah. Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer (Usman 2005).

### 2.2 Pengolahan Citra Digital

Pengolahan citra digital dimulai sekitar awal tahun 1920-an dari pemberitaan dunia cetak, dimana sebuah citra dikirim melalui sebuah kabel bawah laut dari London menuju ke New York. Teknologi pengolahan citra digital dapat masuk ke berbagai bidang seperti kedokteran, industri, pertanian, geologi, kelautan dan lain sebagainya. Dalam pengolahan citra digital akan sering dijumpai beberapa istilah-istilah seperti *image processing*, *image analysis*, *image understanding*, dan *computer vision*. Keempat istilah ini seringkali dibedakan dari *input* dan *output* dari suatu citra seperti penjelasan di bawah ini:

#### 1. *Image Processing*

Memiliki *input* dan *output* berupa citra, sebagai contoh suatu citra ditransformasikan ke bentuk citra yang lainnya.

## 2. Image Analysis

Memiliki *input* berupa citra dengan *output* bukan citra akan tetapi berupa hasil pengukuran terhadap citra tersebut. Sebagai contoh, suatu citra wajah dianalisis untuk mendapatkan fitur wajah seperti jarak kedua mata dan jarak mata dan hidung.

## 3. Image Understanding

Memiliki *input* berupa citra dengan *output* adalah deskripsi tingkat tinggi dari citra tersebut (*output* bukan berupa citra). Sebagai contoh, diberikan suatu *input* citra seseorang, keluarannya deskripsi dari orang tersebut dapat berupa seperti: orang tersebut sedang menangis, sedih, senyum atau tertawa lebar.

## 4. Computer Vision

Bertujuan untuk mengkomputerisasi penglihatan manusia atau dengan kata lain membuat citra digital dari citra sebenarnya (sesuai dengan penglihatan manusia). Hal tersebut dapat disimpulkan *input* dari *computer vision* adalah berupa citra penglihatan manusia sedangkan *output*-nya berupa citra digital.

## 2.3 Format File Citra Digital

Format file citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format ini digunakan dalam menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Dalam penelitian ini penulis menggunakan empat format file citra digital yaitu JPG, dan BMP. Berikut ini adalah penjelasan beberapa format umum digunakan saat ini (Putra 2010):

### 1. Bitmap (.bmp)

Format .bmp adalah format penyimpanan standard tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai piksel.

### 2. JPEG (.jpg)

.jpg sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

### 3. MPEG (.mpg)

Format ini digunakan didunia internet dan diperuntukkan sebagai format penyimpan citra bergerak (video).

### 4. Portable Network Graphics (.png)

Format .png adalah format penyimpanan citra terkompresi. Format ini digunakan pada citra *grayscale*, citra dengan palet warna, dan juga citra *full color*.

### 5. RGB (.rgb)

Format ini merupakan format penyimpanan citra yang dibuat oleh *silicon graphics* untuk menyimpan citra berwarna.

## 2.4 Deteksi Wajah (Face Detection)

Deteksi wajah atau *face detection* berhubungan dengan mencari apakah ada wajah dalam gambar tertentu (biasanya dalam skala abu-abu) dan jika ada, kenali lokasi gambar dan isi dari setiap wajah. Ini adalah langkah awal dari setiap sistem yang sepenuhnya otomatis yang menganalisis setiap informasi yang terdapat pada wajah, misalnya identitas, jenis kelamin, ekspresi, usia, ras, dan pose.

Dalam beberapa tahun terakhir, pengenalan wajah telah menarik banyak perhatian dan penelitiannya telah dengan

cepat diperluas dengan tidak hanya insinyur tapi juga ahli saraf, karena memiliki banyak potensi aplikasi dalam visi komputer komunikasi dan sistem akses kontrol otomatis. Terutama deteksi wajah adalah bagian penting dari pengenalan wajah sebagai langkah awal pengenalan wajah otomatis. Namun, deteksi wajah tidak mudah karena memiliki banyak variasi dari penampilan gambar, seperti menimbulkan variasi (depan, non depan), oklusi, orientasi gambar, menerangi kondisi dan ekspresi wajah.

## 2.5 Jaringan Saraf Tituan (JST)

Jaringan saraf tiruan (*artificial neural network*), atau disingkat JST adalah sistem komputasi di mana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut.

Model saraf ditunjukkan dengan kemampuannya dalam emulasi, analisis, prediksi, dan asosiasi. Kemampuan yang dimiliki JST dapat digunakan untuk belajar dan menghasilkan aturan atau operasi dari beberapa contoh atau *input* yang dimasukkan dan membuat prediksi tentang kemungkinan *output* yang akan muncul atau menyimpan karakteristik dari *input* yang disimpan kepadanya (Hermawan 2006).

Jaringan saraf tiruan seperti manusia belajar dari suatu contoh karena mempunyai karakteristik yang adaptif, yaitu dapat belajar dari data sebelumnya dan mengenal pola data yang selalu berubah. Selain itu, JST merupakan sistem yang tak terprogram, artinya semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran atau pelatihan.

Hal yang ingin dicapai dengan melatih JST adalah untuk mencapai keseimbangan antara kemampuan *memorisasi* dan *generalisasi*. Yang dimaksud kemampuan *memorisasi* adalah kemampuan JST untuk mengambil kembali secara sempurna sebuah pola yang telah dipelajari. Kemampuan *generalisasi* adalah kemampuan JST untuk menghasilkan tanggapan yang bisa diterima terhadap pola-pola *input* yang serupa (namun tidak identik) dengan pola-pola sebelumnya telah dipelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam JST itu dimasukkan informasi baru yang belum pernah dipelajari, maka JST itu masih akan tetap dapat memberikan tanggapan baik, memberikan keluaran yang paling mendekati (Puspitaningrum, 2006).

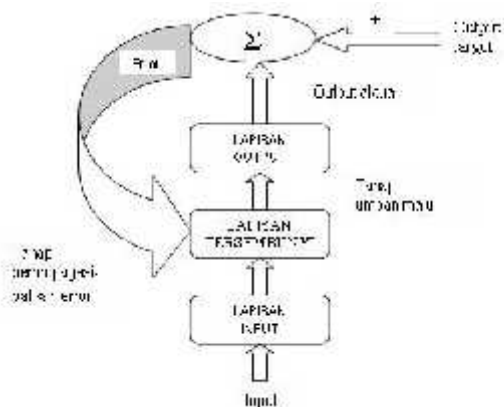
Otak terdiri dari sekitar 10.000.000.000 sel saraf yang saling berhubungan. Sel saraf mempunyai cabang struktur *input (dendrites)*, sebuah inti sel dan percabangan struktur *output (axon)*. *Axon* dari sebuah sel terhubung dengan *dendrites* yang lain melalui sebuah *synapse*. Ketika sebuah sel saraf aktif, kemudian menimbulkan suatu signal *electrochemical* pada *axon*. Signal ini melewati *synapse* menuju ke sel saraf yang lain. Sebuah sel saraf lain akan mendapatkan signal jika memenuhi batasan tertentu yang sering disebut dengan nilai ambang atau *threshold* (Yani 2005).



Gambar 1. Susunan Saraf Manusia

## 2.6 Metode Backpropagation

Perambatan galat mundur (*backpropagation*) adalah sebuah metode sistematis untuk pelatihan *multilayer* jaringan saraf tiruan. Metode ini memiliki dasar matematis yang kuat, obyektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat *error* melalui model yang dikembangkan (Kiki dan Kusumadewi 2009). Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan pada unit-unit dilapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan tanggapan sebagai keluaran JST. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (*backward*) pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukan (Puspitaningrum 2006).



**Gambar 2.** Alur Kerja Jaringan *Backpropagation*

Cara kerja jaringan propagasi balik atau *Backpropagation* adalah sebagai berikut mula-mula jaringan diinisialisasi dengan bobot yang diset dengan bilangan acak. Lalu contoh-contoh pelatihan dimasukkan ke dalam jaringan. Contoh pelatihan terdiri dari pasangan vektor *input* dan vektor *output* target. Keluaran dari jaringan berupa sebuah vektor *output* aktual. Selanjutnya vektor *output* aktual jaringan dibandingkan dengan vektor *output* target untuk mengetahui apakah *output* jaringan sudah sesuai dengan harapan (*output* aktual sudah sama dengan *output* target).

*Error* yang timbul akibat perbedaan antara *output* aktual dengan *output* target tersebut kemudian dihitung dan digunakan untuk mengupdate bobot-bobot yang relevan dengan jalan mempropagasikan kembali *error*. Setiap perubahan bobot yang terjadi diharapkan dapat mengurangi besar *error*. *Epoch* merupakan pengulangan/iterasi pelatihan atau siklus setiap pola pelatihan seperti ini dilakukan pada semua set pelatihan sampai kerja jaringan mencapai tingkat yang diinginkan atau sampai kondisi berhenti terpenuhi. Yang dimaksud dengan kondisi berhenti disini misalnya: pelatihan akan dihentikan setelah *epoch* mencapai 1000 kali, atau pelatihan akan dihentikan sampai sebuah nilai ambang yang ditetapkan terlampaui. Setelah proses pelatihan selesai, barulah ditetapkan algoritma aplikasi. Biasanya sebelum digunakan untuk aplikasi yang sebenarnya, pengujian untuk kerja jaringan dilakukan dengan cara memasukkan set pengujian (satu tes) ke dalam jaringan. Karena bersifat untuk menguji, set pengujian hanya berupa *input* saja. Dari tanggapan jaringan dapat dinilai kemampuan memorisasi

dan generalisasi jaringan dalam menebak *output* berdasarkan pada apa yang telah dipelajarinya selama ini.

Algoritma propagasi balik dapat dibagi ke dalam 2 bagian :

1. Algoritma pelatihan  
 Terdiri dari 3 tahap yaitu tahap umpan maju pola pelatihan *input*, tahap mempropagasibalikan *error*, dan tahap pengaturan bobot.
2. Algoritma aplikasi  
 Yang digunakan hanyalah tahap umpan maju saja.

## 3. METODOLOGI PENELITIAN

### 3.1 Pelatihan *Backpropagation*

Aturan pelatihan jaringan propagasi balik terdiri dari 2 tahap, *feedforward* dan *backward propagation*. Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah vektor *feature* yang disebut dengan vektor *input* yang diasosiasikan dengan sebuah *output* yang menjadi target pelatihannya. Dengan kata lain set pelatihan terdiri dari vektor *input* dan juga vektor *output* target.

Setiap perubahan bobot yang terjadi dapat mengurangi *error*. Siklus setiap perubahan bobot (*epoch*) dilakukan pada setiap set pelatihan hingga kondisi berhenti dicapai, yaitu bila mencapai jumlah *epoch* yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui. Algoritma pelatihan jaringan propagasi balik terdiri dari 3 fase yaitu (Puspitaningrum 2006) :

1. Propagasi maju (*feedforward*) : dalam propagasi maju, sinyal *input* dari lapisan *input* dipropagasikan ke lapisan tersembunyi yang menggunakan fungsi aktivasi yang telah ditentukan. Keluaran dari setiap unit dari lapisan tersembunyi tersebut dipropagasikan maju lagi ke lapisan *output*. Begitu seterusnya sehingga menghasilkan keluaran jaringan sinyal *output* dilapisan *output*.
2. Propagasi mundur (*backpropagation*) : keluaran jaringan sinyal *output* dilapisan *output* dibandingkan dengan nilai target yang sudah ditetapkan dengan perbandingan atau selisih dari nilai sinyal *output* dengan target merupakan kesalahan yang terjadi ini kemudian dipropagasikan mundur. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahann yang terjadi. Nilai kesalahan dilapisan *output* dipropagasikan ke lapisan tersembunyi dilapisan tersembunyi.
3. Perubahan bobot : setelah semua faktor bobot dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor bobot neuron di atasnya.

Algoritma pelatihan untuk jaringan dengan satu lapisan tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut:

**Langkah 0** : inialisasi bobot-bobot, konstanta laju pelatihan (  $\eta$  ), toleransi *error* atau nilai bobot (bila menggunakan nilai bobot sebagai kondisi berhenti) atau set maksimal *epoch* (jika menggunakan banyaknya *epoch* sebagai kondisi berhenti).

**Langkah 1** : selama kondisi berhenti belum dicapai, maka lakukan langkah ke-2 hingga langkah ke-9.

**Langkah 2 :** untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

**Tahap I : Propagasi maju (feedforward)**

**Langkah 3 :** setiap unit *input*  $X_i$  (dari unit ke-1 hingga ke- $n$  pada lapisan *input*) mengirimkan sinyal *input* ke setiap *input* yang berada pada lapisan tersembunyi.

**Langkah 4 :** masing-masing unit di lapisan tersembunyi (dari unit ke-1 hingga unit ke- $p$ ) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya :

$$z_{netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z_{netj}) = \frac{1}{1 + e^{-z_{netj}}}$$

$z_j$  = keluaran unit tersembunyi  
 $v$  = sinyal bias layar masukan  
 $x$  = sinyal pada layar masukan  
 $i = 1, 2, \dots, n$   
 $j = 1, 2, \dots, p$

**Langkah 5 :** masing unit *output* ( $y_k, k=1,2,3,\dots,m$ ) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan biasnya.

$$y_{netk} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}}$$

$y$  = keluaran unir keluaran  
 $w$  = sinyal bias layar tersembunyi  
 $z$  = sinyal layar tersembunyi  
 $k = 1, 2, \dots, m$

**Tahap II : Propagasi Mundur (backward propagation)**

**Langkah 6 :** masing-masing unit *output* ( $y_k, k=1,2,3,\dots,m$ ) menerima pola target  $t_k$  sesuai dengan pola masukan saat pelatihan dan informasi kesalahan lapisan *output* ( $k$ ) dihitung.  $k$  dikirim ke lapisan di bawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias ( $W_{jk}$  dan  $V_{j0}$ ) antara lapisan tersembunyi dengan lapisan *output* :

$$\delta_k = (t_k - y_k) f'(y_{netk}) = (t_k - y_k) y_k (1 - y_k)$$

= error pada unit keluaran  
 $t$  = target

$y$  = keluaran unit keluaran  
 $k = 1, 2, \dots, m$

Hitung suku perubahan bobot  $W_{jk}$  (yang akan digunakan untuk mengubah bobot  $W_{jk}$  dengan laju pelatihan

$$\Delta w_{kj} = \alpha \delta_k z_j$$

$w$  = suku perubahan bobot unit keluaran  
= learning rate  
 $z$  = sinyal layar tersembunyi  
 $j = 1, 2, \dots, p$   
 $k = 1, 2, \dots, m$

**Langkah 7 :** pada setiap unit di lapisan tersembunyi (dari unit ke-1 hingga ke- $p$ ;  $i=1\dots n; k=1\dots m$ ) dilakukan perhitungan informasi kesalahan lapisan tersembunyi ( $j$ ).  $j$  kemudian digunakan untuk menghitung besar koreksi bobot dan bias ( $V_{ji}$  dan  $V_{j0}$ ) antara lapisan *input* dan lapisan tersembunyi.

$$\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj}$$

$$\delta_k = \delta_{netj} f'(z_{netk}) = \delta_{netj} z_j (1 - z_j)$$

= error pada unit tersembunyi  
 $z$  = sinyal layar tersembunyi  
 $j = 1, 2, \dots, p$   
 $k = 1, 2, \dots, m$

Hitung perubahan bobot  $V_{ji}$  (yang digunakan untuk perbaikan bobot  $V_{ji}$ ).

$$\Delta v_{ij} = \alpha \delta_j x_i$$

$v$  = suku perubahan bobot unit tersembunyi  
= learning rate  
 $x$  = sinyal layar masukan  
 $i = 1, 2, \dots, n$   
 $j = 1, 2, \dots, p$

**Tahap III : Perubahan bobot**

**Langkah 8 :** masing-masing unit keluaran ( $y_k, k=1,2,3,\dots,m$ ) dilakukan peng-*update*-an bias dan bobotnya ( $j=0,1,2,\dots,p$ ) sehingga menghasilkan bobot dan bias baru :

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

$w$  = sinyal bias layar tersembunyi  
 $j = 1, 2, \dots, p$   
 $k = 1, 2, \dots, m$

Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai dengan unit ke- $p$  dilakukan peng-*update*-an bobot dan bias :

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

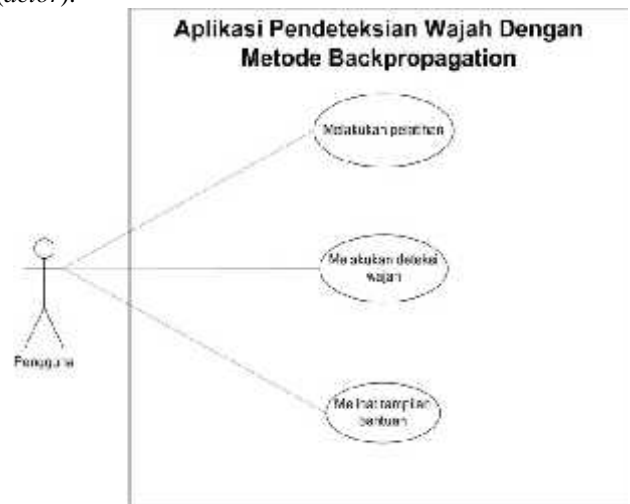
$v$  = sinyal bias layar masukan  
 $i = 1, 2, \dots, n$   
 $j = 1, 2, \dots, p$   
 $k = 1, 2, \dots, m$

**Langkah 9 :** uji kondisi berhenti (akhir iterasi)

**3.2 Pengembangan Sistem**

**3.2.1 Use case diagram**

*Use case diagram* di bawah ini memperlihatkan interaksi yang terjadi antara sistem dengan lingkungannya (*actor*).

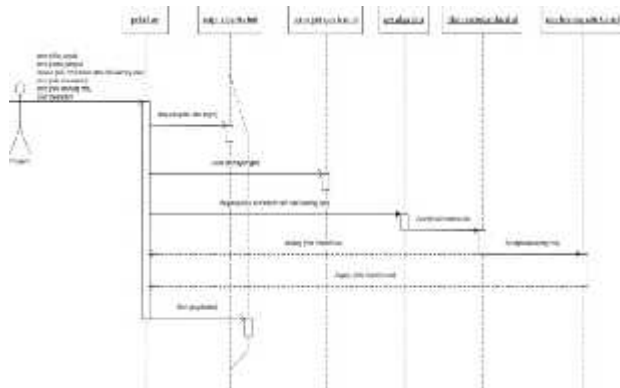


Gambar 3. Use Case Diagram

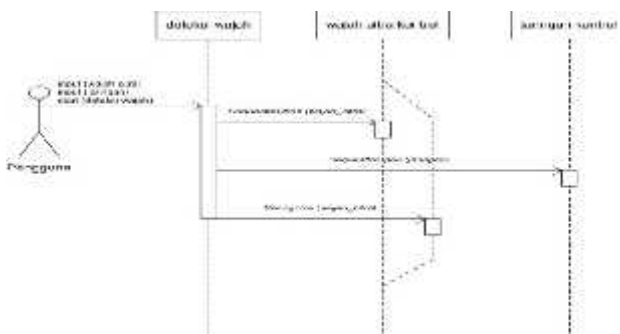
**3.2.2 Sequence diagram**

*Sequence diagram* menunjukkan deretan pesan yang dikirimkan diantara objek dan hal ini juga menunjukkan

interaksi antar objek. Gambar *sequence diagram* menunjukkan rangkaian perubahan aktivitas yang terjadi.



Gambar 4. Sequence Diagram Pelatihan



Gambar 5. Sequence Diagram Deteksi Wajah

## 6. HASIL dan PEMBAHASAN

### 6.1 Pengujian Pelatihan Menggunakan Metode Backpropagation dengan Format .JPG

Tabel 1. Detail dari Citra Wajah .JPG

No	Nama citra	Size	Dimensi citra	Epoch	SSE/Goal	Gradient
1.	Citra Wajah A RGB	103 KB	800 x 600	39	2.51747e-006 / 1e-005	0.00357313 / 1e-006
2.	Citra wajah B Grayscale	154 KB	800 x 600	943	9.84434e-006 / 1e-005	0.00302824 / 1e-006
3.	Citra wajah C RGB	25,1 KB	400 x 300	567	9.77821e-006 / 1e-005	0.00221069 / 1e-006
4.	Citra wajah D Grayscale	57,4 KB	400 x 300	86	7.96753e-006 / 1e-005	0.00888839 / 1e-006

Keterangan Tabel:

Berdasarkan hasil pengujian pada citra digital dengan format .jpg, seperti pada tabel di atas dalam mendeteksi citra wajah A, B, C dan D maka dapat dikatakan bahwa:

1. Citra RGB dengan ukuran file besar cenderung memiliki nilai *epoch* yang lebih tinggi dibandingkan dengan citra RGB dengan ukuran file lebih rendah.
2. Citra RGB memiliki nilai *epoch* yang lebih rendah daripada citra *grayscale*
3. Berdasarkan hasil pengujian, citra RGB dengan *size* tinggi memiliki tingkat nilai *epoch* paling tinggi dibandingkan citra RGB dengan *size* rendah memiliki nilai *epoch* lebih rendah.

### 6.2 Pengujian Peltihan Menggunakan Metode Backpropagation dengan Format .BMP

Tabel 2. Detail dari Citra Wajah .BMP

No	Nama citra	Size	Dimensi citra	Epoch	SSE / Goal	Gradient
1.	Citra wajah E RGB	1,43 MB	720 x 552	57	5.88932e-006 / 1e-005	0.00290761 / 1e-006
2.	Citra wajah F Grayscale	1,43 MB	720 x 552	156	8.0664e-006 / 1e-005	0.00155013 / 1e-006
3.	Citra wajah G RGB	1,48 MB	720 x 540	67	8.60356e-006 / 1e-005	0.0024062 / 1e-006
4.	Citra wajah H Grayscale	1,48 MB	720 x 540	155	9.73971e-006 / 1e-005	0.00223645 / 1e-006

Keterangan tabel:

Berdasarkan hasil pengujian pada citra digital dengan format .bmp, seperti pada tabel 6.4 dan 6.5 maka dapat dikatakan bahwa:

1. Citra RGB lebih rendah nilai *epoch*, dibandingkan dari *grayscale*
2. Berdasarkan hasil pengujian, citra RGB memiliki nilai *epoch* lebih rendah dibandingkan *grayscale*.

### 6.3 Pengujian Pendeteksian Wajah Menggunakan Metode Backpropagation dengan Format .JPG

Tabel 3. Detail dari Citra Wajah .JPG

No	Nama citra	Size	Dimensi citra	Hasil deteksi wajah
1.	Citra wajah A RGB	103 KB	800 x 600	Wajah
2.	Citra wajah B Grayscale	154 KB	800 x 600	Wajah
3.	Citra wajah C RGB	25,1 KB	400 x 300	Wajah
4.	Citra wajah D Grayscale	57,4 KB	400 x 300	Wajah

Keterangan tabel :

Berdasarkan hasil pengujian pada citra digital dengan format .jpg, seperti pada tabel di atas, maka dapat dikatakan bahwa:

- Hasil pengujian seluruh citra wajah dengan format JPG menunjukkan bahwa hasil tersebut adalah “Wajah”.

#### 6.4 Pengujian Pendeteksian Menggunakan Metode Backpropagation dengan Format .BMP

**Tabel 4. Detail dari Citra Wajah .BMP**

No	Nama citra	Size	Dimensi citra	Hasil deteksi wajah
1.	Citra wajah E RGB	1,43 MB	720 x 552	Wajah
2.	Citra wajah F Grayscale	1,43 MB	720 x 552	Wajah
3.	Citra wajah G RGB	1,48 MB	720 x 540	Wajah
4.	Citra wajah H Grayscale	1,48 MB	720 x 540	Wajah

Keterangan tabel :





Berdasarkan hasil pengujian pada citra digital dengan format .bmp, seperti pada Tabel 4 maka dapat dikatakan bahwa:

- Hasil pengujian seluruh citra wajah dengan format BMP menunjukkan bahwa hasil tersebut adalah “Wajah”.

#### 6.5 Pengujian Perbandingan Citra Wajah

Pengujian berikut dilakukan terhadap citra dengan nama file yang sama dengan citra namun telah dimodifikasi. Modifikasi yang dilakukan adalah penambahan *noise* pada citra wajah. Perbandingan dilihat pada hasil pendeteksian wajah.

**Tabel 5. Pengujian Citra Wajah yang Tidak dan yang Telah Dimodifikasai**

No.	Modifikasi	Citra Wajah Pendeteksian	Hasil Pendeteksian Wajah
1.	Tidak dimodifikasi		Wajah
2.	Noise		Wajah
3.	Tidak dimodifikasi		Wajah
4.	Noise		Wajah

Berdasarkan hasil pengujian pada tabel di atas maka dapat disimpulkan bahwa citra wajah yang telah dimodifikasi memiliki hasil pendeteksian wajah yang sama dengan citra wajah yang tidak dimodifikasi. Hal ini dikarenakan tidak adanya perbedaan vektor *input* yang dihasilkan antara citra asli dengan citra yang sudah dimodifikasi pada proses pengolahan citra.

#### 6.6 PEMBAHASAN

Pengujian yang dilakukan dalam penelitian ini adalah pengujian *blackbox* dan pengujian metode *backpropagation* pada aplikasi. Pengujian *blackbox* menunjukkan fungsionalitas aplikasi berjalan dengan baik sehingga pengujian metode pada aplikasi dapat dilanjutkan. Citra yang digunakan pada proses pengujian adalah citra dengan format JPG dan BMP dan kemudian akan dilatih dan dideteksi perbandingan citra wajah pada citra digital.

Analisis pada metode *backpropagation* dilihat dari pengujian yang sudah dilakukan maka dapat dikatakan untuk melatih dan mendeteksi format .jpg tidak bisa melatih *size* di atas 2.5 MB dan dimensi di atas 3072 x 2304 sedangkan untuk format .bmp tidak bisa melatih dan mendeteksi *size* 2.5 MB dan dimensi di atas 3000 x 2250.

Melatih citra wajah dengan metode *backpropagation* lebih baik jika nilai *epoch* lebih rendah sedangkan dalam mendeteksi citra wajah gambar wajah harus menghadap ke depan. Dimensi citra wajah dari metode *backpropagation* ini tidak berubah dan untuk format citra wajah yang sudah dilatih dan dideteksi adalah sama dengan format citra awal yang belum dilatih dan dideteksi.

#### 7. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan yaitu:

- Metode *backpropagation* mampu mendeteksi wajah pada citra digital.
- Metode *backpropagation* melakukan pelatihan citra wajah dengan nilai *epoch* yang lebih rendah cenderung terjadi pada citra RGB dengan *size* yang rendah dibandingkan *grayscale* dengan *size* yang berbeda-beda yang nilai *epoch* lebih tinggi, sedangkan dalam pendeteksian wajah baik citra RGB maupun *grayscale* dengan *size* yang berbeda-beda menunjukkan hasil tersebut “Wajah”.
- Metode *backpropagation* tidak bisa melatih dan mendeteksi citra wajah dengan format file citra digital .jpg dengan *size* lebih dari 2.5 MB dan dimensi di atas 3072 x 2304 sedangkan untuk format .bmp tidak lebih dari 2.5 MB dan dimensi 3000 x 2250.

#### DAFTAR PUSTAKA

- Al Fatta Hanif, 2007, *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*, Penerbit Andi, Yogyakarta.
- Galuh Dian Maulana, 2005, *Penentuan Spesifikasi Perangkat Literatur Metode RUP*, 26 September 2012 <http://eprints.lib.ui.ac.id/4376/1/126546-PA-52-Penentuan%20spesifikasi-Literatur.pdf>.

3. Herawati, Ira 2011, *Pemodelan Wajah 3D Melalui Pendeteksian Fitur Wajah 2D Menggunakan Teknik Morphing*, 21 Februari 2012, <[http://digilib.ittelkom.ac.id/index.php?option=com\\_content&view=article&id=840:definisi-citra-digital&catid=18:multimedia&Itemid=14](http://digilib.ittelkom.ac.id/index.php?option=com_content&view=article&id=840:definisi-citra-digital&catid=18:multimedia&Itemid=14)>.
4. Kiki, Sri, Kusumadewi 2009, 'Jaringan Saraf Tiruan Dengan Metode Backpropagation Untuk Mendeteksi Gangguan Psikologi, *Jurnal Media Informatika*, vol.2, no.2, hal 1-11.
5. Maduko, E, 2007, *Development and Testing of A Neuro-Fuzzy Classification System for IOS Data ini Asthmatic Children*, Proquest, Norwood.
6. Neneng, Kusworo .A, Isnanto .R.R, 2016 Support vector Machine untuk Klasifikasi Jenis Daging Berdasarkan Tekstur Menggunakan Ekstrasi Ciri Gray Level Co-Occurrence Matrices (GLCM). *Jurnal Sistem Informasi dan Bisnis*, Vol 1 hal 1-10.
7. Nurmila, N., Sugiharto, A., Sarwoko, E. A. (2010). Algoritma Backpropagation Neural Network untuk Pengenalan Pola Karakter Huruf Jawa. *Jurnal Masyarakat Informatika*. Vol. 1, No. 1.
8. Puspitaningrum, Diyah, 2006, *Pengantar Jaringan Saraf Tiruan*, Penerbit Andi, Yogyakarta.
9. Putra, Darma, 2010, *Pengolahan Citra Digital*, CV. Andi Offset, Yogyakarta
10. Rinaldi, Munir 2008, 'Image Watermarking untuk citra berwarna dengan metode berbasis korelasi dalam ranah DCT', *Jurnal Ilmiah Realtech Fakultas Teknik UNIKA DE La Salle Manado*, vol.4, no.2, hal 1-116.
11. Saubira .N, Isnanto R.R, Kuworo .A, 2016 Jaringan Saraf Tiruan Perambatan Balik Untuk Pengenalan Wajah. *Jurnal Sistem Informasi dan Bisnis*, Vol 1, hal 30-37.
12. Schach, S.R, 2008, *Object Oriented Software Engineering*. McGraw Hill, Singapore.
13. Sedyono, Eko 2006, 'Penentuan Lokasi Fasilitas Gudang Menggunakan Fuzzy C – Means ( FCM )', *Jurnal Informatika*, vol.2, no.2, 155–166.
14. Usman Ahmad, 2005, *Pengolahan Citra Digital dan Teknik Pemrogramannya*, Penerbit Graha Ilmu, Yogyakarta.